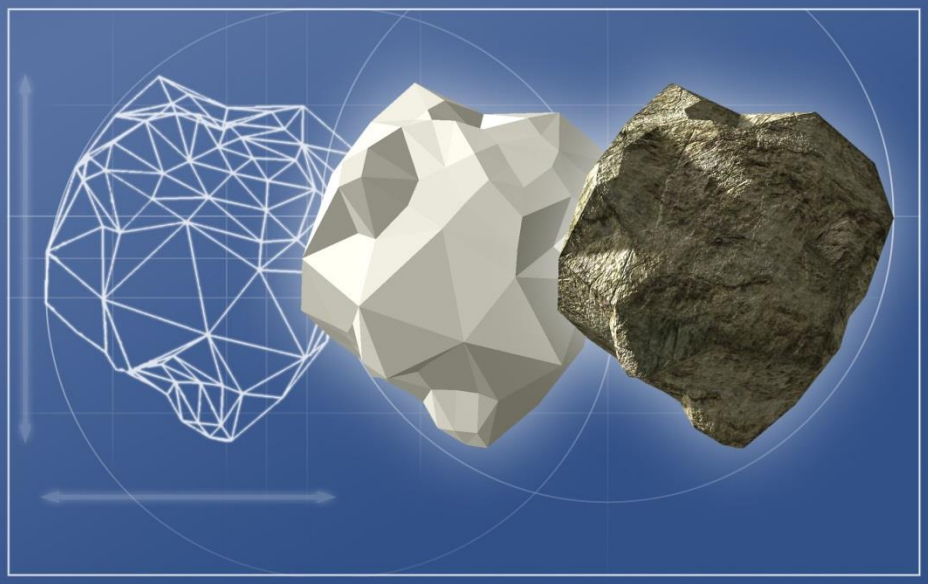
ROCK FACTORY

Owner's manual



by AtomicCrew.com
games & Unity development

AtomicCrew

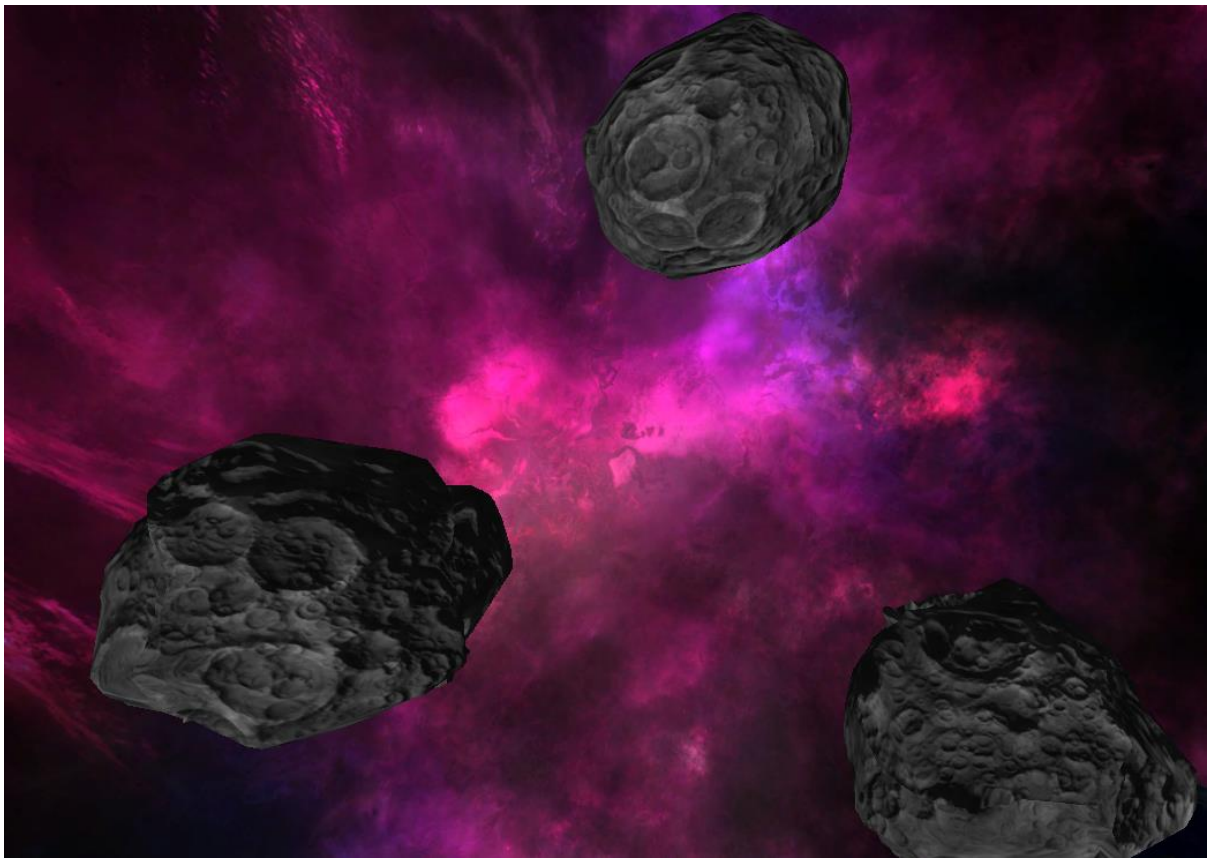
unity

Contents

1 Introduction

Rock Factory has been designed to automate the design, modelling and texture mapping processes, when creating unique rocks for game development purposes within the Unity game engine.

It consists out of a RFRock class for generating the mesh and a Rock Factory 'über-shader' that can be used to enhance visual quality by bringing a toolset of different shader techniques to add detail to your rock.



2 Modeling Concept

2.1 Overview

A rock is generated by roughly following one of the many modelling techniques for creating a rock.

The internal modeling concept step by step, using the most basic types for illustrational purposes. **This is what happens internally 'under the hood'**:

<u>Step 1 create a cube.</u>

This ensures texture UV mappings are consistent and non-polar with the flexibility to apply any amount of smoothing without a great amount of texture distortion.

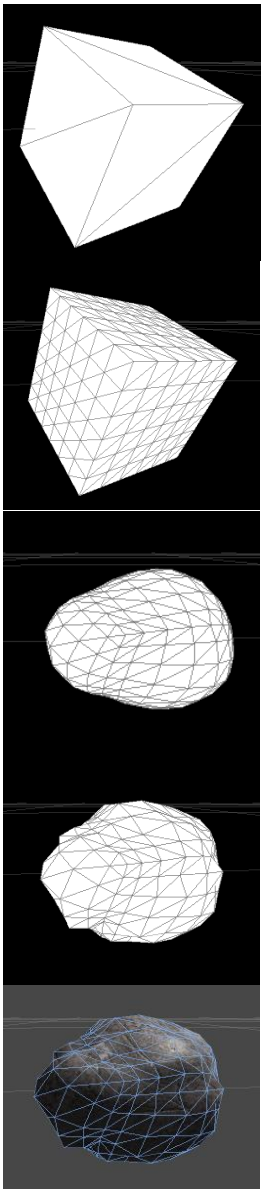
<u>Step 2 subdivide the cube.</u>
Generate the topology to apply a displacement map to. There are currently three different systems for creating geometry.

Note: The most basic type 'SlicedEqually' is used in the illustrations on the right.

<u>Step 3 deform the cube.</u>
Apply deformation like spherize and taper to smooth and alter the overall shape of the cube to morph into a deformed sphere.

<u>Step 4 apply displacement.</u>
Generate a displacement map and apply it to all vertices.

<u>Step 5 optimize new mesh.</u>
Recalculate all normals, tangents. Smooth and merge all vertices that share a normal or sharpen any edge that exceeds a given threshold. It will also generate an edge angle encoded map for the Rock Factory shader to be used to render the 'edge wear'-effect.



3 RFRock component

3.1 Overview

The RFRock class, a behaviour class, is the main component for generating rocks.

There are two modes for editing the parameters: basic and advanced mode.

Basic mode is meant for every person that is using Rock Factory for the first time and for basic usage.

Advanced mode is meant when artistic freedom is needed at some point. Where basic mode doesn't expose all parameters, advanced mode does. It is also meant for everyone that is comfortable with the basic parameters and knows their way around.

Basic mode uses all systems and all parameters of Rock Factory. The quality of rocks is not different from the ones produced in advanced mode. Advanced mode only brings more flexibility. For example, you can override global parameters for each side of the rock in advanced mode.

Note: To cover all parts in one go, we will continue using Rock Factory in advanced mode.

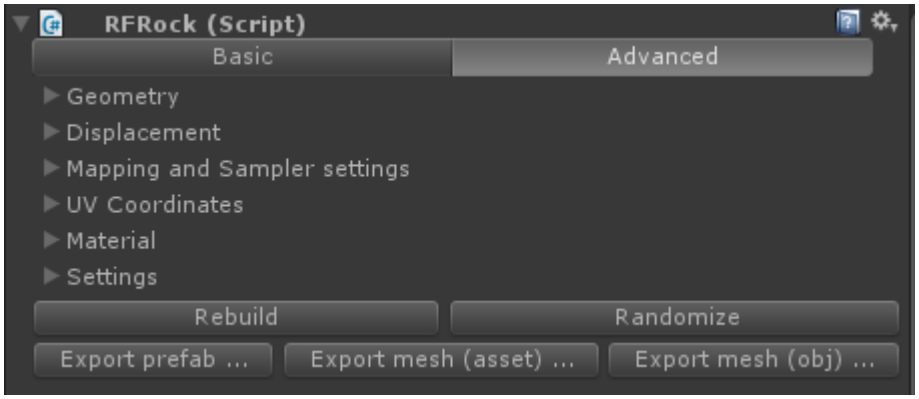


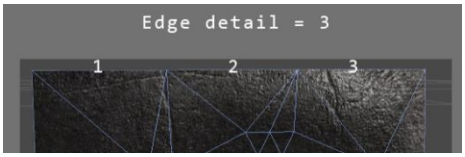
Fig. 3.1.1: Overview of all RFRock class inspector parts

3.2 Geometry

The geometry section of Rock Factory defines the method of creating topology by generating and connecting vertices into a solid mesh of connected triangles.

The scale applied to the local coordinates of each vertex can be specified in the '***Size***' parameter. It will not affect the transform scale, which can be left at (1,1,1) for optimal performance. If you want to alter the scale at runtime for non-static rocks, using the transform scale will give you better performance as no vertices are changed inside the mesh.

Rock Factory currently contains three systems for geometry creation. Each system will have an option to enable/disable a side. Disabling a side will completely ignore that side during the entire flow inside the Rock Factory engine.



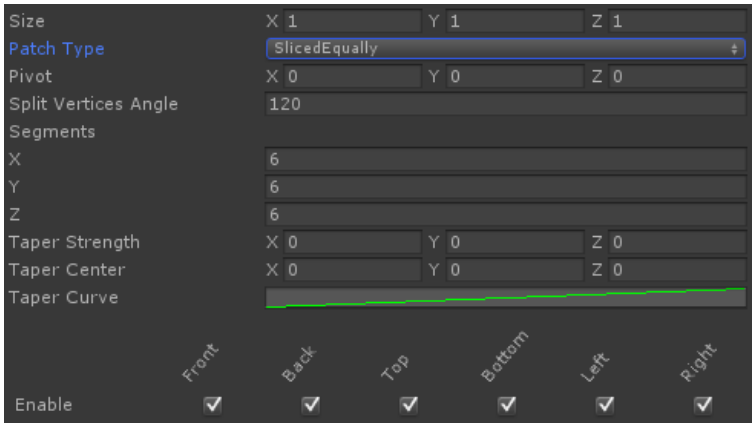
There is a setting for specifying a fixed amount of vertices at the outer edges of each face of the cube. It's fixed amount because these vertices will be used to connect sides and properly align and match UV coordinates. It's called '***Segments***' for 'Sliced Equally' patch type (because it also defines the number of segments when the cube is subdivided) or '***Edge detail***' for all other types.

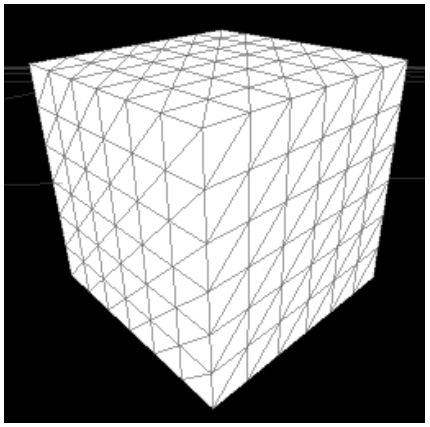
	Complexity	Quality	Uses height map
Sliced Equally	*	*	no
Delaunay	***	**	no
Delaunay by Height	*****	*****	yes

3.2.1 Patch types

Sliced Equally

Subdivide the cube faces into equally sized grid patches. You can specify the amount of patches along an axis with the 'Segments' parameter.



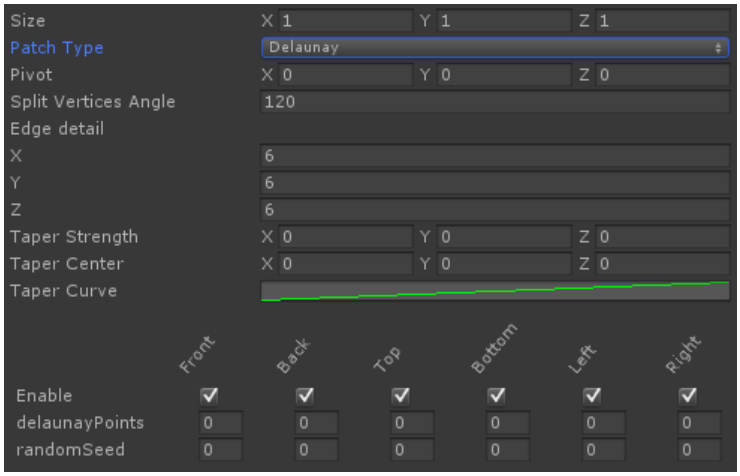


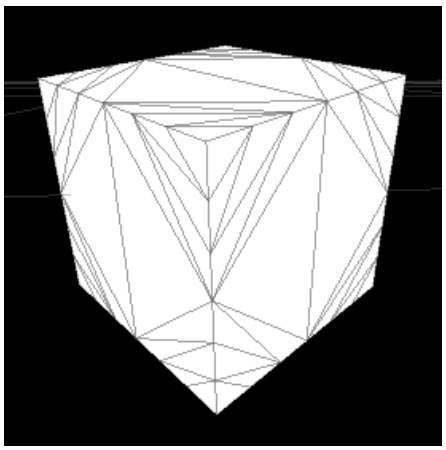
Delaunay

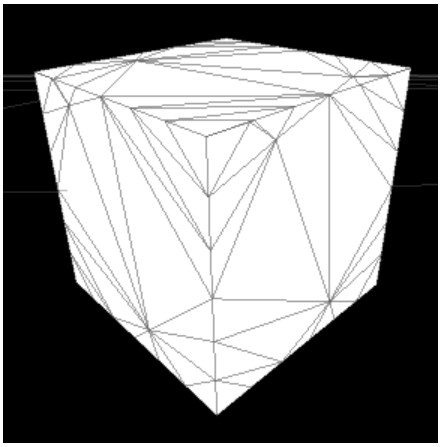
By placing random points and applying a triangulation algorithm based on a Delaunay graph, a more natural pattern can be reached.

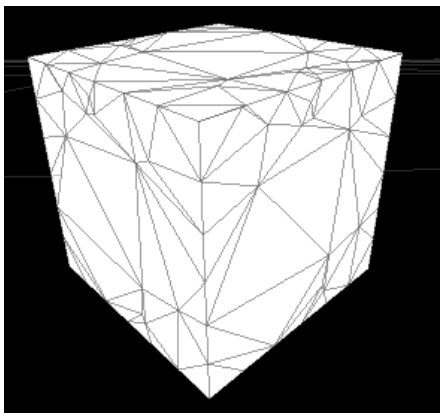
In mathematics and computational geometry, a Delaunay triangulation for a set P of points in a plane is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation; they tend to avoid skinny triangles. The triangulation is named after Boris Delaunay for his work on this topic from 1934.

An illustration of what happens when you add more delaunay points to the sides:





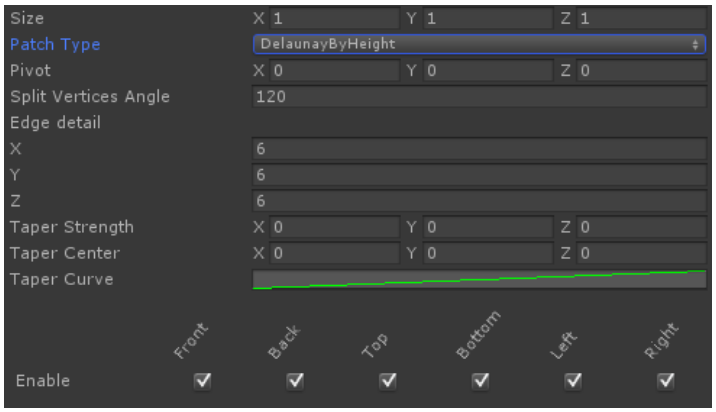


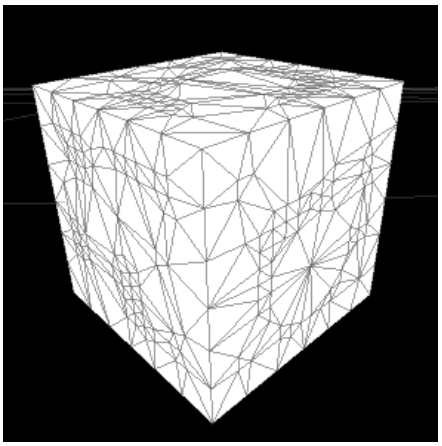


Delaunay by Height

This patch type uses the height (displacement) map to generate vertices. It will follow a given contour line (line that connects points of equal height) using the mapping and sample settings.

We will go more in-depth later on in chapter Mapping and Sampler settings.





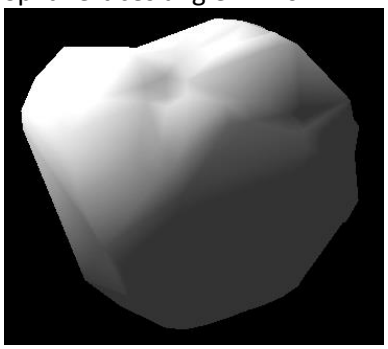
3.2.2 Pivot

Change the center point of rotation. It will offset the local vertex coordinates and will not have any impact on performance.

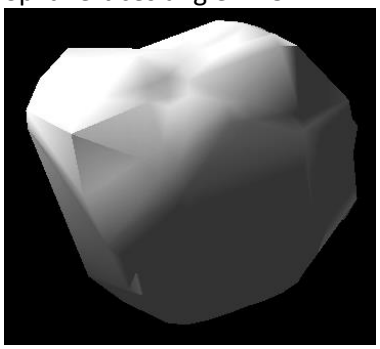
3.2.3 Split vertices angle

Create hard (sharp) edges when the angle between two edges exceeds this threshold.

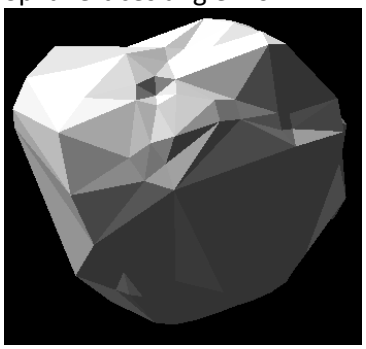
Split vertices angle = 120



Split vertices angle = 75



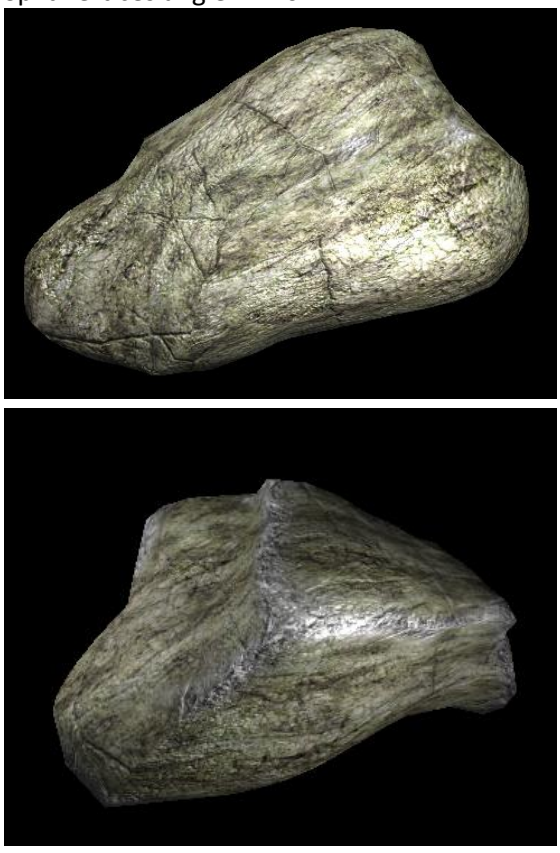
Split vertices angle = 0



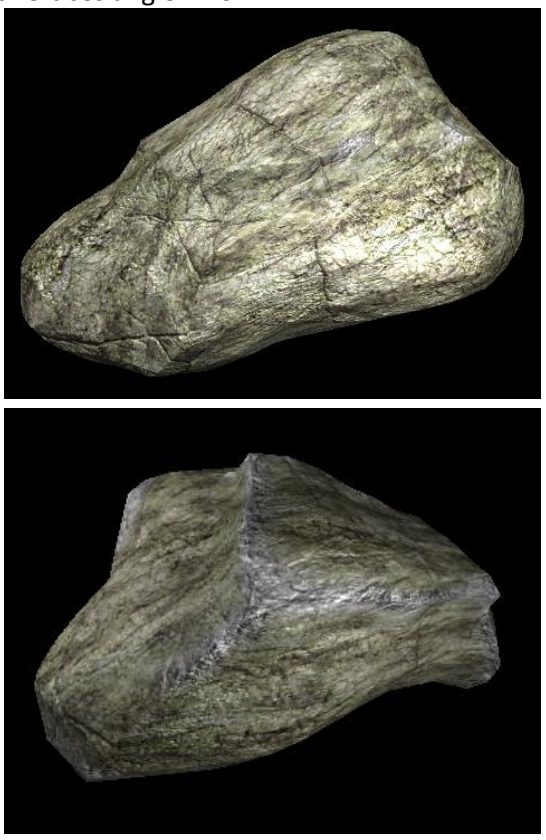
Use a good balance for a realistic rock by emphasizing sharp younger rock edges and smoothen out the older worn edges. Choose a high value to hide low poly rocks by smoothing them out. Or use a low value when you're looking for that low-poly retro style rock!

On smooth geometry rocks it's also great for adding more contrast and depth at micro level, because the light reflection angle is more aggresive:

Split vertices angle = 120



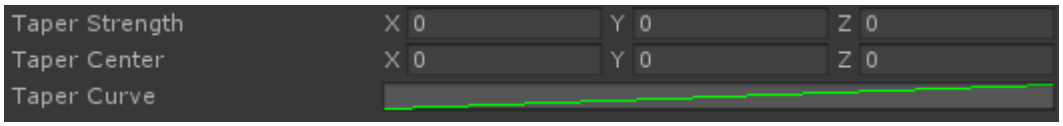
Split vertices angle = 40



3.2.5 Taper deformation

Taper deformation is the type of deformation where you scale vertices linearly along a given axis.

In our case it will deform the basic shape of the cube, before it gets displaced, allowing us to create any shape we want.



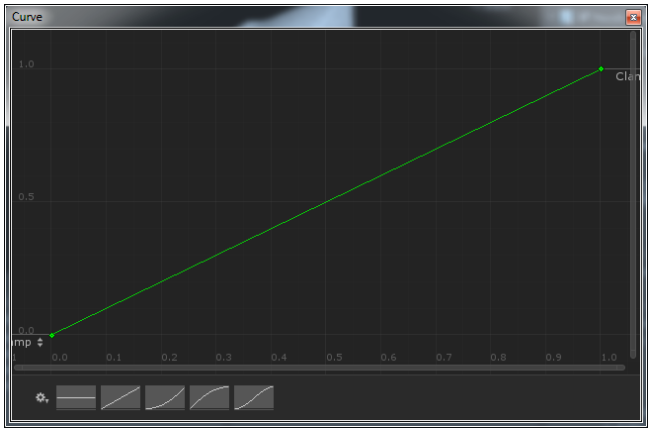
In advanced mode the following parameters can be used for deformation:

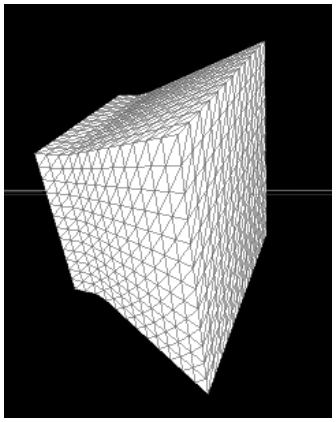
Taper Strength: amount of deformation. Can be negative to scale down, positive to scale up along the axis.

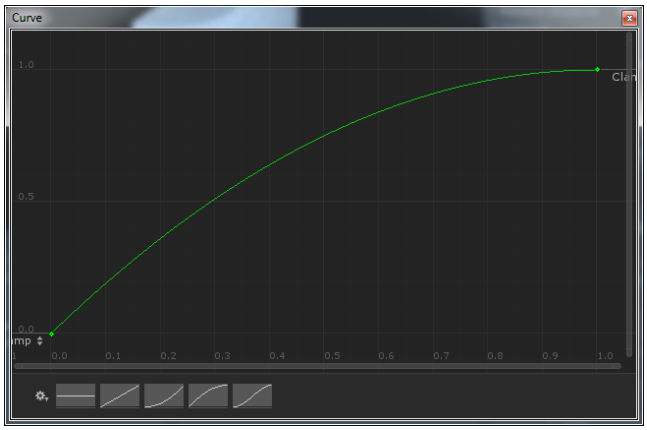
Taper Center: Shift the center from or towards the center of the mesh. 0 means to scale starting from the center.

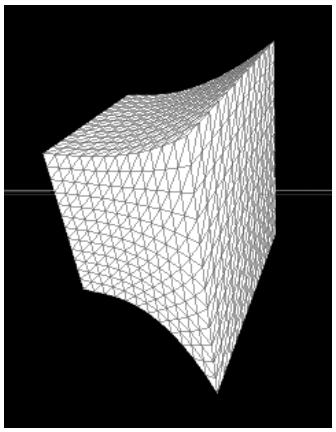
Taper Curve: Shape the curve of our scale to non-linearly scale, allowing you to actually draw the shape of the deformation.

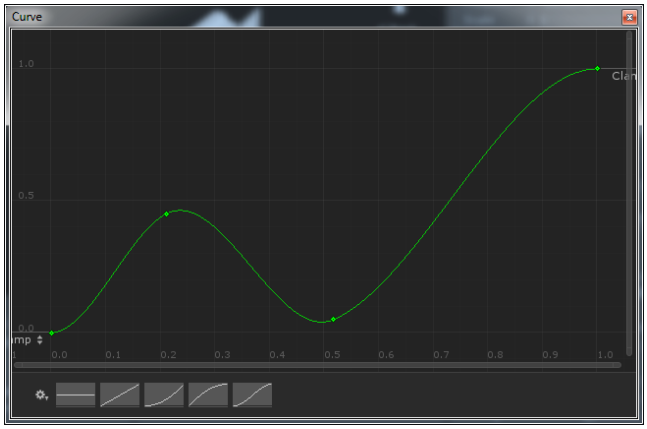
An example of what curves are capable of:

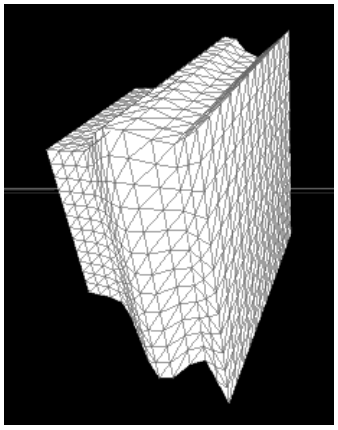




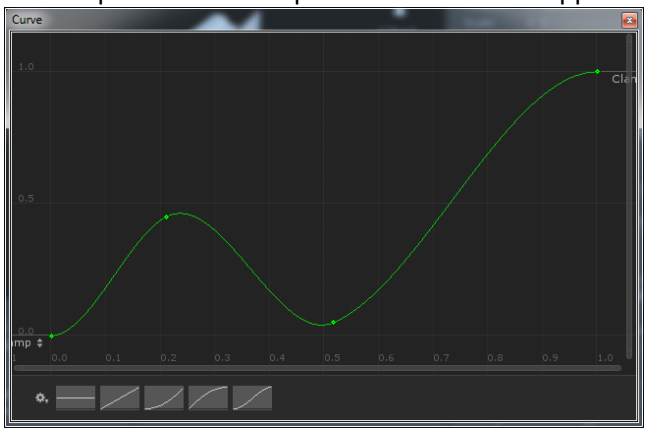








with displacement and spherize deformation applied:

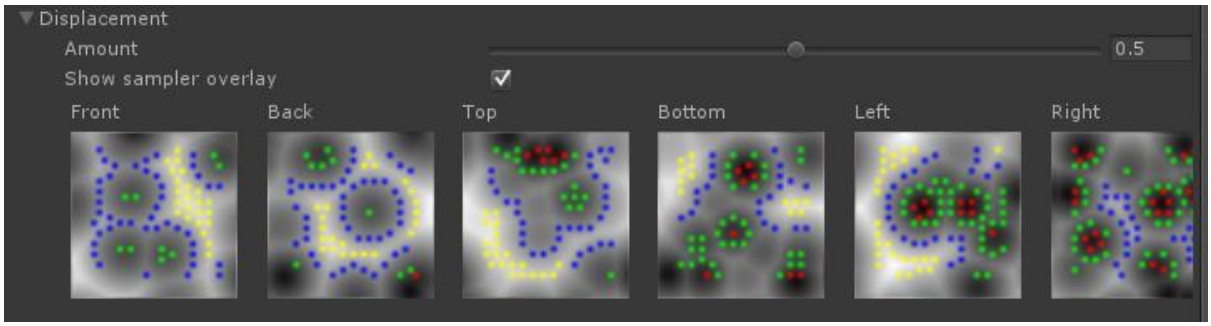


3.3 Displacement

Rock Factory generates a texture where each pixel has a greyscale value from black 0 to white 255. The value of each pixel specifies the amount of displacement along the normal vector for each vertex. Vertices are mapped in normalized UV space (0 to 1) per side.

The height is multiplied by the amount slider. You can use the slider to specify the amount of displacement a rock has in general. Use height scale in the section 'Mapping and sampler settings' to pre multiply a given side.

When using patch type 'Delaunay by Height', the vertices of the geometry are generated based on the pattern of the displacement map. This can be visualized by the '***Show sampler overlay***' checkbox. *For more details see 'Mapping and sampler settings'.*



3.4 Mapping and Sampler settings

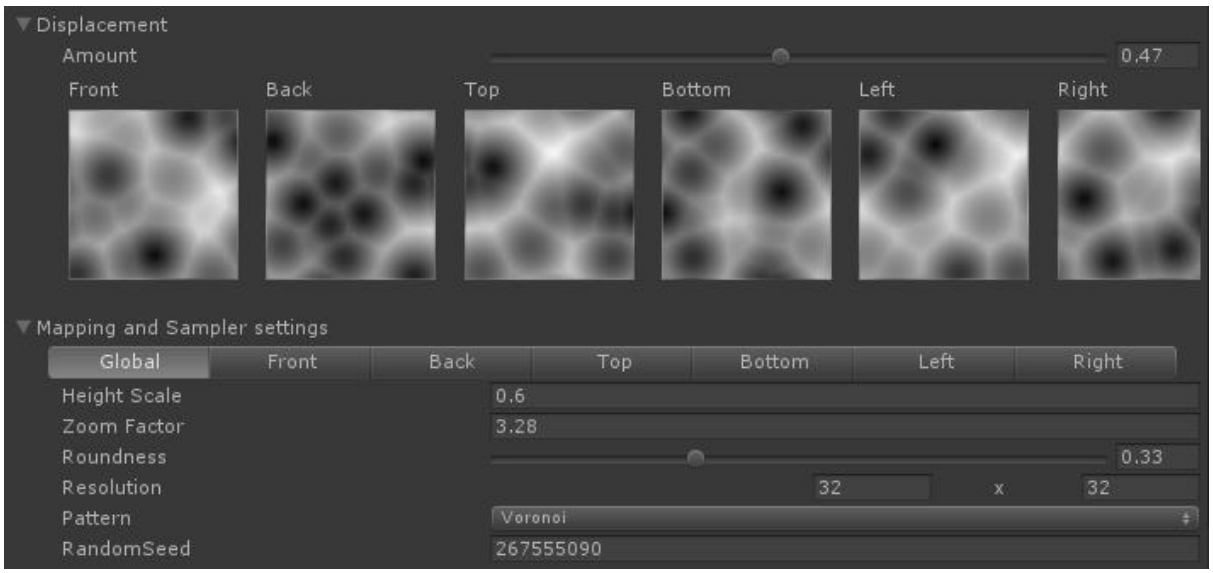
When geometry is created, each vertex has a distance from the center of the cube. This we call the 'height' of the vertex. The height is calculated from the value sampled from a height map (a.k.a. displacement map) generated in the displacement section.

In computer graphics, a height map or height field is a raster image used to store values, such as surface elevation data, for display in 3D computer graphics. A height map can be used in bump mapping to calculate where this 3D data would create shadow in a material, in displacement mapping to displace the actual geometric position of points over the textured surface.

The section 'Mapping and Sampler settings' covers all parameters used to generate the height map, sample the height and apply per side deformation on the vertices.

The way of sampling differs between the patch types. This is because the 'Delaunay by Height'-method uses the sampling to generate vertices based on specific rules, where the other two methods use the height map only as lookup reference for displacement.

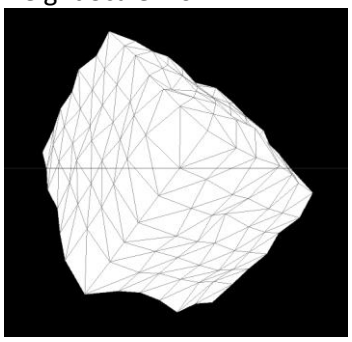
In advanced mode the 'Mapping and Sampler settings' shows a row of buttons to switch sides. You can override the global settings for each side individually, creating even more distinct rocks.



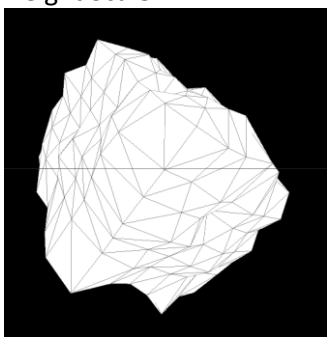
3.4.1 Generic settings

Height Scale is the multiplier to multiply the sampled height by.

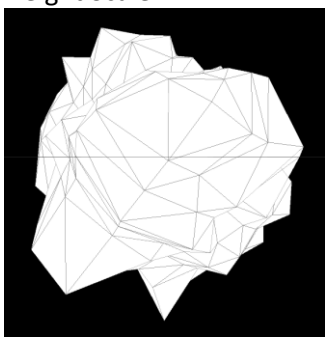
Height Scale = 0.1



Height Scale = 1

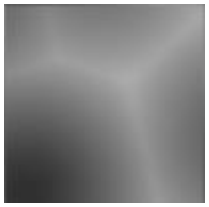


Height Scale = 2

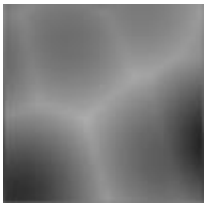


Zoom Factor specifies the scale of the generated height map.

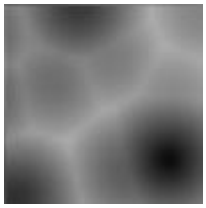
Zoom Factor = 1



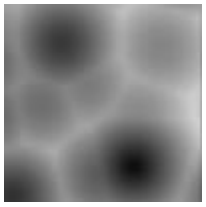
Zoom Factor = 1.5



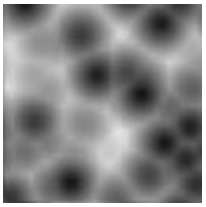
Zoom Factor = 2



Zoom Factor = 2.5

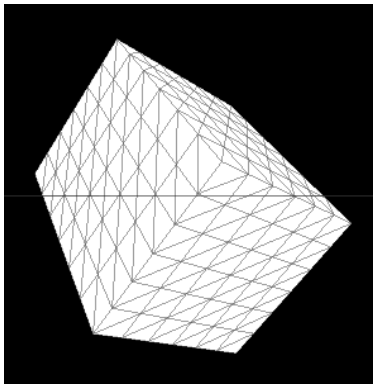


Zoom Factor = 5

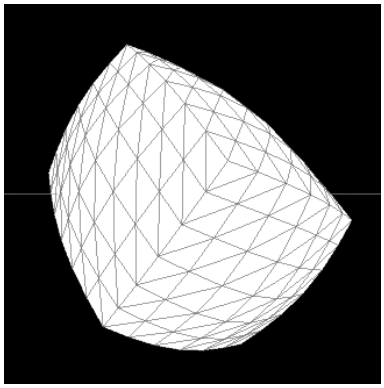


Roundness normalizes the vector offsets from the origin, making the vertices more spherical.

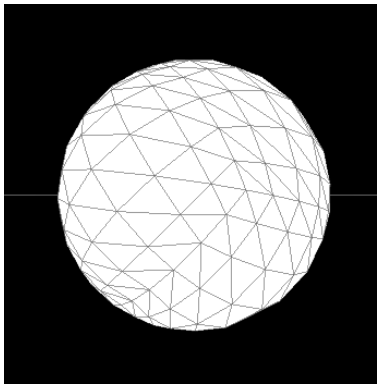
Roundness = 0



Roundness = 0.5

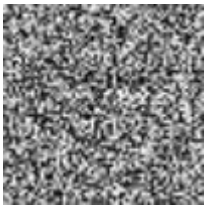
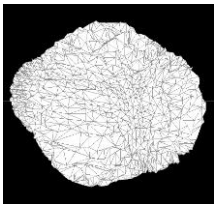
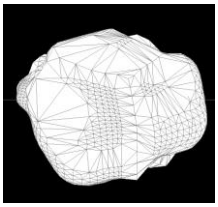
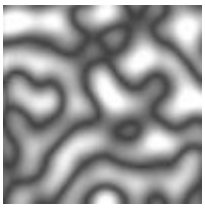
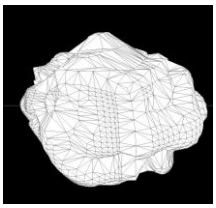
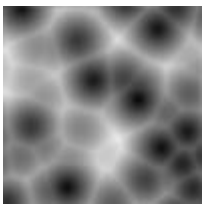
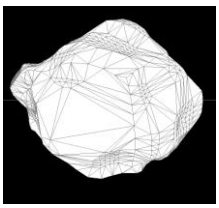


Roundness = 1



Resolution is the size in pixels used for the height map. This can be any value you'd like and doesn't have to be a power of two and can be any ratio desired. The internal sampling however will treat the texture data as a filtered square and will sample the interpolated texel, not the pixel. Higher values have more detail, but take longer to process when rebuilding a rock. *Rebuilding will only happen at design time so there will be no drawback at runtime.*

Pattern is the type of noise generated.

Noise	Random white noise. Use it only for sharp spikey surfaces.			
Perlin noise	Perlin noise is a computer-generated visual effect developed by Ken Perlin, who won an Academy Award for Technical Achievement for discovering the algorithm. It can be used to simulate elements from nature, and is especially useful in circumstances where computer memory is limited. Great for smooth river pebbles and old rocks.			
Billow	Usable for all different kinds of rock. With an almost canyon like difference between heights.			
Voronoi	Usable for all different kinds of rock. Having faces of different elevation. Great for sandstone.			

Random Seed A random seed (or seed state, or just seed) is a number (or vector) used to initialize the pseudorandom number generator of Rock Factory. It uses its own generators so it's perfectly safe to use Unity's internal generator for other purposes without influencing the one from Rock Factory.

When you have generated an awesome rock, write down this value to restore the rock at a later time. You can restore the rock, when all other settings are left alone, by setting this value.

A nice workflow might be: create some promising rock. Keep hitting the randomize button and write down the 'Random Seed' values for the rocks you really like. *Keep in mind the side overrides have their own 'Random Seed' parameters. If you override a side, make sure to write down those 'Random Seed' values too.*

3.4.2 Delaunay by Height

As mentioned earlier in the geometry patch type paragraph of chapter 3.2, the Delaunay by Height type will create geometry based on some specific rules by analyzing the height map.

Brief overview of the whole process:

- First you define contour lines.
- The engine will create point markers for all contour lines.
    - The distance between the points, per contour line, will be defined by the distance parameter.
    - The minimal distance to the border will be defined by the border parameter.

Keep in mind that the more points you create, the more vertices there are, the higher the polycount will be. With that knowledge, you can also work the other way around. Because you know all contour lines are responsible for registering heights and are responsible for the amount of triangles created in the end, we can eliminate lines that have the least influence on steepness (are closest together) to create a lower detailed version. Cloning the rock and disabling the desired contour lines will provide you with the meshes to be put in a LOD group.

See chapter 6 Tips & Tricks > 6.1 Level Of Detail for a practical example.

Contour lines

A contour line (also isoline, isopleth or isarithm) of a function of two variables is a curve along which the function has a constant value. In physical geography and cartography it's a line that connect points of equal height.

You can define a contour line by using the sliders inside the contour line array. The array can be expanded by sliding the parameter 'Number of lines'.

In the image above there are two contour lines defined. The colors will correspond to the colors in the sample overlay within the height map preview images inside the displacement section. Each contour line slider has a start and end height value. The start height value of contour line 0 is 0. This means the contour line will accept all heights from 0 (all black in a height map). The end height value is somewhere around 0.2. This will be the last accepted height for this contour line.

Creating vertices for every point that qualifies for these values will generate a vertex for each texel sampled. This will result in incredibly dense high poly models, which is not the desired result. Therefore we specify the Distance parameter.

Distance (0 to 1) will define the minimal distance in texture space between points of equal height.

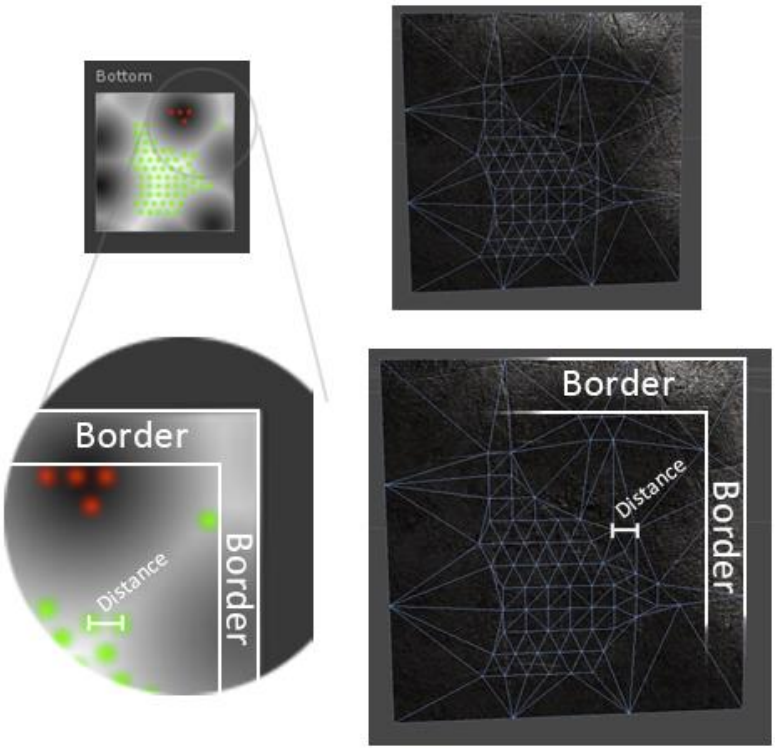
In the example below there is a minimal distance of 0.05 between all green and red dots. This means a distance of 5% of the total width or height from the texture, which is 1 (100%).

Border (0 to 1) will define the minimal distance to the edge of the height map.

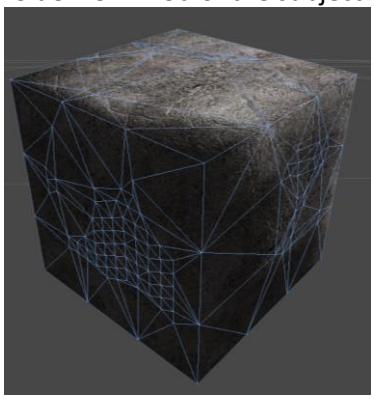
In the example below there is a minimal distance of 0.1 between the edge and the colored dots. This means a distance of 10% of the total width or height from the texture, which is 1 (100%). Make sure you keep a good distance from the border with steep elevation changes to avoid texture stretching.

Note: the triangulation process might add or remove vertices to maintain a valid topology.

Height map and corresponding side view of geometry:

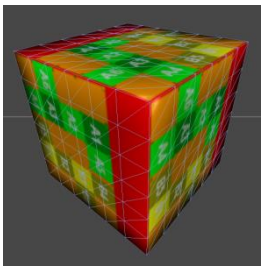


Side view in 3d of the subject:



3.5 UV Coordinates

3.5.1 Mapping



Box mapping uses the entire texture resolution on each side. However all sides show the same texture. Using a natural texture will hide this enough and enabling a detail overlay and edge wear will hide it even more.

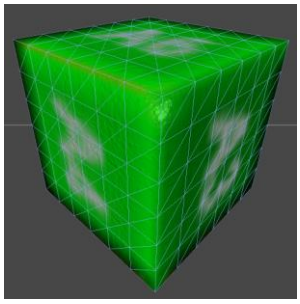
Foldout mapping uses only a part of the texture on each side. This will result into a completely unique texture per side, without extra cost.

It uses a texture with the following unwrap pattern:

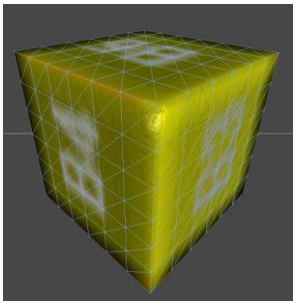
Texture A is the standard choice, texture B is the alternate texture. This way we can have two rocks using the same texture space in video memory.

Use Alternate You can switch between texture A and texture B by checking the 'Use Alternate' checkbox in the inspector.

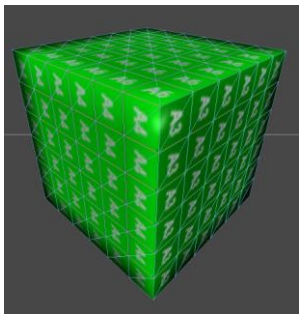
Use Alternate = false



Use Alternate = true



Tile Foldout parameter is still experimental, but can be used to tile the foldout texture.



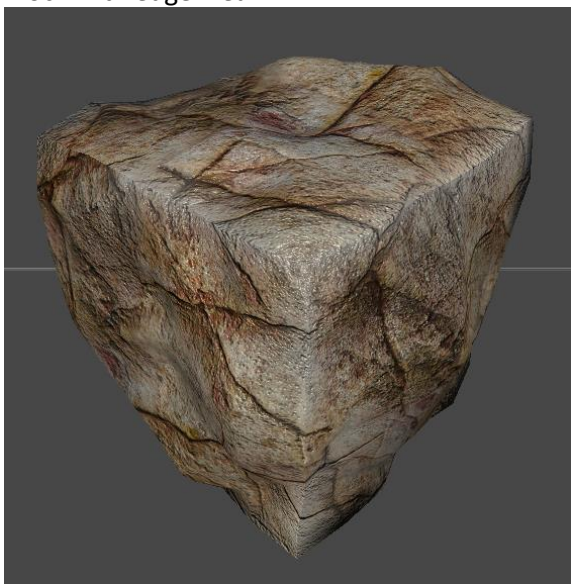
Tiling and ***Offset*** parameters are currently only available for Box type mapping. They provide axis aligned tiling and offset in all three dimensions.

3.6 Material

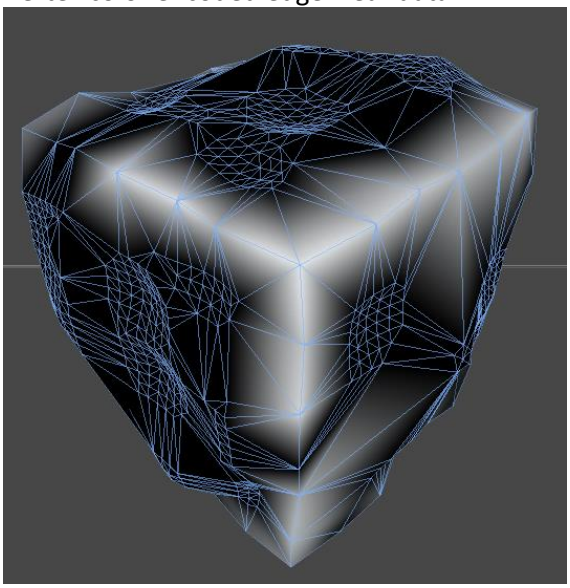
Edge wear is a visual effect where edges are damaged due to wear and erosion. This often results in a different texture than the rest of the rock.

Enable Edge Wear will encode the amount of edge wear to apply into vertex colors which can be used by any shader. The Rock Factory shader also has support for this effect.

Block with edge wear



Vertex color encoded edge wear data



Edge Wear Angle will set the threshold at what angle edge wear should be applied. A higher value means less wear because angles should be sharper before edge wear gets applied.

See chapter 5.6 for more edge wear related information.

3.7 Settings

Max rebuild time (0=infinite) defines the maximum time you allow a rock to be generated. The time is in seconds. When the rebuild exceeds the time limit, the build will be aborted and a message will be displayed in the console window. This is a safety measure to avoid you having to wait too long when you accidentally change a parameter that would result in a too complex rock. You can increase or decrease this value; however it is not recommended to set it to 0.

Auto rebuild will hit the rebuild button for you automatically with every change you make. Usually this will result in an instant update. When your computer is too slow, or the rocks get to complex to handle, uncheck this option and hit the rebuild button yourself.

Debug info will output timings to the console for each subsystem. You can use it to determine what is causing the delay when generating a rock.

Contour settings contain the maximum amount of possible contour lines with their corresponding colors.

Sub Systems allow you to switch off an entire subsystem. Use for debugging purposes only.

3.8 Notification panel

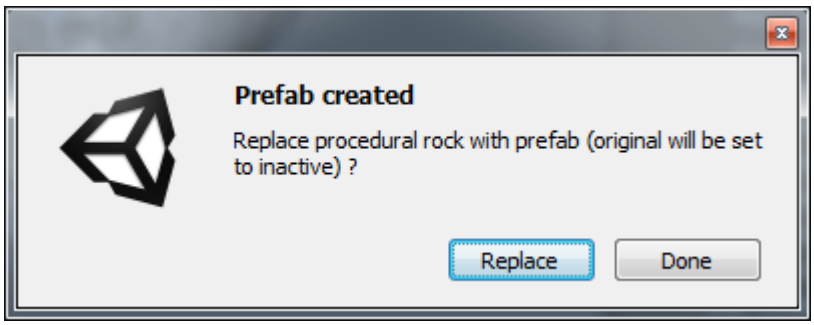
The notification panel will inform you about whether or not dynamic batching is available, the number of vertices and faces created and how long the entire (re)build process took on the last run. It also shows the version number.

4 Export

When you are switched to advanced mode, the button bar will show you three types of export.

Export prefab will export a .prefab file that will contain the plain mesh data and the attached material. This will eliminate the need for Rock Factory when your rock is ready. Use this to share finished rocks with your coworkers or friends when they do not own Rock Factory.

After exporting to prefab it will ask if you want to replace the rock with the 'baked' rock.



If you choose to replace, the original rock will be kept in your hierarchy but will be disabled.

Export mesh (asset) will export a unity mesh as an asset file. This will contain only mesh data (vertices, faces, normals, vertex colors, etc) and no trace of Rock Factory. It will also discard connected materials. Use 'Export prefab' if you want to preserve the entire setup.

Export mesh (obj) will export a text based Wavefront mesh object file.

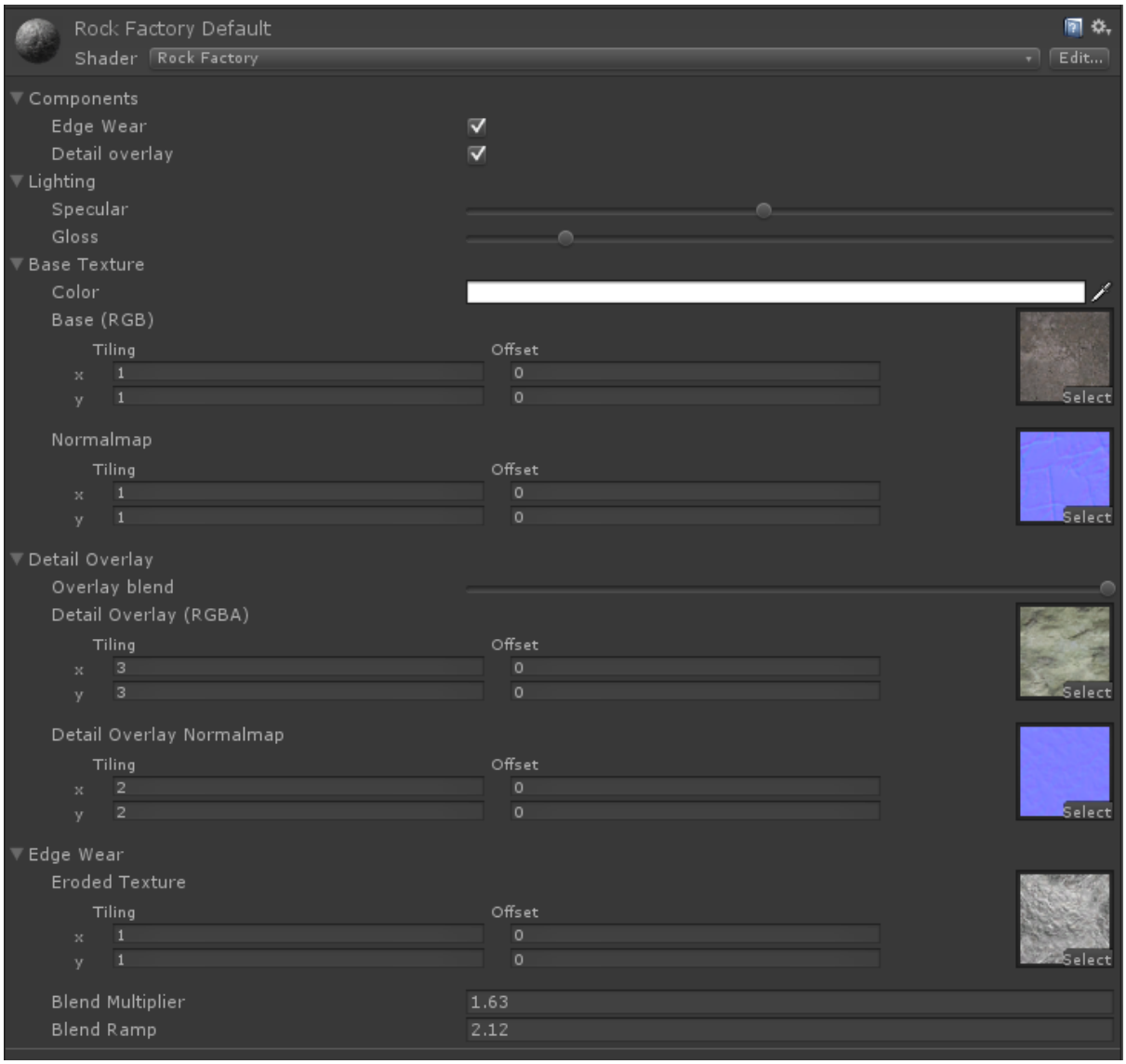
5 Rock Factory shader

5.1 Overview

Rock Factory comes with an über shader. Über shaders are rendering programs that can handle many different rendering scenarios. In our case it supports edge wear and detail overlay.

If you uncheck any of the components an entirely different shader will be used where all source of that particular component is left out of compilation, meaning there is zero overhead.

It is not necessary to use the Rock Factory shader. However it might be the only shader that combines all necessary components for rendering rocks into one switchable shader at this moment in time.



5.2 Components

Edge Wear checkbox enables or disables the edge wear rendering system.

Detail overlay checkbox enables or disables the detail overlay rendering system.

5.3 Lighting

Specular is the amount of highlight reflection.

Gloss defines the size of the highlight.

5.4 Base Texture

Color is a color tint that will be applied to the rock.

Base (RGB) is the base texture.

Normalmap is the bump map of the base texture.

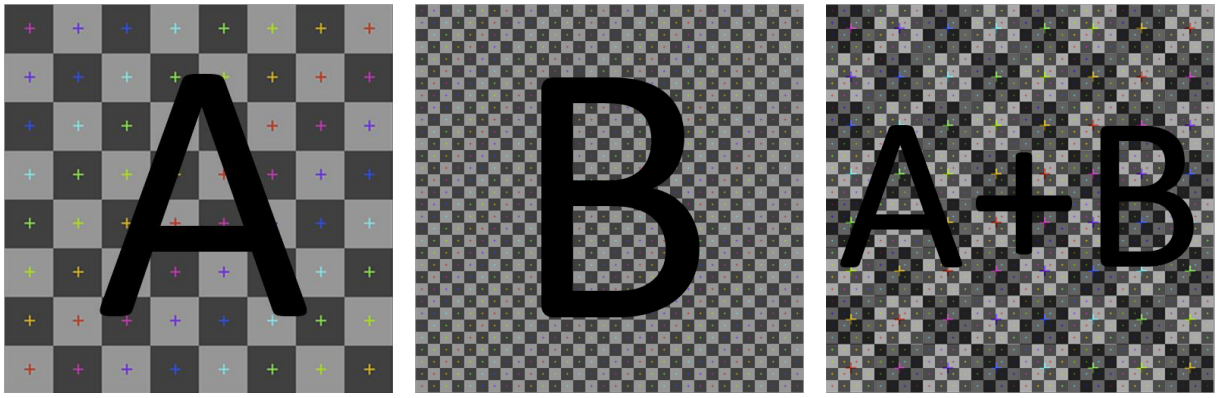
5.5 Detail Overlay

A detail texture is used to cover up a low resolution base texture. Usually the base texture holds the low resolution base structure and the detail overlay is a tiled texture which will be blended over the base texture.

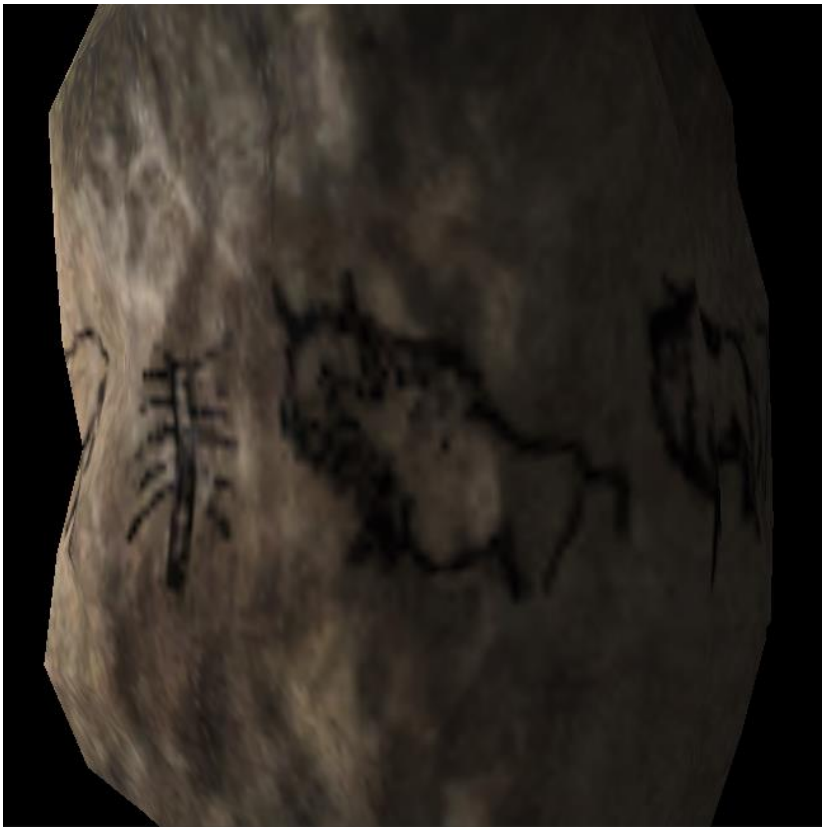
It uses the common overlay blend mode; where light in the base texture becomes lighter and dark becomes darker.

$$f(a,b) = \begin{cases} 2ab, & \text{if } a < 0.5 \\ 1 - 2(1-a)(1-b), & \text{otherwise} \end{cases}$$

Where ***a*** is the base texture and ***b*** is the detail overlay texture. They can both remain low res textures, because the second texture is tiled to create the detail.



Base foldout texture
no detail overlay



Base foldout texture
Detail overlay 50%
Tiled 5x5

Base foldout texture
Detail overlay 100%
Tiled 5x5

5.6 Edge Wear

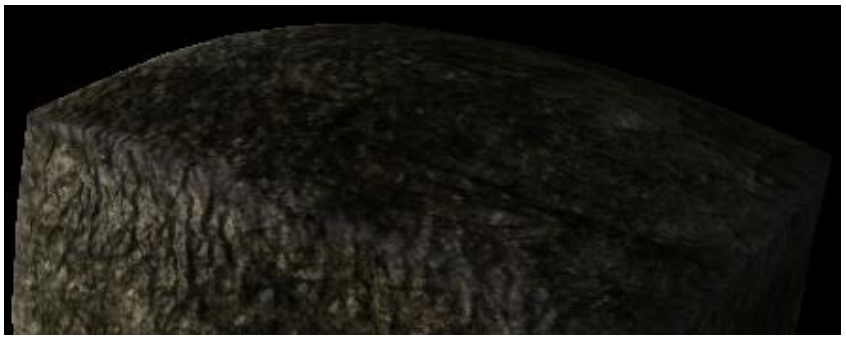
As mentioned in chapter 3.6, edge wear mimics the damaged edges of a rock. The material part of the engine encodes all angles into vertex colors which are passed to the shader.

The shader blends the ***Eroded Texture*** and calculates the blend factor and blend ramp according to the ***Blend Multiplier*** and ***Blend Ramp*** material parameters.

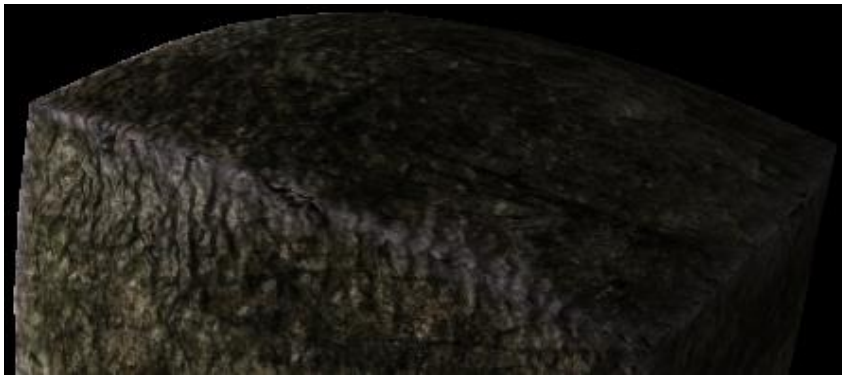
Blend Multiplier is responsible for the amount of transparency for the edge wear texture layer. It's a multiplier and can be any value.

Notice the edge getting lighter and more distinct as the blend multiplier increases.

Blend Multiplier 0.5
Blend Ramp 2



Blend Multiplier 1
Blend Ramp 2

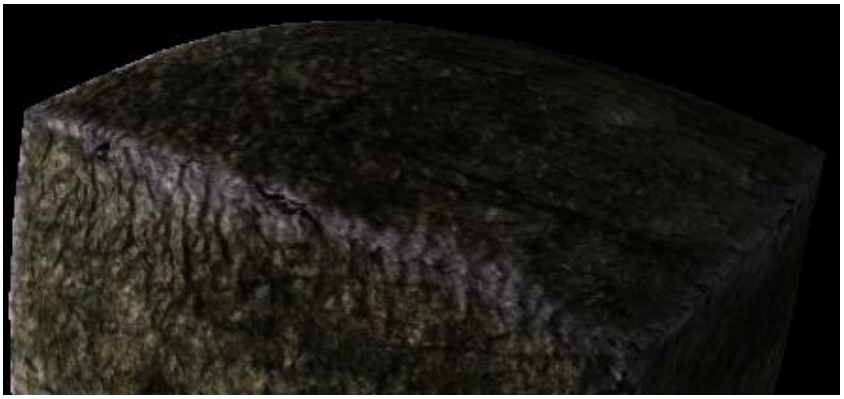


Blend Multiplier 2
Blend Ramp 2

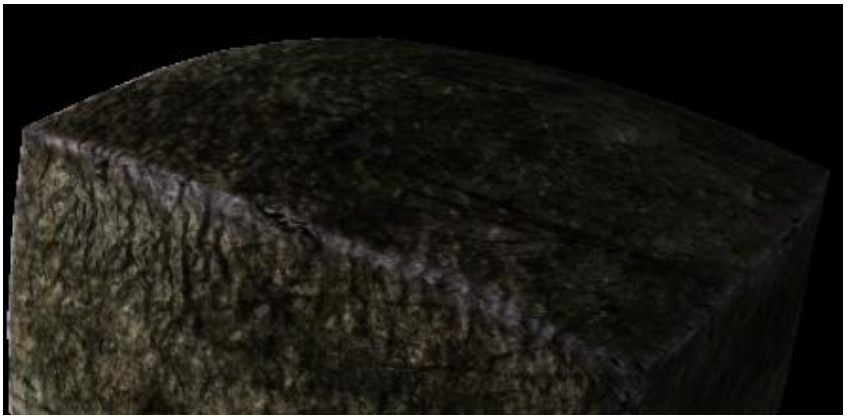
Blend Ramp is used as exponent to calculate the ramp curve of the edge wear. The weight factor calculated and encoded in the vertex colors are smoothly interpolated over each face. Sometimes it's desired to harden the edges of the edge wear layer and choose not to fade over the entire face, but end earlier (higher value) or later on (lower value).

Blend Multiplier 2
Blend Ramp 0.5

Blend Multiplier 2
Blend Ramp 3



Blend Multiplier 2
Blend Ramp 8



6 Tips & Tricks

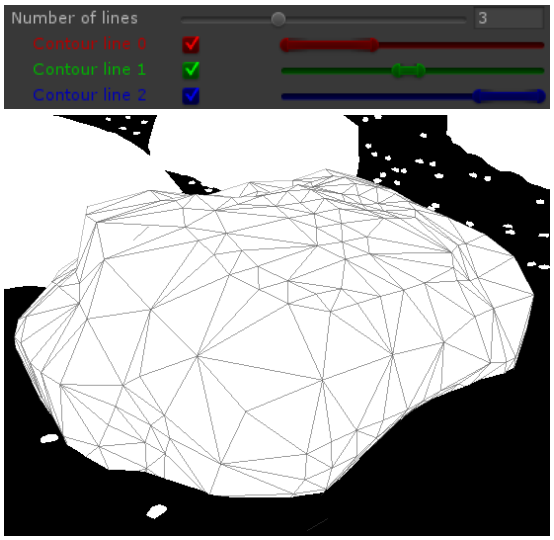
6.1 Level Of Detail

In computer graphics, accounting for level of detail involves decreasing the complexity of a 3D object representation as it moves away from the viewer or according to other metrics such as object importance, viewpoint-relative speed or position. Level of detail techniques increases the efficiency of rendering by decreasing the workload on graphics pipeline stages, usually vertex transformations. The reduced visual quality of the model is often unnoticed because of the small effect on object appearance when distant or moving fast.

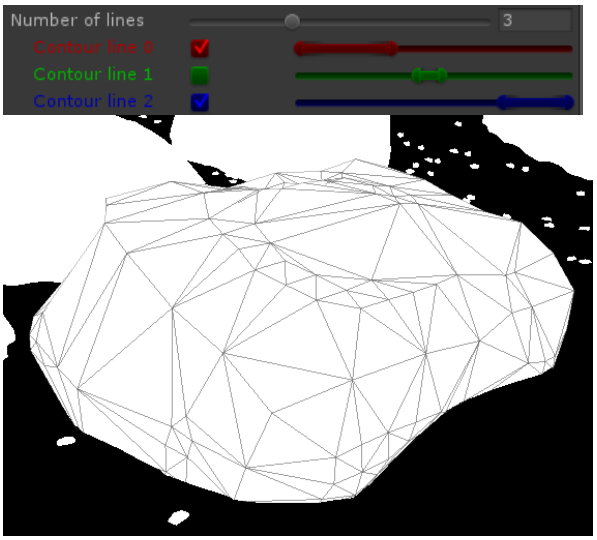
Level of detail should maintain the basic shape of a mesh at all level. For a rock it means keeping the largest amount of elevation changes between vertices. Since the contour lines in Rock Factory are connecting points of equal height and are sorted from lowest to highest points, we can easily disable any amount of contour lines between the first and the last one to eliminate unnecessary detail.

There are alternative ways like decreasing the sample range of each contour line or widen the distance between sample points. But disabling contour lines gives great results and is incredible easy to do.

LOD0: 962 faces



LOD1: 590 faces



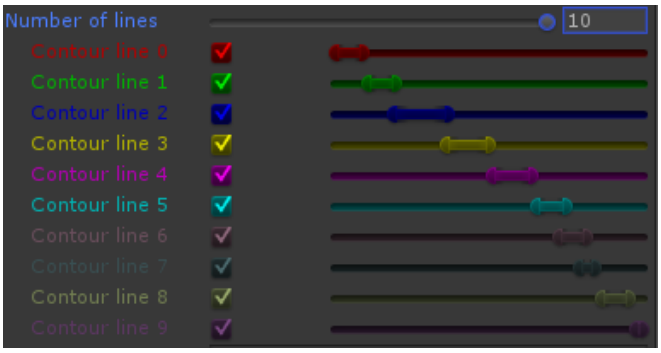
The basic workflow for LOD groups and Rock Factory is:

- Create an awesome rock
- Duplicate the rock
- Disable the contour lines (never the first or last)
- Throw them straight in a LOD group this way and you're done!

Have a look at the included Rocky Lake example to see the LOD groups in action.

If you want to create more LOD levels. You can add more contour lines and disable them for example like this:

LOD 0



LOD 1

LOD 2

LOD 3

Index

NOTES