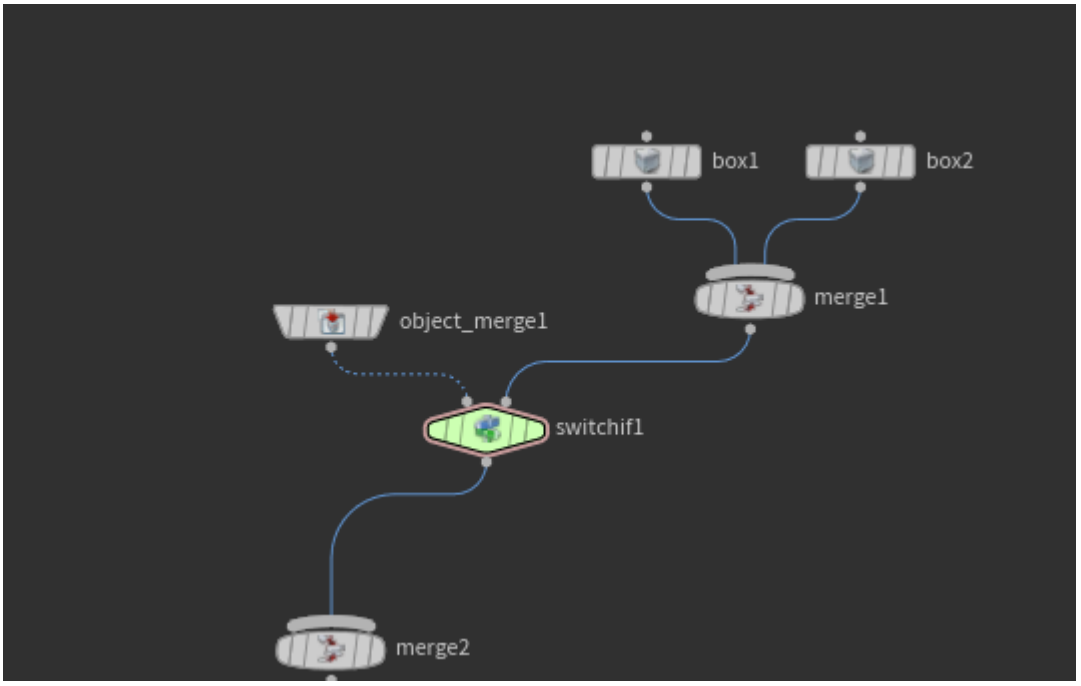
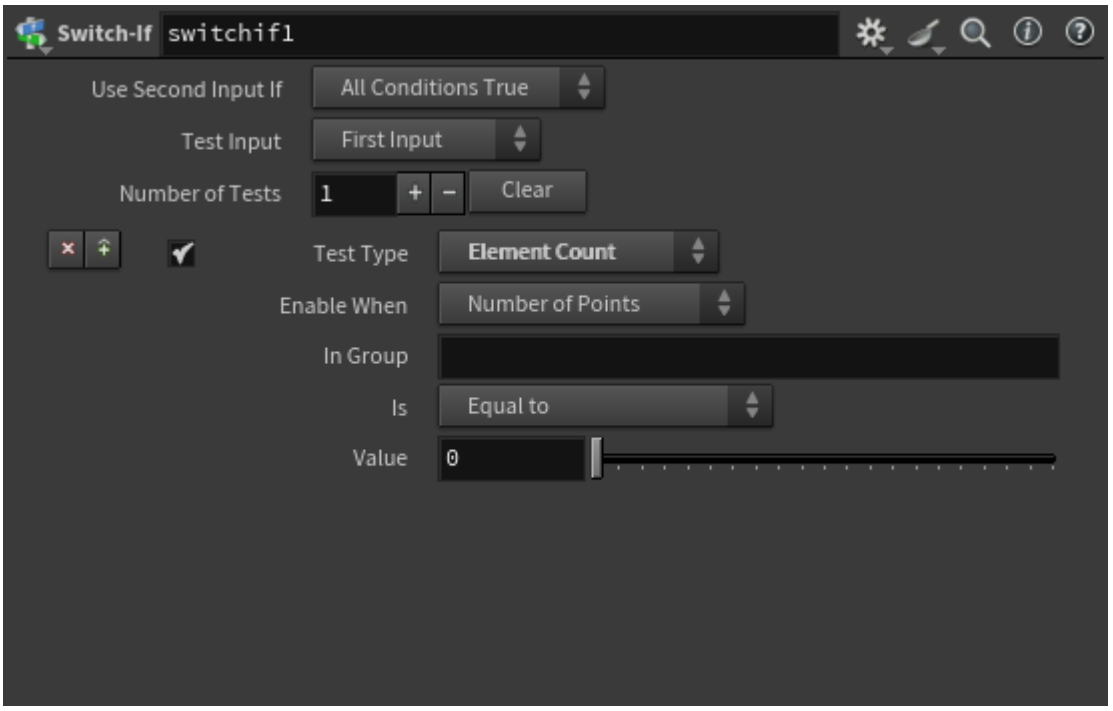
Platform Maker



首先利用box做出一个基本的平台形状当做制作HDA的参考，作为预览的效果，然后在利用switchif节点来判断一下object merge里面的输出是否为空，如果为空就输出预览的结果，如果不为空就以object_merge作为input。

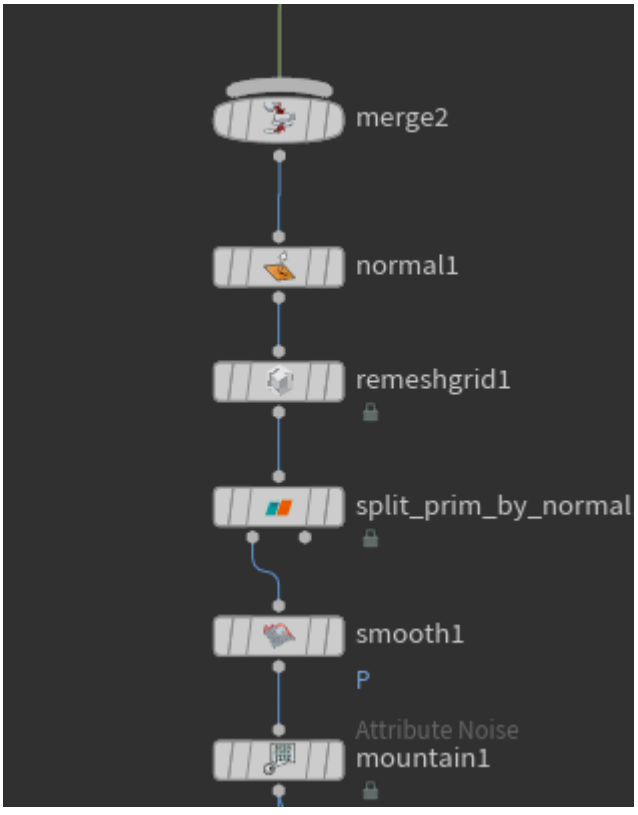
#Switchifsetting



Element count 元素数量

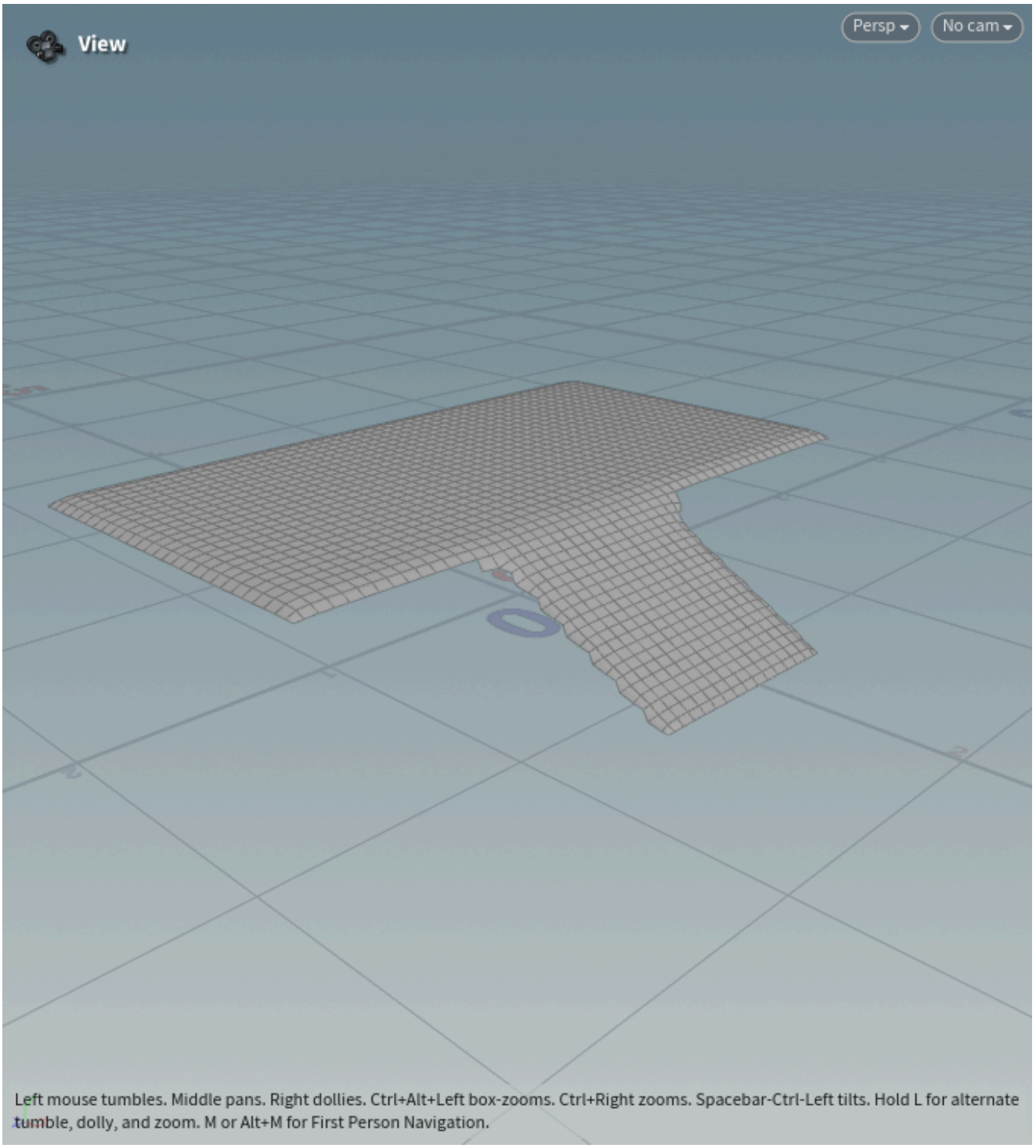
Tests based on the number of components (points, primitives, vertices) in the geometry. Configure this test using the **Enable when**, **In group**, **Is**, and **Value** parameters.

根据几何体中的组件数量（点、图元、顶点）进行测试。使用“启用时”、“在组中”、“是”和“值”参数配置此测试。



然后连了normal以及remeshgrid节点，这两个节点的目的就是清理input进来的模型

split_prim_by_normal节点则是根据模型法线的方向进行split，这边的目的就是剪出input进来的模型朝上的那个范围面片



#LabsSplitPrimitivesbyNormal

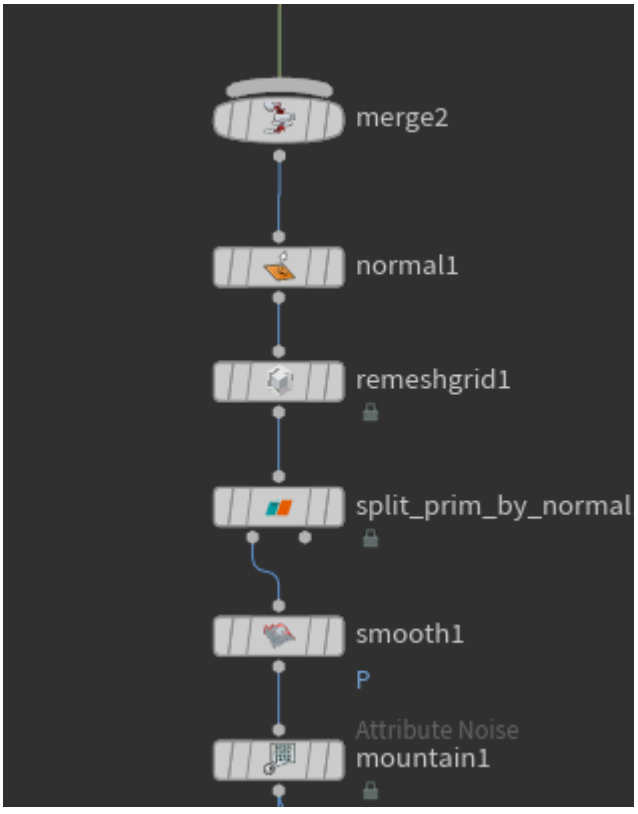
The split primitives by normal looks at the primitive normals of the input geometry, and splits them into separate outputs based on the normals. This is a very useful tool for hard-surface modeling, scattering or even just isolating geometry.

按法线分割图元会查看输入几何体的图元法线，并根据法线将它们分割成单独的输出。这对于硬表面建模、散射，甚至仅仅是隔离几何体来说，都是一个非常有用的工具。

PARAMETERS 参数

Settings 设置

Direction 方向	This allows you to specify the direction of the normal for the primitives that should be split off. 这允许您指定应分离的图元的法线方向。
Invert Selection 反向选择	Inverts the selection generated based on the Direction parameter. 反转根据方向参数生成的选择。
Tolerance 宽容	This allows you to control the deviation from the direction specified in the Direction parameter that is allowed for the splitting. 这使您可以控制允许分割的方向参数中指定的方向的偏差。

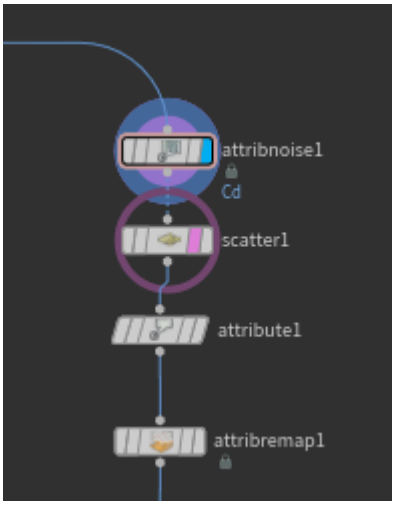


smooth节点就是smooth的效果。
mountain节点的目的便是让面片，也就是让这个platform有一点起伏
在这里的mountain节点其实就是attribute noise节点，然后勾选 Noise along vector 让模型的每个点沿法线方向移动，在配合noise map 就是制作起伏的基本。

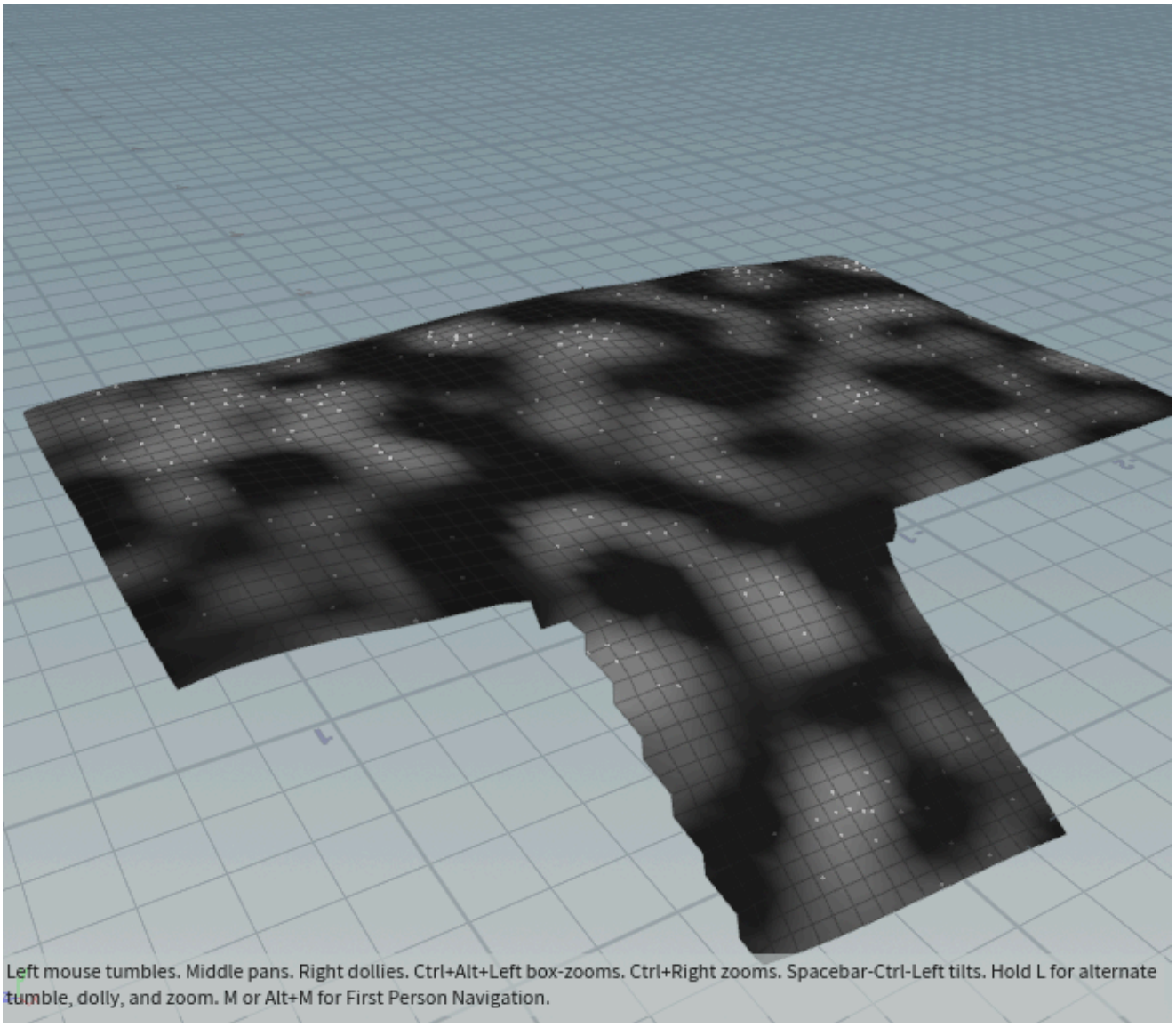
#Mountain

Noise Along Vector 沿向量的噪声	When the attributes are vector attributes, and the checkbox is on, the direction of the noise will be controlled by the **Vector Attribute** parameter and the noise will change the length of this vector. This is useful when you need to displace geometry along it's normal direction. 当属性为矢量属性且复选框处于勾选状态时，噪声的方向将由**矢量属性**参数控制，并且噪声将改变该矢量的长度。当需要沿法线方向移动几何体时，此功能非常有用。
Attribute Names 属性名称	The name of the attribute to generate noise into. The class of attribute (such as point, face, or vertex) is controlled by the **Attribute Class** parameter. If the attribute does not exist, this node will create it. You can specify a space-separated list of names to add noise to multiple attributes at once. 要生成噪声的属性的名称。属性的类别（例如点、面或顶点）由“**属性类别**”参数控制。如果该属性不存在，此节点将创建它。您可以指定以空格分隔的名称列表，以便一次性将噪声添加到多个属性。

平台小石子撒点



整体的思路就是将上面的面片作为input然后做一个attribute noise，然后是根据Cd属性做一个attribute noise，简单来说，就是在这个platform上画区域，利用noise map 随机画一些区域。



再利用scatter节点进行撒点，在白色区域的地方才进行撒点，黑色部分点比较少。
需要注意的是在scatter中需要指定生成点的密度属性，也就是我们刚刚指定的Cd属性进行点的生成。

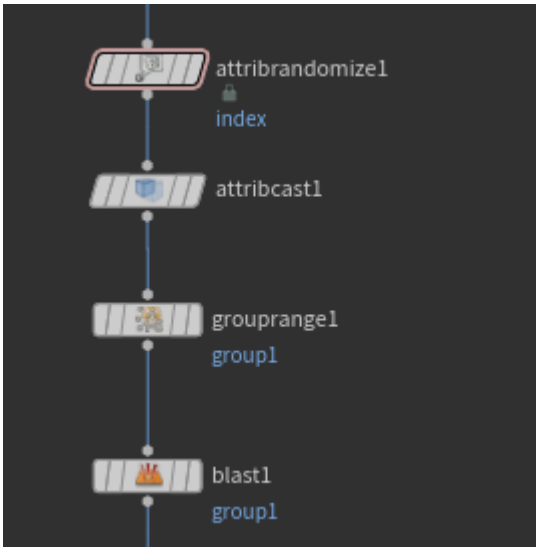
#Scatter

Density Attribute 密度属性	An attribute specifying the density of points to generate, when **Generate** is **By Density**. The attribute can be a point, vertex, primitive, or detail attribute, and in the case of point and vertex attributes, the density will be linearly interpolated over the area of the primitive. This is ideal for use with the Paint node. When **Force Total Count** is enabled, this specifies relative distribution of point density, not actual density. 当 “**生成**” 设置为 “**按密度**” 时，指定要生成的点的密度的属性。该属性可以是点、顶点、图元或细节属性。对于点和顶点属性，密度将在图元区域上进行线性插值。这非常适合与 “绘制”节点一起使用。启用 “**强制总数**” 时，这指定的是点密度的相对分布，而不是实际密度。

然后在利用 attribute rename节点将Cd 转成 pscale ，正好Cd属性的数值其实可以替换成pscale

	P[x]	P[y]	P[z]	Cd	N[x]	N[y]	N[z]
0	-0.432492	0.535216	0.152142	0.18859	0.0705367	0.99731	0.0183979
1	0.641413	0.404156	0.538644	0.56	0.450315	0.889978	-0.0470676
2	1.05191	0.144724	0.274207	0.407236	0.457867	0.875408	-0.154501
3	-1.05144	0.546094	0.103104	0.466822	-0.0922243	0.993668	0.0595575
4	-0.825015	0.510103	-0.764072	0.299225	0.0335624	0.995325	-0.0896218
5	-1.09896	0.462644	-1.48254	0.228451	0.0144787	0.997873	-0.06347
6	-0.493546	0.483627	-1.1365	0.28277	0.0577313	0.99826	0.0023613
7	-0.438586	0.571024	1.06559	0.0724939	-0.138306	0.988431	-0.0601431
8	-0.600203	0.487198	-1.09864	0.435849	0.0308345	0.999429	5.4938e-5
9	-0.0283493	0.501138	0.122099	0.0956477	0.0785449	0.996498	-0.0285512
10	-1.02656	0.492446	-1.11792	0.295221	0.0226629	0.9932	-0.114006
11	-0.743661	0.47895	-1.26108	0.251852	-0.0407769	0.996703	-0.0699598
12	-0.666377	0.492833	-1.00113	0.0845776	0.0633574	0.997229	-0.0378609
13	-0.206427	0.571956	0.86441	0.0488739	-0.0150893	0.994321	-0.10375
14	-0.299725	0.516668	-0.184753	0.191928	0.145823	0.986408	-0.0749408
15	1.09249	0.0797639	-0.018449	0.0373333	0.442677	0.89023	-0.107016
16	-0.5831	0.486564	-1.02195	0.214773	0.0625548	0.997785	-0.00622437
17	-1.15671	0.508066	-0.953607	0.0539727	-0.0576677	0.996768	-0.0501156
18	-0.880877	0.558607	0.00510493	0.317715	-0.0277012	0.996849	0.0738793
19	0.00801441	0.494749	-0.033985	0.332457	0.0814872	0.995746	-0.0406538
20	-1.08173	0.529977	-0.533649	0.256191	-0.0561975	0.994267	-0.0902006
21	0.067931	0.451369	-1.61492	0.551312	-0.0485348	0.992665	-0.109696

由图可见，在spreadsheet中，Cd属性的数值其实挺符合我们预期的pscale值的，所以就直接利用上，不用再多创建一个属性了，想进行一些控制的话，便再利用 attribute remap节点控制一下就好。



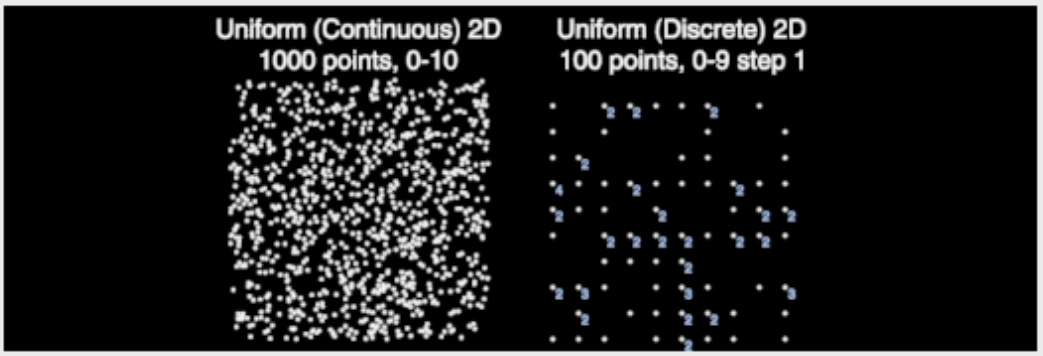
然后在利用attributerandomize节点创建一个新的index属性，然后在Distribution选择Uniform（Discrete）,这一步的目的主要是利用index这个属性对我们之后input进来两个不同的小石子进行随机摆放，Uniform（Discrete）则是output出来的值是离散的，就类似于当我们选择在0-1之间，他只会input出0跟1两个值，这样在后面我们给每个点随机分配0和1两个值来判断选择哪个小石子。

#attributerandomize

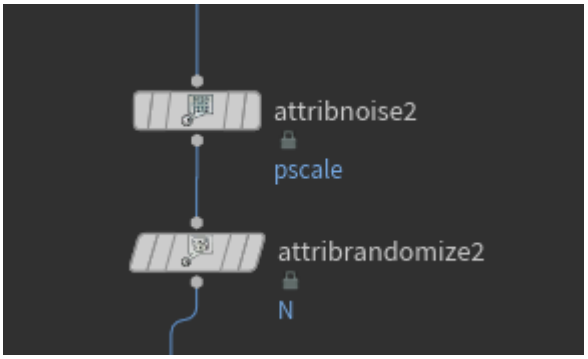
Uniform (Discrete) 均匀（离散）

The generated values will be between **Min Value** and **Max Value** (inclusive), landing only on values that are some multiple of **Step Size** greater than **Min Value**.

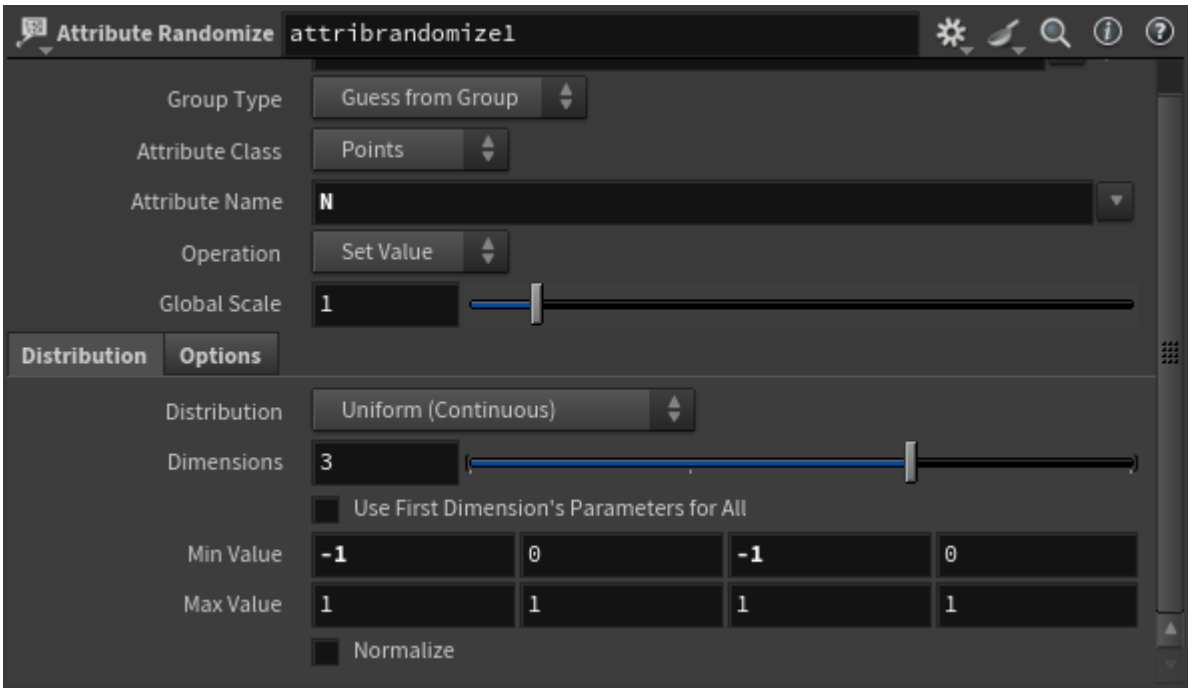
生成的值将介于**最小值**和**最大值**（含）之间，且仅落在**步长比最小值**大某个倍数的值上。

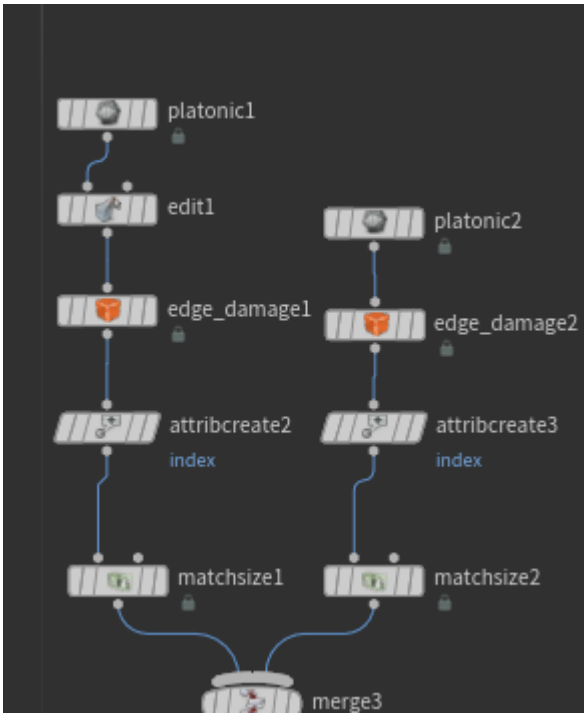


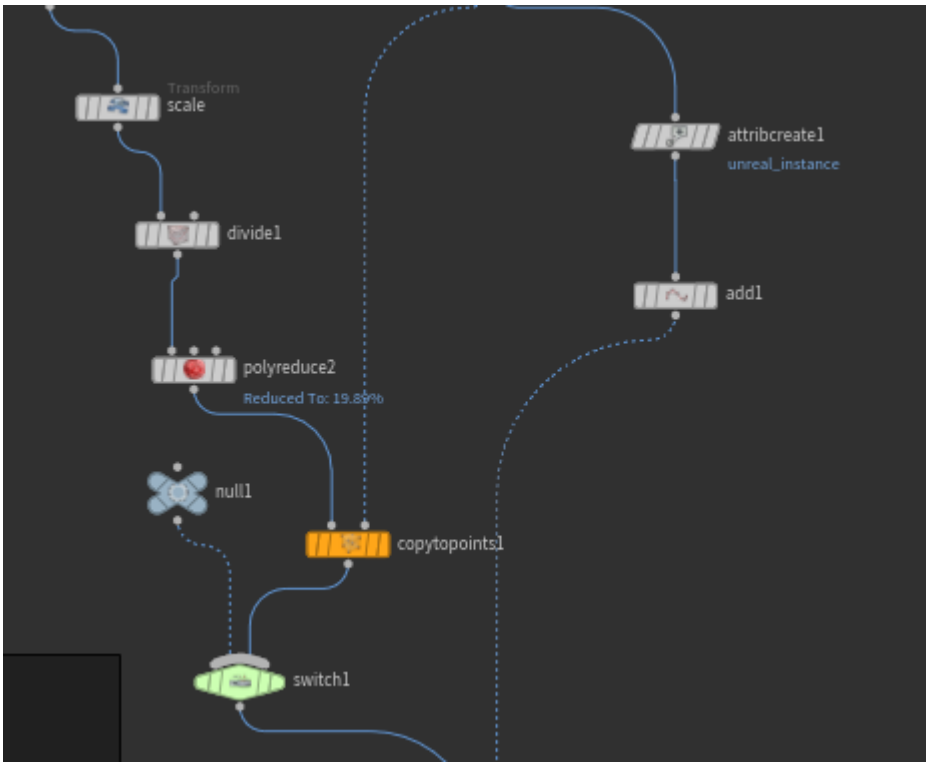
attributecast节点，将index从float值转为int值。免得出现意外，进行数据验证。
然后在添加grouprange筛选一些点，并用blast将其删除掉,减少一些小石子的数量。



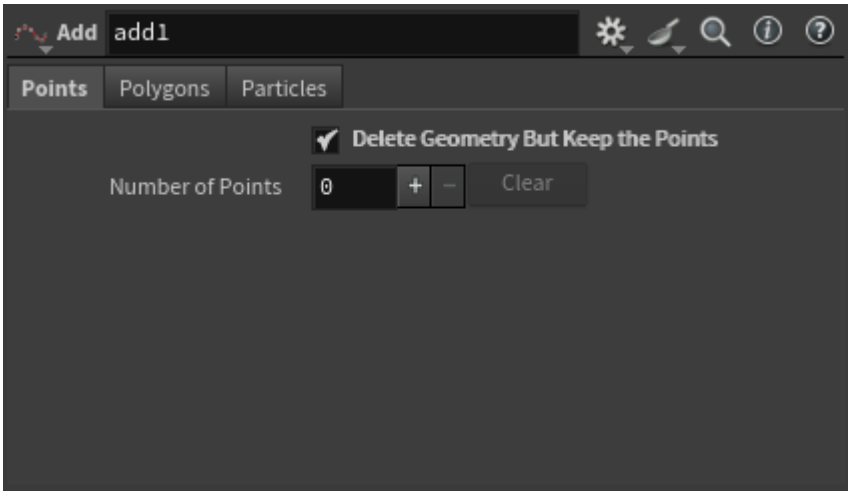
在利用attributenoise节点对pscale的数值做一些rand，以及对小石子的法线，也就是方向做一些randomize。

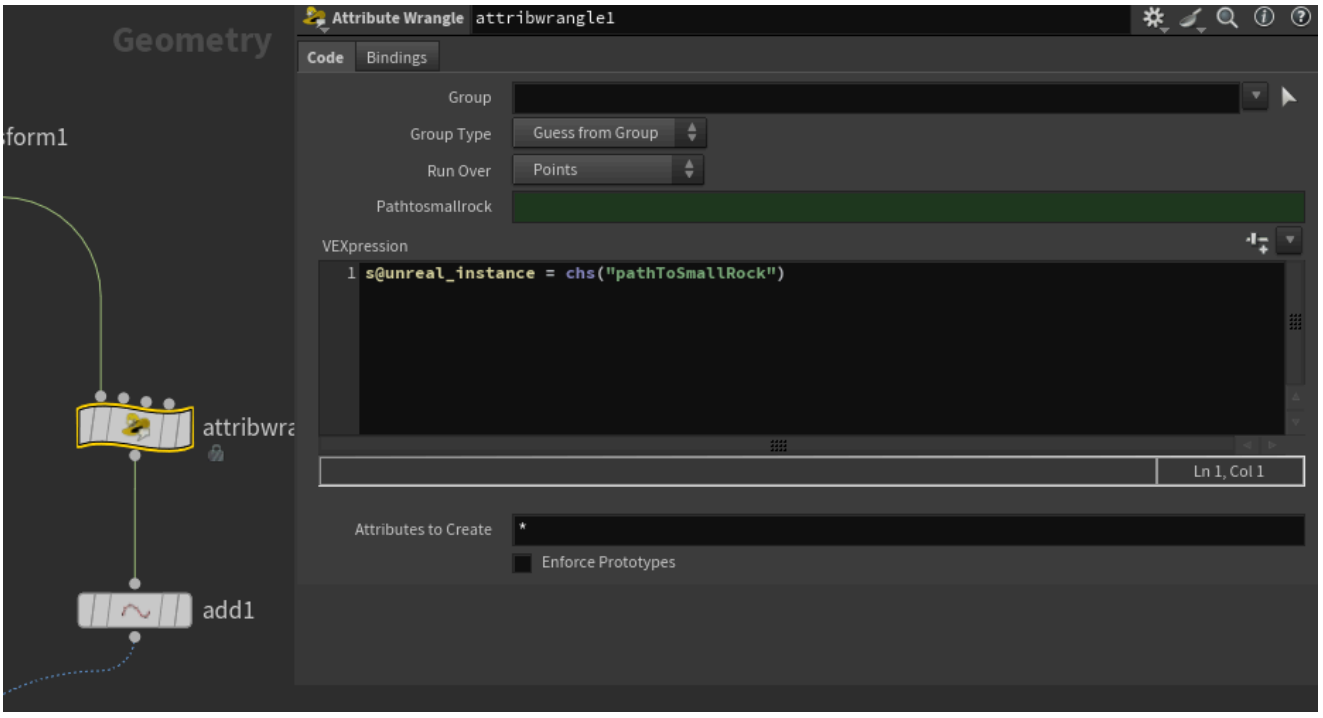

Attribute Randomize
attribrandomize1
Group Type
Guess from Group
Attribute Class
Points
Attribute Name
N
Operation
Set Value
Global Scale
1
Distribution
Options
Distribution
Uniform (Continuous)
Dimensions
3
Use First Dimension's Parameters for All
Min Value
-1
0
-1
0
Max Value
1
1
1
1
Normalize


platonic1
edit1
edge_damage1
attribcreate2
index
matchsize1
platonic2
edge_damage2
attribcreate3
index
matchsize2
merge3



这边有创建一个unreal_instance属性，这个属性主要是将houdini数据转换成unreal输出，在unreal当中进行instance的操作。加了个add节点则是为了确保delete了其他的geo，只保留点。





后面用回attribute wrangle了，用attribute create也是可以的。

This table lists special attributes to set in an HDA and the plugin recognizes when you translate the Houdini data to Unreal outputs. You can setup these attributes on your assets to control and override some properties and behaviours of the generated Unreal outputs. See Attributes and Groups for more information.

Instancers

Attribute Name	Owner	Type	Description
unreal_instance	point, detail	string	Unreal path to an Asset that shall be instanced at the given transform. In version2, this attribute can also be a Class Name (PointLight for example)

Attribute Instancers 属性实例器

Besides instancing geometry generated by the HDA, instances can also be created from pre-existing Unreal assets.

除了实例化 HDA 生成的几何图形之外，还可以从预先存在的虚幻资产创建实例。

This can be done using the unreal_instance attribute and setting the attribute value to the asset's reference.

这可以通过使用 unreal_instance 属性并将属性值设置为资产的引用来完成。

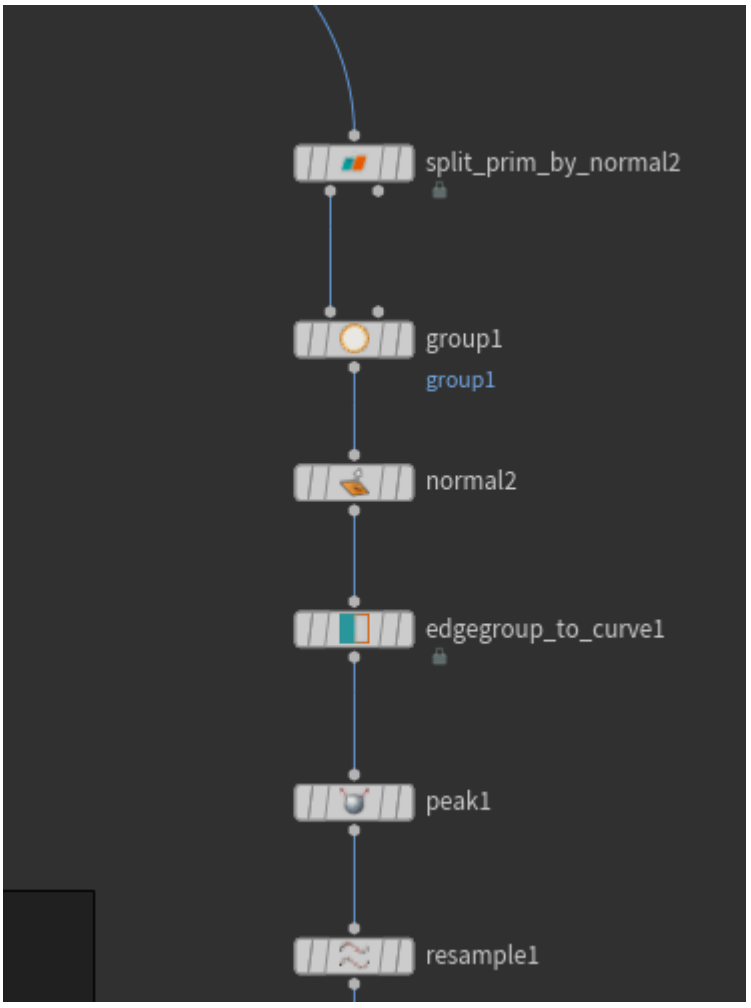
You aren't limited to static meshes, but the instanced asset must be convertible to a native Actor type. In that case, multiple instances of that Actor type is created and the generated instancer component will be a HoudiniInstancedActorComponent. This can be used to spawn pre-existing assets in the level using the generated instancer's point cloud.

您不限于静态网格体，但实例化资源必须能够转换为原生 Actor 类型。在这种情况下，系统会创建该 Actor 类型的多个实例，并且生成的实例化器组件将为 HoudiniInstancedActorComponent。这可用于使用生成的实例化器的点云在关卡中生成预先存在的资源。

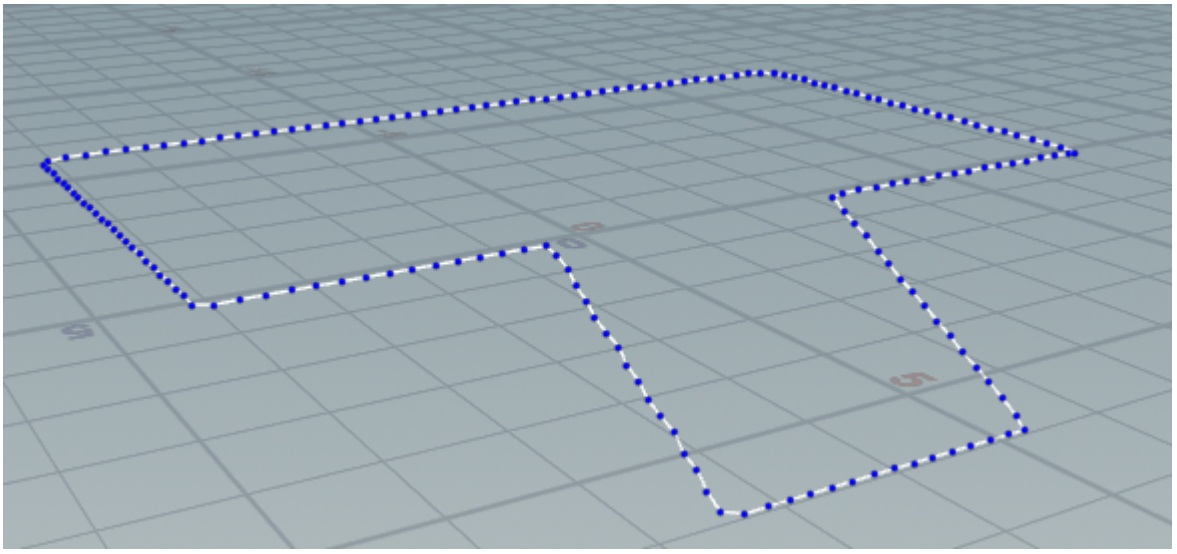
You can also instantiate classes directly by using attribute instancers. This is useful to create instances of a class which you can't save as a Uasset in the Content Browser, but can be instaniated as an Actor(for example, Lights, Volumes etc..).

您还可以使用属性实例器直接实例化类。这对于创建无法在内容浏览器中保存为 Uasset 但可以实例化为 Actor（例如，光源、体积等）的类的实例非常有用。

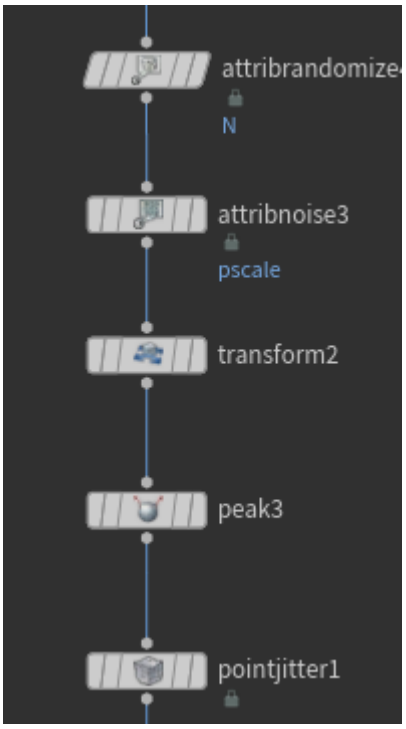
石墙撒点制作



依然是split平台出来，但是invert这个结果，再利用group节点 的 unshared edges将边进行分组，摘选出来，在利用edgegroup to curve将他转为多边形曲线。peak节点重新计算点的法线，resample节点控制石柱的数量。



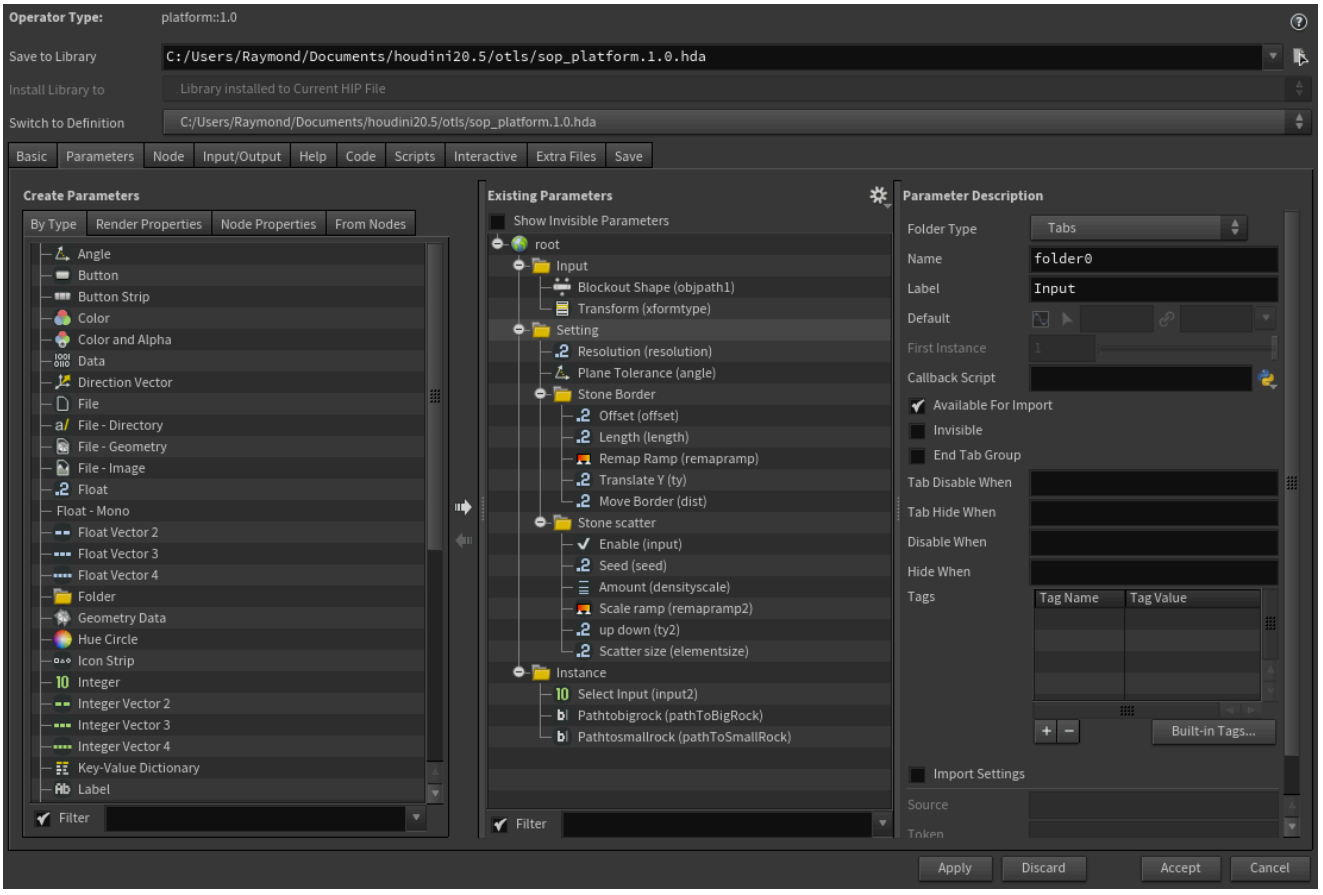
点弄出来了基本上整体思路就跟小石子一样，copy to point石柱到点上就行，一些些微的区别就是利用point jitter和 peak节点做一些方向和位置上的rand



#attributerandomize 这边的randomize将Y完全设为0，只randomize X和Z的值，就能达到石柱左右的一些rand效果

HDA制作

HDA list



HDA的话主要还是看个人吧，这边我就展示一下我自己的list，做做参考。