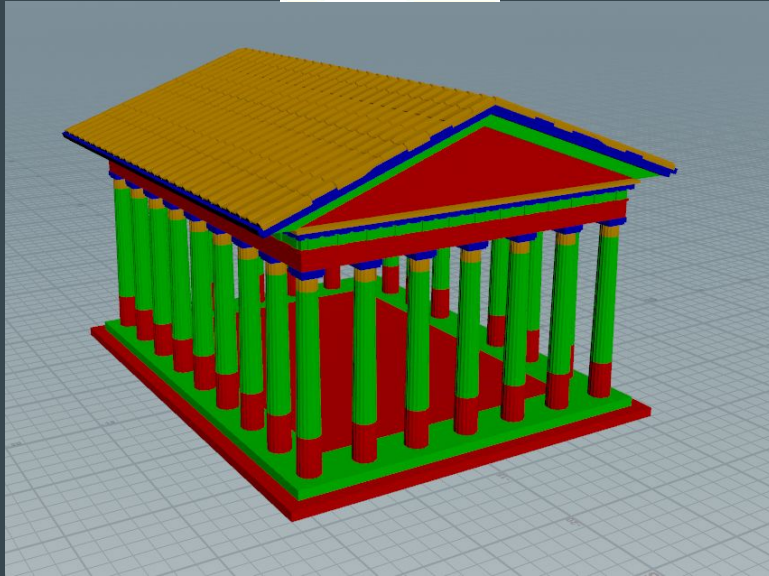
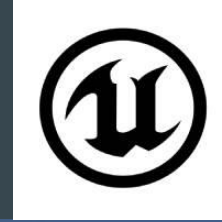


# Houdini Unreal PCG

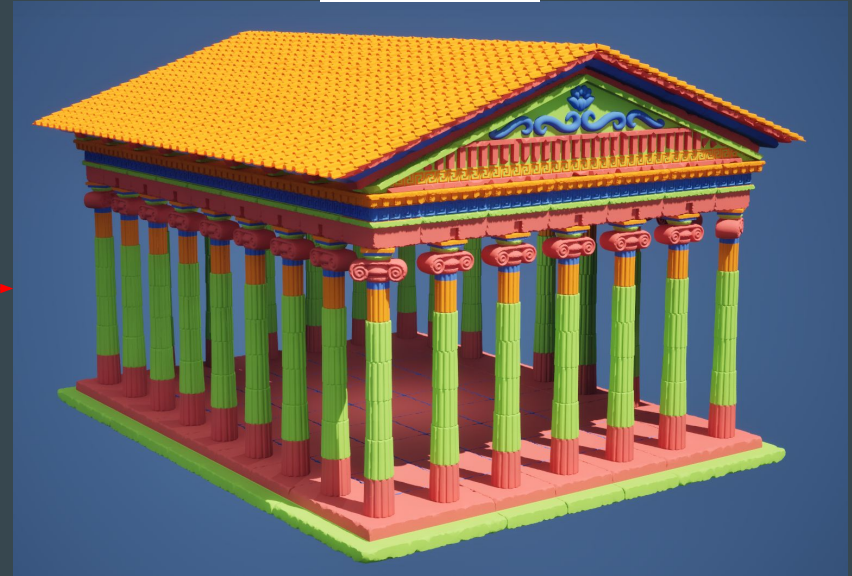
...

# Table of Contents

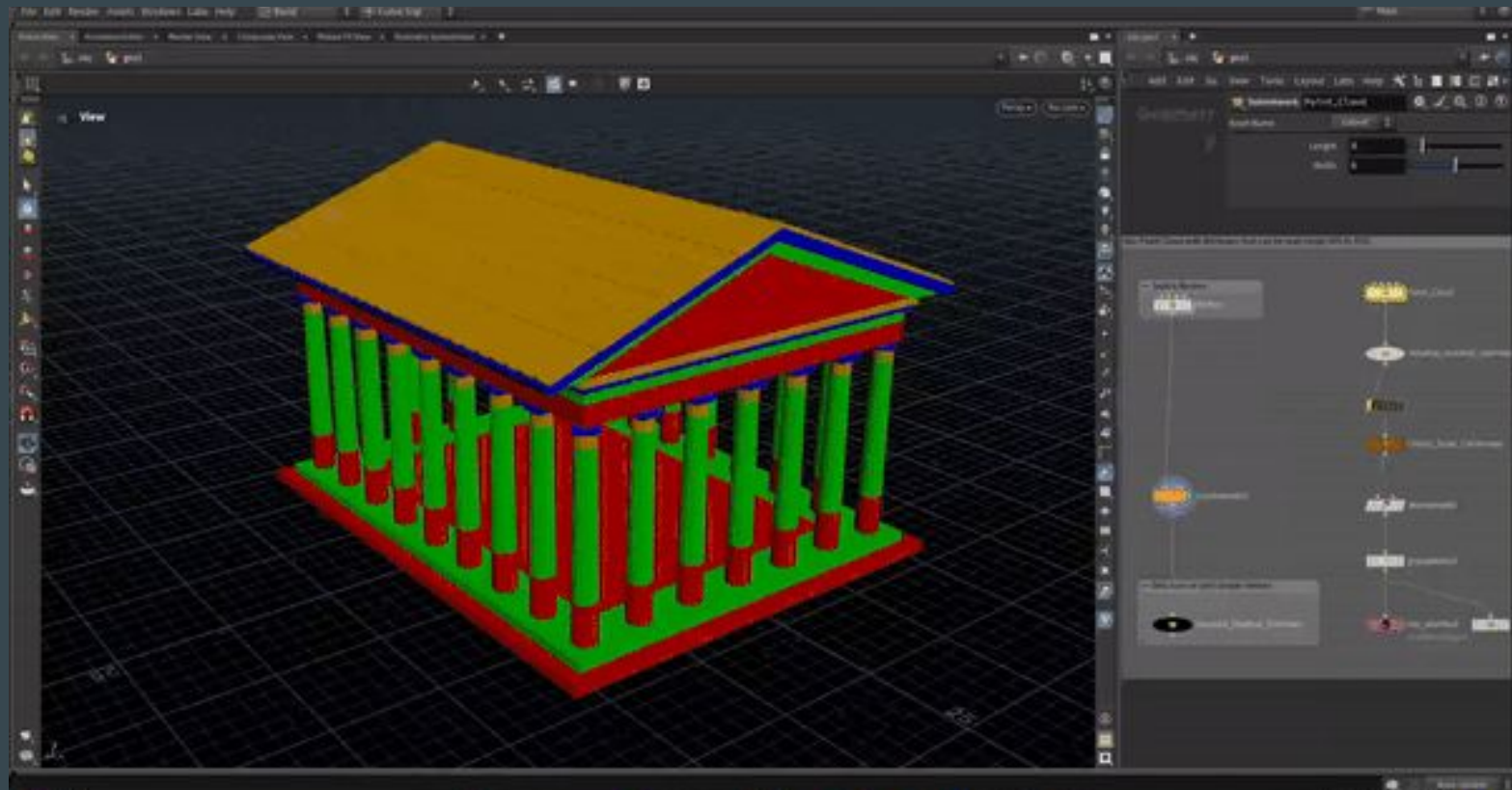
- > Workflow
- > Export Format
- > Export Setup
- > Lego Example
- > Material Overrides
- > Matrix Demo City Sample
- > Alembic Format
- > Roads
- > Prefabs
- > Environment Artists
- > Colosseum



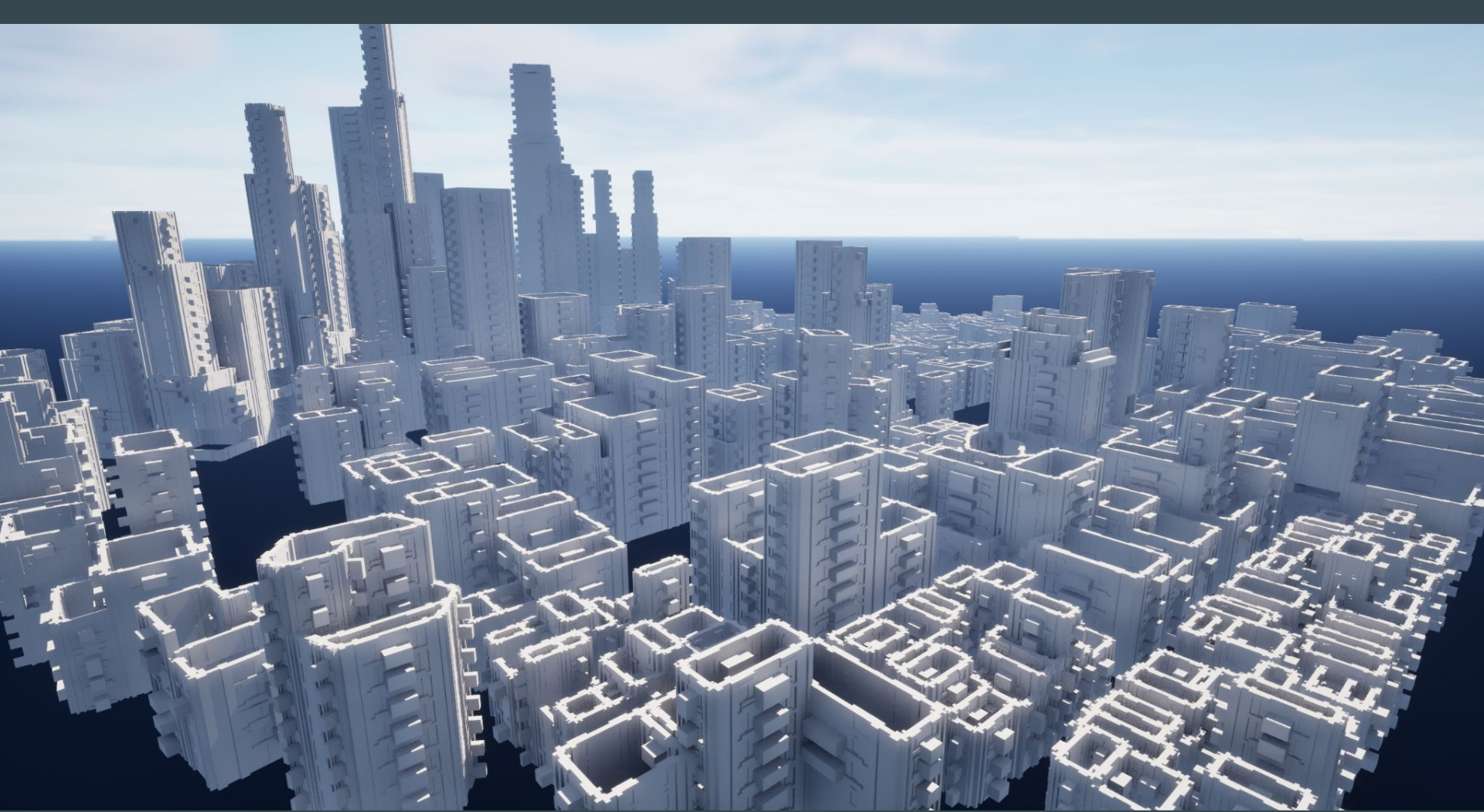
PCG



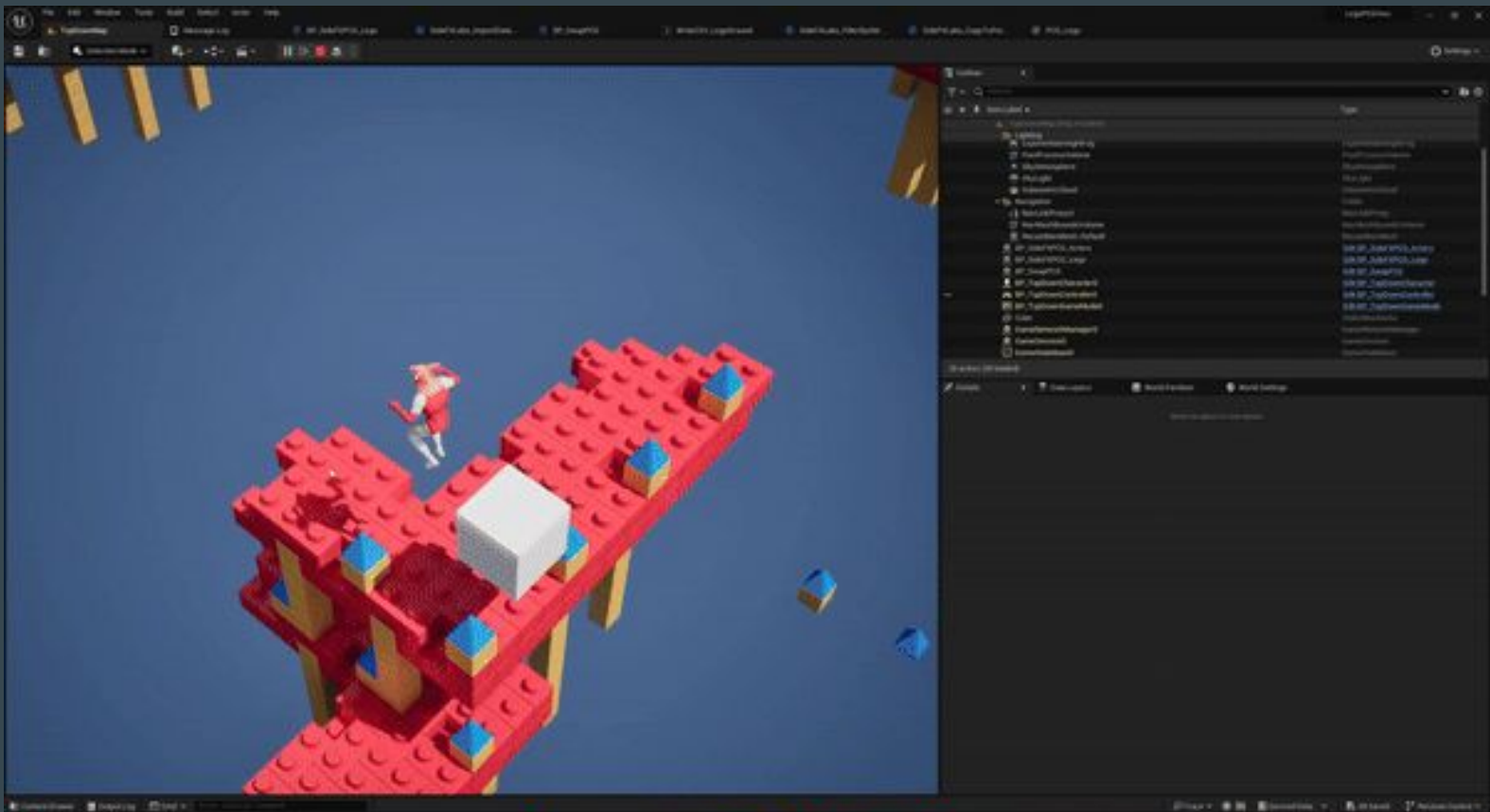
# Procedural Parameters







# Real-time PCG Player Input Controls



# Real-time PCG Animation



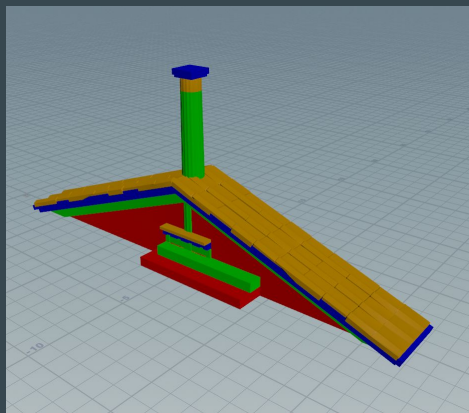
\*Too Slow\*



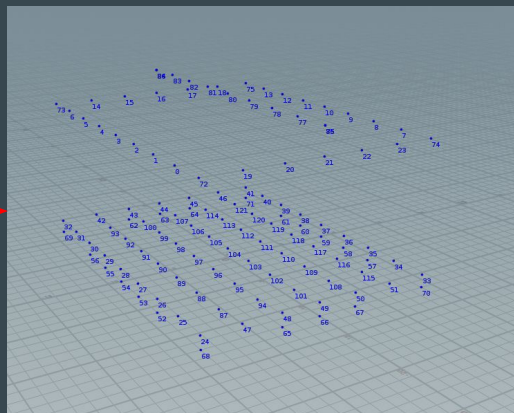


# Workflow

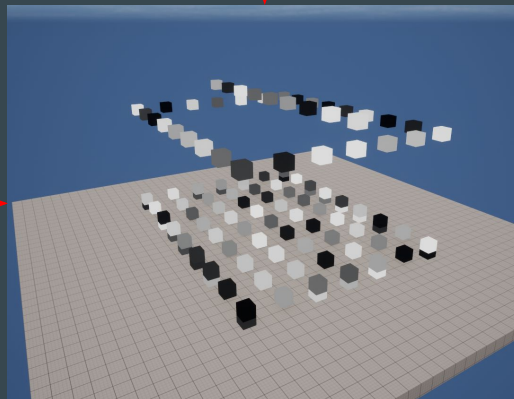
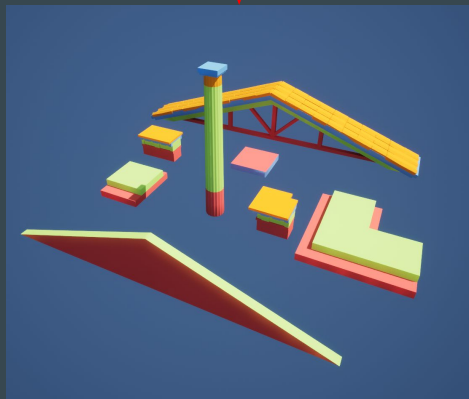
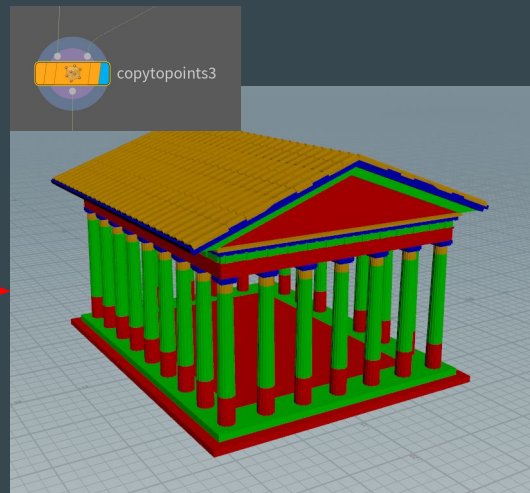
...



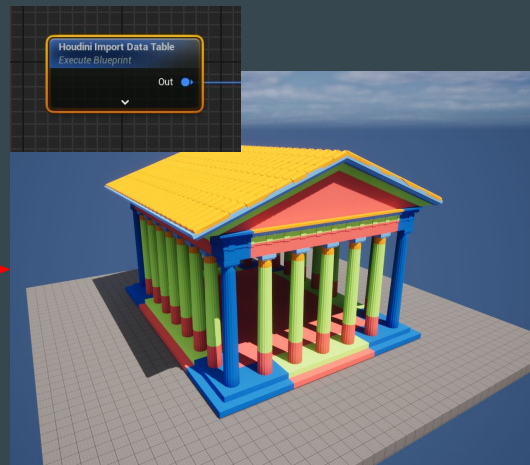
FBX



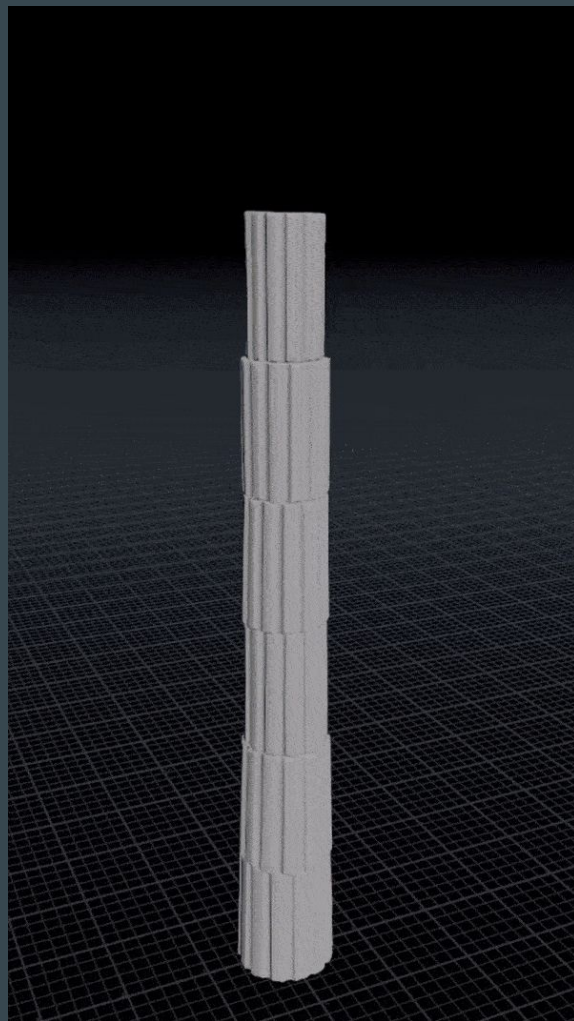
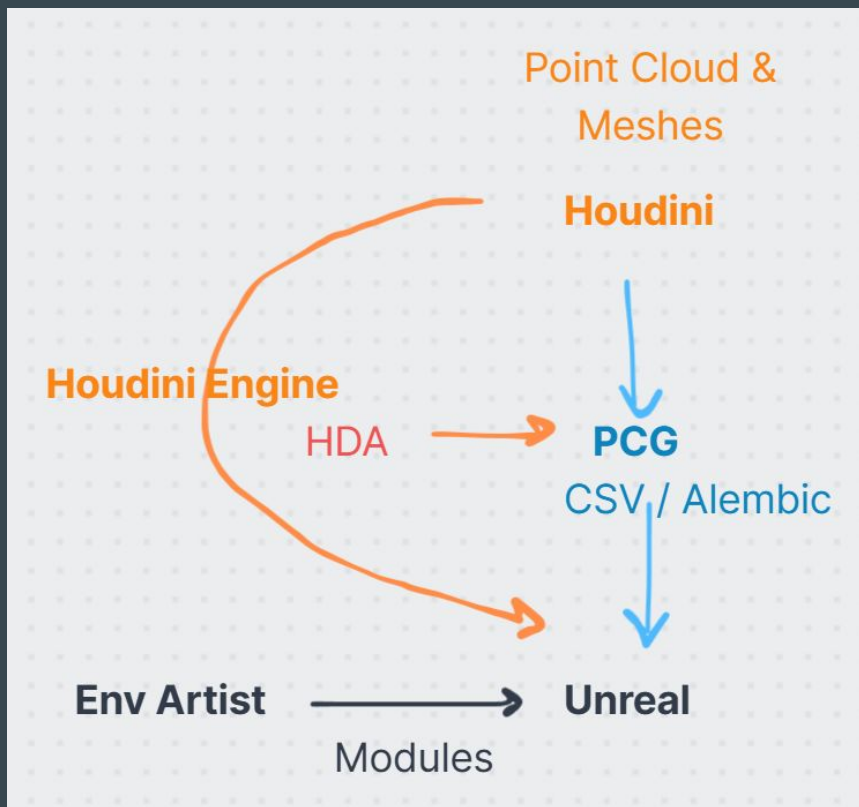
Alembic CSV



PCG

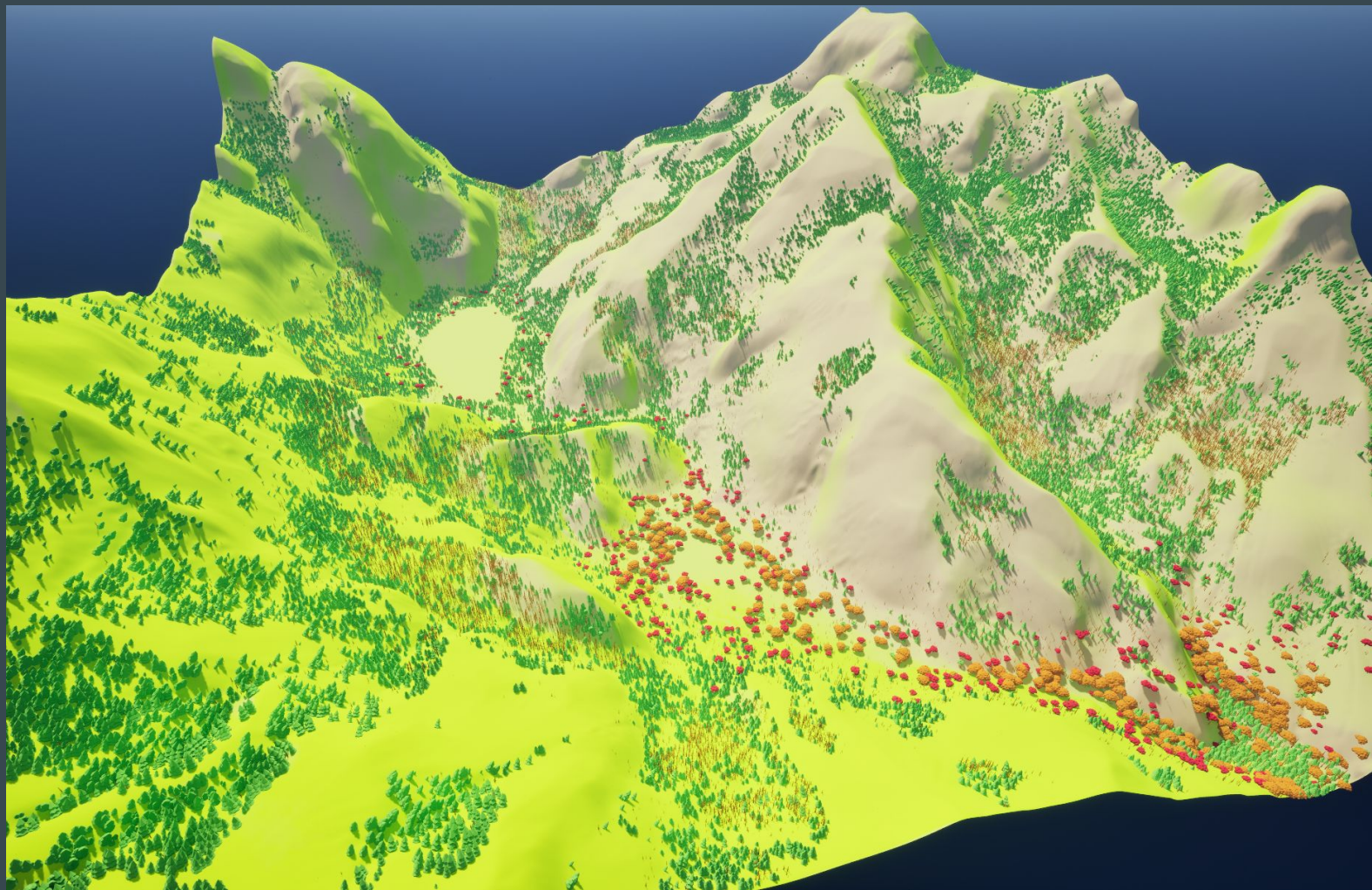


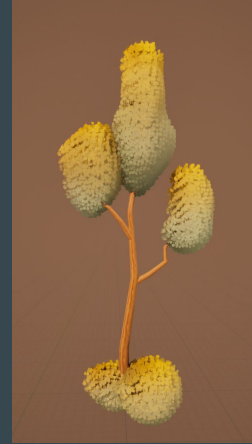
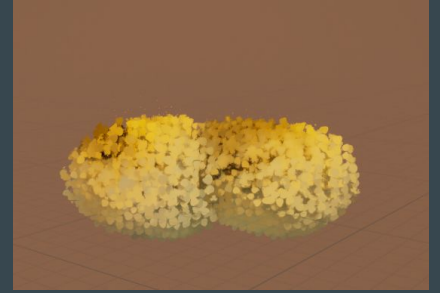
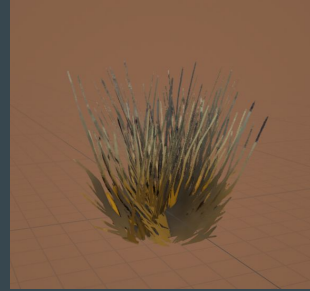
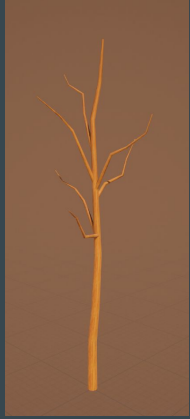
# Workflow













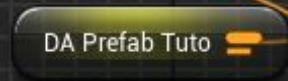
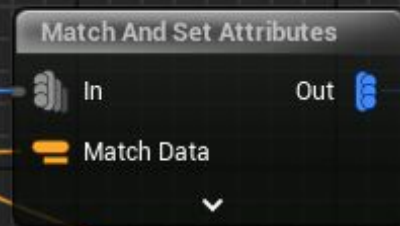
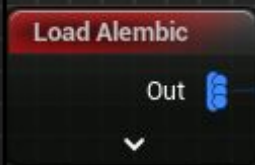
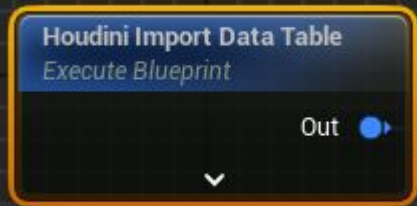
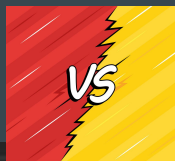




# Export Format

...

# CSV vs Alembic



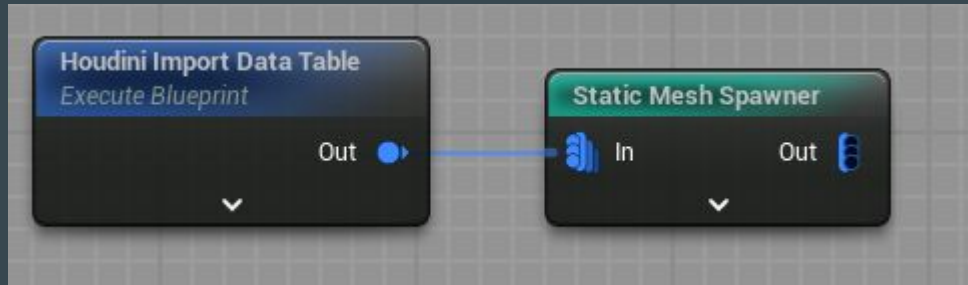
## Houdini To PCG Pipeline

CSV UE5.2+

Alembic UE5.4+

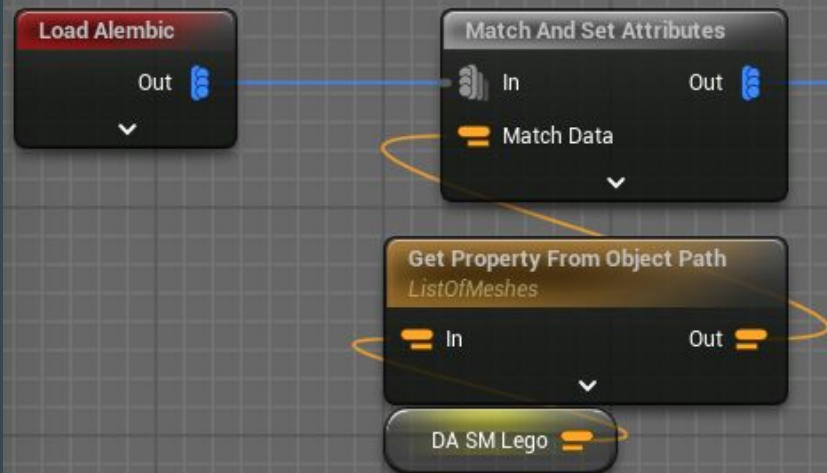
Static Meshes/ Actors/ Prefabs/ Material Overrides

# CSV

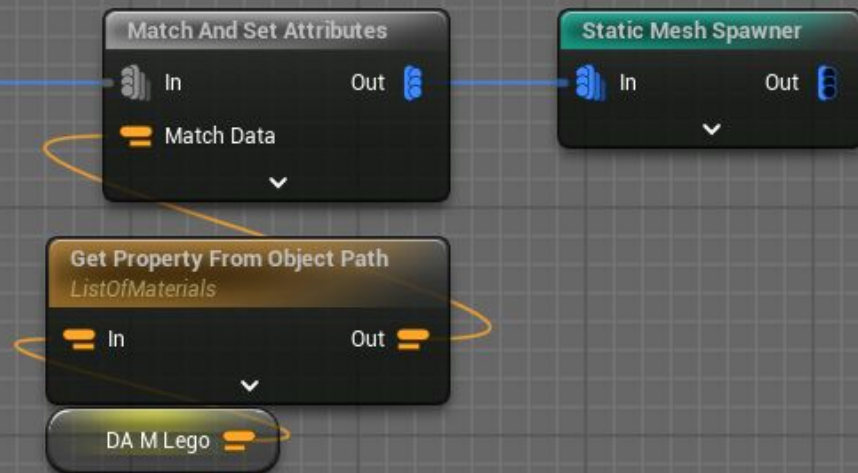


# Alembic

## Mesh



## Material

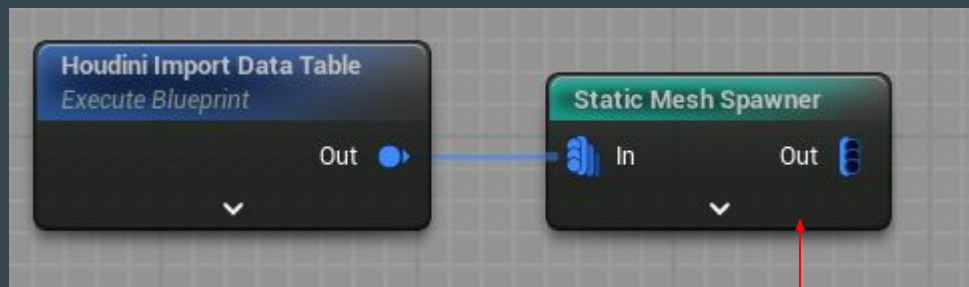




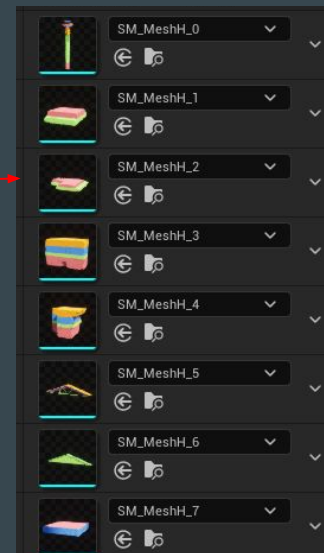
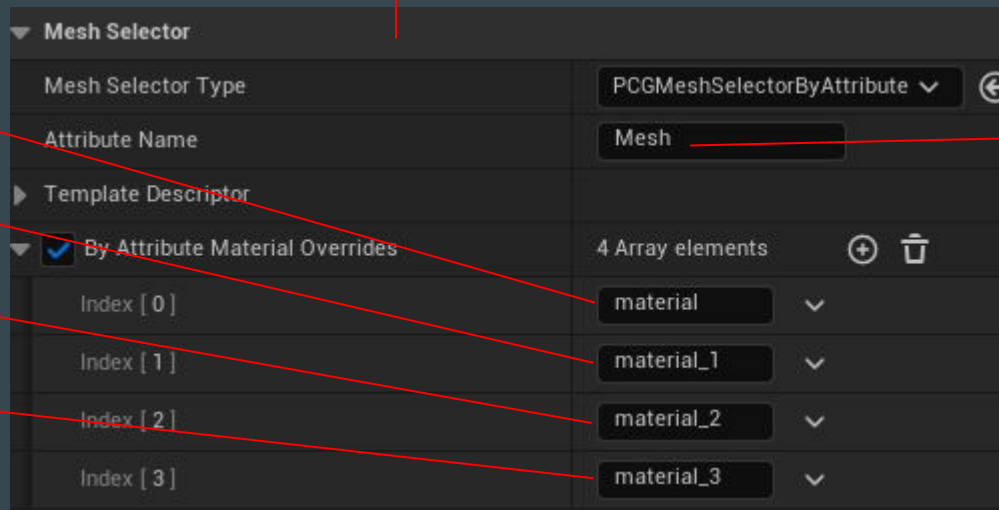
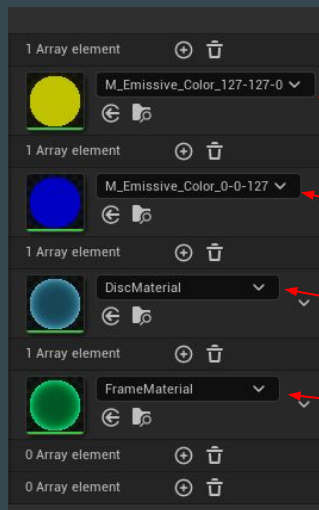
# Export Setup

...

# CSV Spawning Static Meshes

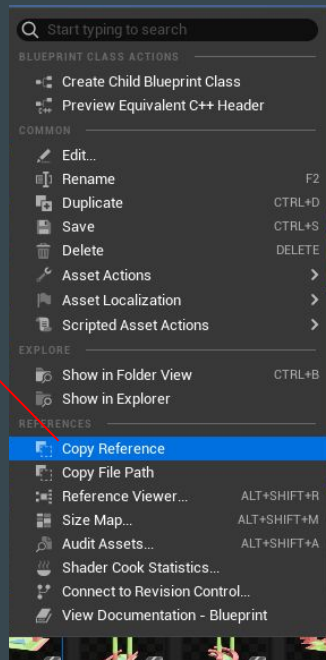


Spawn by Attribute

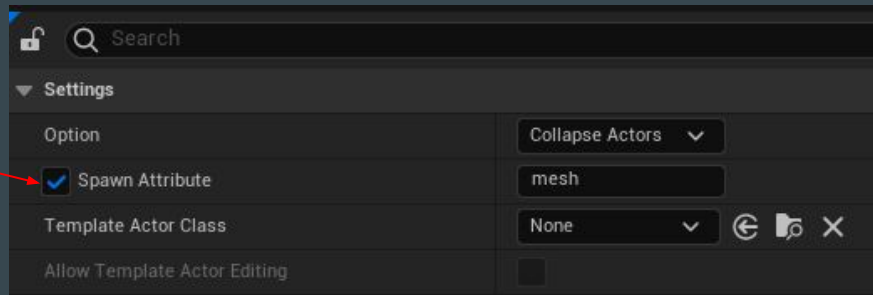
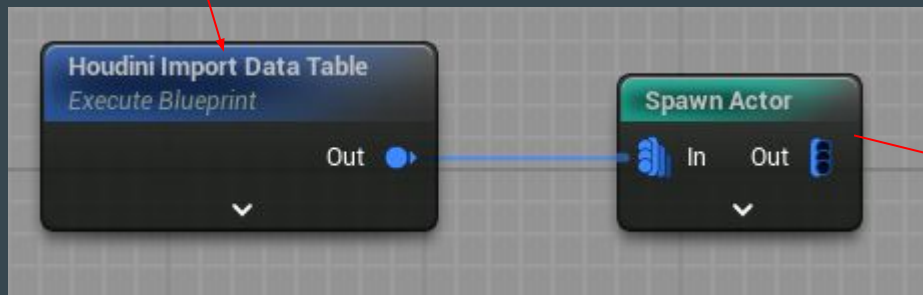


# CSV Spawn Actors

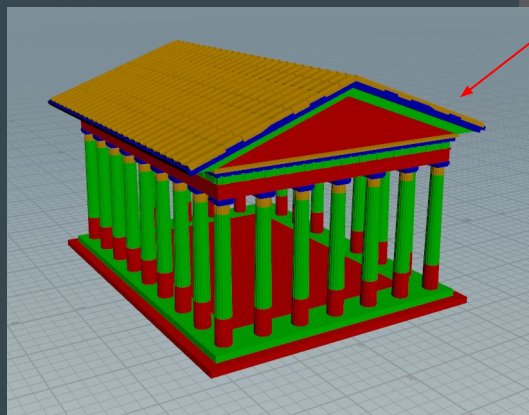
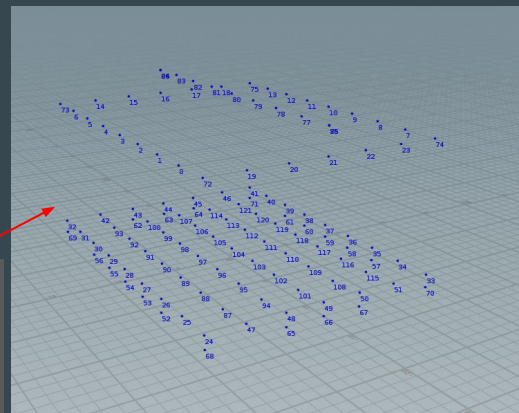
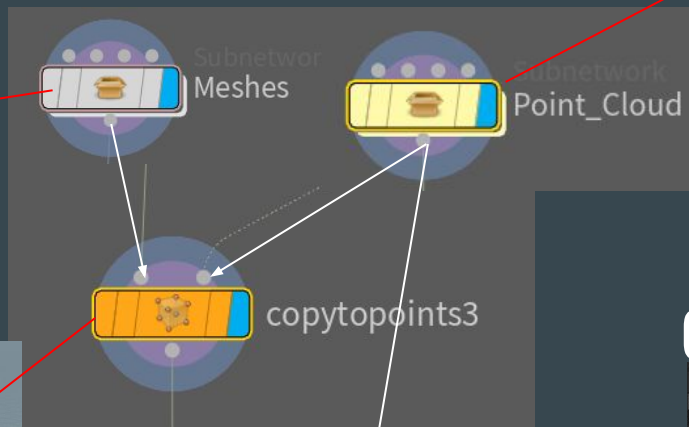
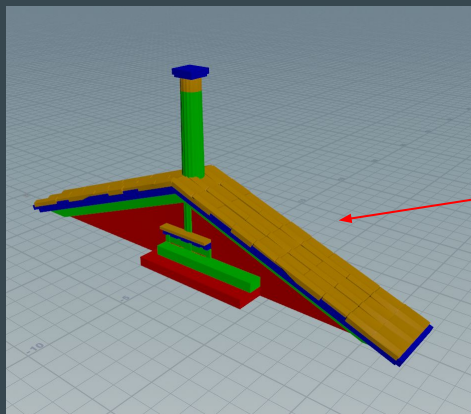
Actors	9 Array elements	+	-
Index [ 0 ]	/Script/Engine.Blueprint'/Game/PCG/DataAss		
Index [ 1 ]	/Script/Engine.Blueprint'/Game/PCG/DataAss		
Index [ 2 ]	/Script/Engine.Blueprint'/Game/PCG/DataAss		
Index [ 3 ]	/Script/Engine.Blueprint'/Game/PCG/DataAss		
Index [ 4 ]	/Script/Engine.Blueprint'/Game/PCG/DataAss		
Index [ 5 ]	/Script/Engine.Blueprint'/Game/PCG/DataAss		
Index [ 6 ]	/Script/Engine.Blueprint'/Game/PCG/DataAss		
Index [ 7 ]	/Script/Engine.Blueprint'/Game/PCG/DataAss		
Index [ 8 ]	/Script/Engine.Blueprint'/Game/PCG/DataAss		



Spawn by Attribute



# PCG Export Setup



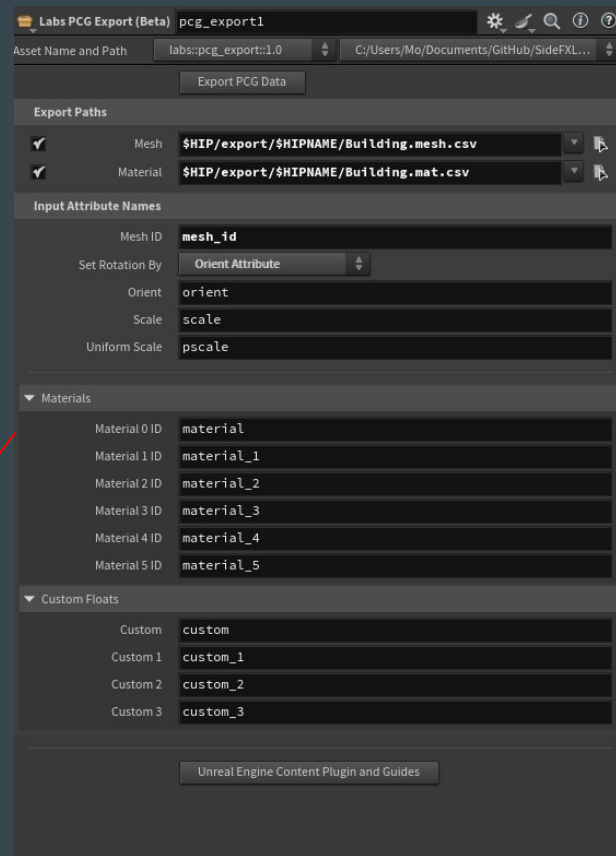
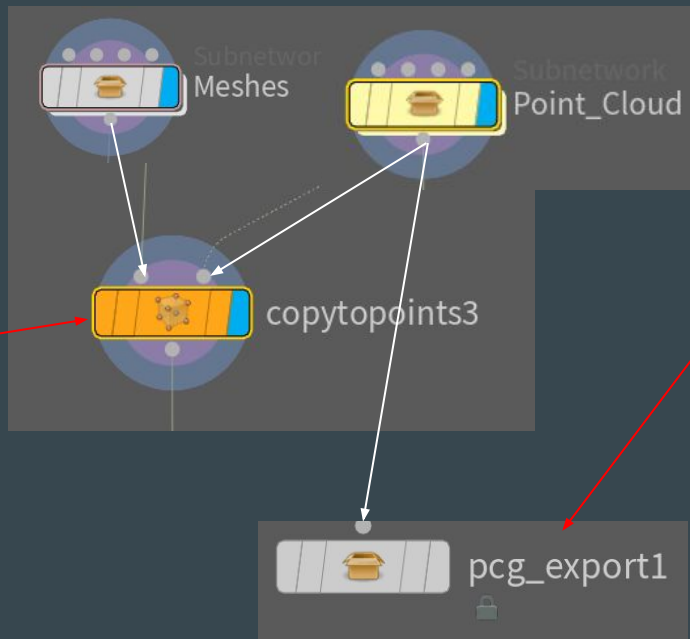
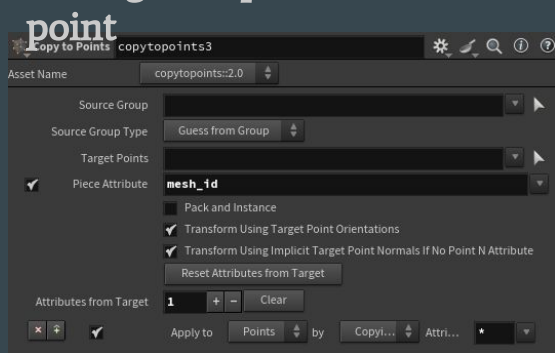
CSV

	Row	Px	Py	Pz	Nx	Ny	Nz	Uvx	Uvy	UVz	pscale
1	0	-900.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
2	1	-600.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
3	2	-300.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
4	3	0.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
5	4	300.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
6	5	600.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
7	6	900.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
8	7	-900.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
9	8	-600.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
10	9	-300.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
11	10	0.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
12	11	300.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
13	12	600.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
14	13	900.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
15	14	1200.000000	1100.000000	-600.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000
16	15	1200.000000	1100.000000	-300.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000
17	16	1200.000000	1100.000000	0.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000
18	17	1200.000000	1100.000000	300.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000
19	18	1200.000000	1100.000000	600.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000
20	19	1200.000000	1100.000000	900.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000

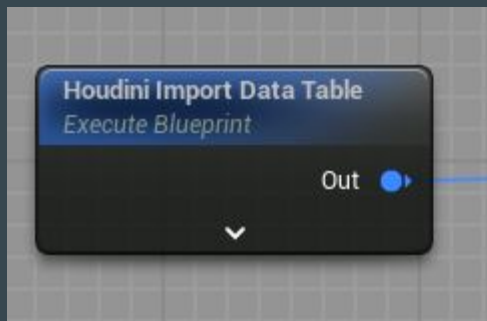


# PCG Export Settings (Mesh\_id is an integer “No Hard Reference”)

Piece Attribute is Mesh\_id  
which determines which  
mesh gets copied to which  
point



# CSV Import Settings

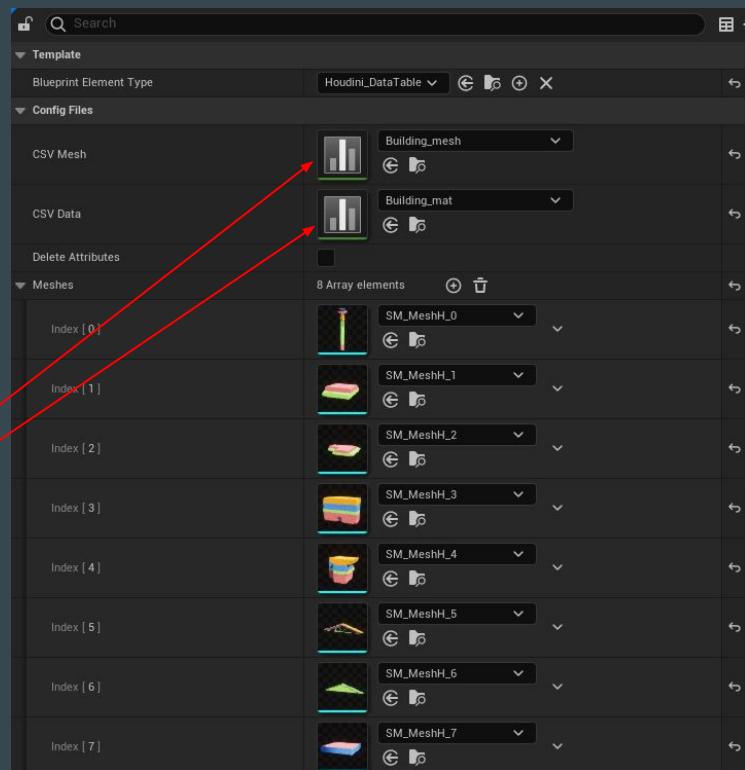


# CSV Point Cloud Data

	Row	Px	Py	Pz	Nx	Ny	Nz	UVx	UVy	UVz	pscale
1	0	-900.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
2	1	-600.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
3	2	-300.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
4	3	0.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
5	4	300.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
6	5	600.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
7	6	900.000000	1100.000000	-900.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	3.000000
8	7	-900.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
9	8	-600.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
10	9	-300.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
11	10	0.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
12	11	300.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
13	12	600.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
14	13	900.000000	1100.000000	900.000000	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	3.000000
15	14	1200.000000	1100.000000	-600.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000
16	15	1200.000000	1100.000000	-300.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000
17	16	1200.000000	1100.000000	0.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000
18	17	1200.000000	1100.000000	300.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000
19	18	1200.000000	1100.000000	600.000000	0.000000	-0.707107	0.000000	1.000000	1.000000	1.000000	3.000000

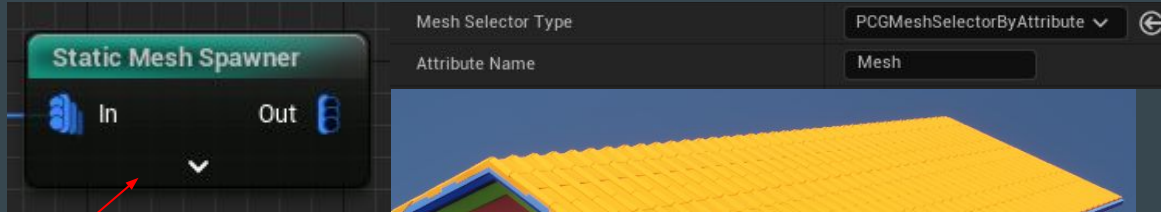
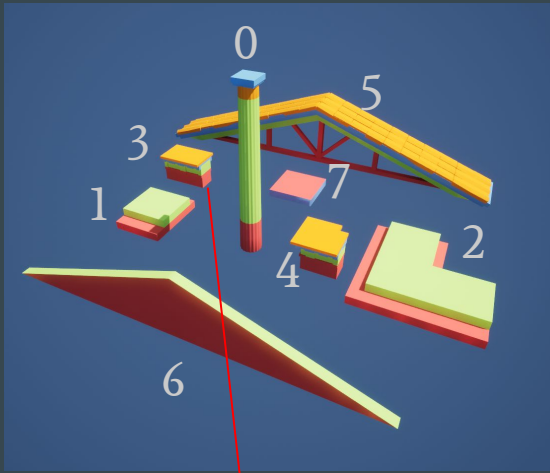
## CSV PCG:

Meshes are included within the node in an array.

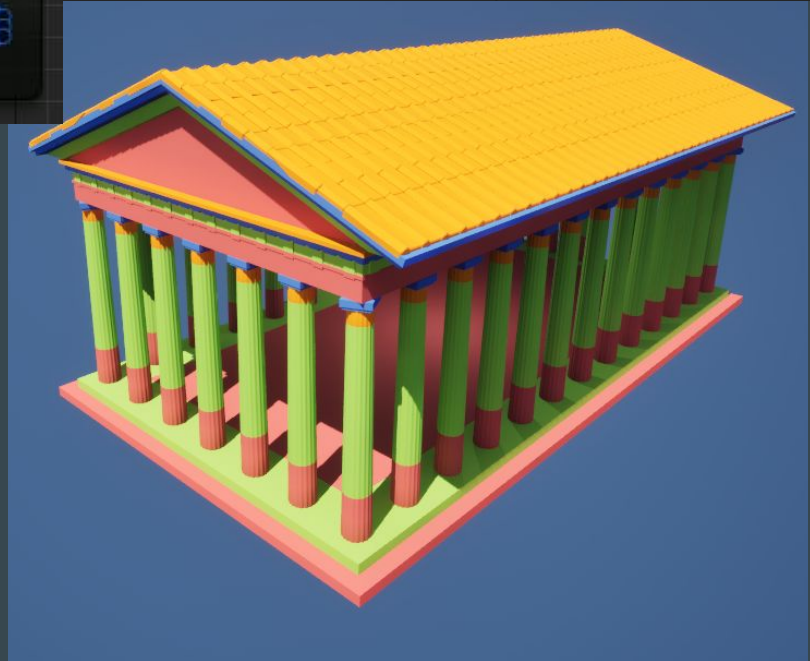




# Mesh\_id Gets Converted to Mesh Path



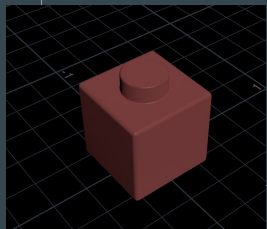
Index ▲	mesh_id	Mesh
0	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
1	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
2	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
3	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
4	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
5	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
6	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
7	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
8	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
9	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
10	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
11	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
12	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3
13	3	/Game/PCG/Meshes/SM_Mesh_3.SM_Mesh_3

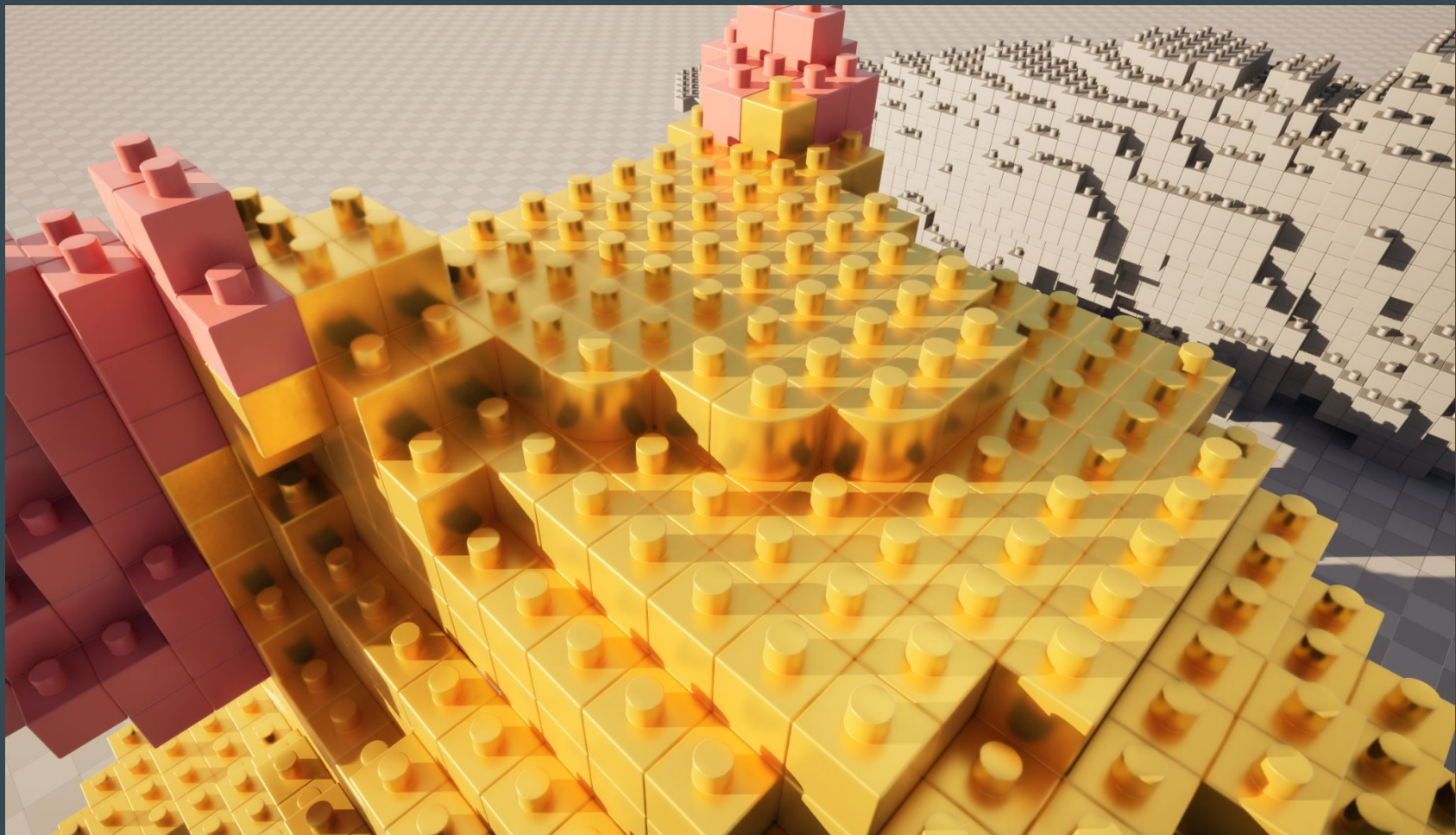


# Lego Example

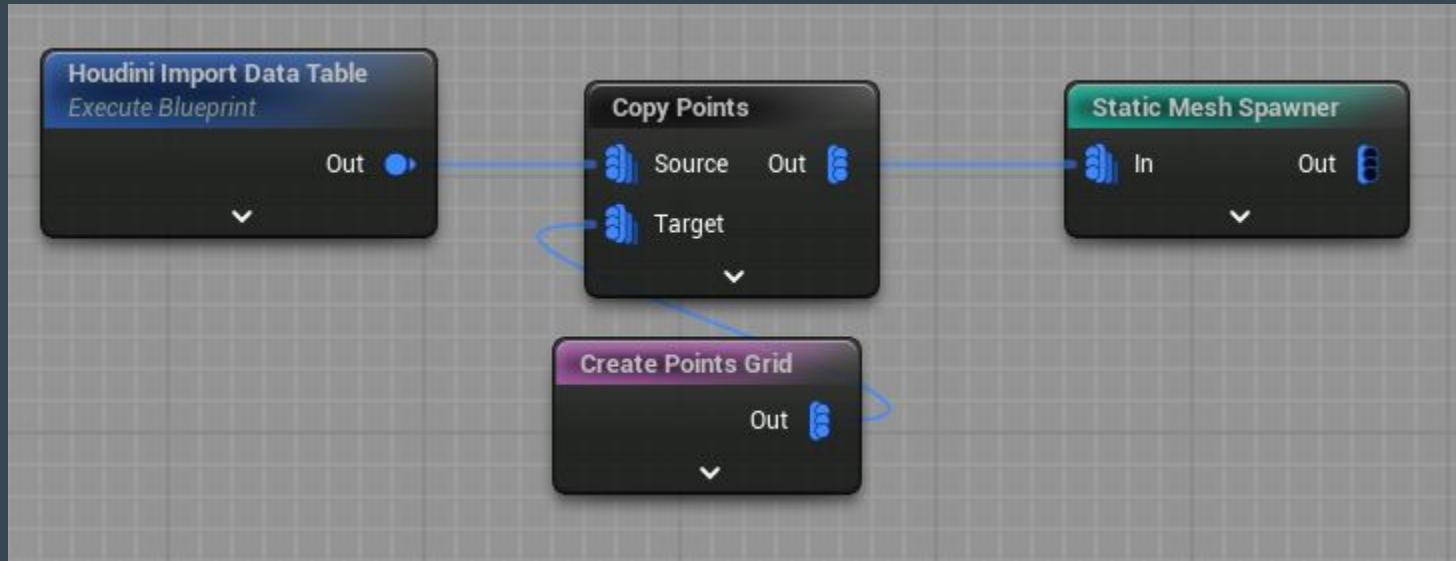
...



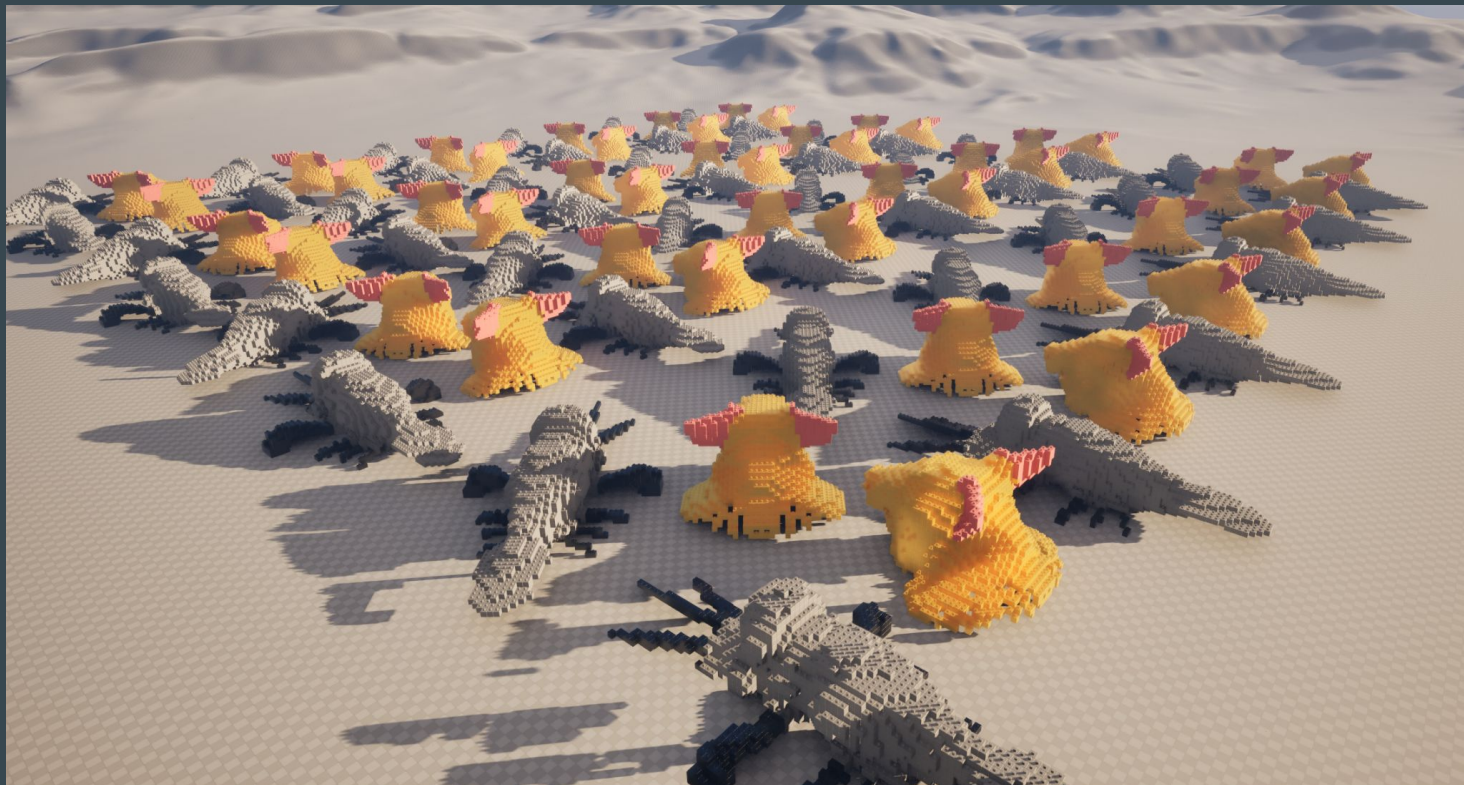




# Copy Houdini Point Cloud To Grid of Points in PCG



# Copy Houdini Point Cloud To Grid of Points in PCG

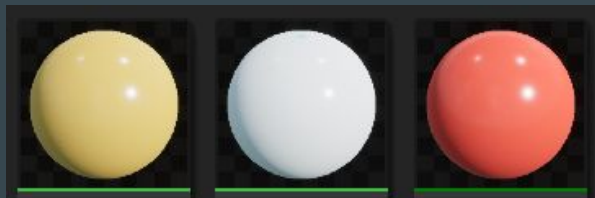


# Material Overrides

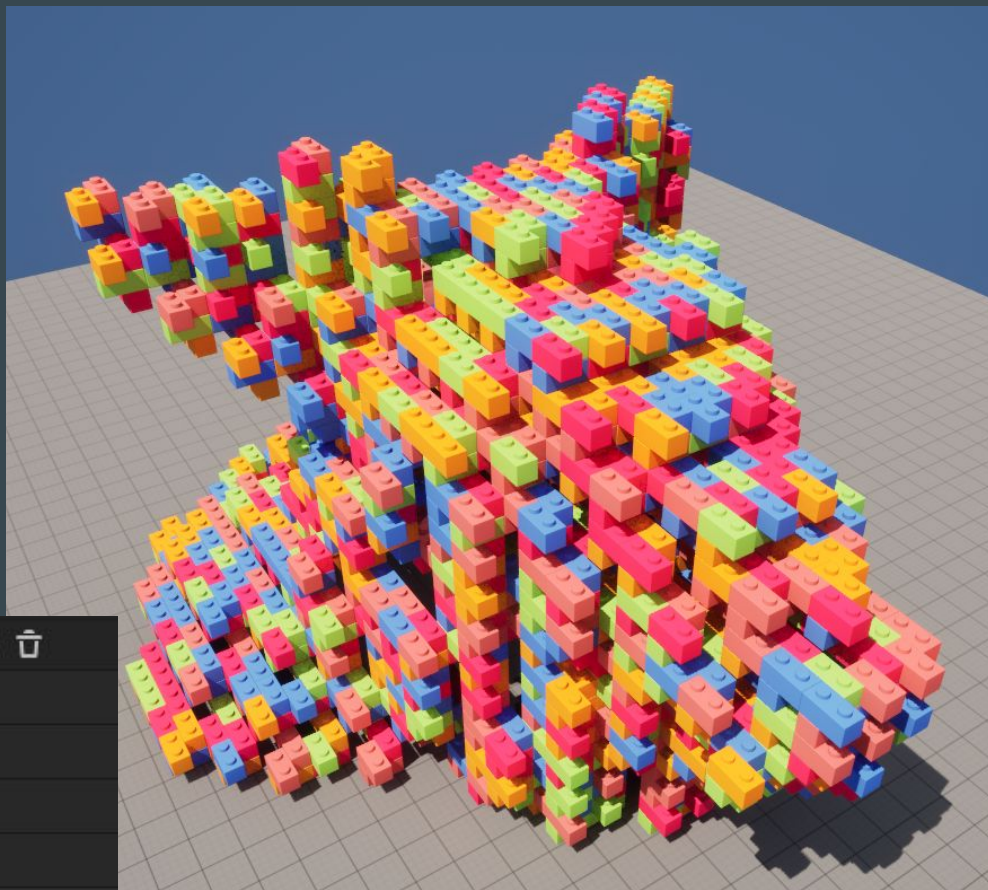
...



# Material Overrides

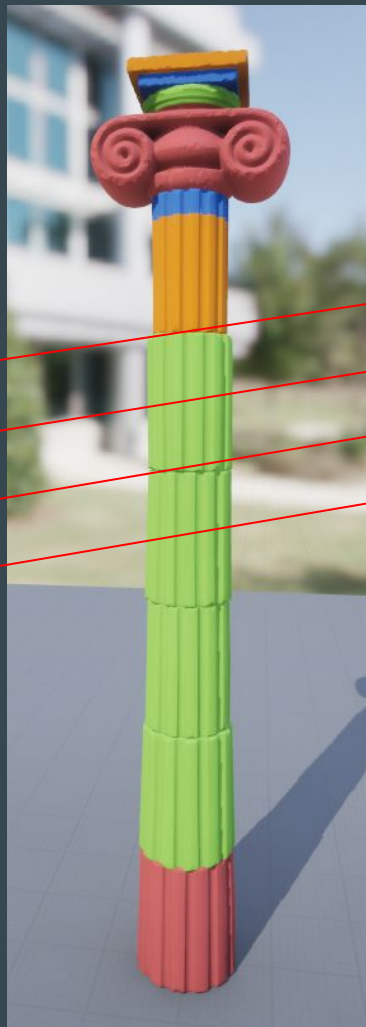
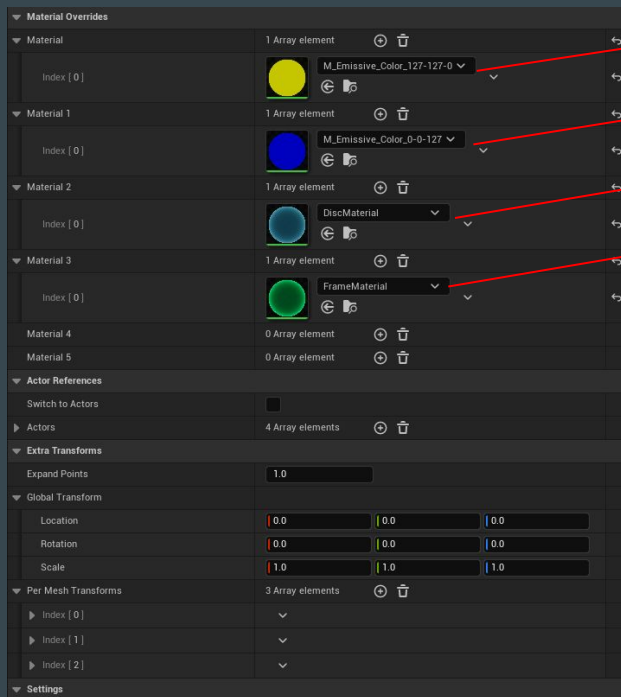


<input checked="" type="checkbox"/> By Attribute Material Overrides	4 Array elements	 
Index [ 0 ]	material	▼
Index [ 1 ]	material_1	▼
Index [ 2 ]	material_2	▼
Index [ 3 ]	material_3	▼

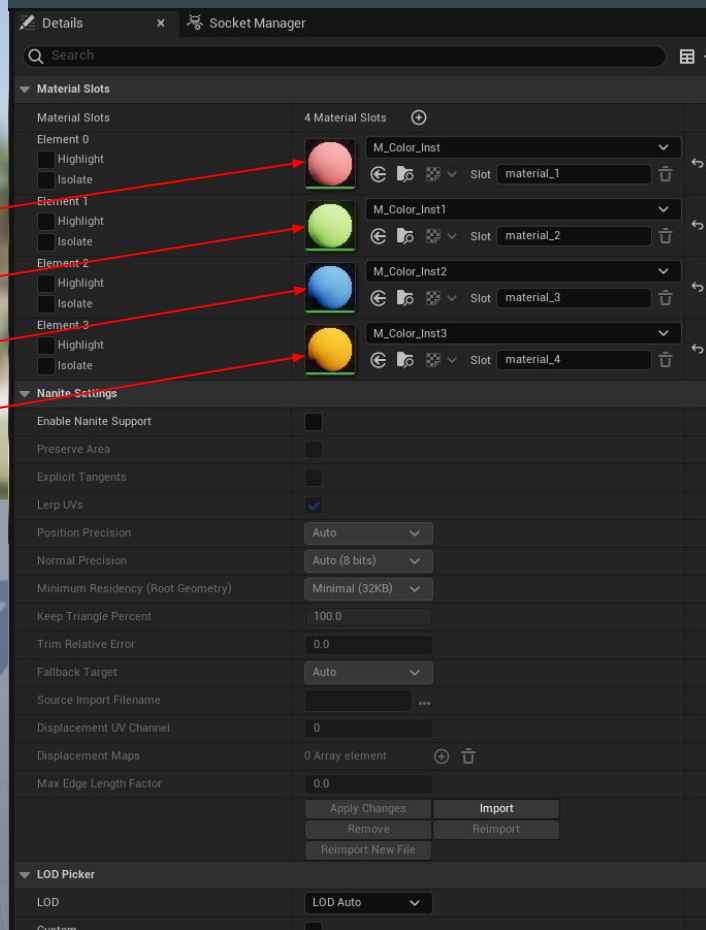


# Multi-Material Overrides

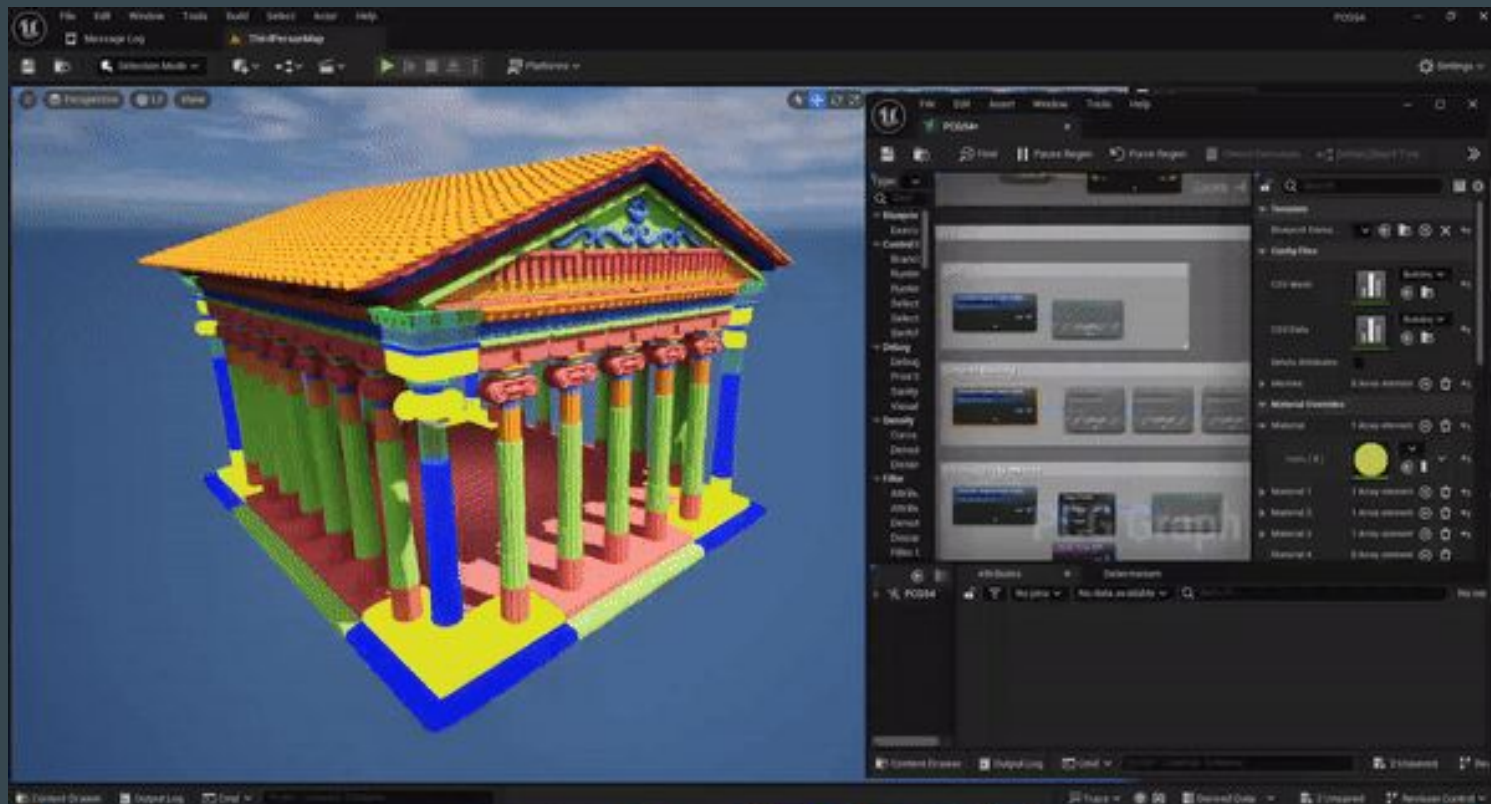
## PCG Details



## Mesh Details

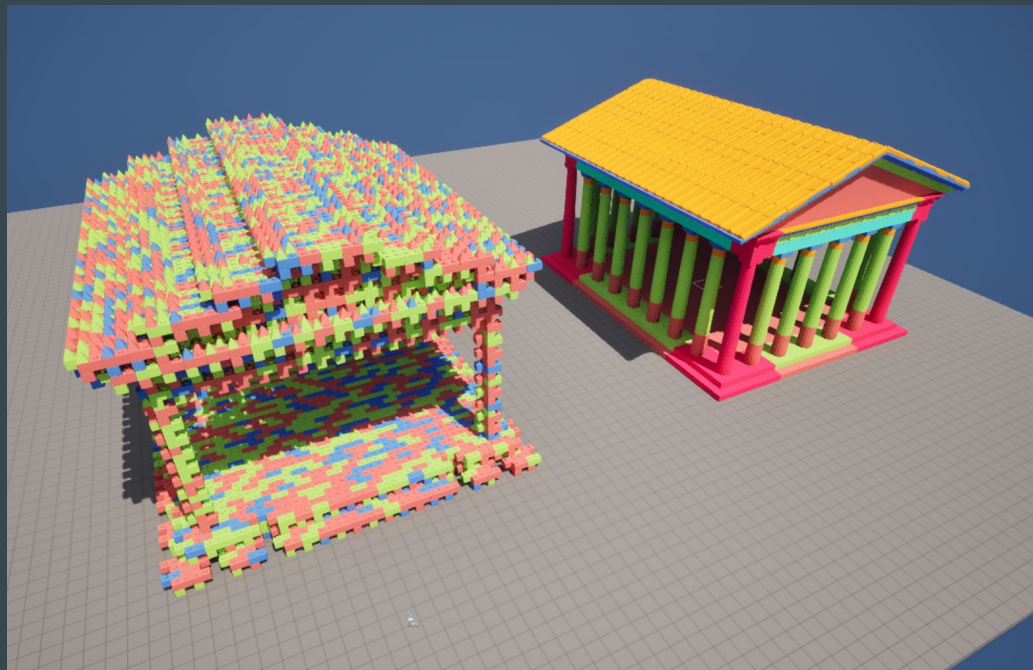
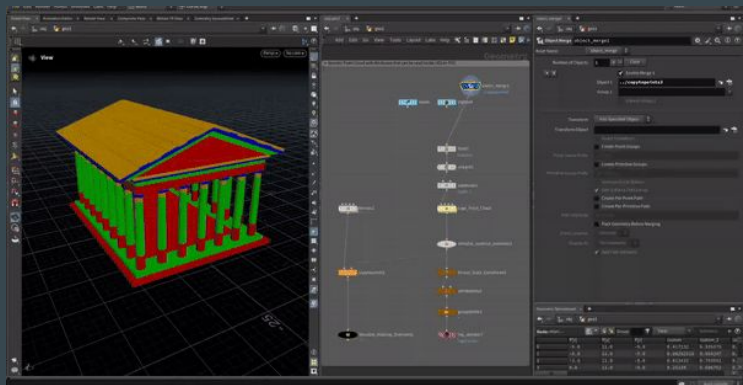
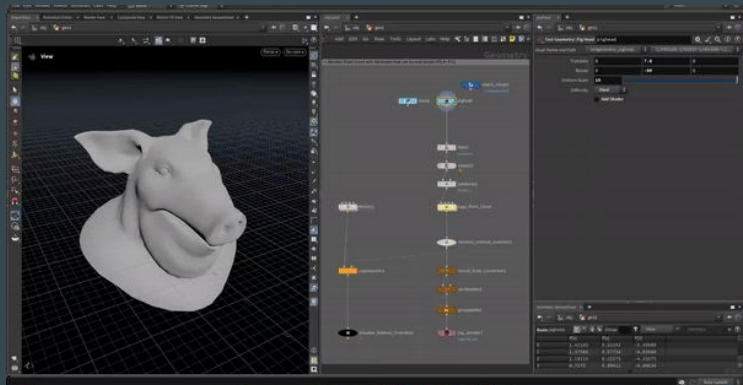


# Multi-Material Overrides

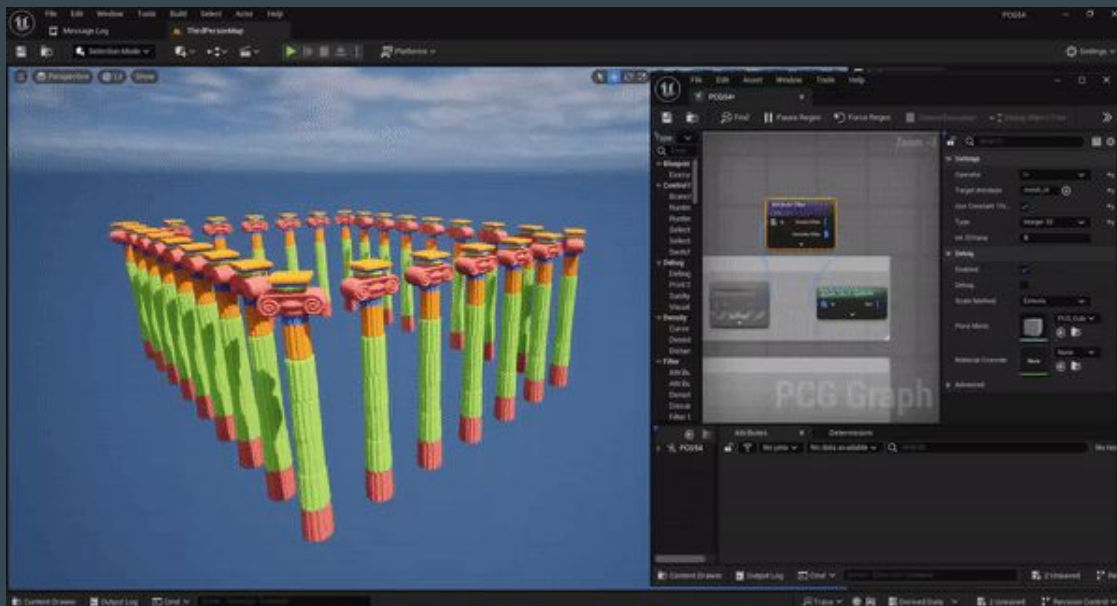




# Mixing Houdini Graphs (Turning Building into Lego)



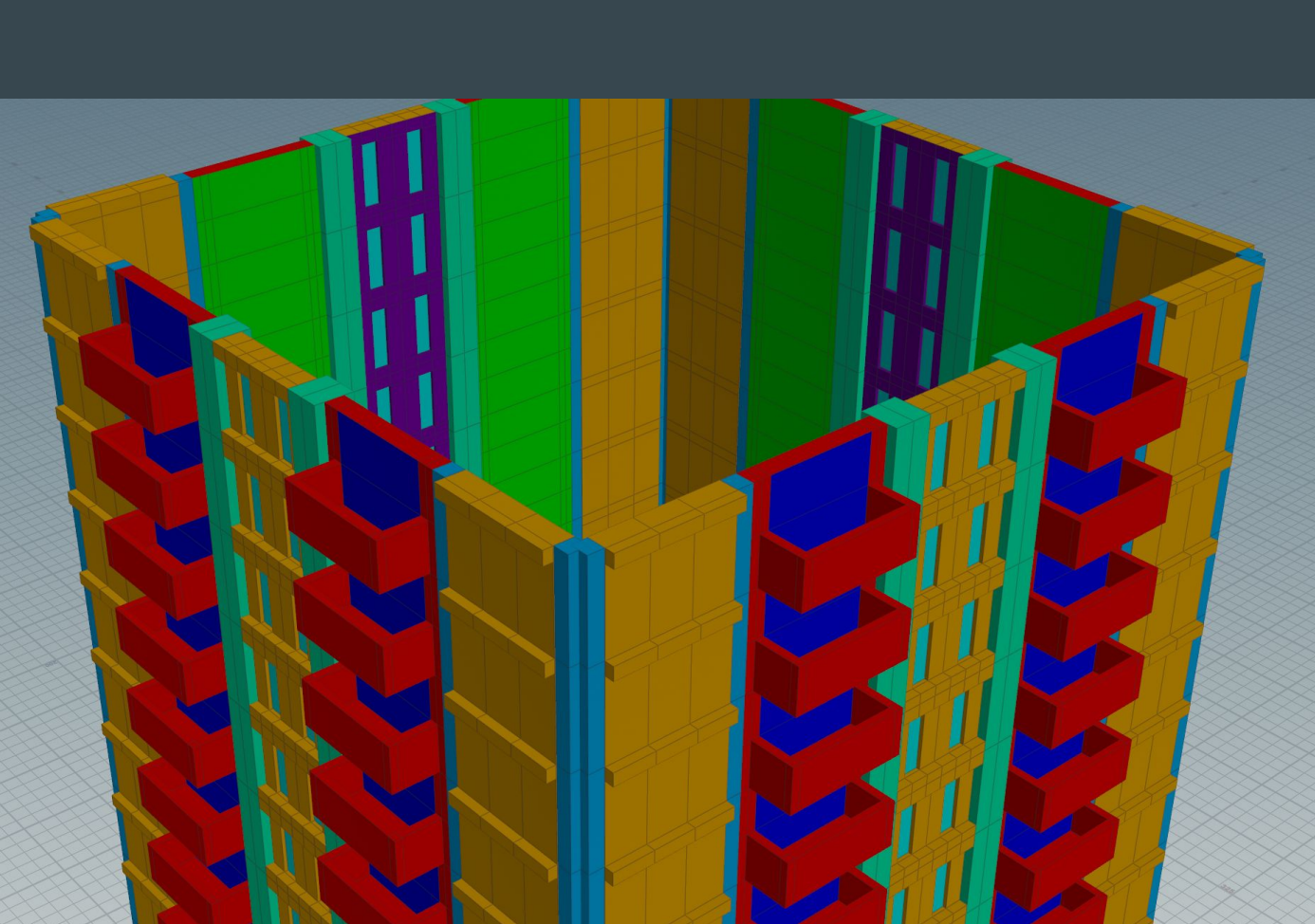
# Attribute filter Mesh\_id



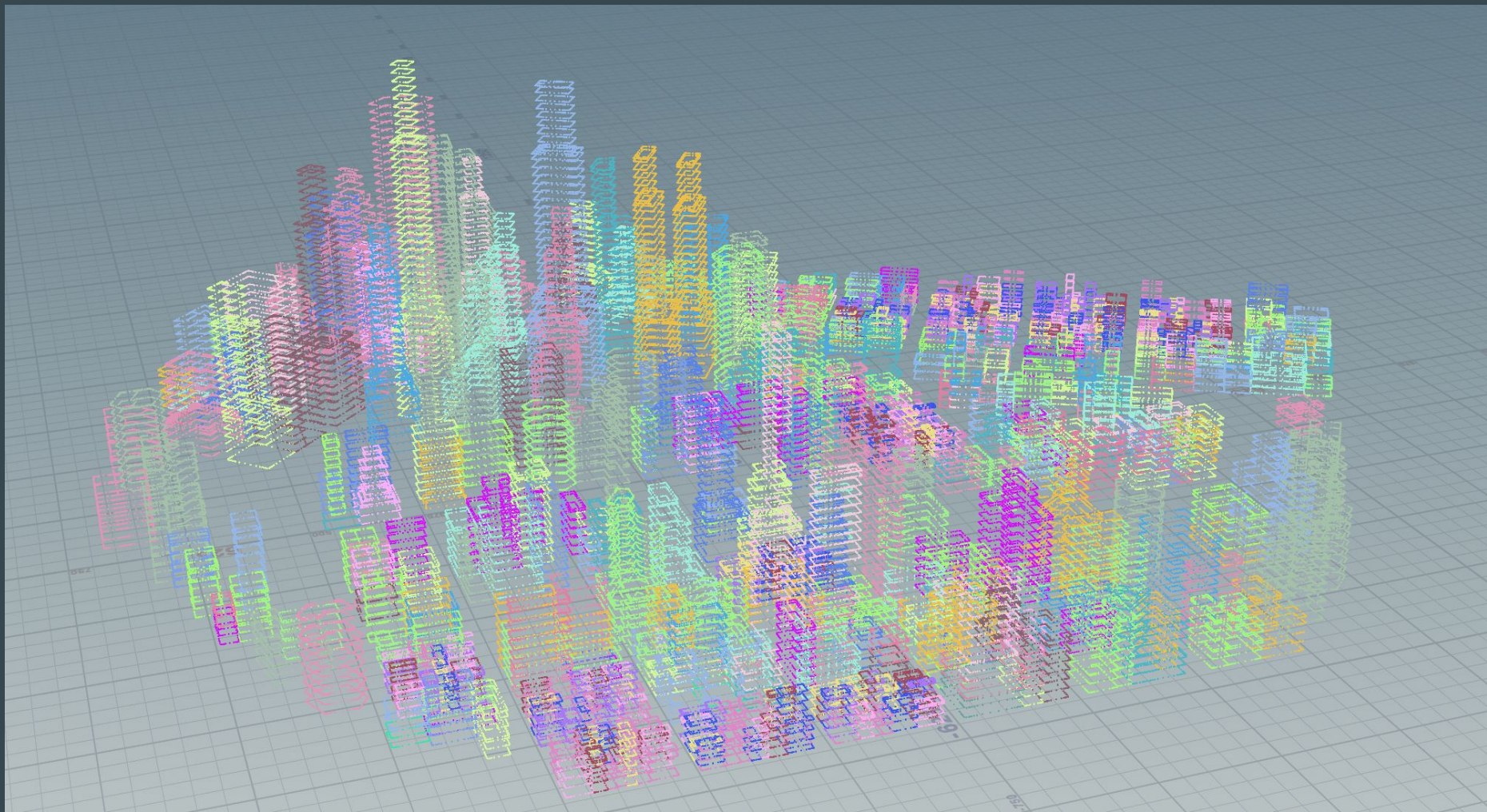
Settings	
Operator	!=
Target Attribute	mesh_id
Use Constant Threshold	<input checked="" type="checkbox"/>
Type	Integer 32
Int 32Value	0

# Matrix Demo City Sample

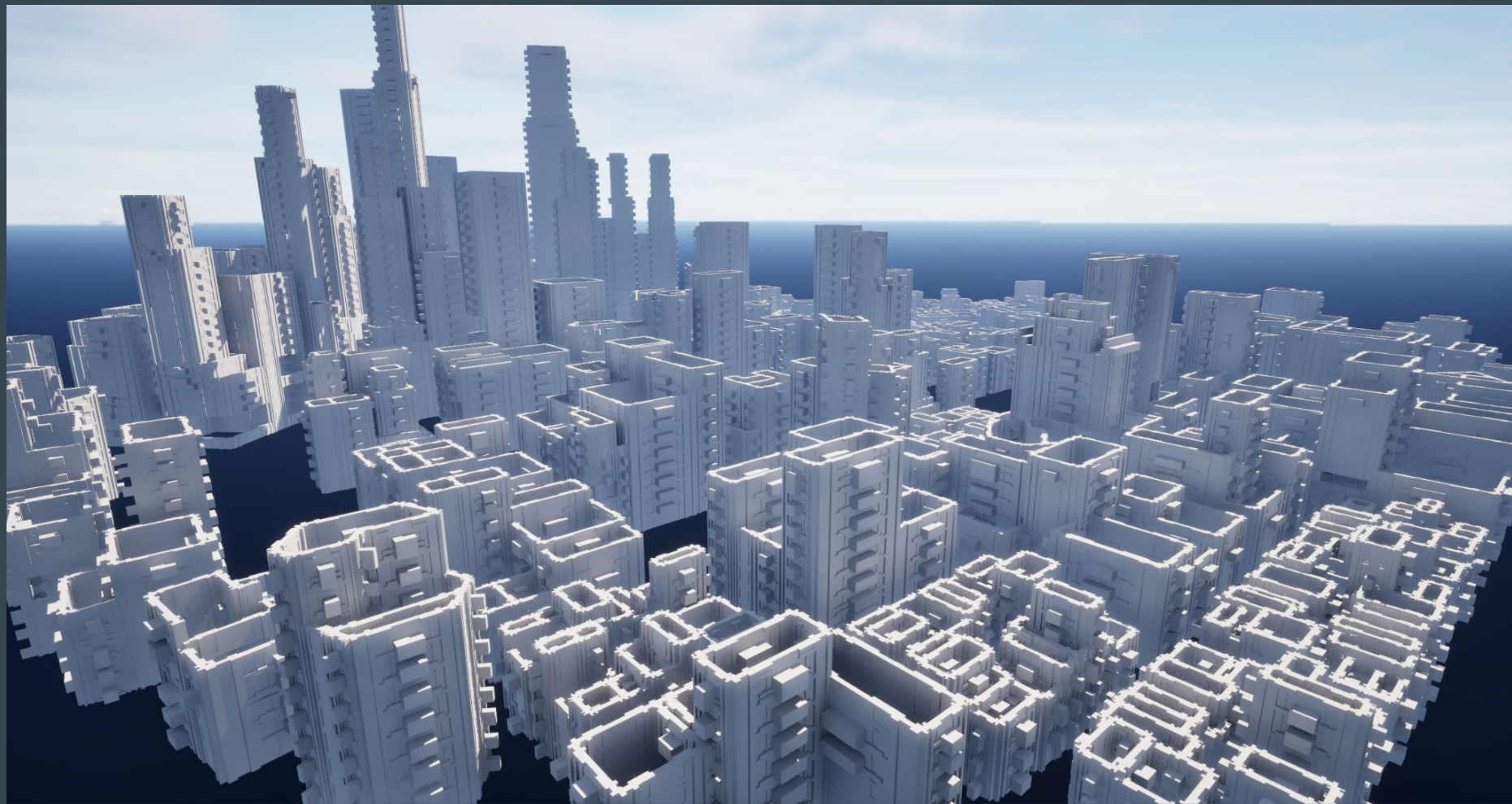
...













# Alembic Format

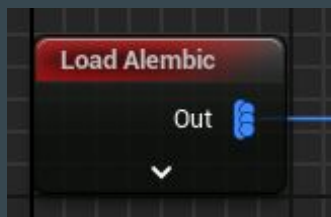
...

# Alembic Export Settings



@orient has to be normalized  
If it is greater than 0 to 1 value  
The engine crashes on import.


# Alembic Import Settings



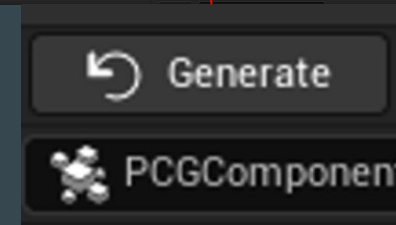
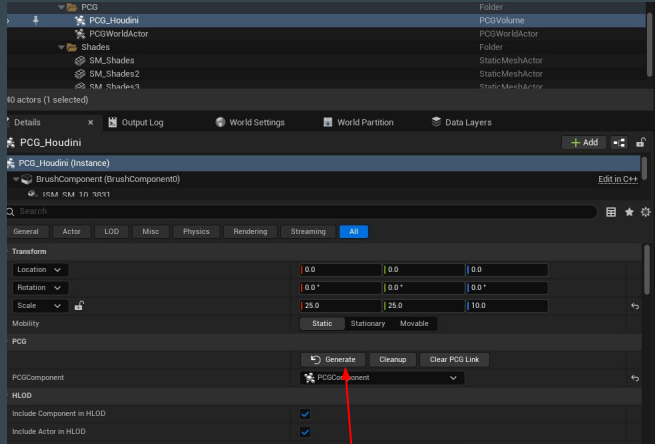
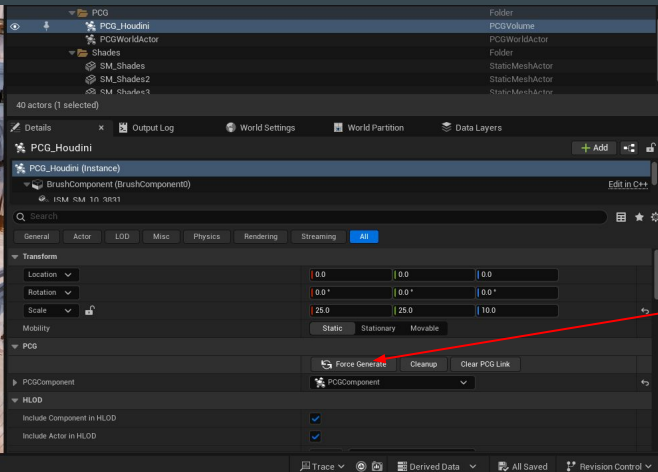
▼ Alembic			
Alembic File Path	../../../../Users/Mo/Documents/SideFX/I ... ↩		
▶ Conversion Scale	1.0	1.0	1.0 ↩
▶ Conversion Rotation	0.0	0.0	0.0
Conversion Flip Handedness	<input checked="" type="checkbox"/>		↩
Setup from standard	None ▼		
▼ Settings	None		
▼ Attribute Mapping	3 Map elements + - ↩		
position	\$Position.xzy	+	▼ ↩
scale	\$Scale.xzy	+	▼ ↩
orient	\$Rotation.xzyw	+	▼ ↩

# Alembic Re-import

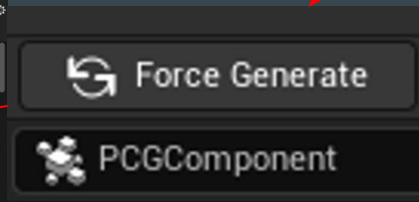
Use `pcg.FlushCache` command to force Alembic to reload file.

 Cmd `pcg.FlushCache`

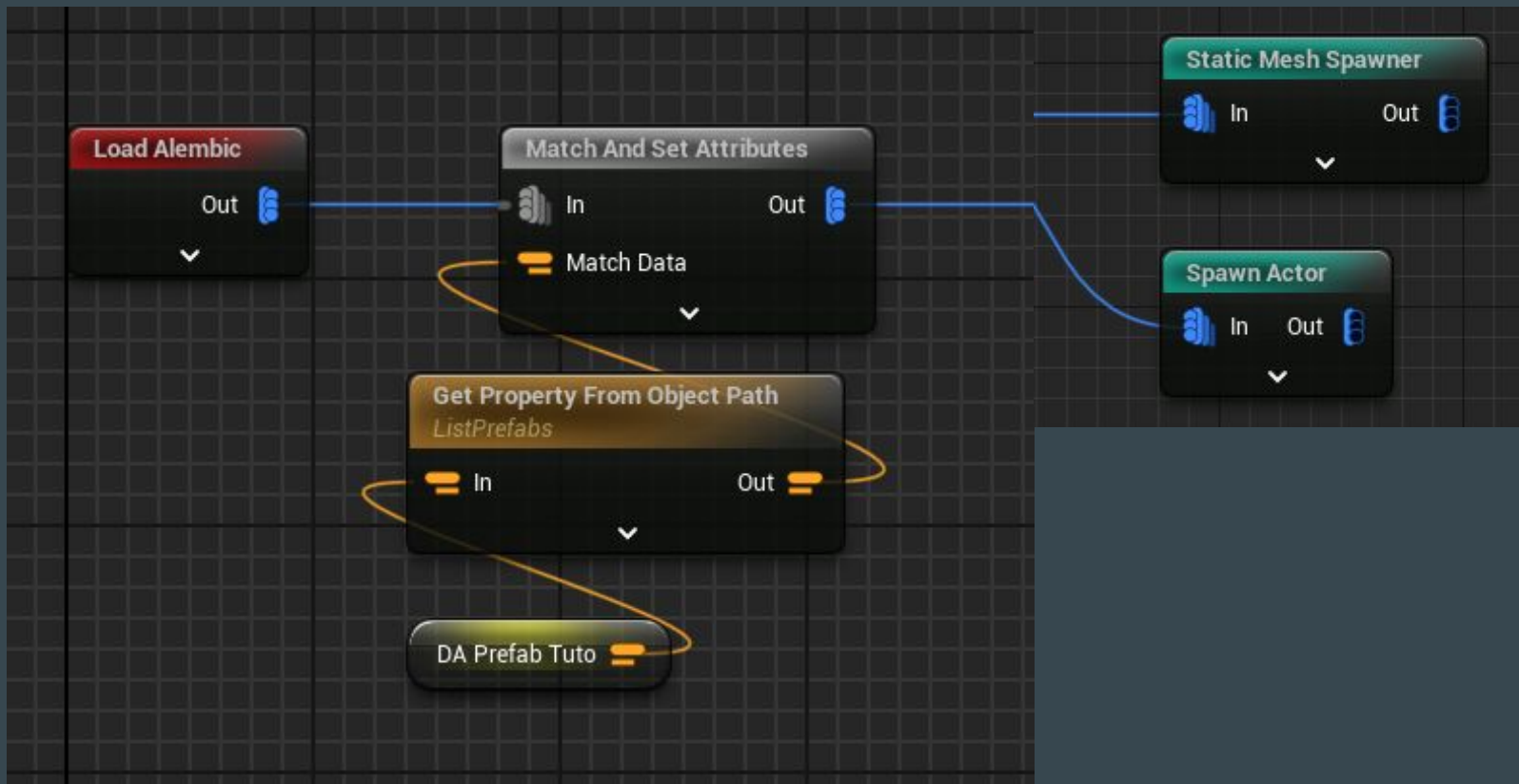
Or select the PCG graph in the outliner and hold Ctrl for the Generate button to turn into Force Generate.



Hold Ctrl



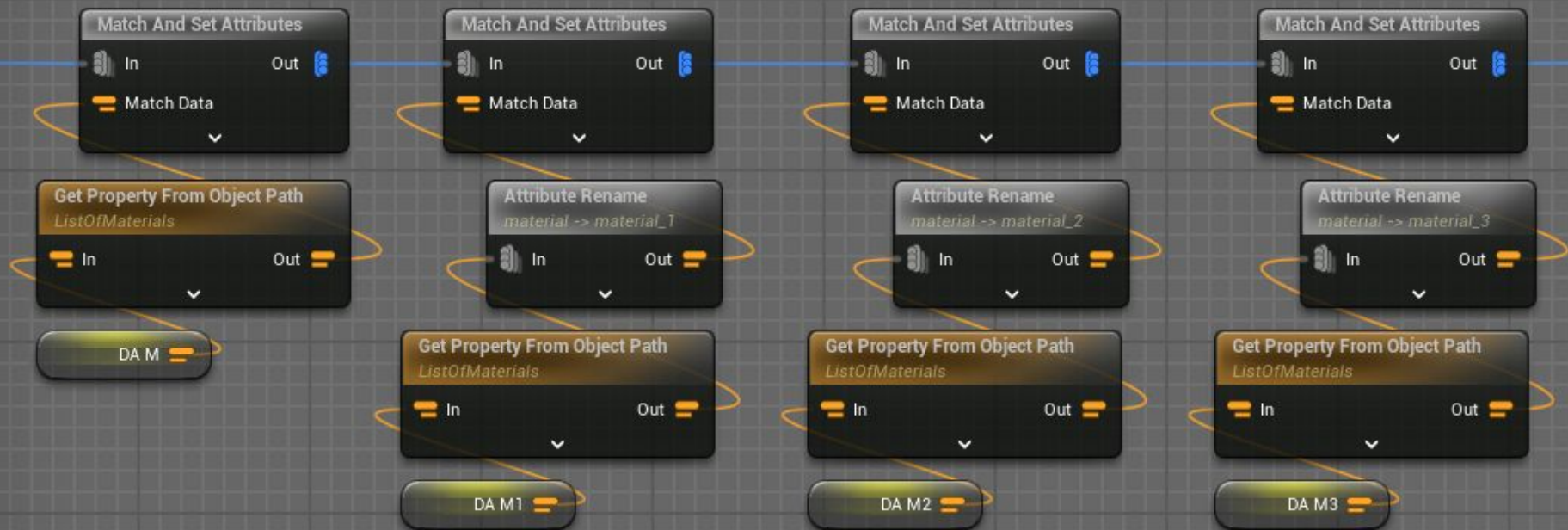
# Alembic PCG Setup



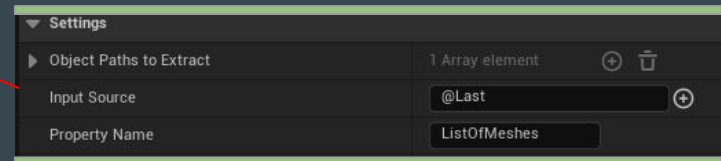
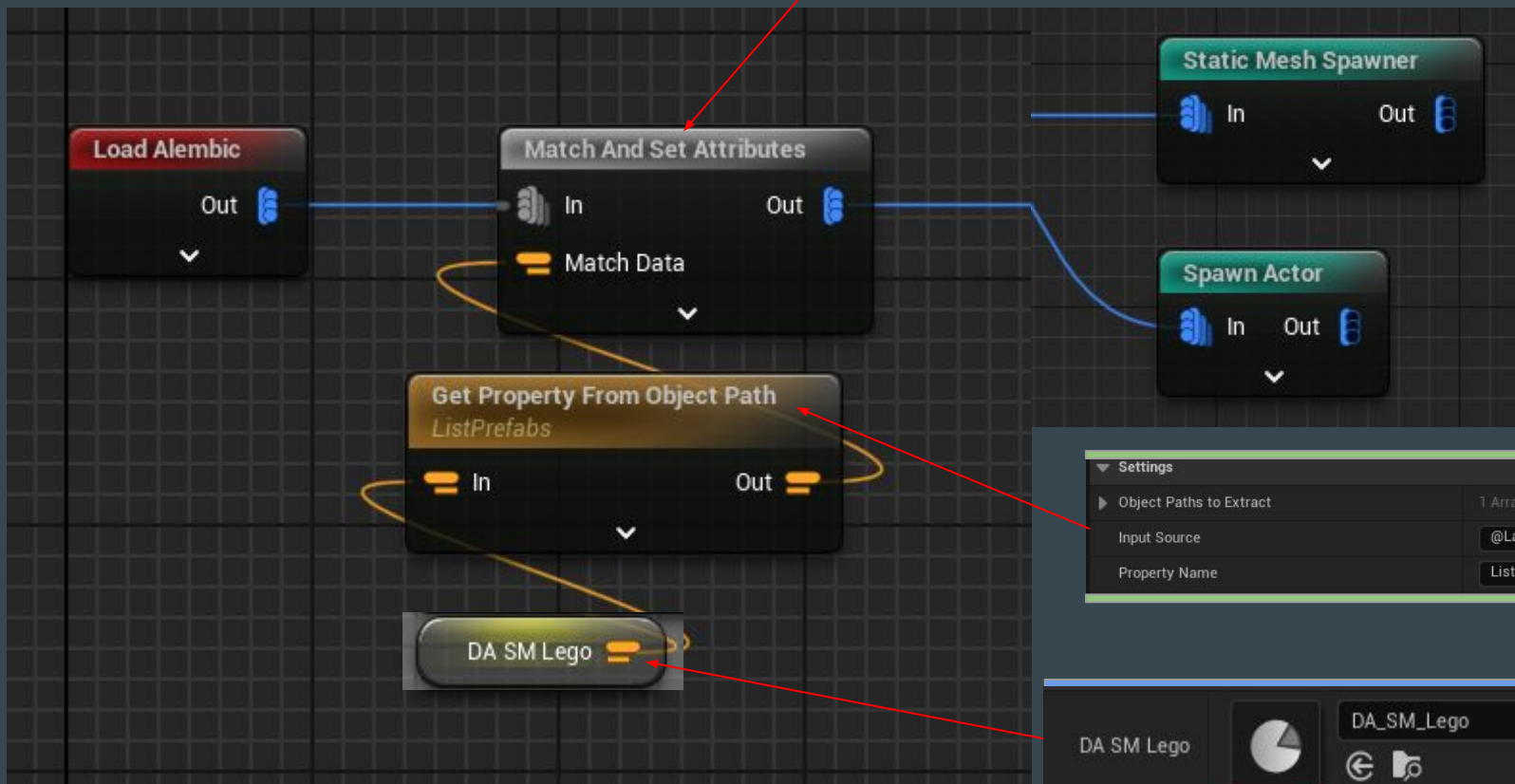
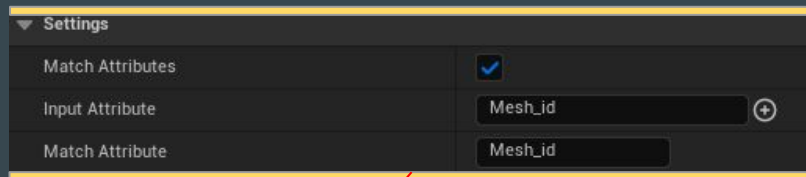


# Multi-Material Overrides using Alembic and Data Assets

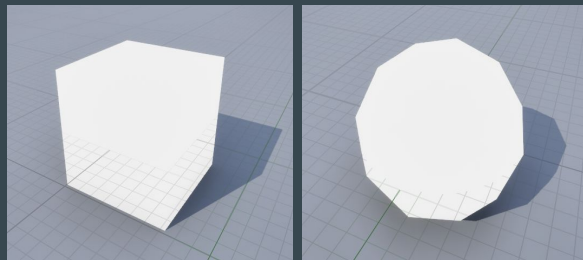
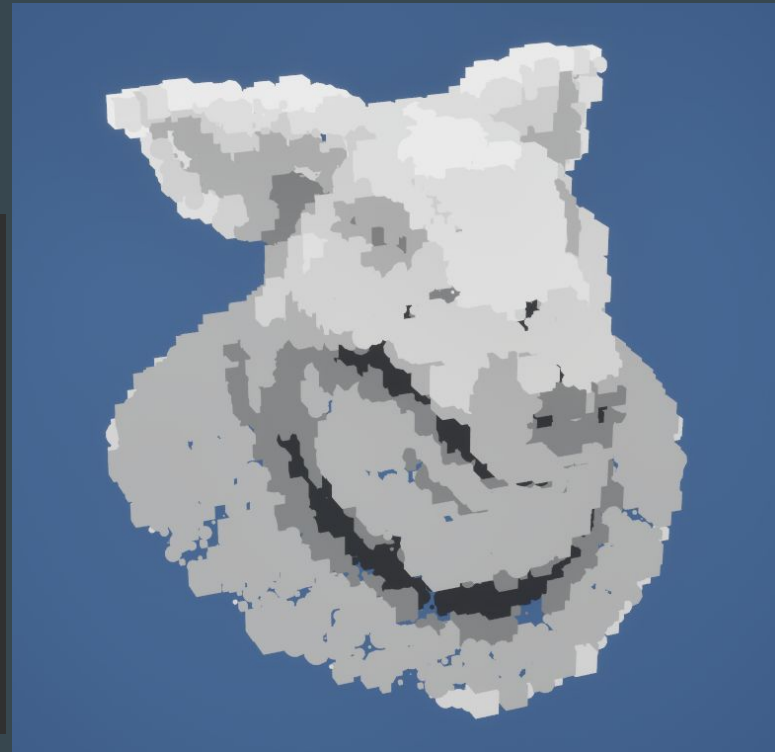
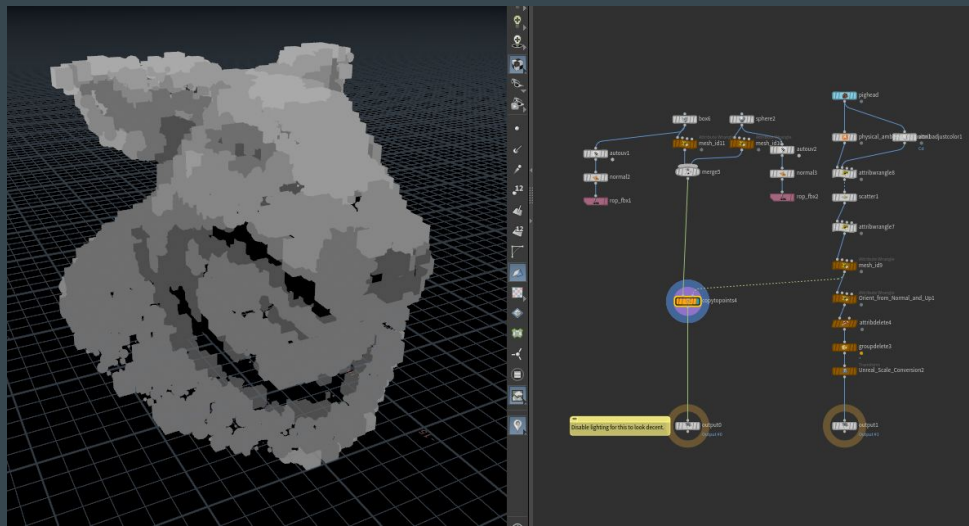
## Material Overrides



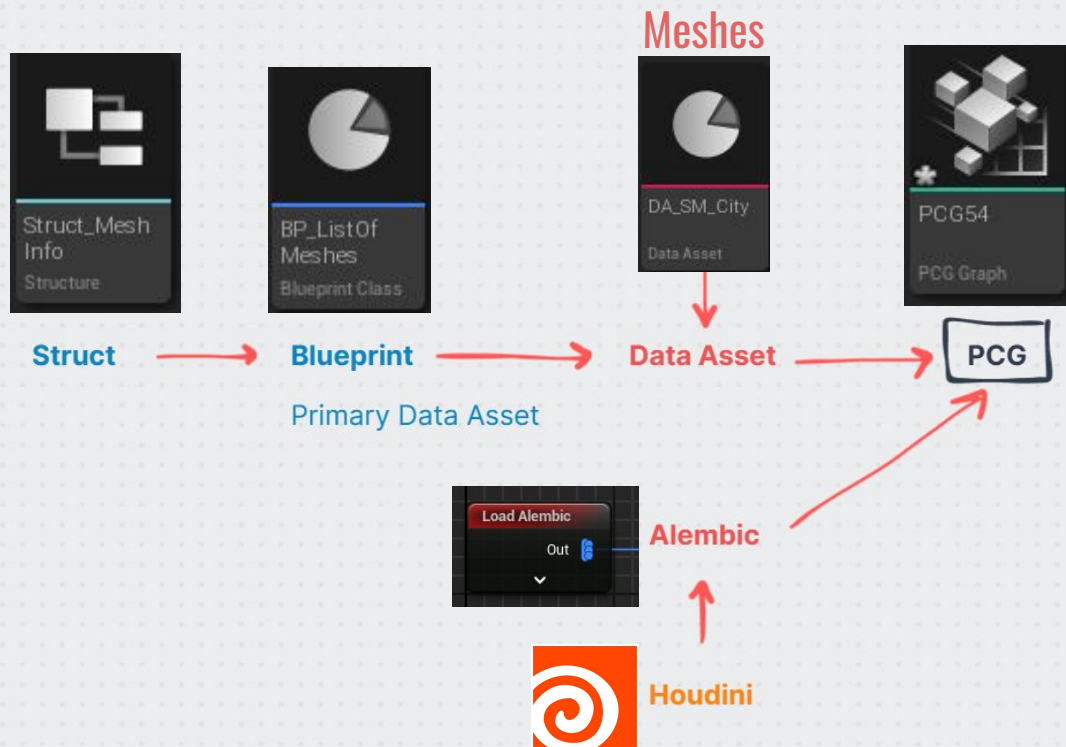
# Alembic Mesh



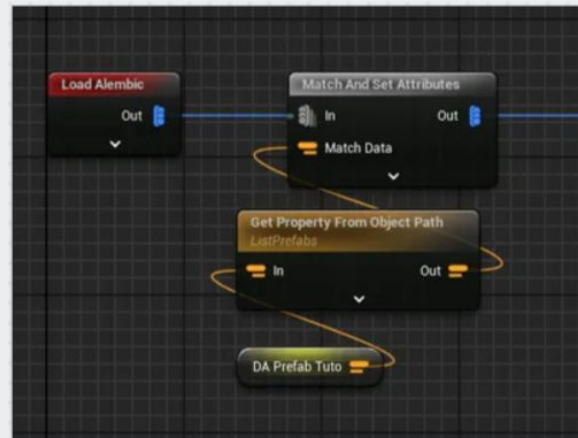
## Stylized Example



# Alembic PCG Data Asset format



UE5.4+

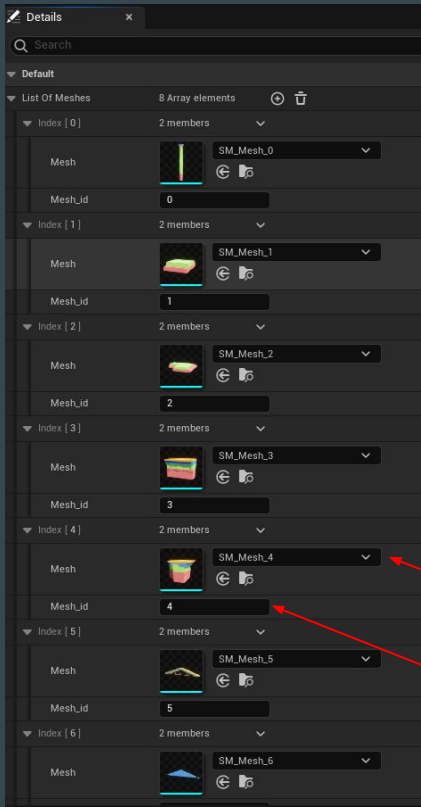


# Alembic PCG Data Asset format

Data Assets  
for Alembic  
PCG:

Mesh  
(Static Mesh)

Mesh\_id  
(integer)



Attributes

Determinism

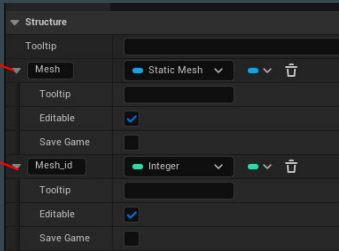
Output: Out [0] PCGPoint Data

Search

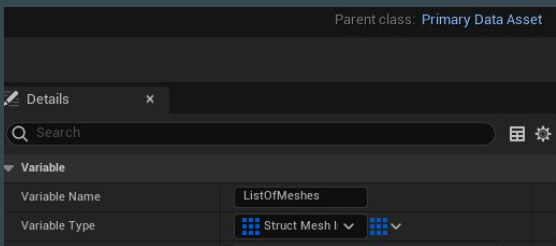
Load Alembic Last attribute: mesh\_id / Numb

s	Seed	custom	custom_1	custom_2	custom_3	material	material_1	material_2	material_3	material_4	material_5	mesh_id
0.5	0	0.417	0.935	0.331	0.408	1	-1	-1	-1	-1	-1	3
0.5	0	0.003	0.964	0.351	0.79	1	-1	-1	-1	-1	-1	3
0.5	0	0.813	0.744	0.582	0.668	1	-1	-1	-1	-1	-1	3
0.5	0	0.262	0.697	0.607	0.597	1	-1	-1	-1	-1	-1	3
0.5	0	0.002	0.877	0.949	0.88	1	-1	-1	-1	-1	-1	3
0.5	0	0.189	0.448	0.261	0.048	1	-1	-1	-1	-1	-1	3
0.5	0	0.116	0.143	0.951	0.258	1	-1	-1	-1	-1	-1	3
0.5	0	0.618	0.08	0.181	0.161	1	-1	-1	-1	-1	-1	3
0.5	0	0.916	0.534	0.215	0.02	1	-1	-1	-1	-1	-1	3
0.5	0	0.52	0.004	0.122	0.841	1	-1	-1	-1	-1	-1	3
0.5	0	0.094	0.036	0.015	0.188	1	-1	-1	-1	-1	-1	3
0.5	0	0.12	0.828	0.369	0.011	1	-1	-1	-1	-1	-1	3
0.5	0	0.974	0.111	0.151	0.463	1	-1	-1	-1	-1	-1	3
0.5	0	0.922	0.431	0.557	0.094	1	-1	-1	-1	-1	-1	3
0.5	0	0.149	0.219	0.84	0.01	1	-1	-1	-1	-1	-1	3
0.5	0	0.086	0.129	0.089	0.8	1	-1	-1	-1	-1	-1	3
0.5	0	0.805	0.706	0.848	0.354	1	-1	-1	-1	-1	-1	3
0.5	0	0.869	0.655	0.023	0.946	1	-1	-1	-1	-1	-1	3

Struct (Mesh & Mesh\_id)



Blueprint of type: Primary Data Asset

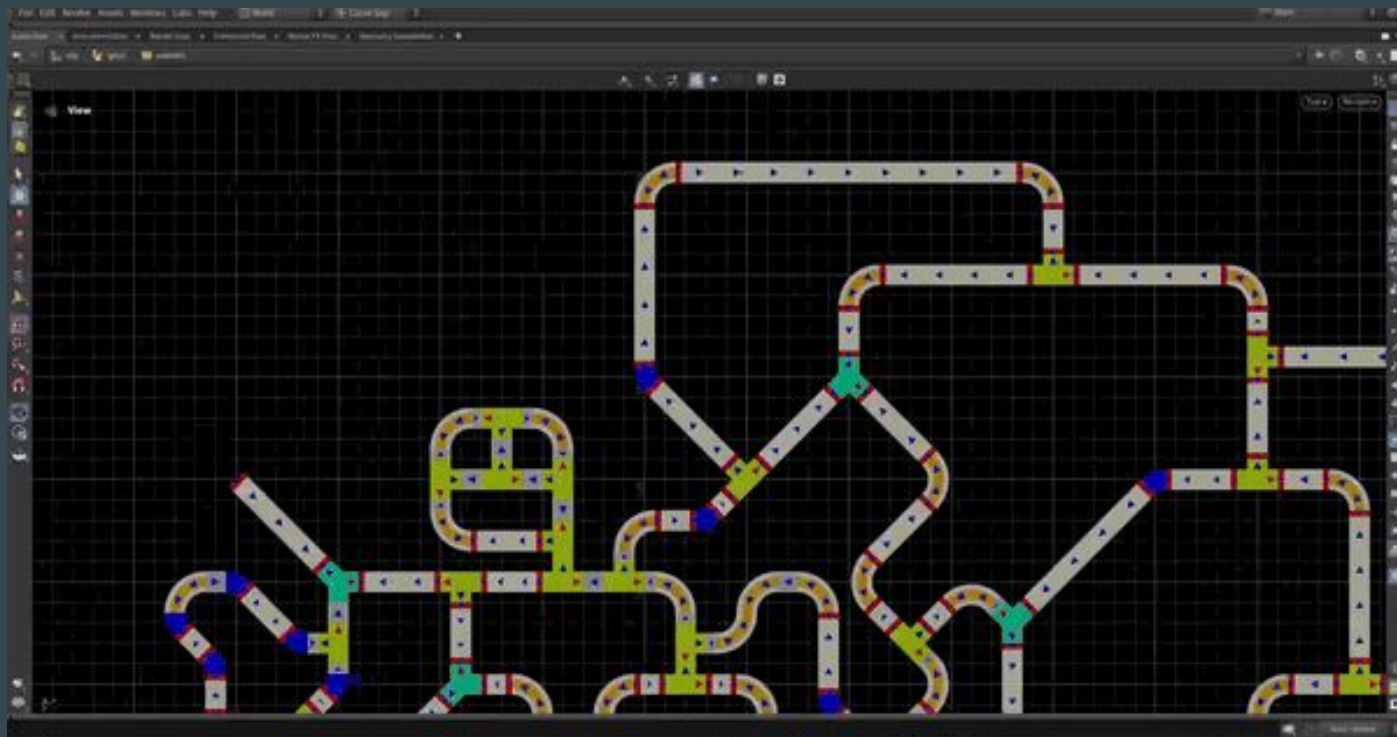




# Roads

...

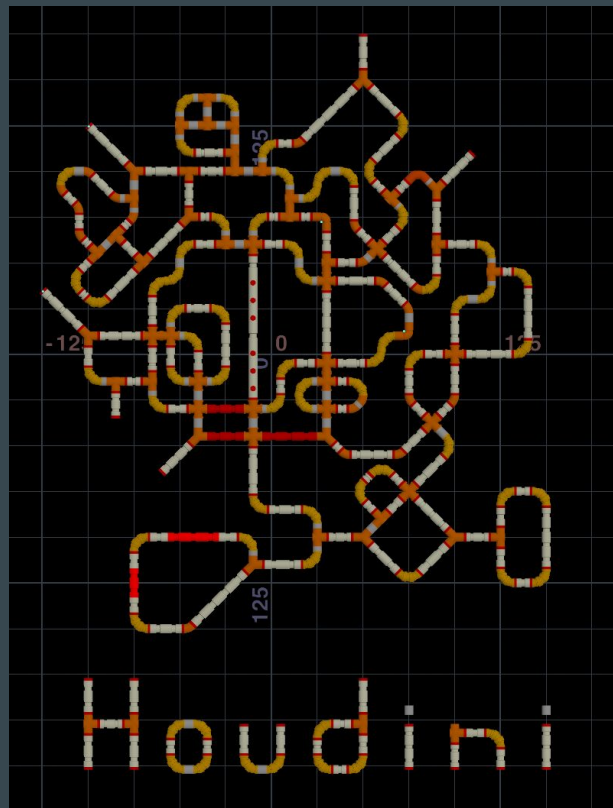
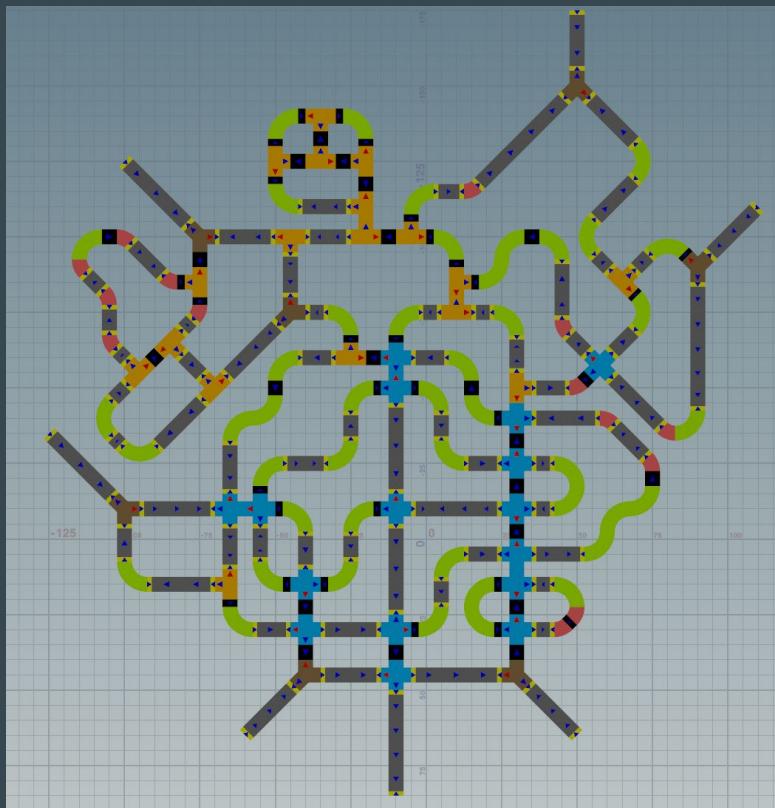
# Houdini Procedural Roads



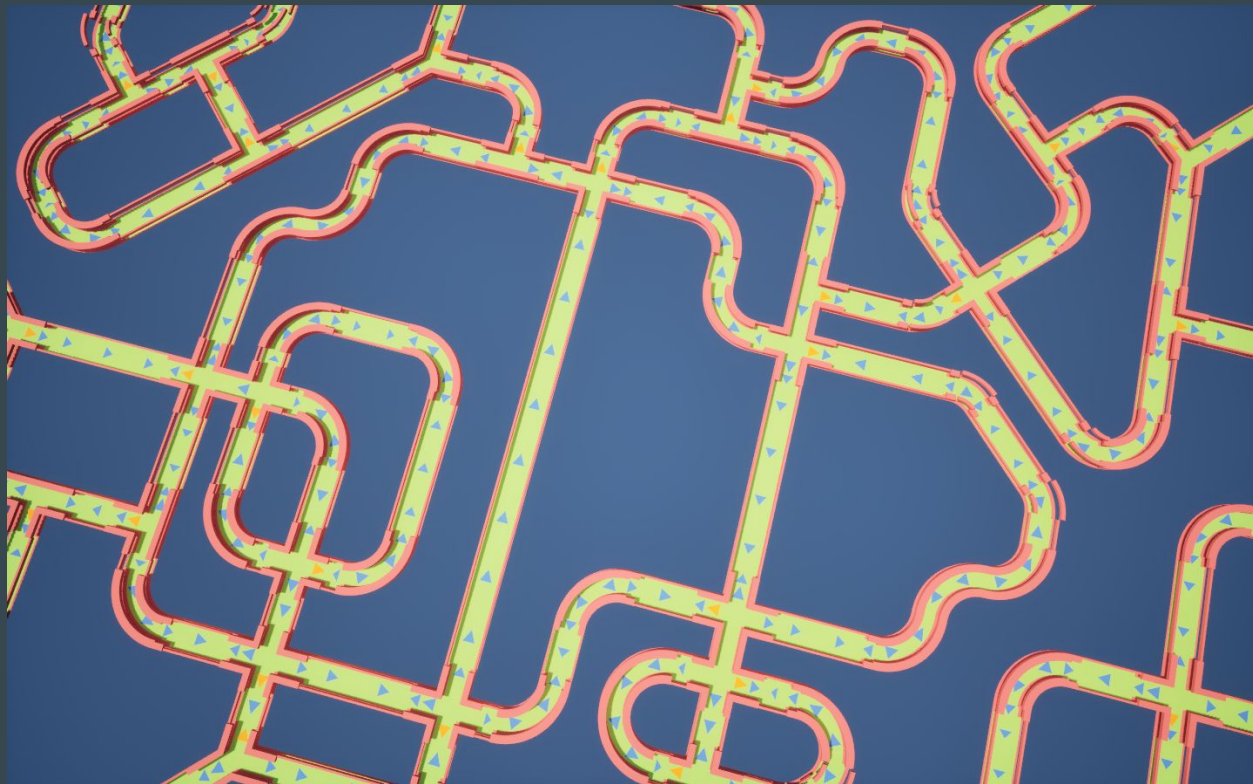
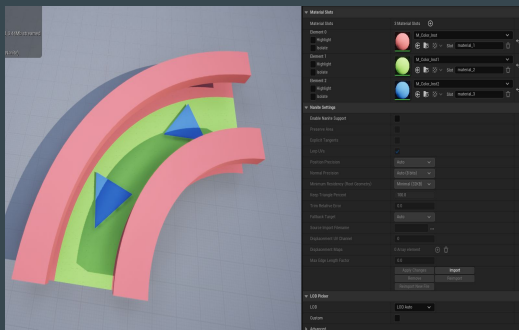
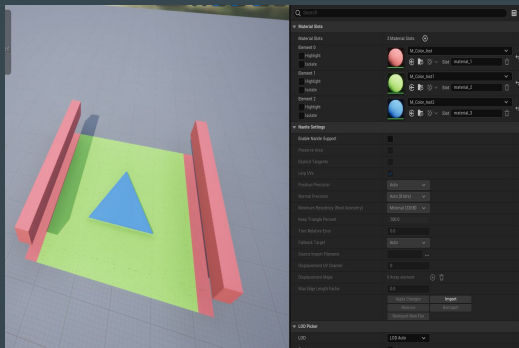
# Houdini Procedural Roads Override meshes



# Roads / Pipes

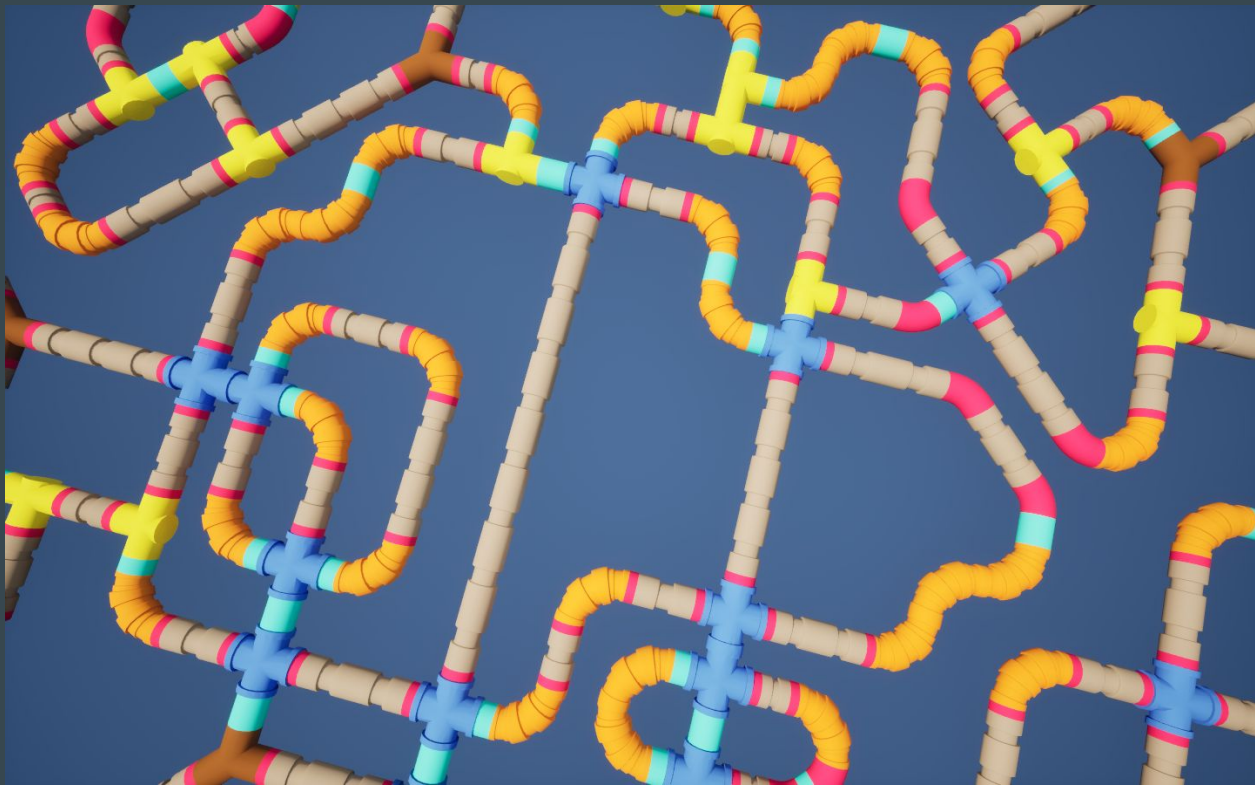
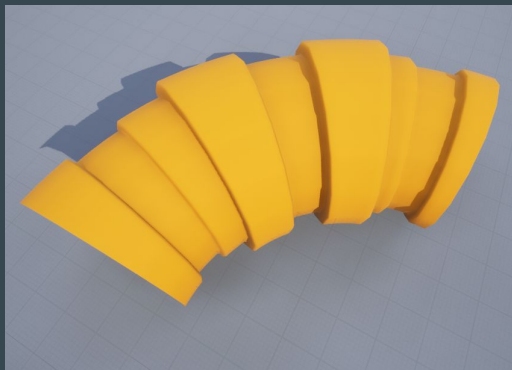
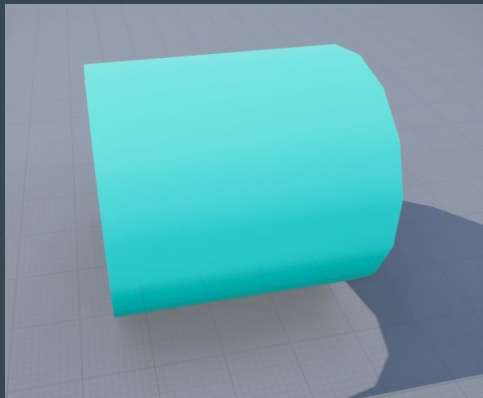


# Roads

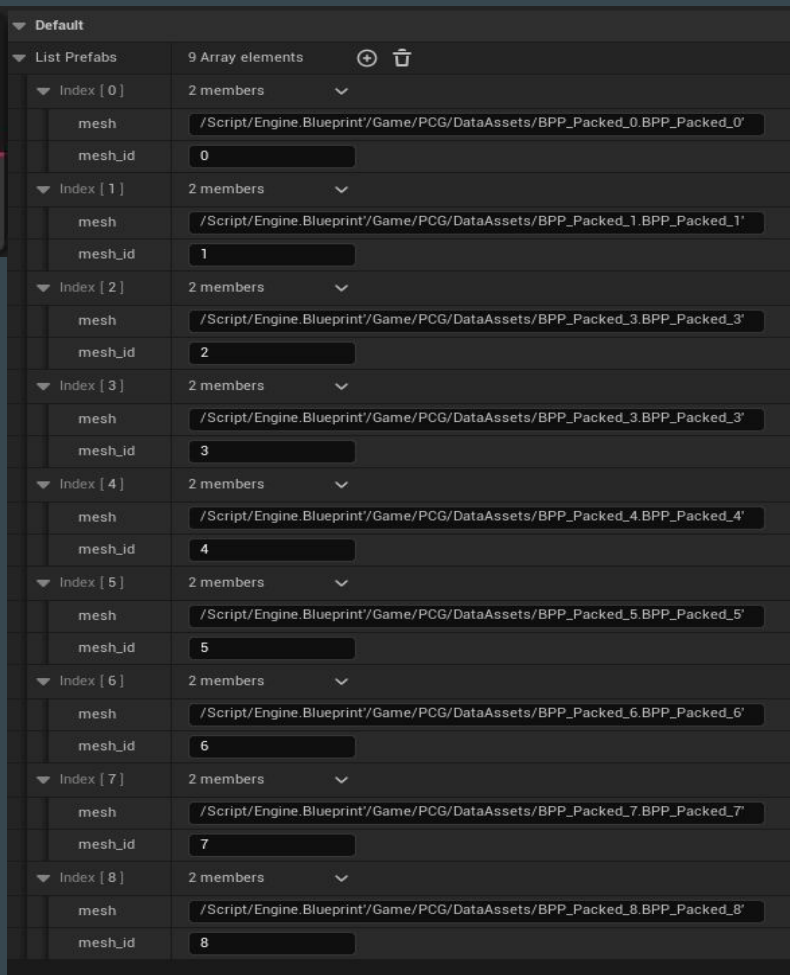
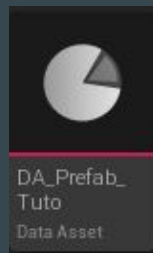
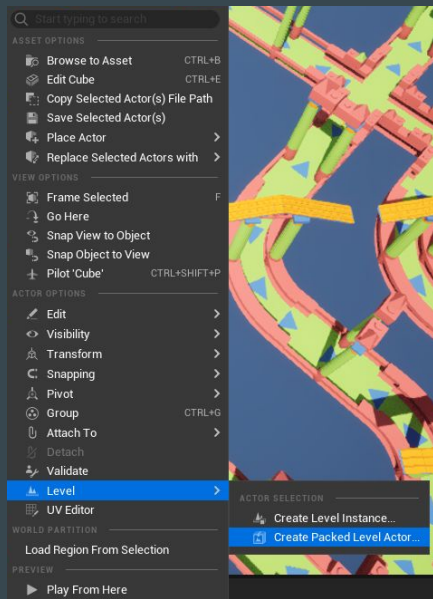
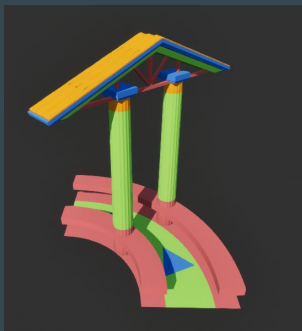
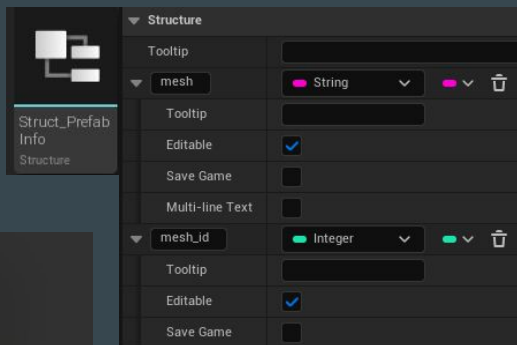




# Pipes



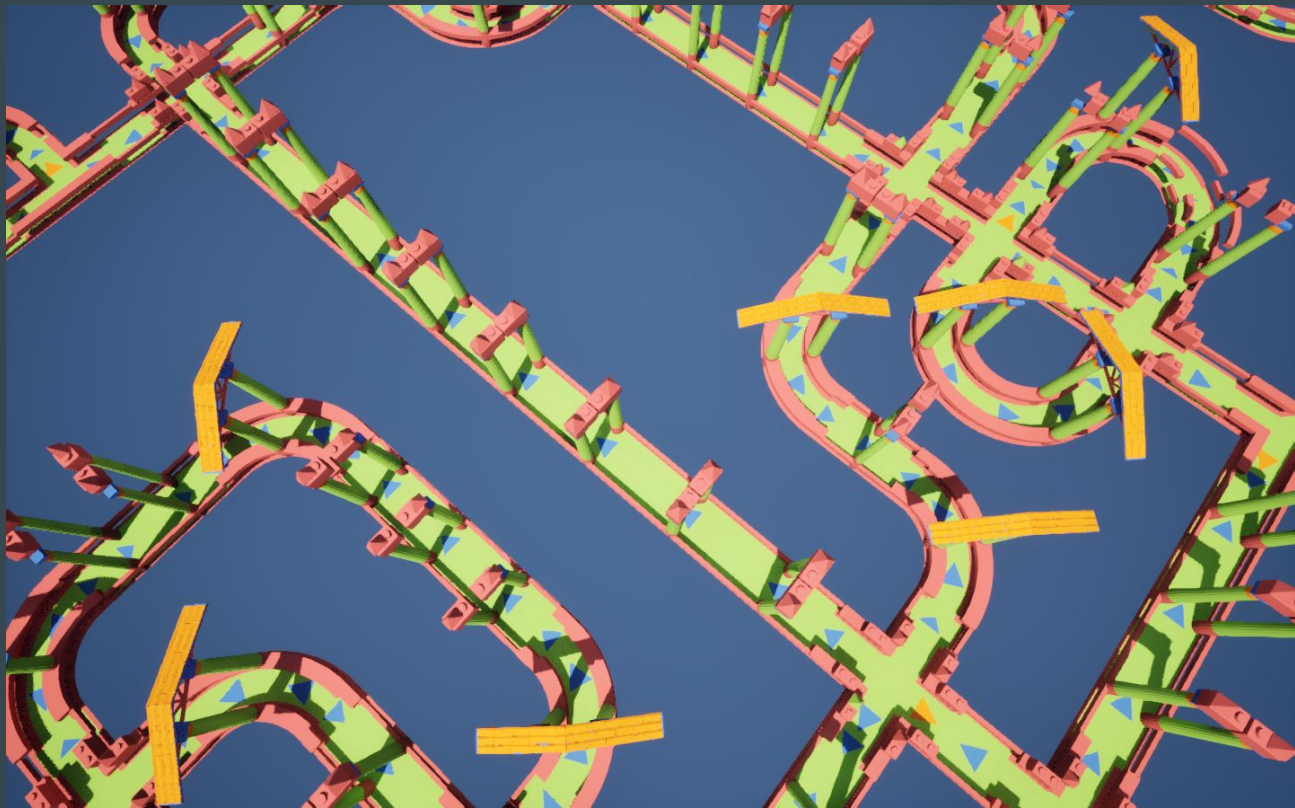
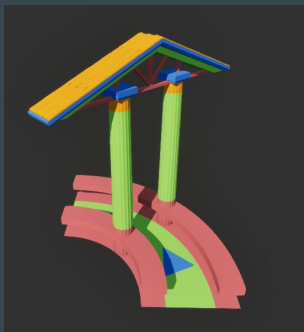
# Prefabs



# Prefabs

...

# Prefabs

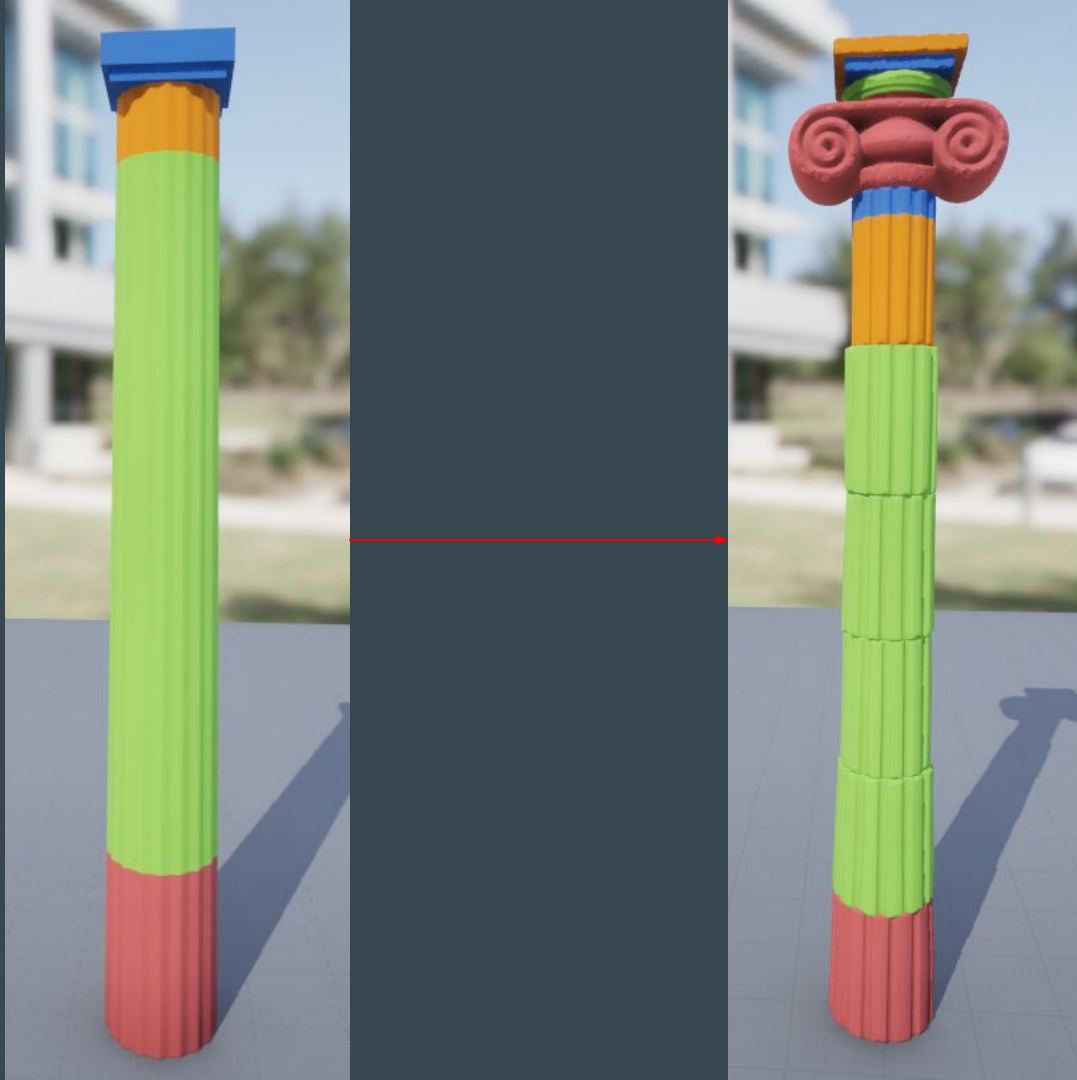


# Environment Artists

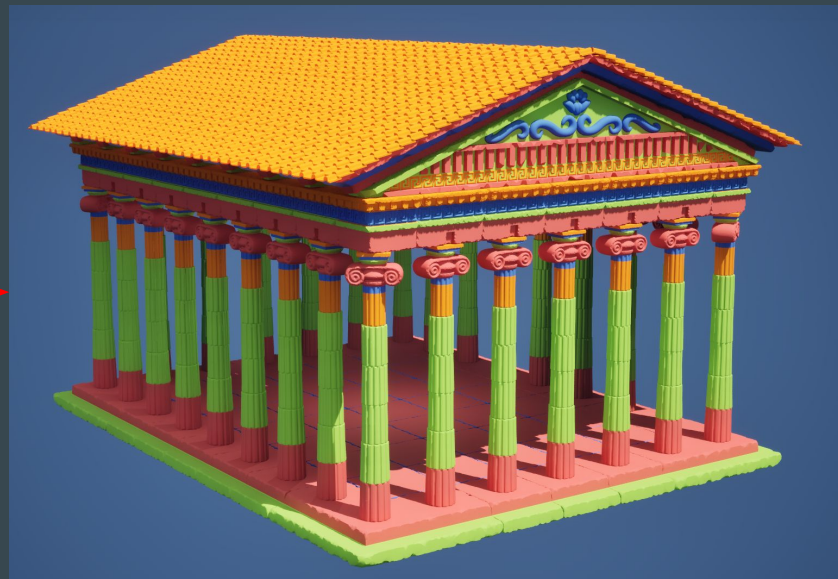
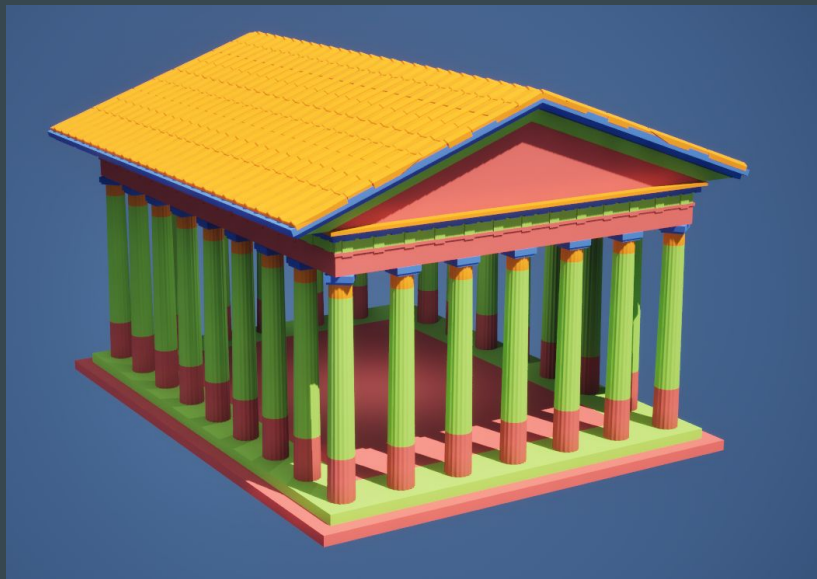
...



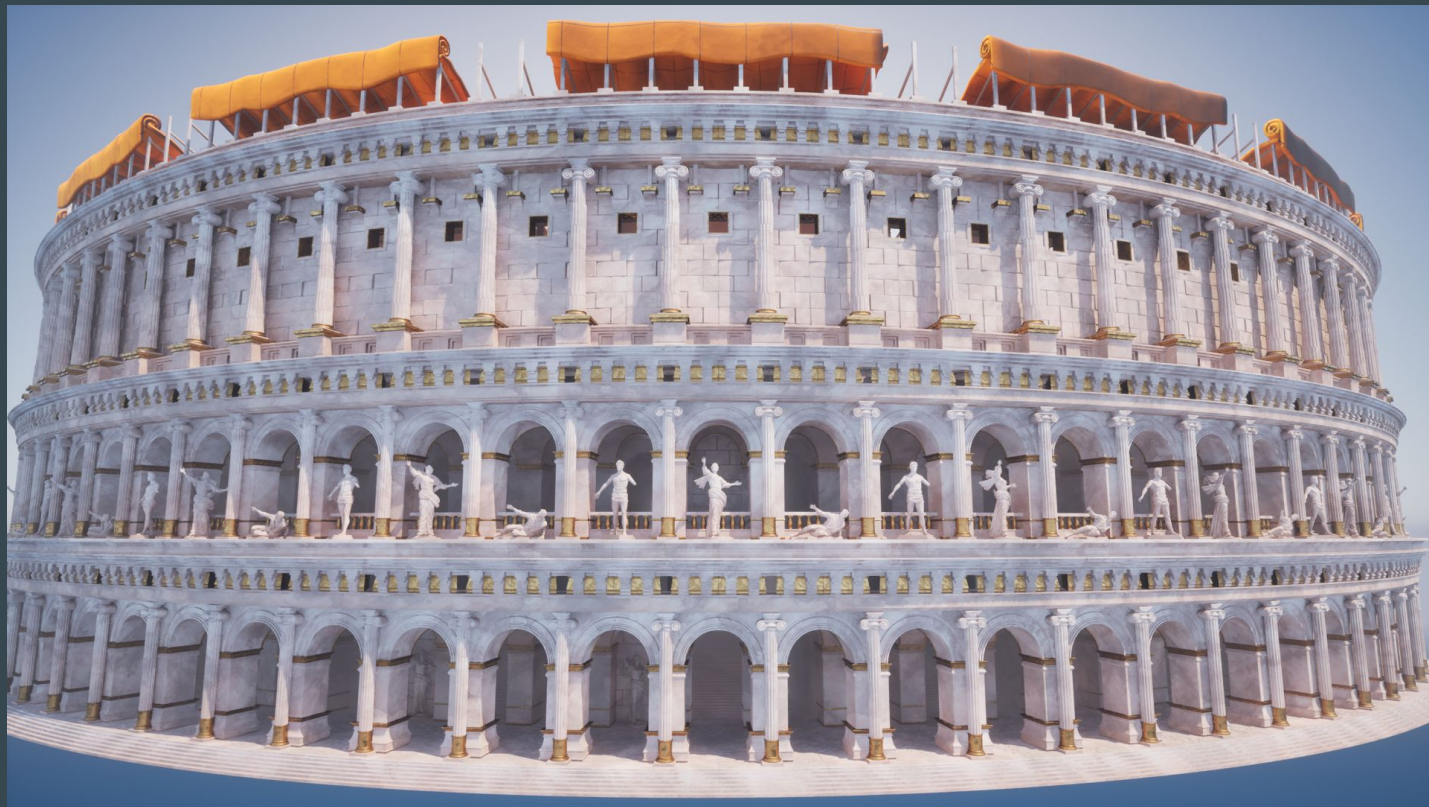
# Env Artists



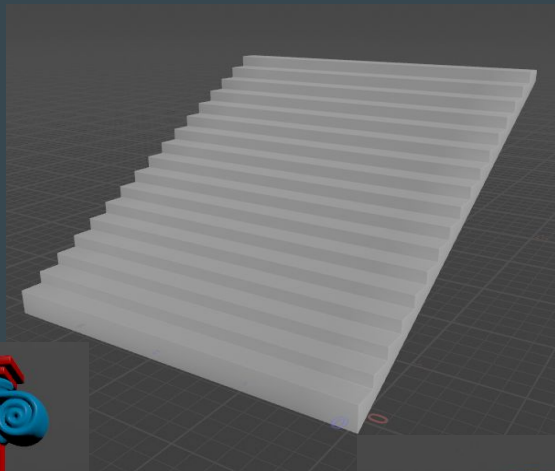
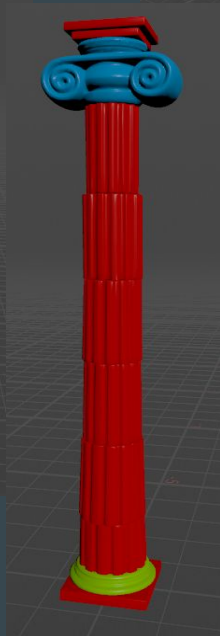
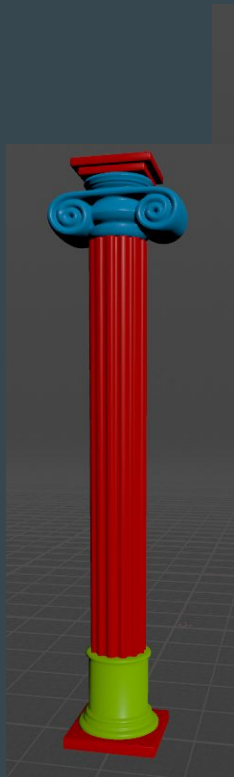
# Env Artists



# How to make a Colosseum with Houdini and Unreal PCG

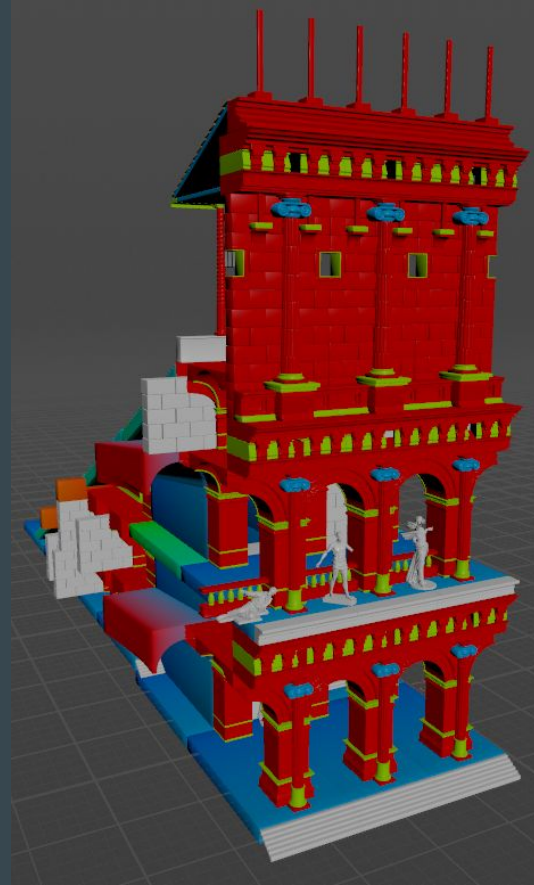
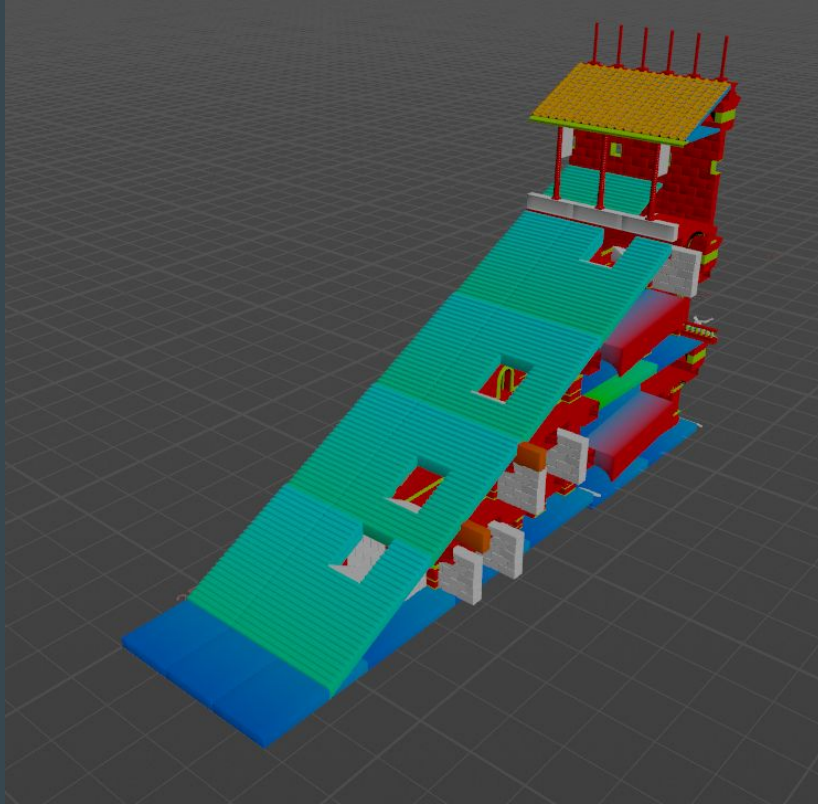


# Make Modular Meshes





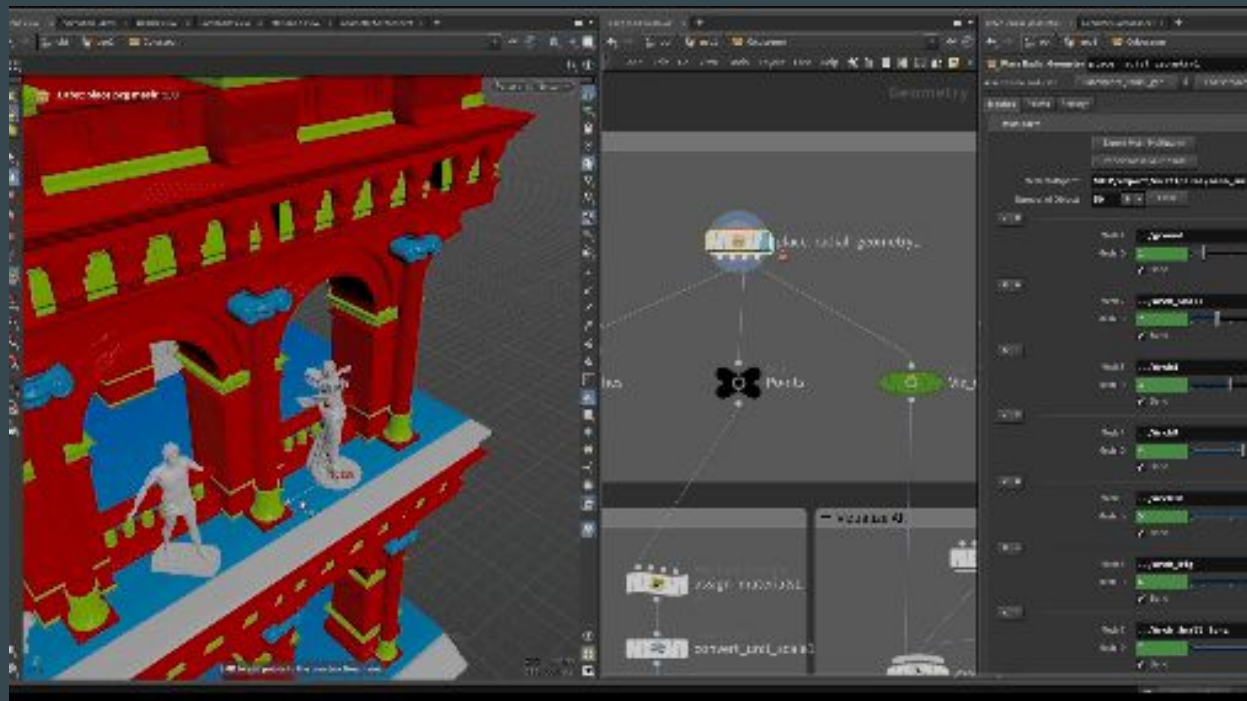
# Place Meshes to make 3 unique lanes



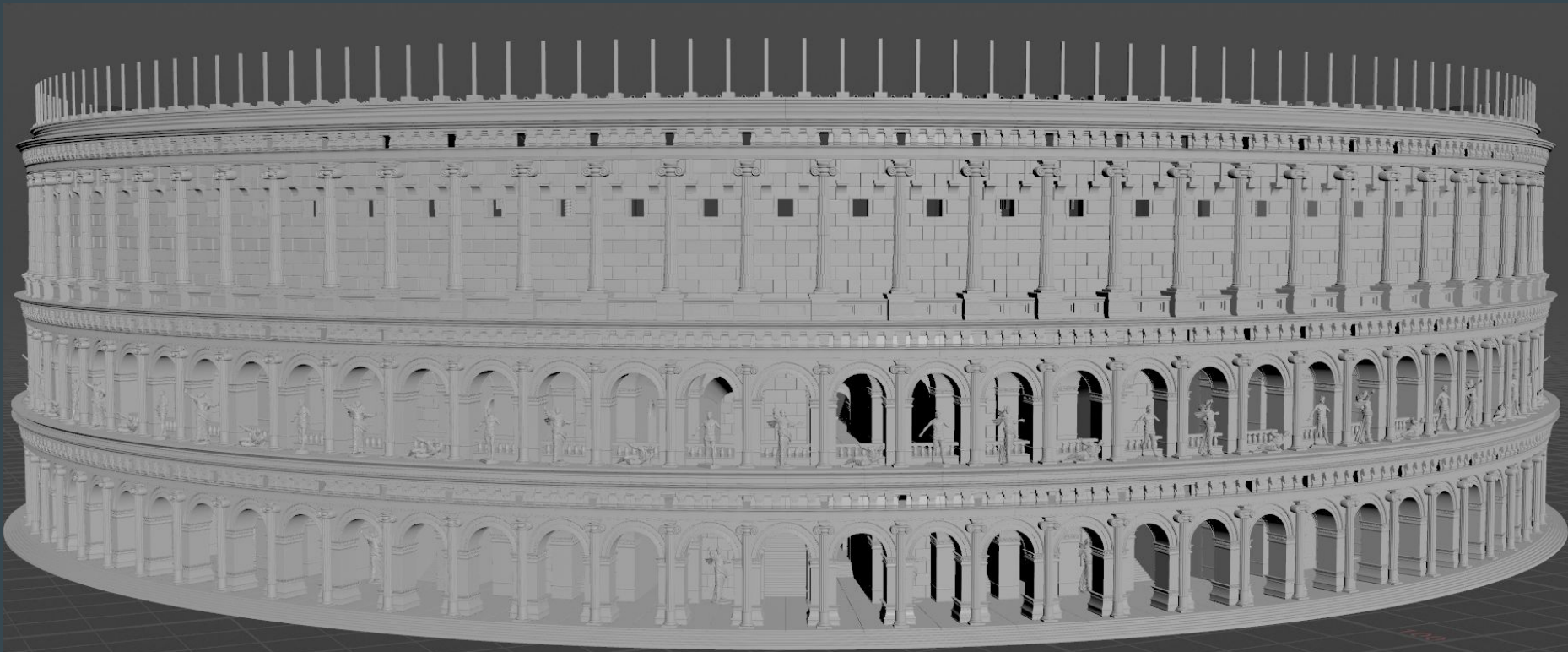


## Place Meshes to make 3 unique lanes

# You can use Python States to place meshes in Houdini Viewport.



# Bend the lanes and Instance in a circle

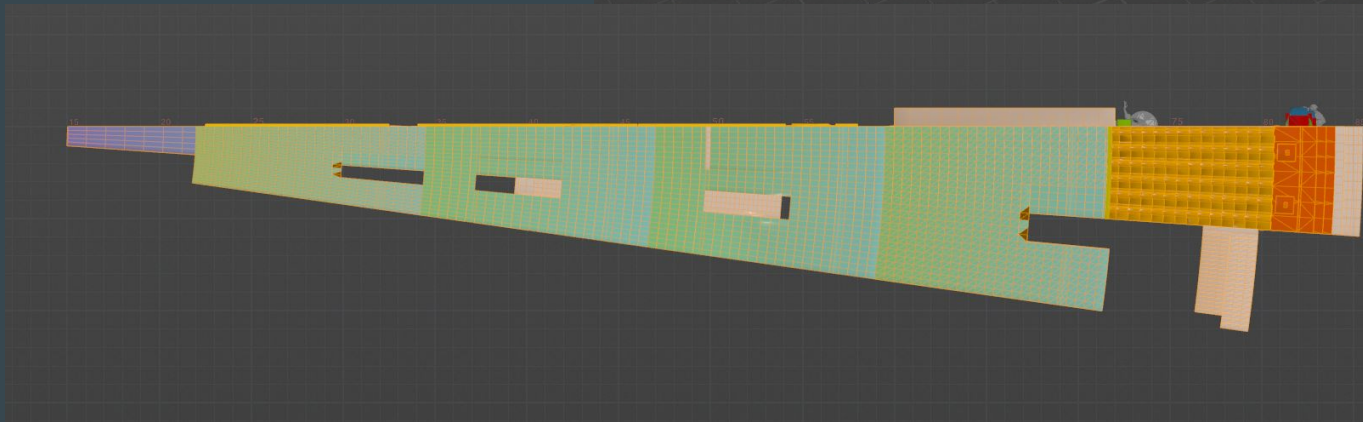
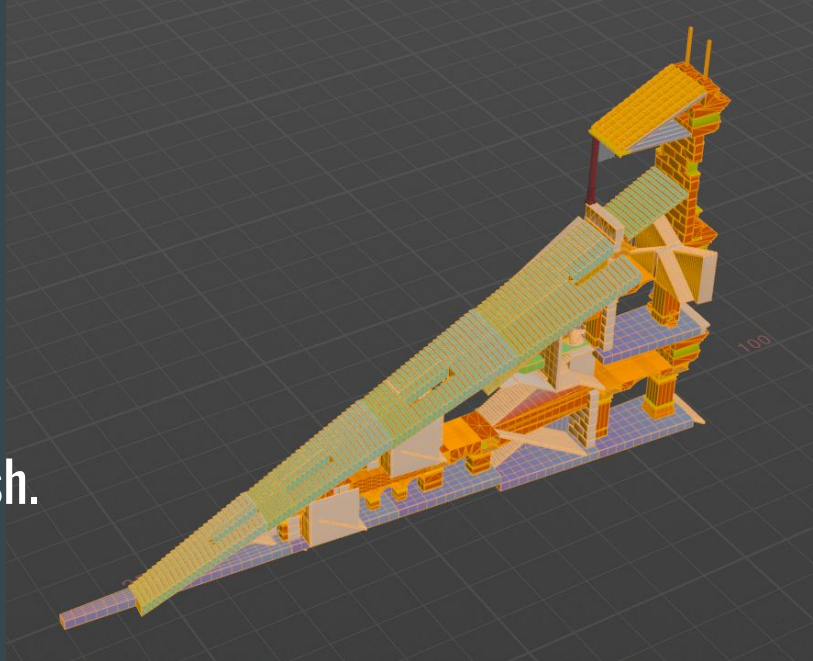


## PROBLEM after Bending:

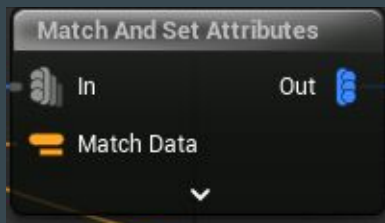
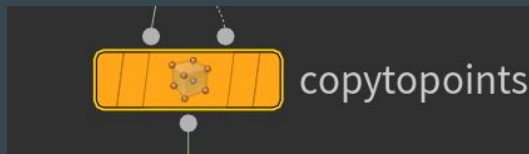
The Meshes get narrower at the center.

So moving a mesh on the x-axis will create a new mesh.

New meshes need a new id for piece attribute.



# Piece Attribute

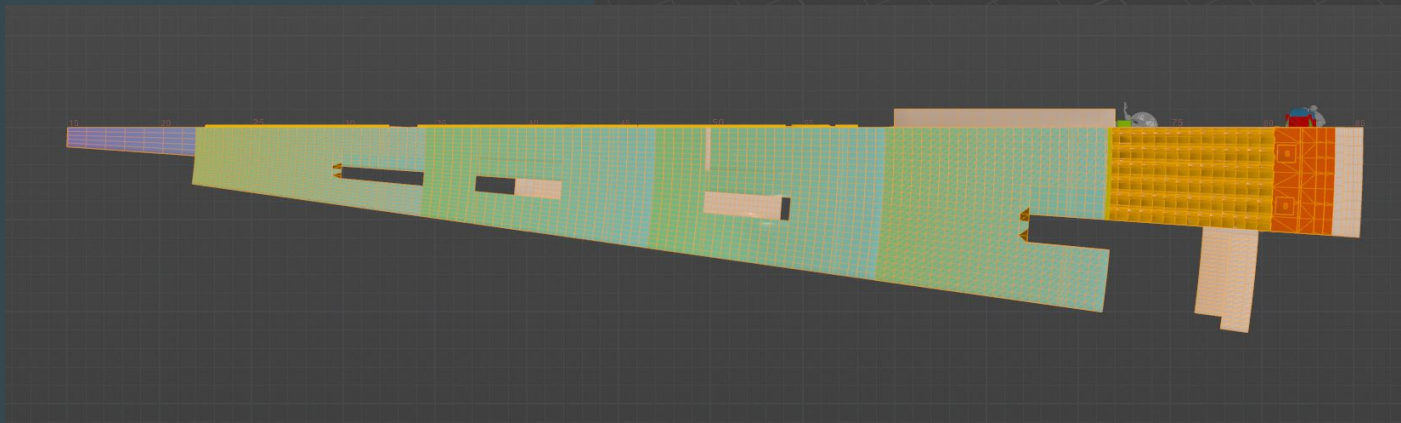
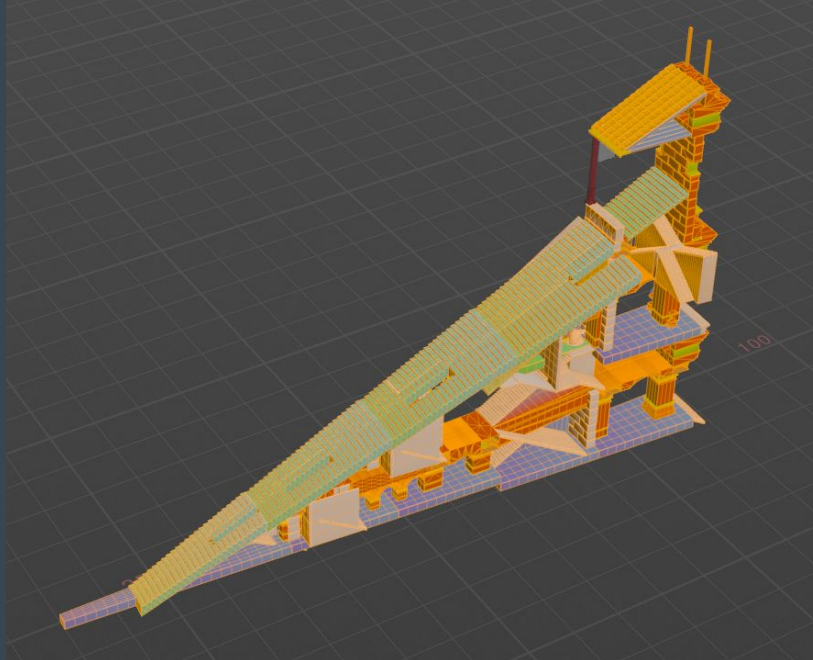


Each Mesh has an id: 1, 2, 3...

After the bend, the mesh will have variants, so..

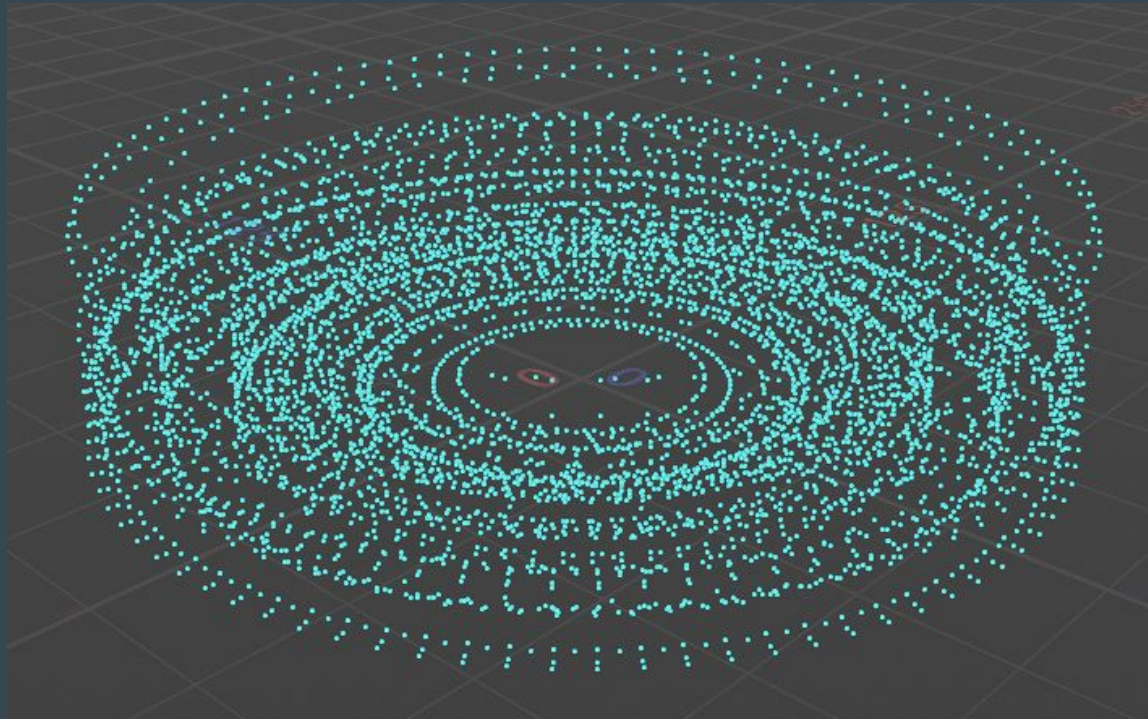
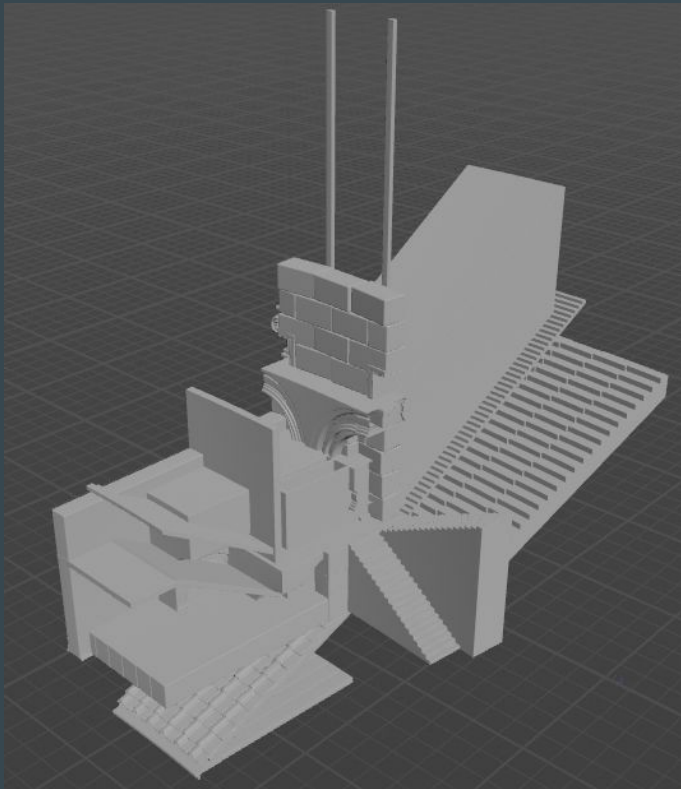
We use a string id instead and concatenate the 2nd id

Like this: 1\_1, 1\_2, 1\_3...



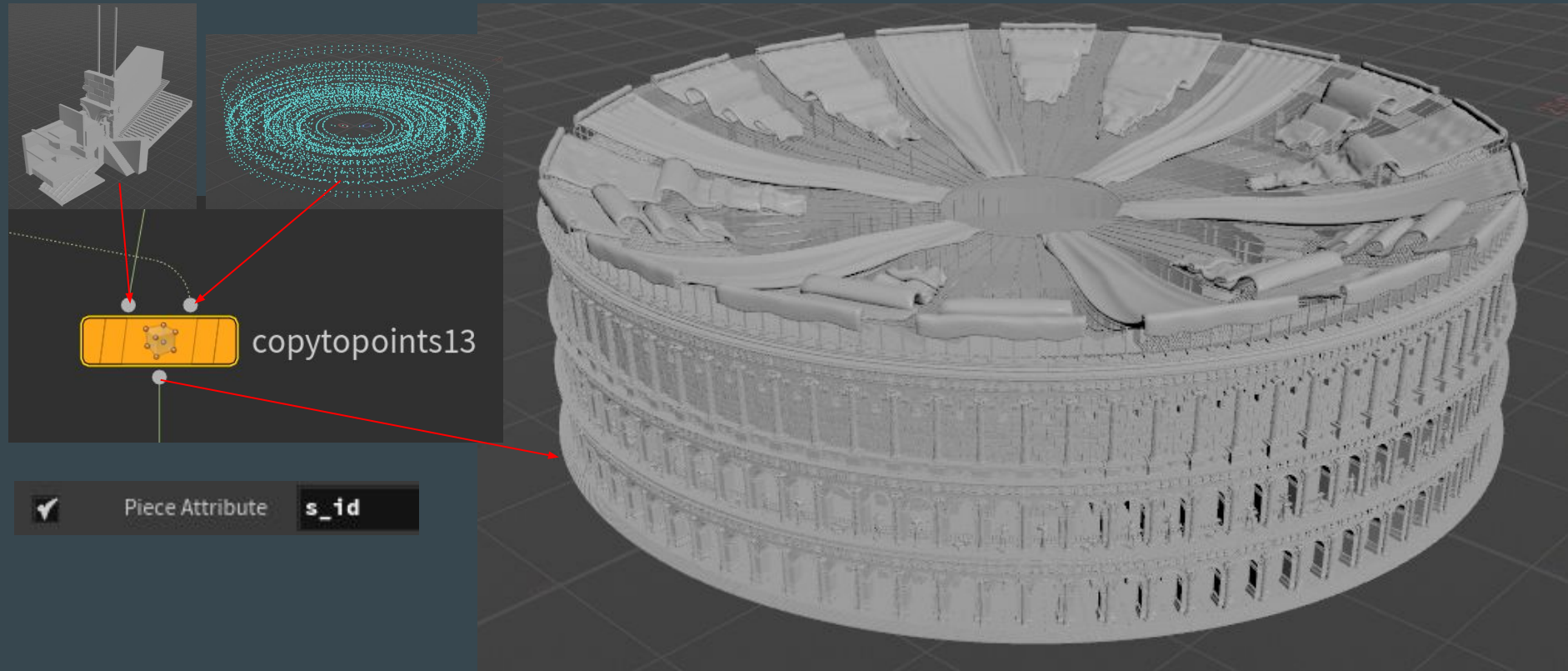


# Extract unique bent meshes and the point cloud with attributes

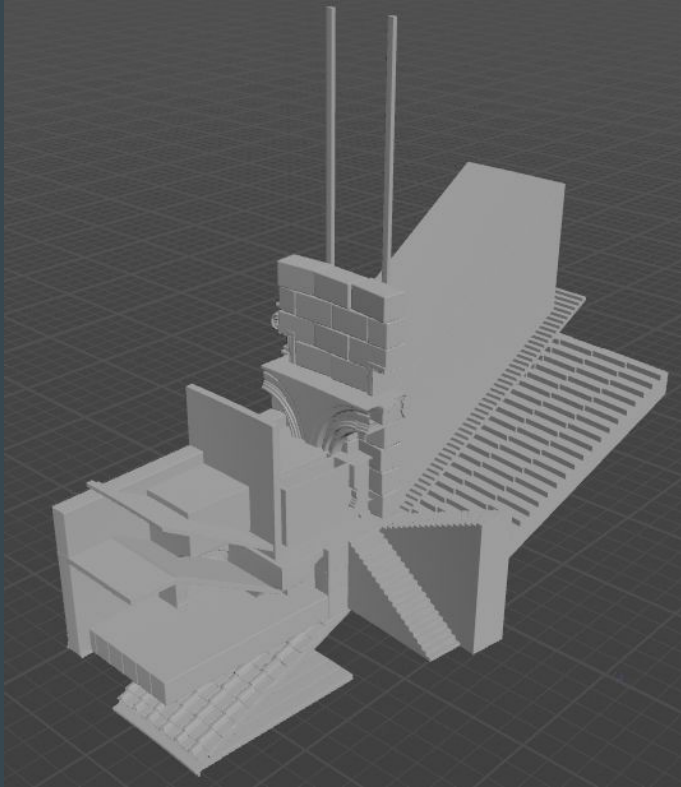




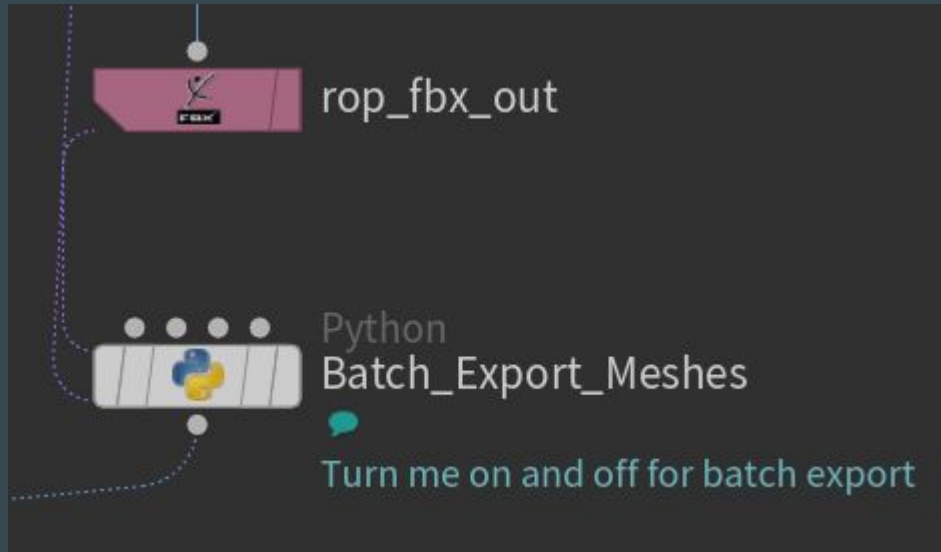
# Use Copy to Points with Piece Attribute which will be used in PCG



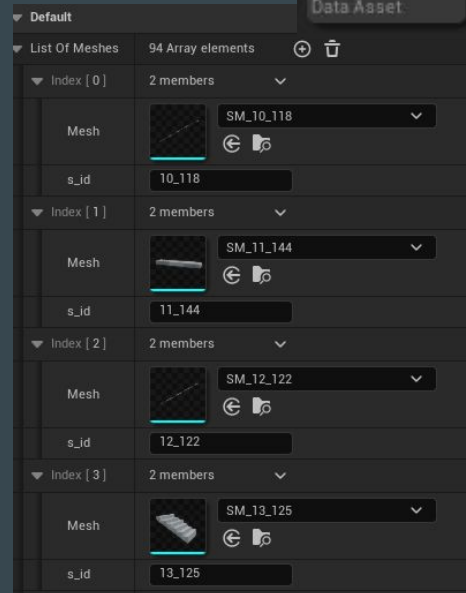
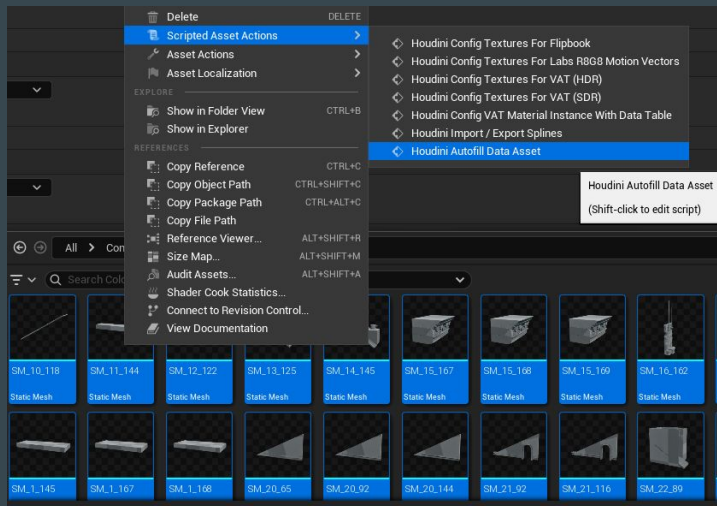
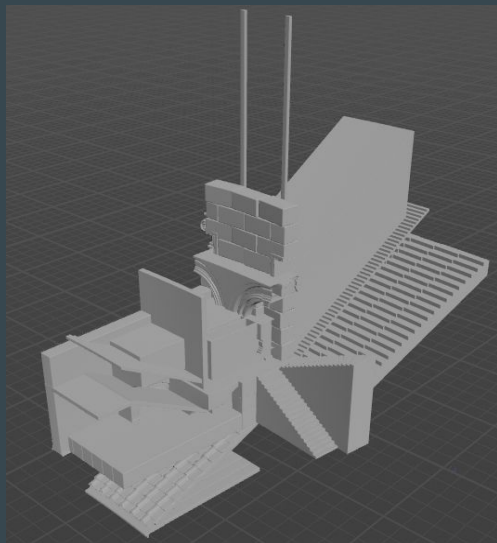
# Batch Export Unique Meshes



You can use a python sop to click the “Save to Disk” button for you automatically for each mesh.  
Or use TOPs which is faster!



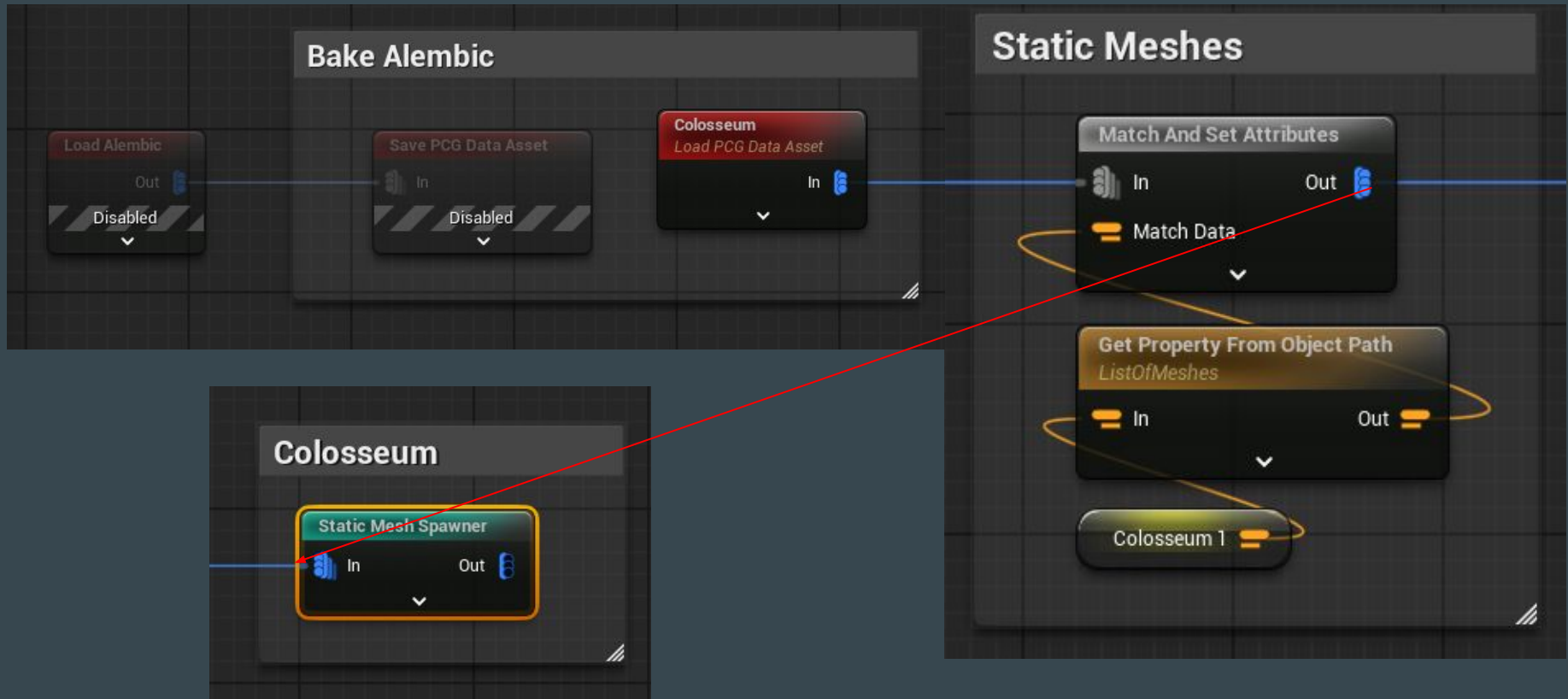
# Importing Meshes into a Data Asset



**PROBLEM:** 98 Meshes need to be manually filled in a Data Asset.

**SOLUTION:** Use Scriptable Tools to Autofill Meshes.

# Import Points using Alembic with piece attribute





# Force Regenerate Graph to re-import Alembic









