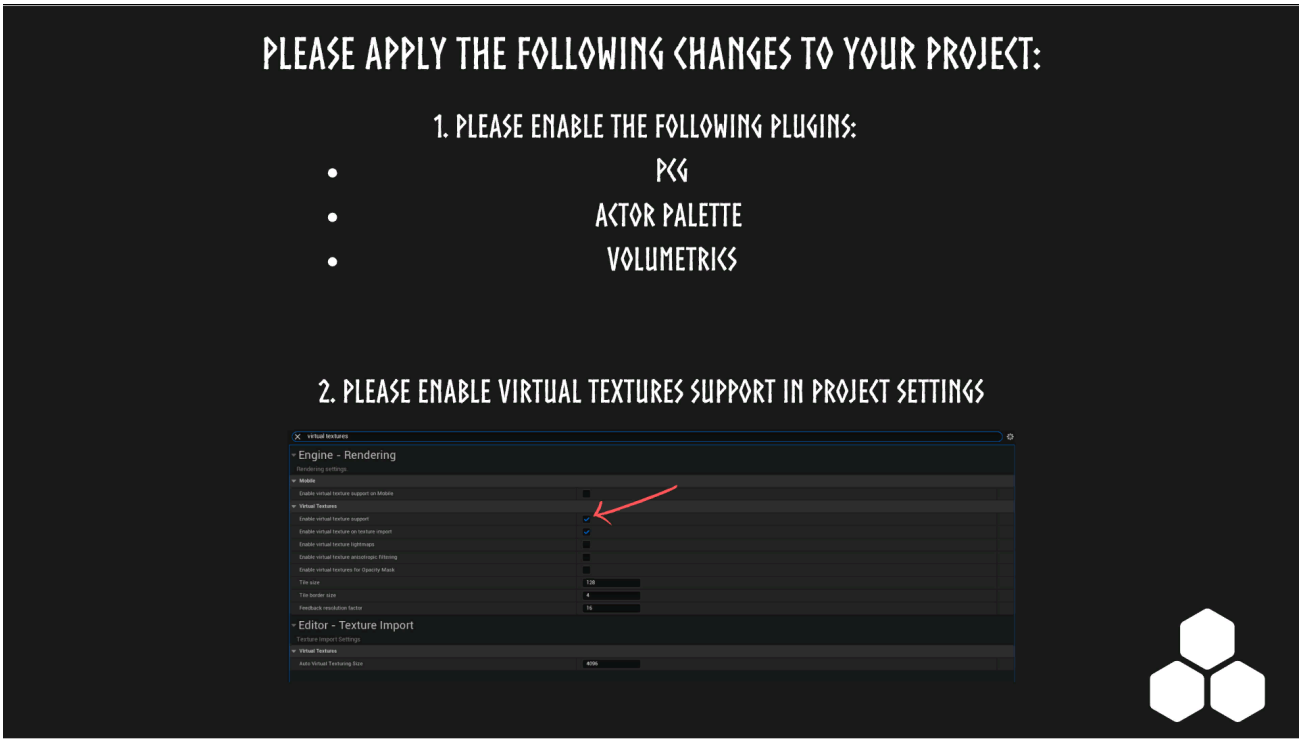
Toskov: Silent Zone Pack Documentation

Welcome to the Toskov: Silent Zone documentation.

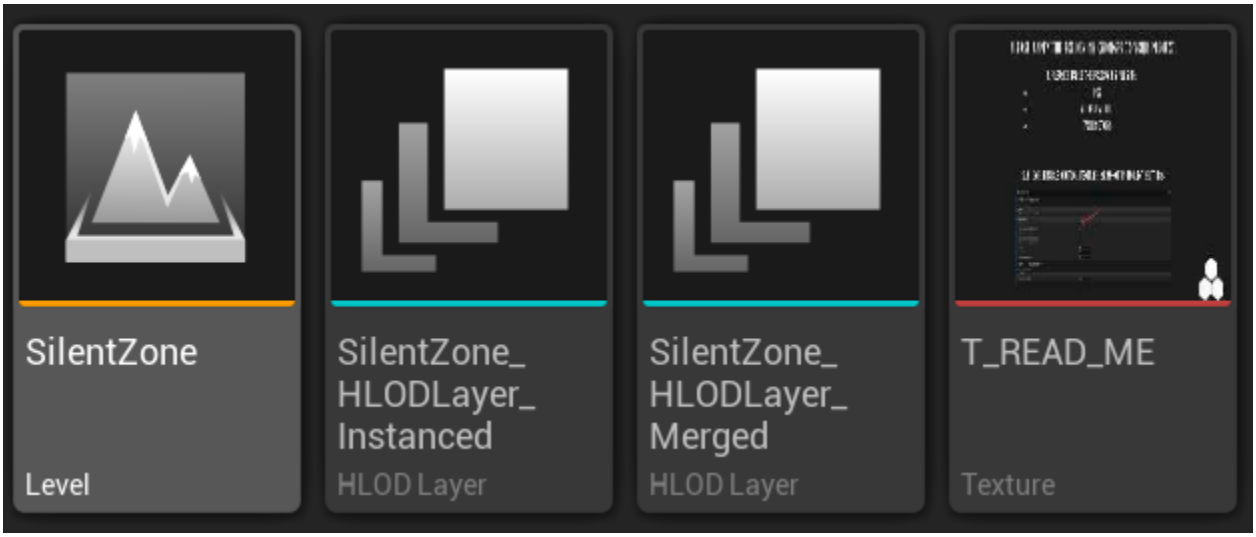
This guide will help you navigate the project, understand the structure, and get the most out of the assets and tools included in the pack. Whether you're exploring the environment or building your own scenes, everything here is designed to make your workflow faster and more efficient

First time installation (Please follow each step carefully to properly open the project)

1) **Follow the instructions from T_READ_ME file**



2) **Open the "SilentZone" Level**



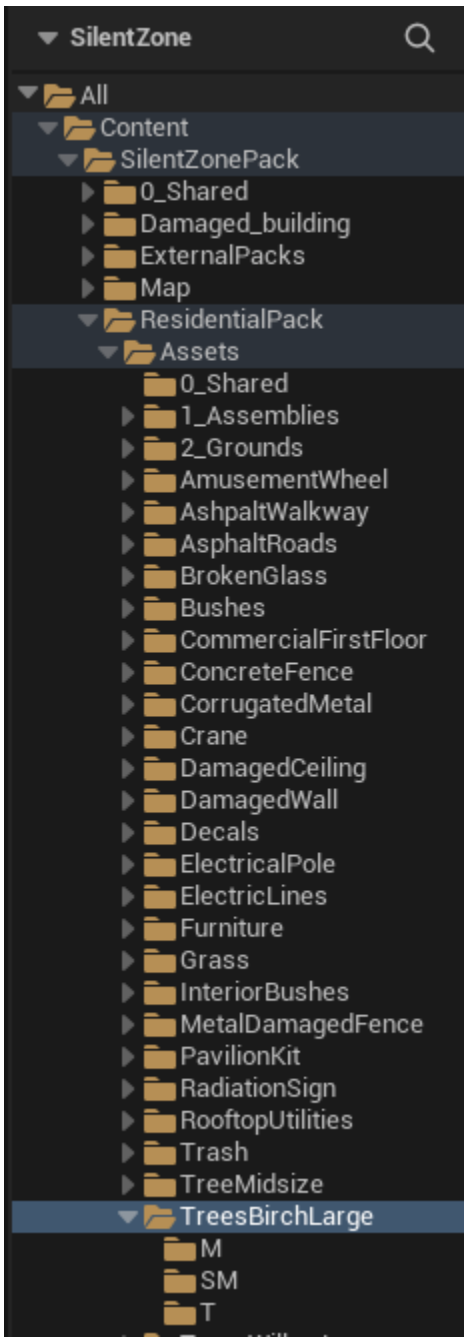
Let’s get started.

1. Navigate Content Browser
   1.1. Asset Folder Structure
2. Navigate Silent Zone level
   2.1. Load map
   2.2. Navigate Outliner
   2.3. Amusement wheel folder
   2.4. How to rotate Amusement wheel
   2.5. Cranes folder
   2.6. Electric pole folder
   2.7. Grounds folder
   2.8. Pipes folder
   2.9. Residential_Buildings folder
3. How to make your own building
   3.1. Building modular kit
   3.2. Why and how to pack in PLA
   3.3. Adding foliage to building
4. Shaders.
   4.1. Master Materials
   4.2. TwoTextureSets_Master
   4.3. Glass_TwoTextureSets_Master

1.Navigate Content Browser

1.1.Asset folder structure

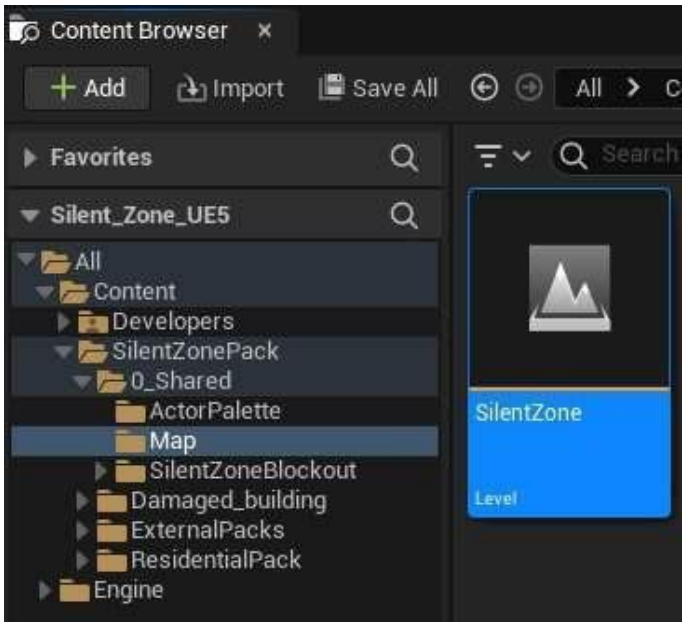
All assets have respective folders inside Damaged _building or ResidentialPack folders.
On example of **TreesBirchLarge** you can see the default subfolder structure:
M - Materials
SM - Static Meshes
T - Textures



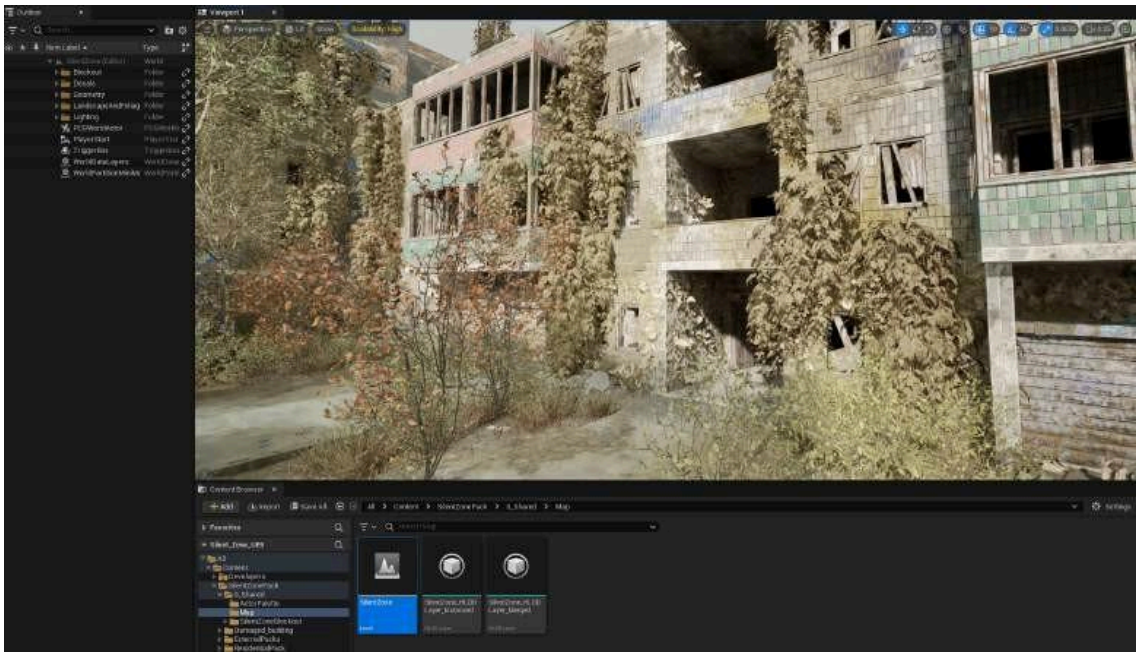
2.Navigate The Toskov: Silent Zone level

2.1.Load map

You can find the most important map **Silent Zone** at /Content/SilentZonePack/Map/L_SilentZone_Day.umap



Congratulations! You are almost done! Joking, no. But it is a good starting point. I recommend navigating assets from here as it is the most convenient way.



2.2. Navigate outliner

Everything in the **Toskov Silent Zone** map is structured in few categories.



Blockout.

A collection of background assets, meant only for background use.

Decals.

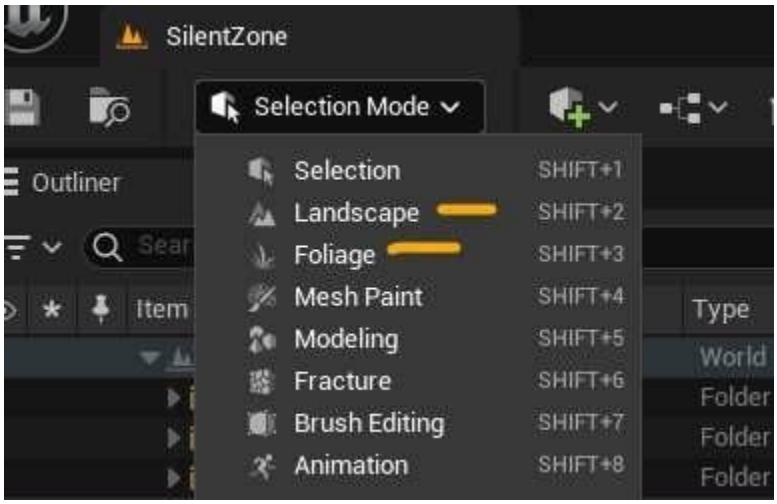
They live outside of packed level actors in this folder due to technical limitations of UE5. Still it helps to quickly select all of the same category and adjust parameters.

Geometry

That's where most of the juicy stuff lives. I will expand on this one in the next chapters. Large topic.

Landscape and Foliage.

Here lives all automatically created blueprints by UE for foliage and landscape. You should not edit this in the outliner. Please use respective editing menus. Actors in outline here will update automatically.



Lighting.

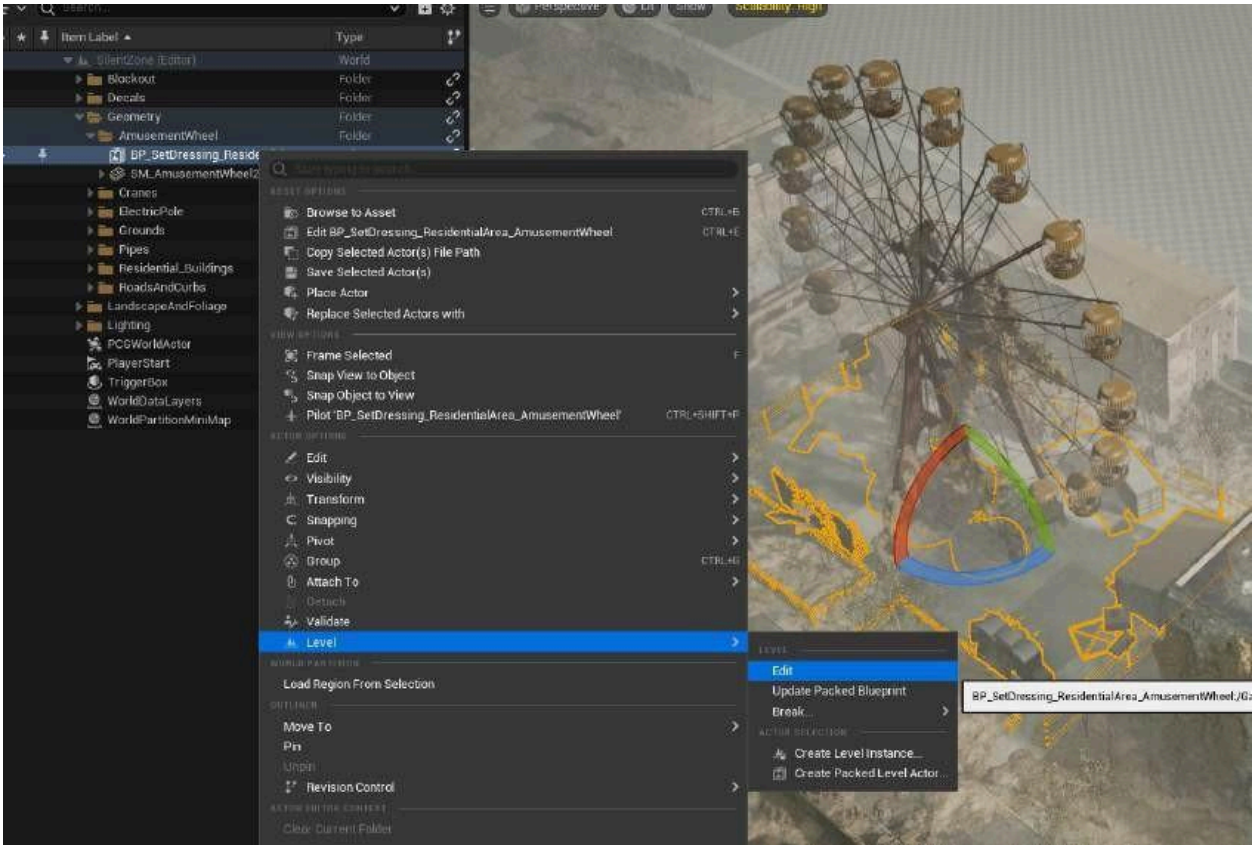
Here you can find all essential lighting related stuff. Directional light, Atmosphere light, Post Process Volume, Exponential height fog.

2.3. Editing Packed Level Actors - AmusementWheel

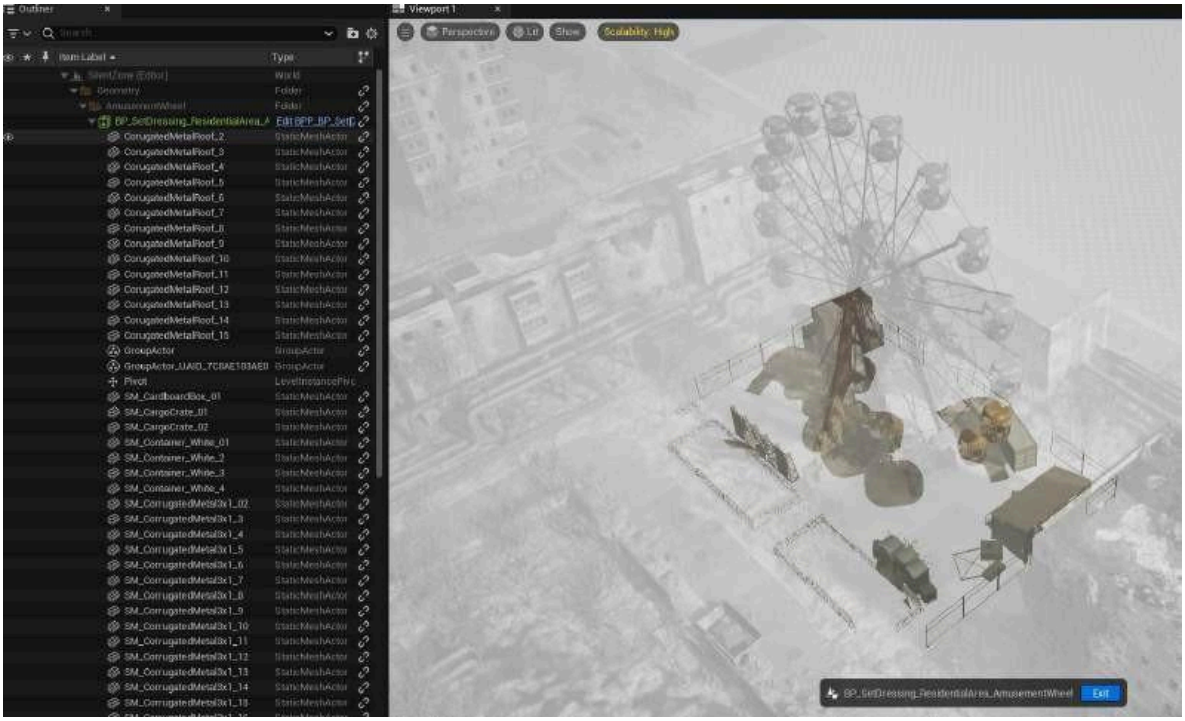
The landmark and legendary wheel lives in this folder.



Set dressing around the wheel is compressed in a single packed level actor.
Don`t be scared. It is easy to edit if you want to. Just press **Edit** in viewport or outliner. Or use hotkey **Ctrl+E**.

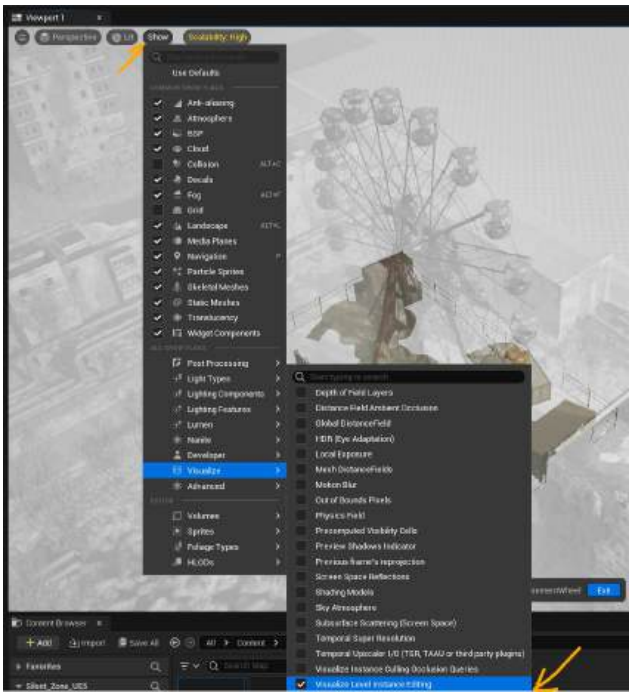


Once you press it you are in edit mode and you can adjust the position of each static mesh individually like you usually do in viewport. No annoying limitations like in blueprint editor.



Please note that by default it will highlight in viewport only the portion which is inside this packed level actor. So on one hand it is handy to see what is inside, but can be confusing in artistic terms if you want to see it as is. Feel free to disable/enable this mode.

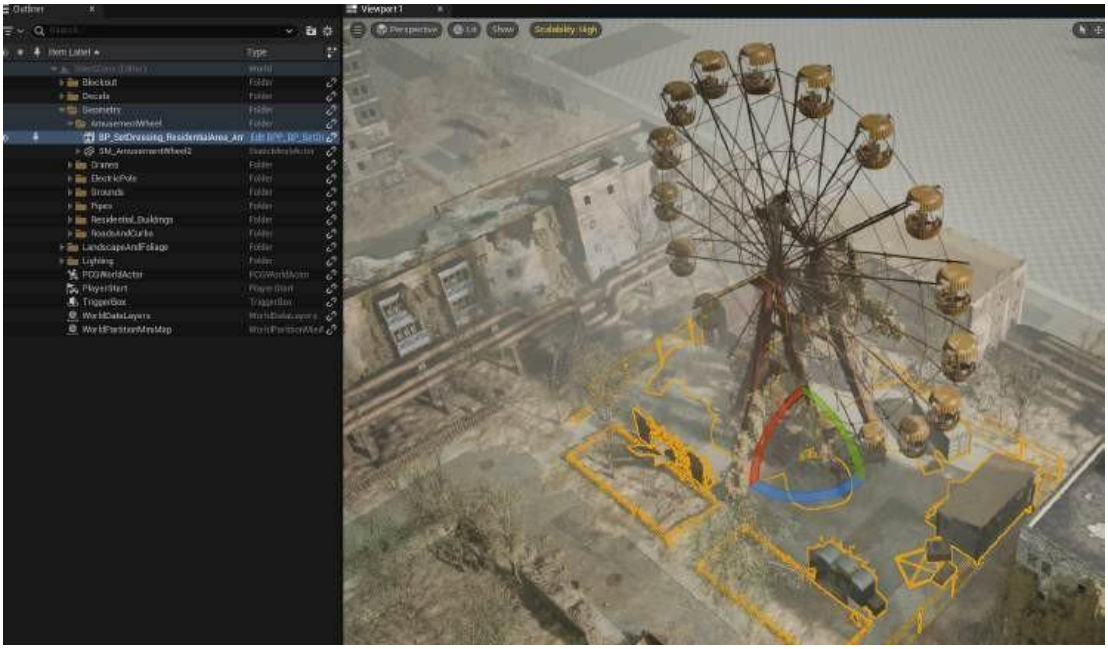




This approach also has an extra advantage. Selecting all actors with the same mesh (**Shift+E**) or **same material** will select only actors inside this PLA, which may be handy to speed up editing.

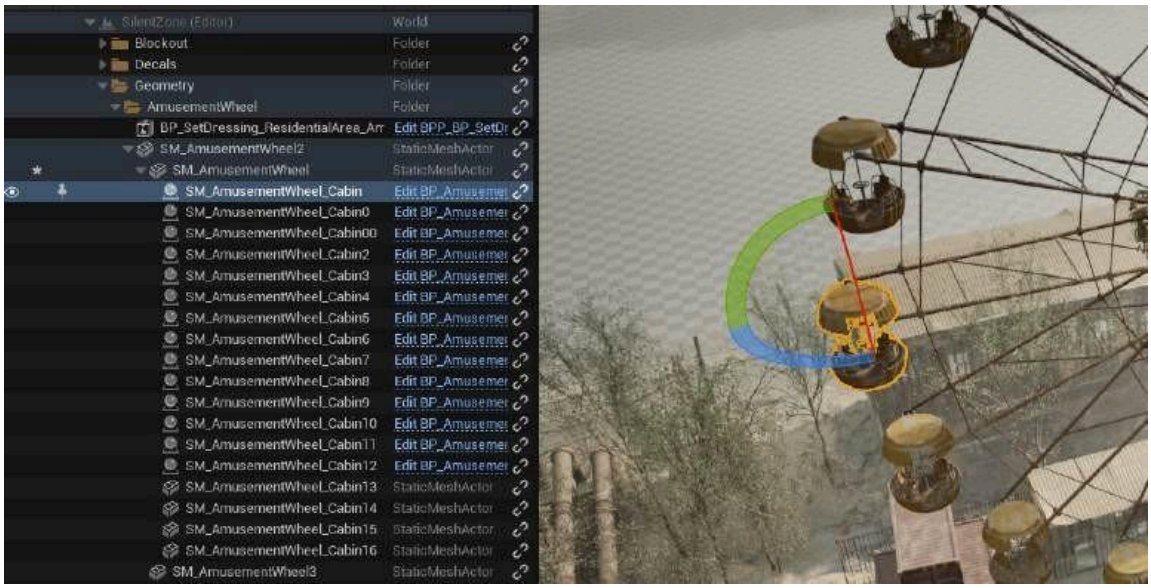
Once you are happy with edits you have made inside a packed level actor and you want to return to regular scene editing press **EXIT** or **Esc** keyboard button.



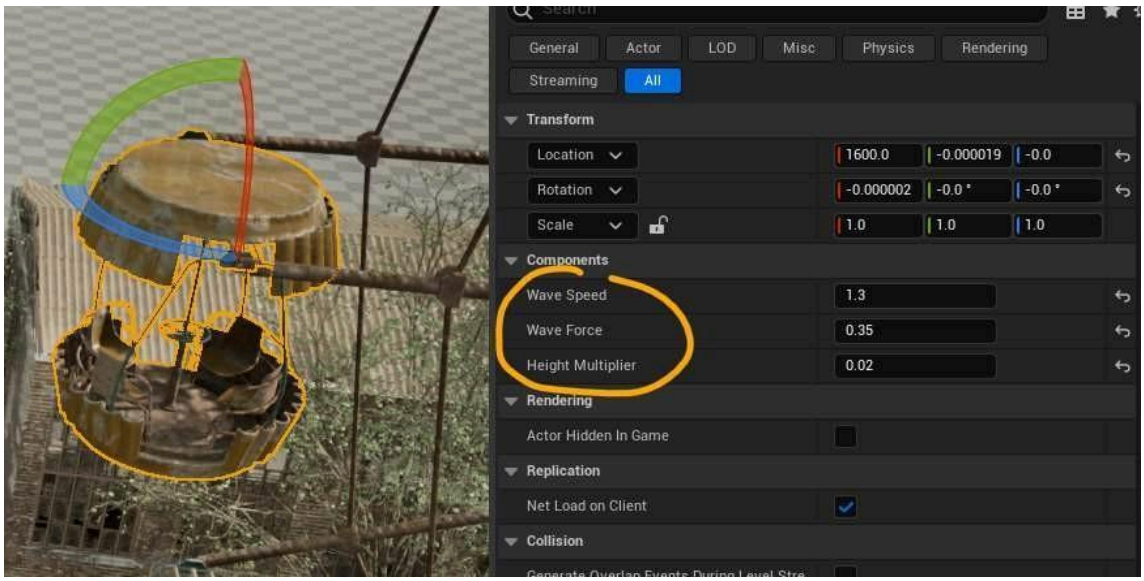


That will return you to regular mode. Using PLA increases performance dramatically, so it is impossible to avoid them in large scale scenes. It has the advantage of instantly detecting actors with the same static mesh and converting it into an instanced static mesh without need of any manual adjustments. In scenarios where there are lots of repeating static meshes it can be most effective.

Regarding the elephant in a room. Why then is the wheel itself not inside PLA?
The answer is short. Not so many repeating static meshes + need of individual wheel carts to swing on wind. Packing all of them inside of PLA would make it impossible for the waving blueprint to detect each individual pivot and change its rotation during runtime. So in this case it is an acceptable tradeoff.



Regarding carts swinging. When you select any of cart you will see that they have 3 parameters.



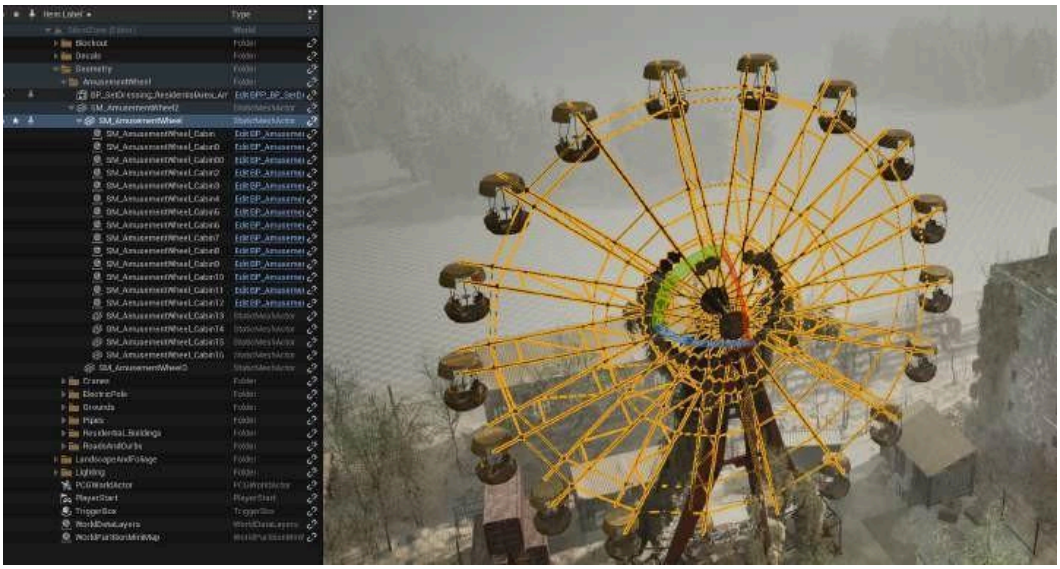
Wave Speed is how quickly it will swing from side to side.
Wave Force defines amplitude of swing. Aka how far it will swing.
Height Multiplier defines how much effect will increase depending on world Z position. You see that it would be too tedious to manually set values for each cart each time position changes. So logic is simple: the higher the cart - the more it swings on the wind.
Feel free to play with values. Best part of this approach - you can set the same value for all carts.

2.4. How to rotate Amusement Wheel

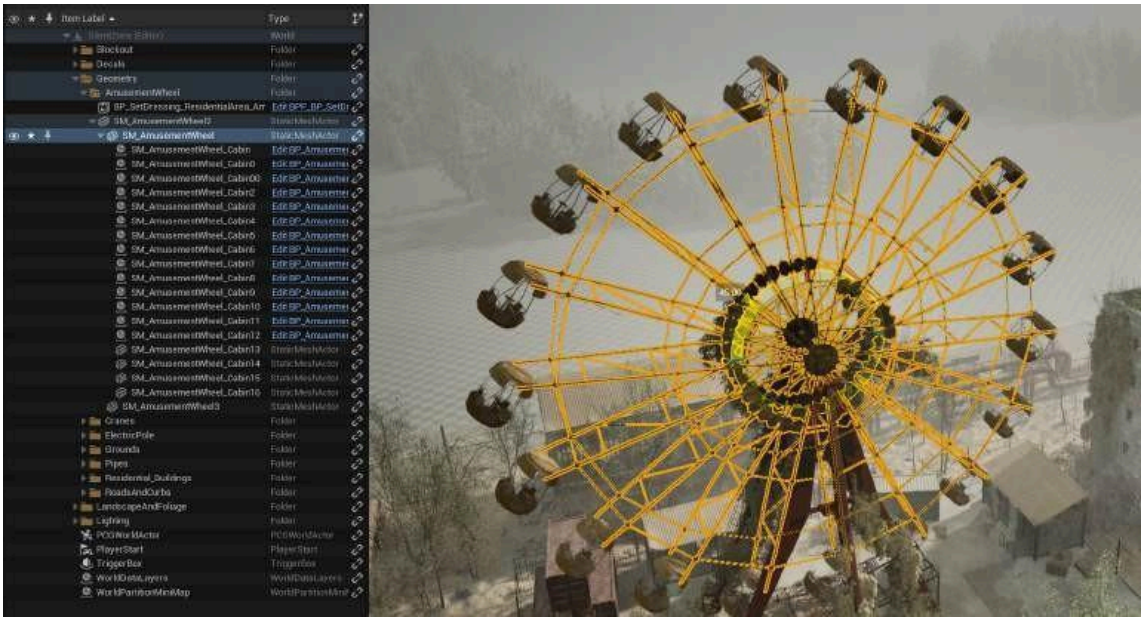
I mean this is a ***post-apocalyptic*** wheel…It will not rotate. Topic closed.



Just kidding. If you still want to rotate there are few ways.
First select the wheel.



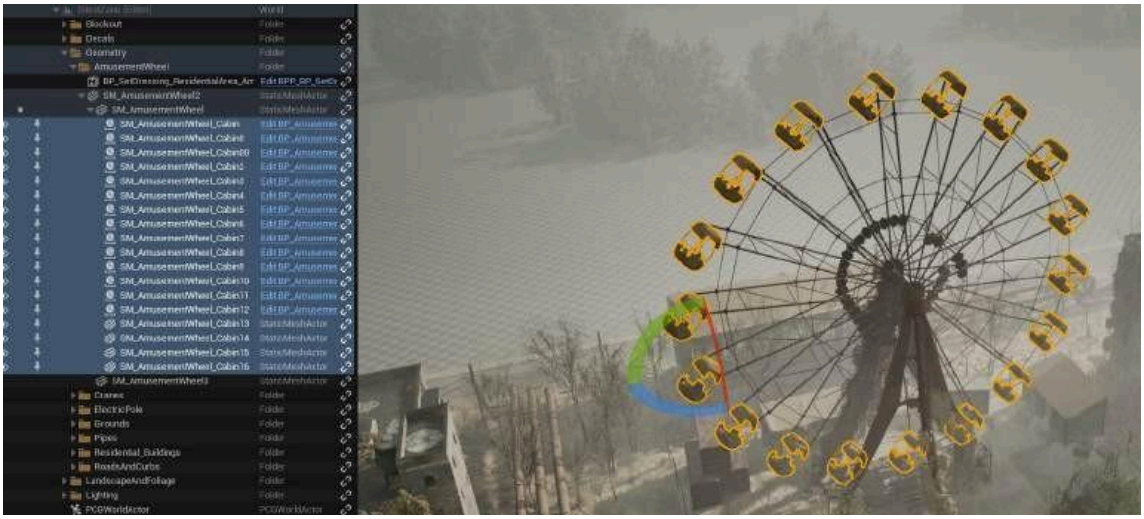
Rotate for desired angle.



But wait, what is it? How to fix it? Do not worry. Here is a quick and effective way. And yes, they will swing on the wind in the correct position after these steps. I promise.



Select all cabins in outliner (quickest way).



And press **Detach.**

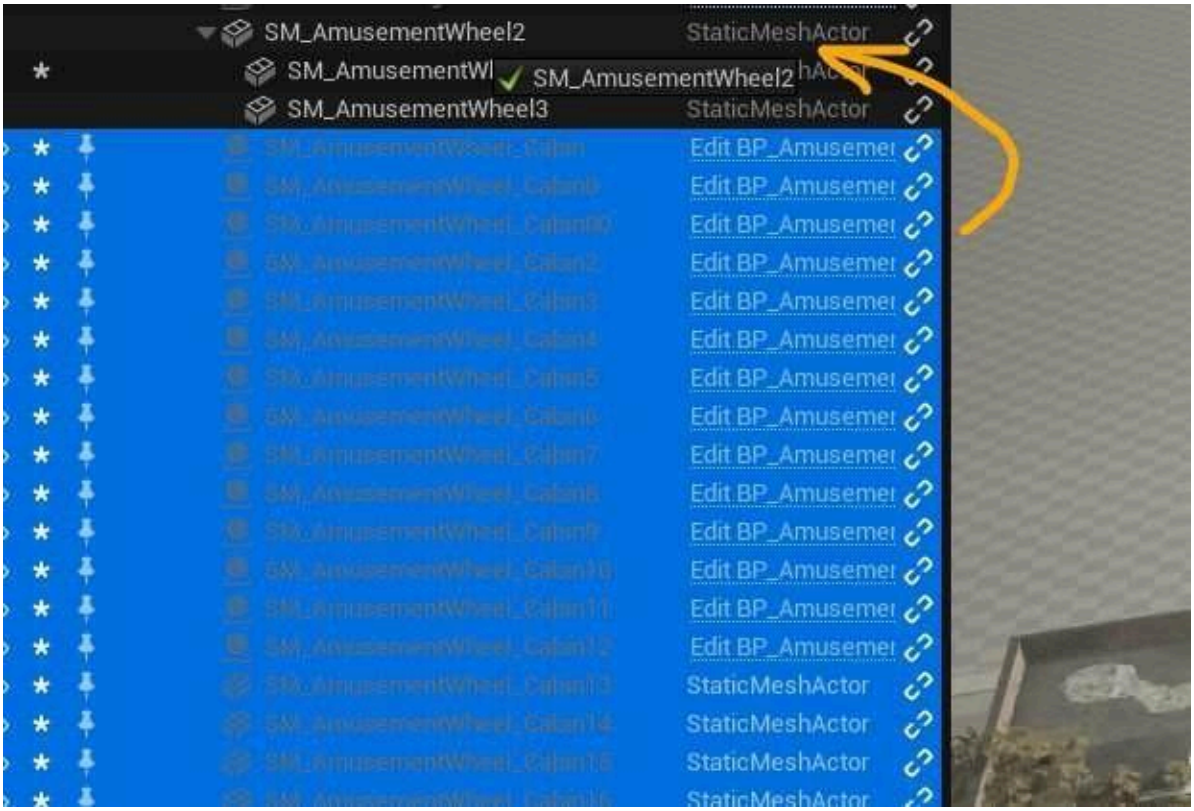


With all them still selected set rotation to {0,0,0}





Now they rotated properly.
And lets parent them back to wheel by drag-n-dropping them back.



This way carts will be back to normal position. And swing on the wind as they should. No need to adjust wind parameters.

2.5. Crane folder

Contains Packed Level Actor with all crane related static meshes inside



To edit please press **Edit** on selected Packed Level Actor (in short PLA) or **Ctrl+E**



Now you can select and edit each Static mesh individually.



To make viewport look “normal”, with no gray highlight please see 2.3.

2.6. Electric pole folder

Contains all Static Meshes of electric poles and cables.



EndPoints and **Supports** contains static meshes of electric poles and electric boxes. Also it has **SM_Electric_Cables** which is mesh with all cables connecting these supports and boxes together. Please note that this static mesh actor was created in DCC outside UE5 specifically for this environment, so it can not be edited directly in UE.

Though if you want to create ***custom cables arrangement inside UE*** you have such an option using a modular kit of cables. They already have the waving effect applied. Here you can find a set of different lengths to suit any scenario.

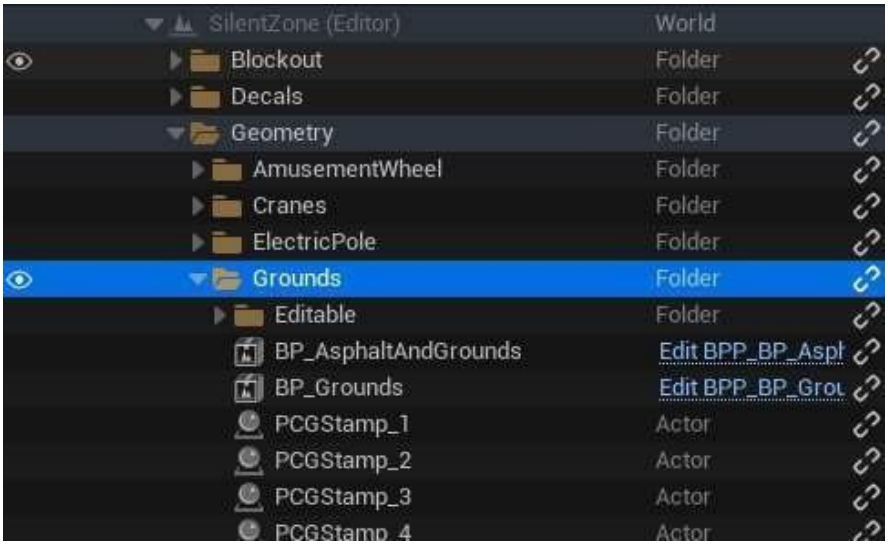


Also it has a handy pivot system, so you can scale them to fit any scenario!

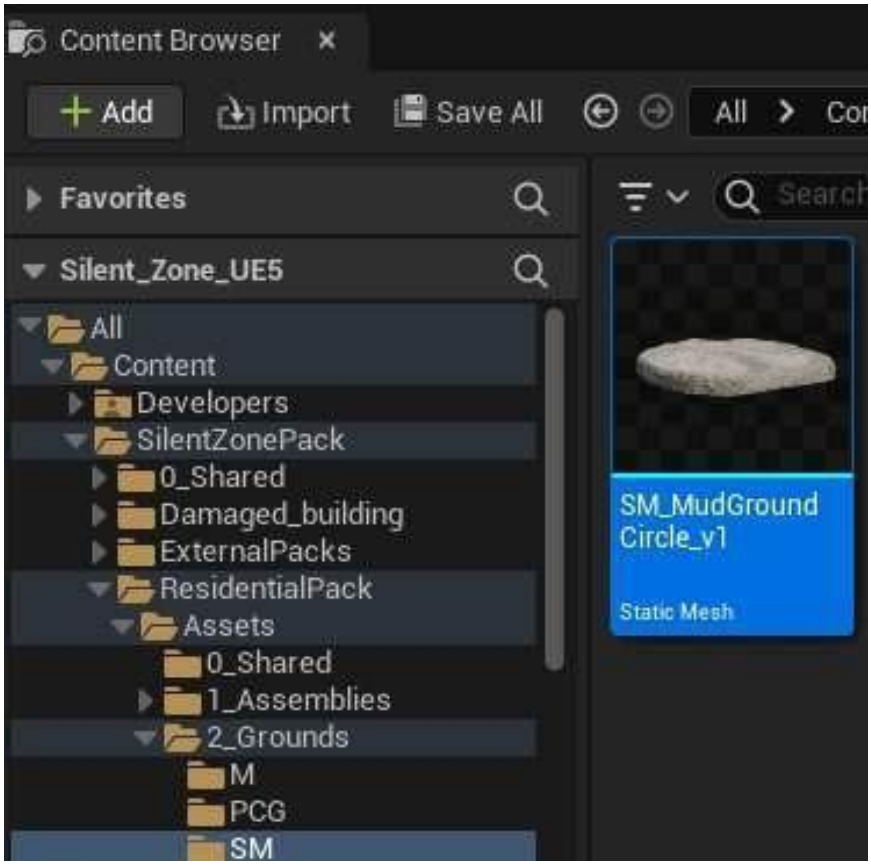


2.7. Grounds folder

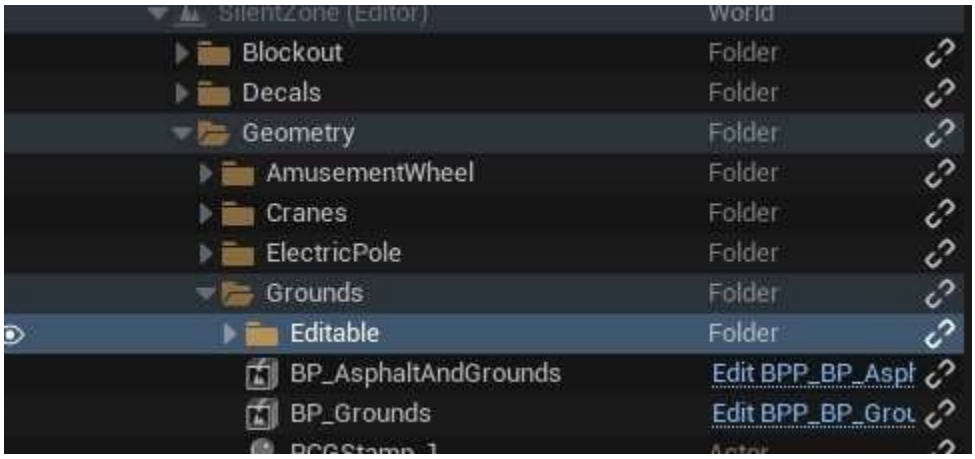
Contains all Static Meshes related to mud and asphalt roads.



99% of ground mud was created by instancing the same single static mesh. Here it is this mighty mud pancake. Look at this beauty.



So basically all hierarchy you see here is ways to speed up scattering of this pancake, as well as manual placement packed in PLA.



Editable folder contains a simple PCG actor which places this pancake on terrain and adjusts its rotation randomly. Not all clients will have enabled PCG plugin by default. Also baking in ISM assures if anyone opens it in another version, or in a version where PCG nodes were changed by Epic it still looks the same way. For this purpose it was baked in PCG stamps. Whenever you want to adjust position you can make it in an editable folder, delete all PCG stamps and rebake them back.

Or if this approach sounds too complicated for you, feel free to place these mud pancakes manually and pack them in Packed Level Actor once you are happy with the result.

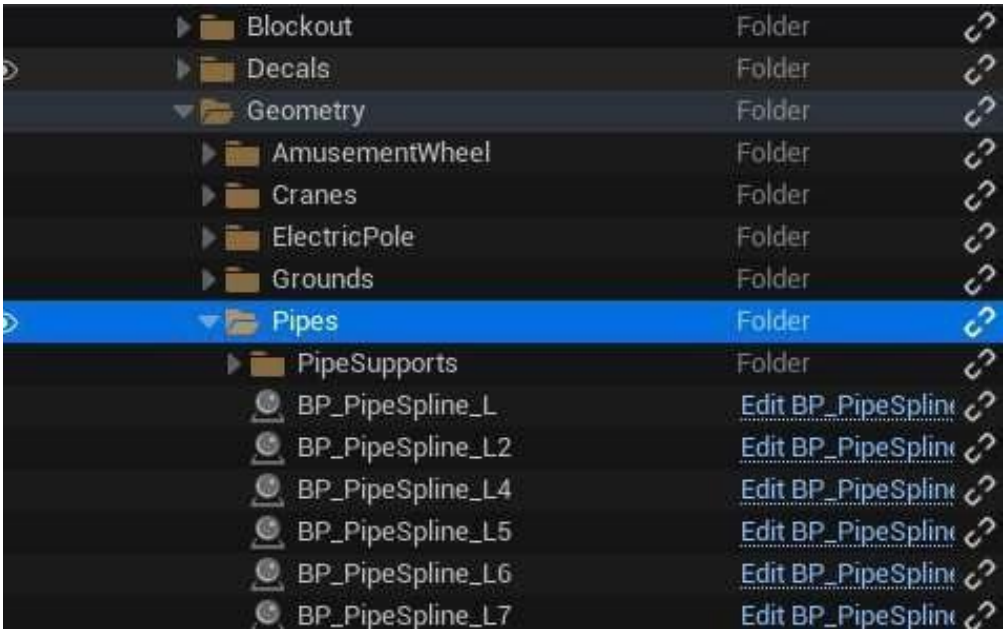


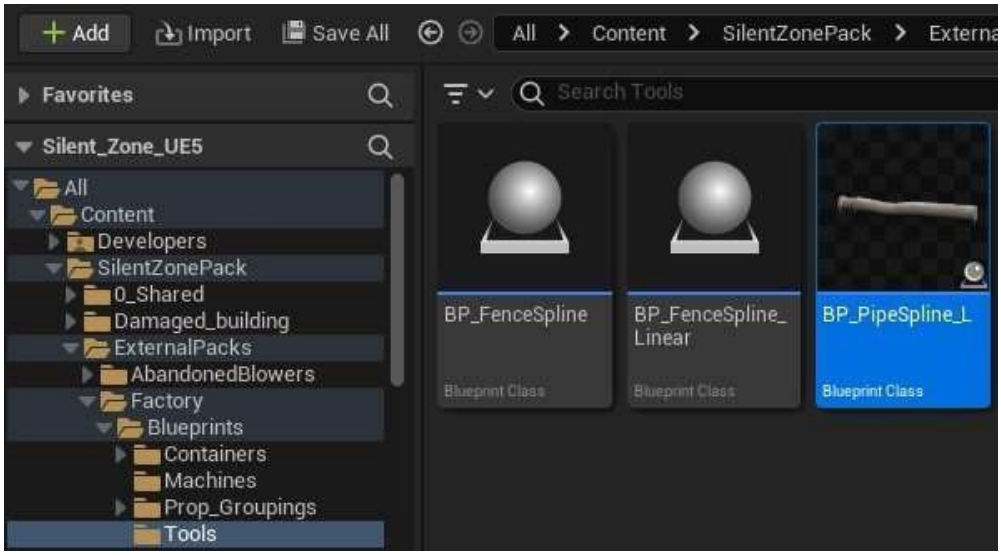
Exactly how I did with this case.

Btw if you want to edit this Actor look at 2.3. Where all this process is described in detail.

2.8. Pipes folder

Contains all Blueprints and PLA related to pipes.



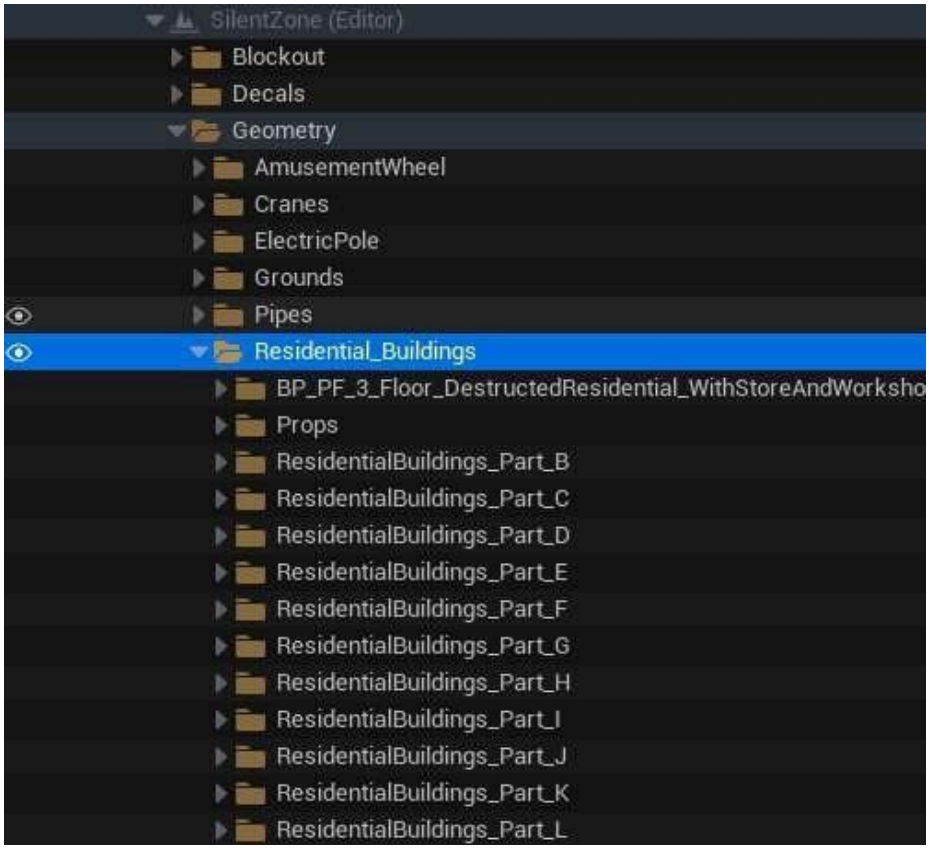


Also it has PLA with concrete supports. Editing of PLA is described in article **2.3**



2.9.Residential_Buildings folder

Contains everything related to residential buildings and industrial pavilions as well as small architecture.



Each section of map is split in a separate folder to make navigation easier.



Lets take for example chunk B for example



Here you can find each Packed Level Actor for buildings, ground set dressing.
Editing of these PLA in viewport is described at article **2.3**.



A bit special in this hierarchy is chunk A.

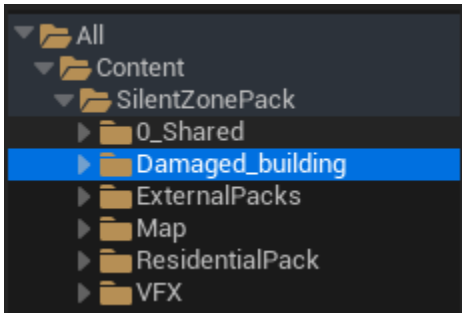


It has separate PLA for building, exterior and interior set dressing.

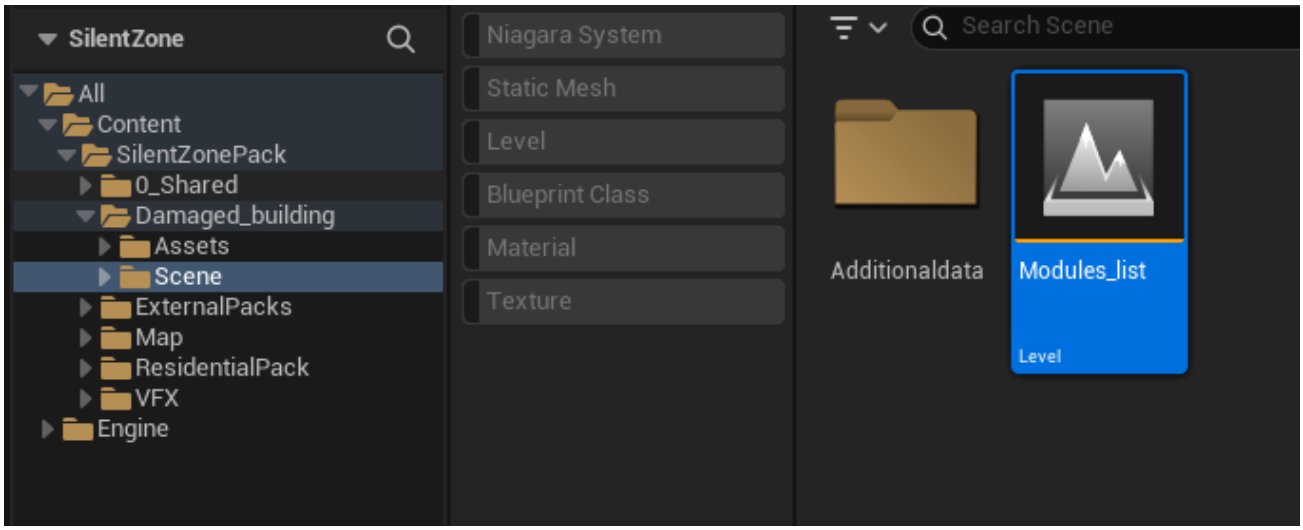
3.How to make your own building

3.1.Building modular kit

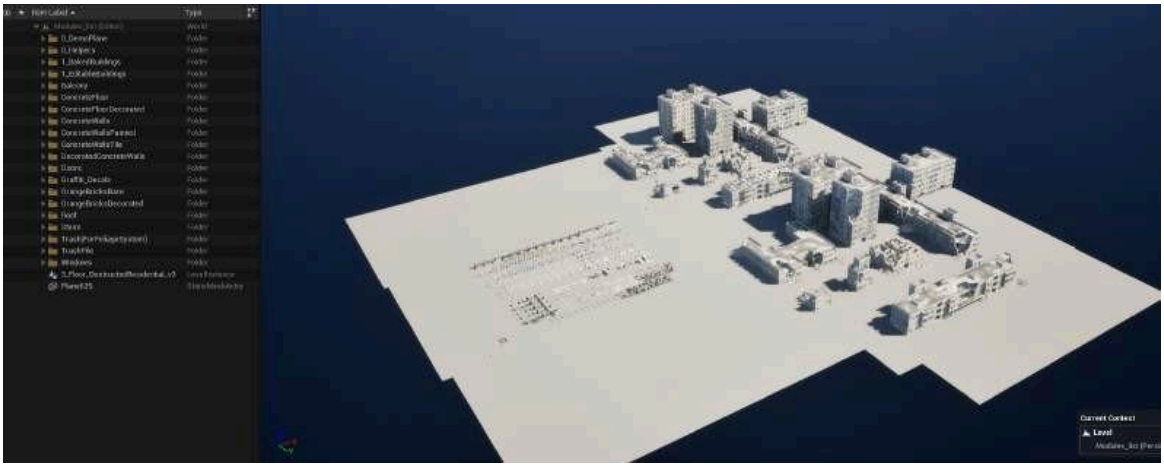
If you find it too overwhelming working with PLA and you *just want to assemble buildings by yourself* do not worry, I got you.



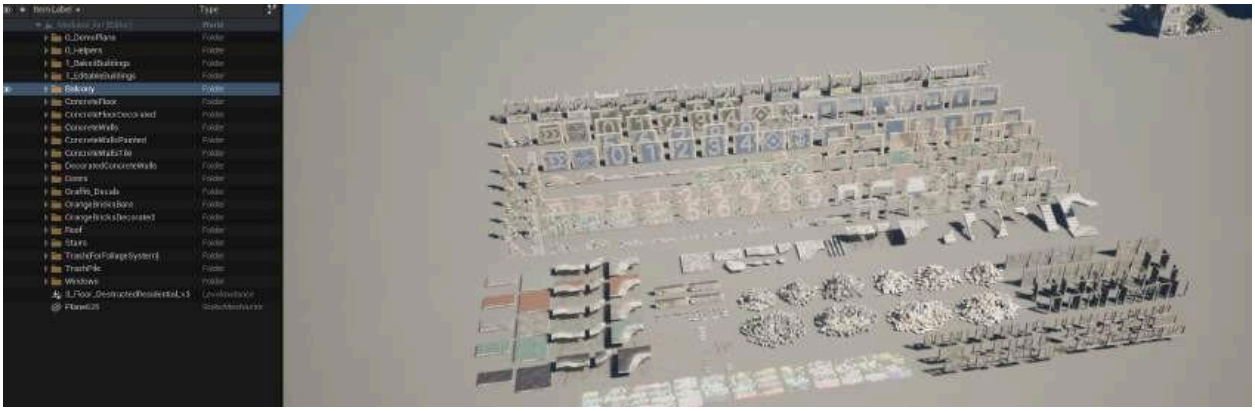
That is exactly why we have a Damaged_building folder in the Content browser for.
Open **Modules_list** level. To see how it works.



Here you can find both modules to assemble any buildings from as well as baked packed level actors from them.



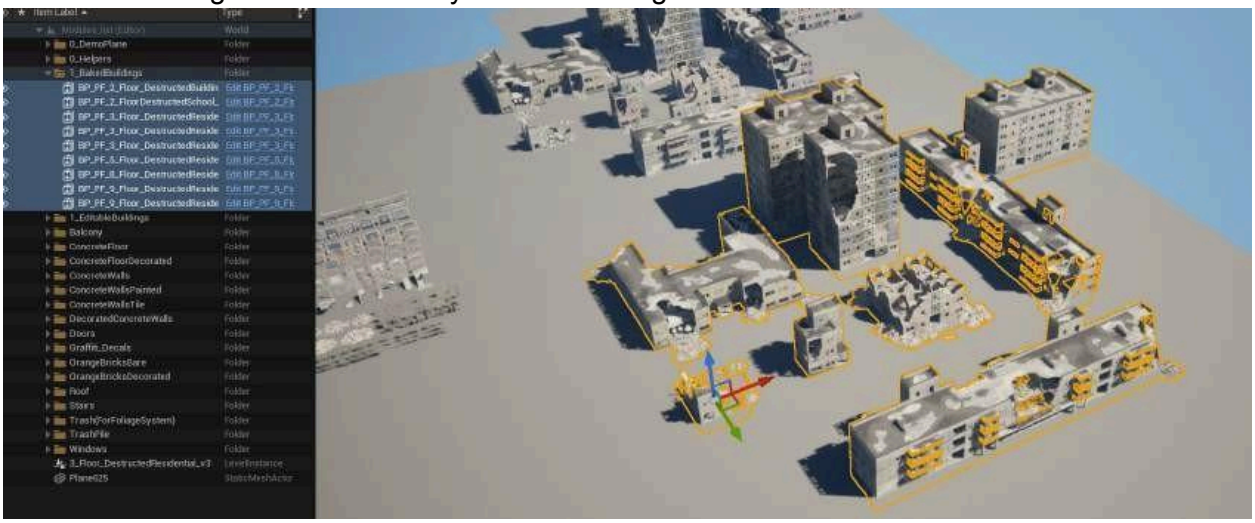
Feel free to use a static mesh palette to assemble your own building.



Copy any of existing assembled levels. ATTENTION. These are not performant. Easy to edit, but will affect performance in the same way as putting hundreds of static meshes in a scene.



Or copying baked Packed Level Actors. RECOMMENDED These are so performant, you can put hundred of these buildings in one scene and UE will crush it like cookies. This is the pipeline used in all AAA games and the way UE5 was designed for.

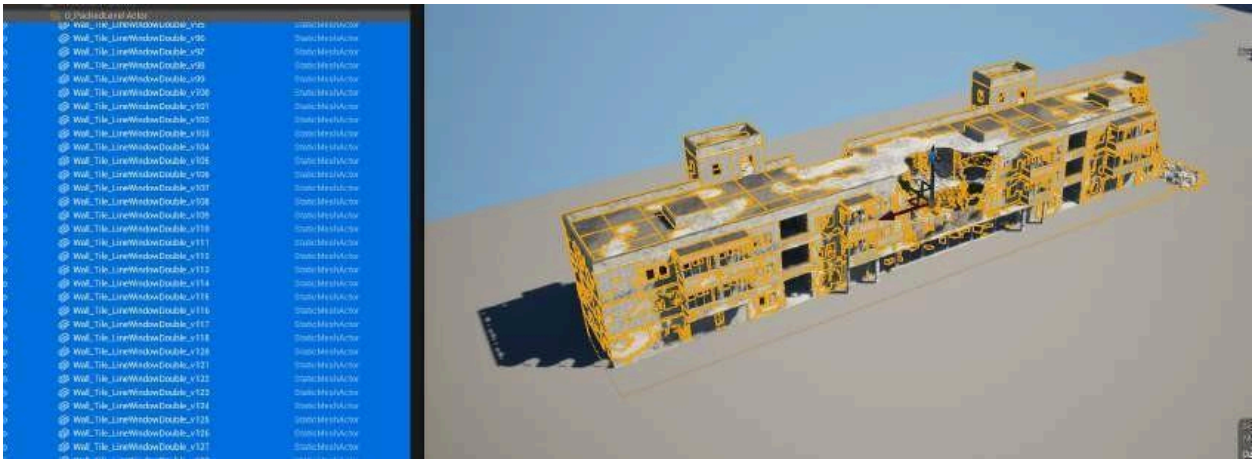


3.2.Why and how to pack in PLA

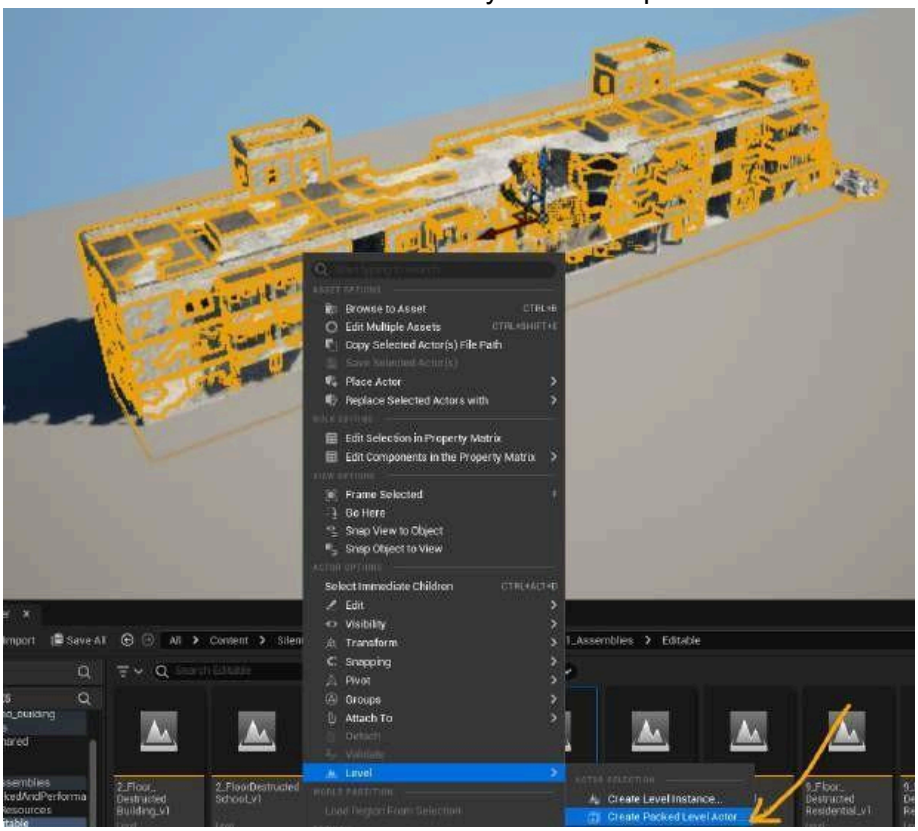
So yeah you may be tired of hearing again and again “Packed level actor this Packed level actor that”.

Enough.

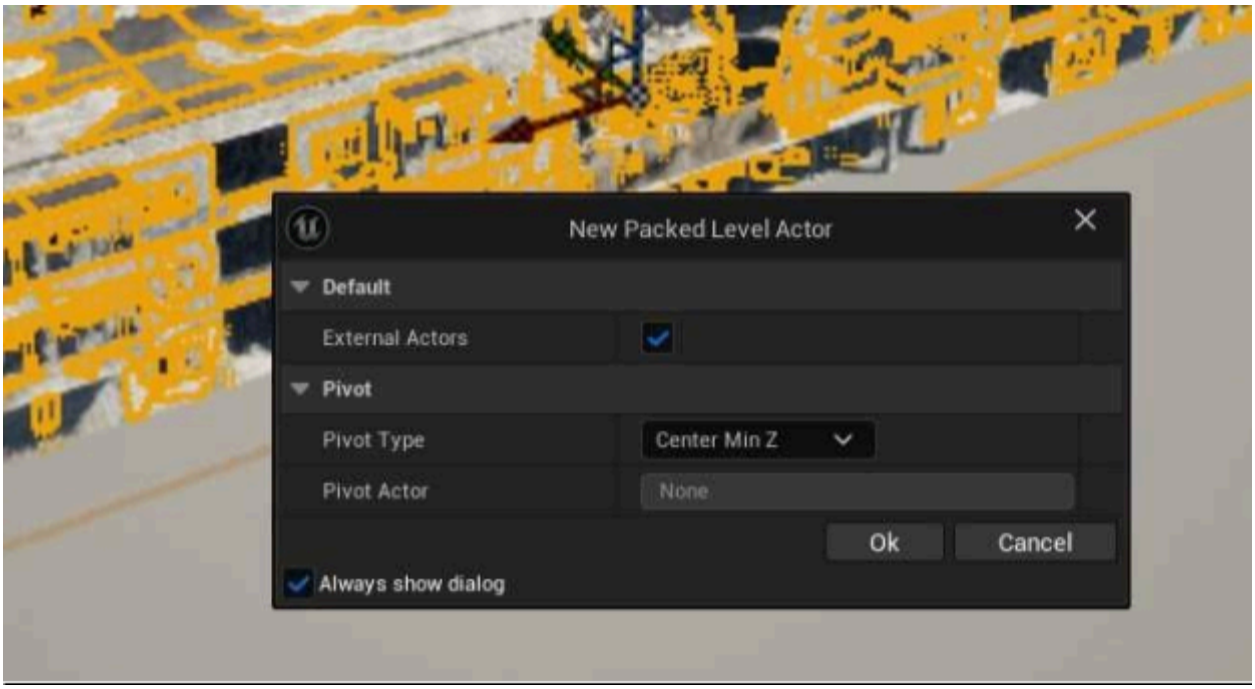
What is it and how to make them?



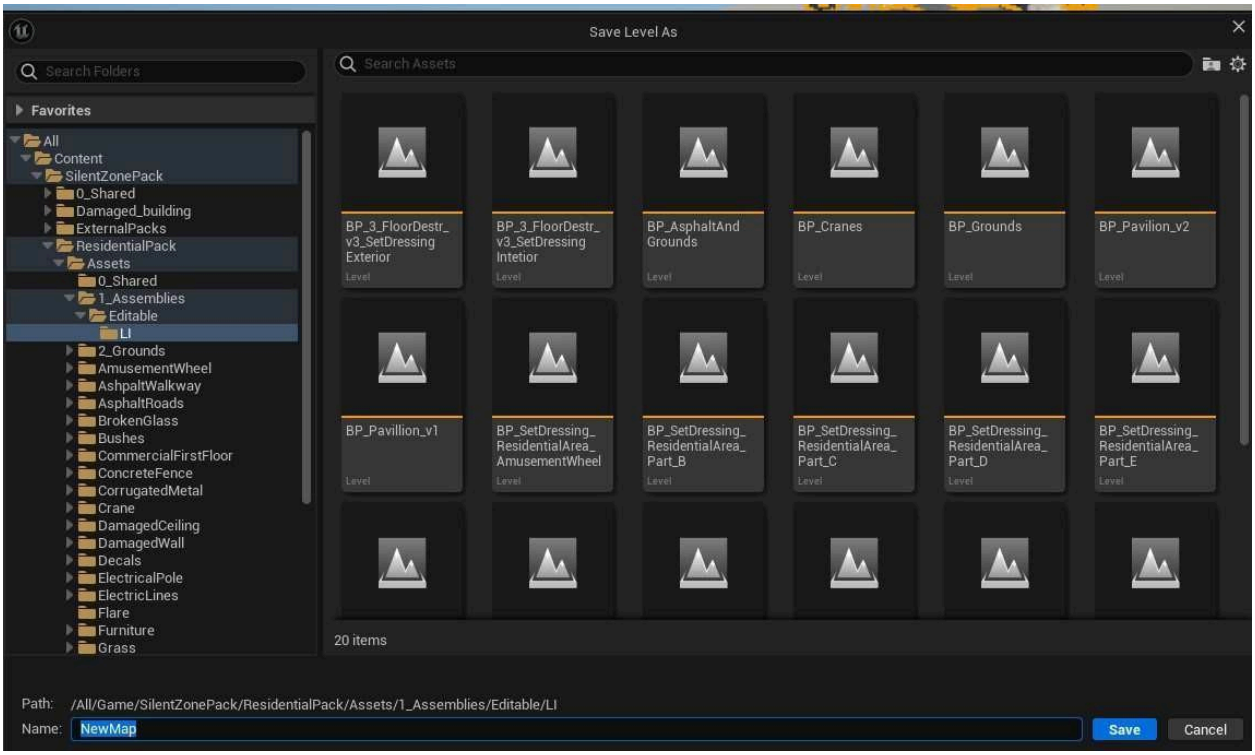
Select all hundreds of Static Meshes you want to pack.



Press **right click**-**Level**-**Create PackedLevel Actor**

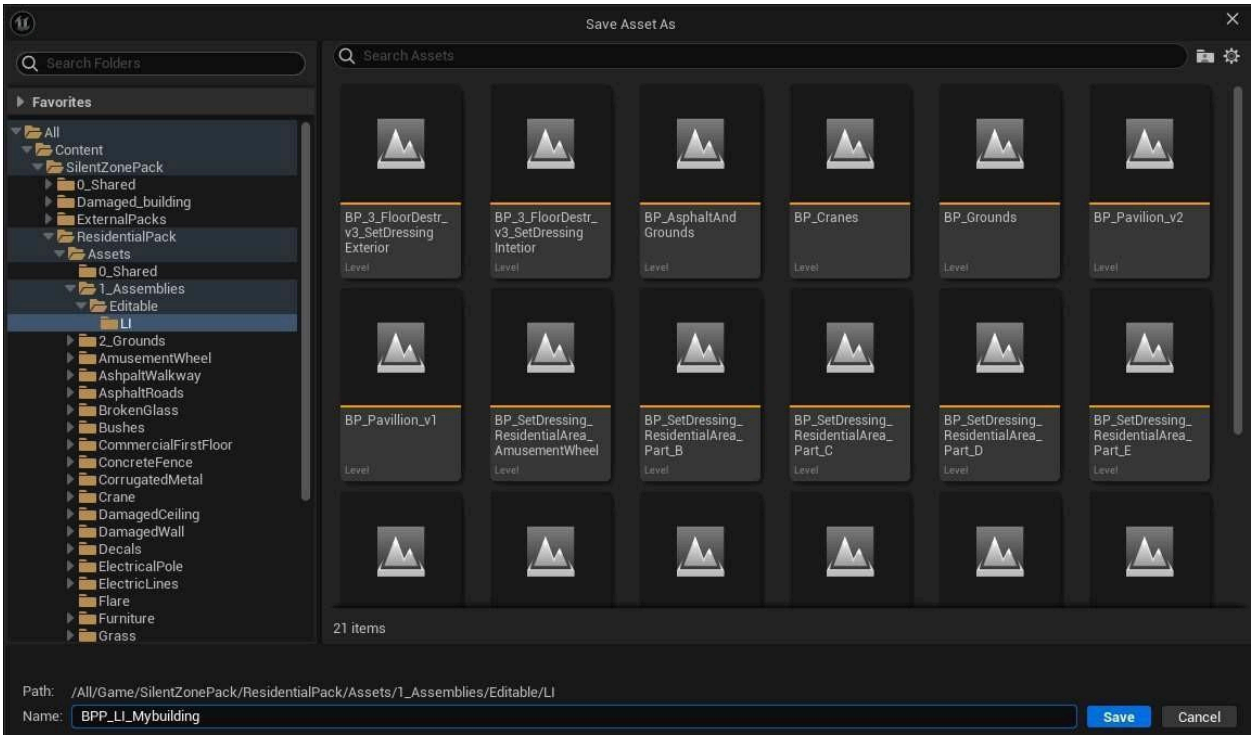


It asks where the pivot of a new actor should be. I recommend using Center Min Z. It's the most predictable one from my experience.



It will ask where to store Level Instance. Save it wherever you want and name it for example **LI_MyBuilding**

Then UE will ask where to store **Packed Level Actor**.



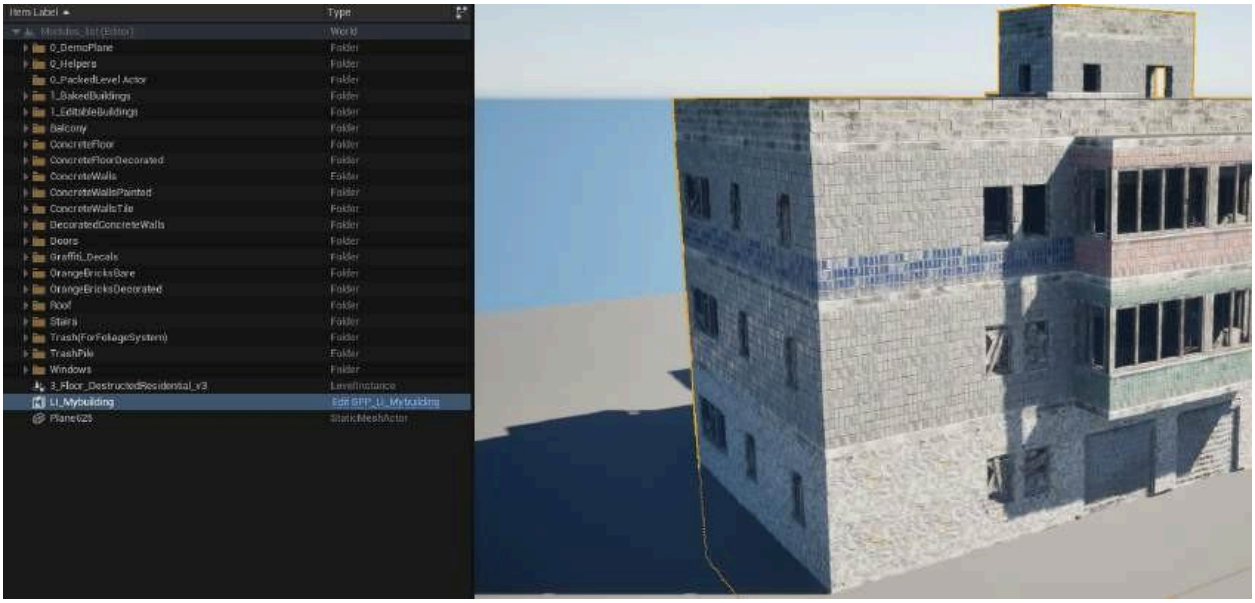
You can save it right next to the Level instance and name it this time **BPP_LI_MyBuilding.**



Congratulations!

You have made it. Now instead of hundreds of Static Meshes you will have a single PLA actor which will have inside a few Instanced Static Mesh actors for each of Static Mesh type. This will reduce significantly overhead on the CPU side.

But wait a second.



All my glorious vertex paint is gone now! What to do?!
Do not worry. That is expected. Packed level actors support only Static Meshes and will use vertex color from asset itself. The one from the Content browser. That is the price we pay for such a dramatic performance boost.
However there is a workaround.

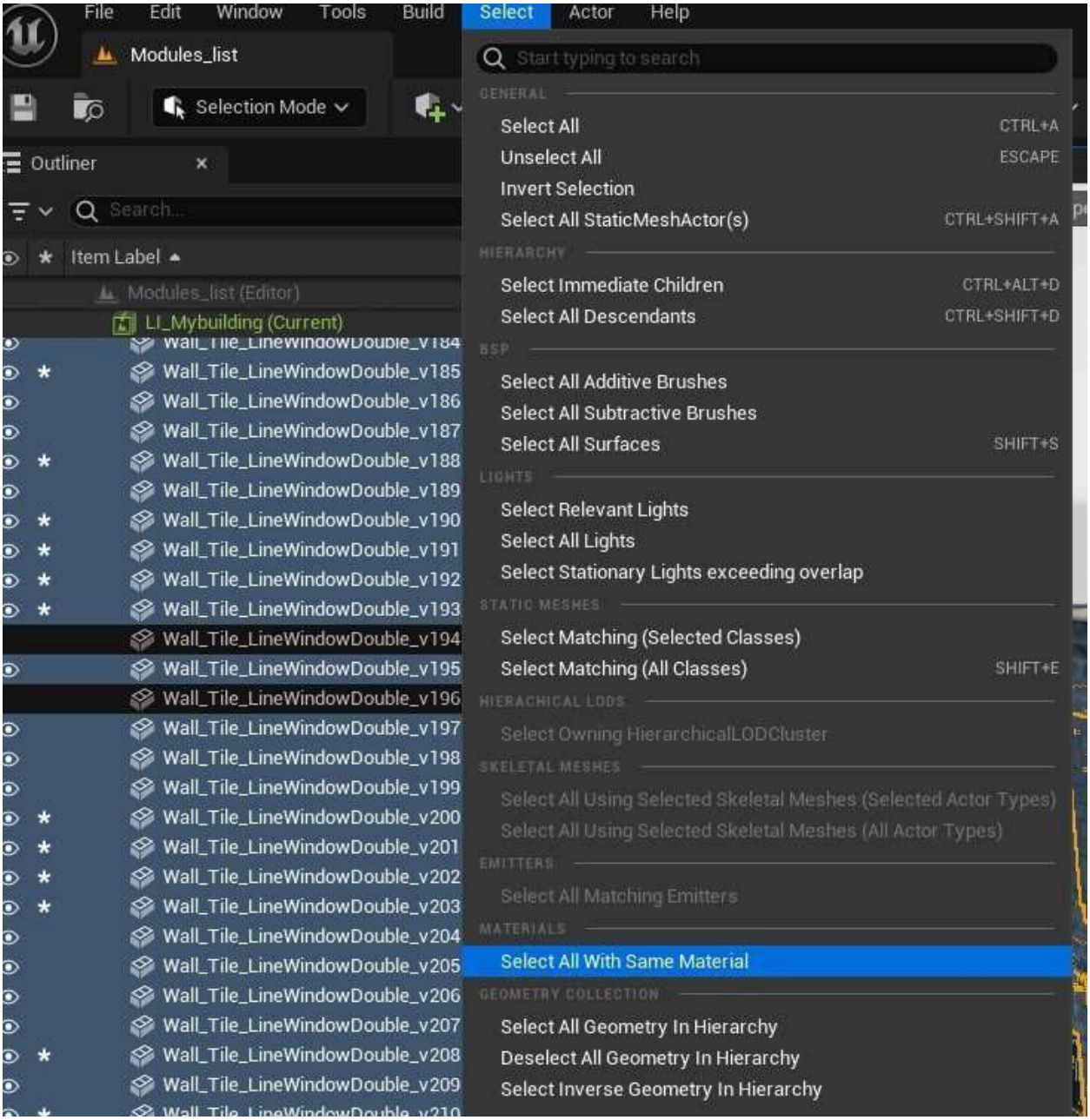
Press **Ctrl+E** on PLA you have just created.



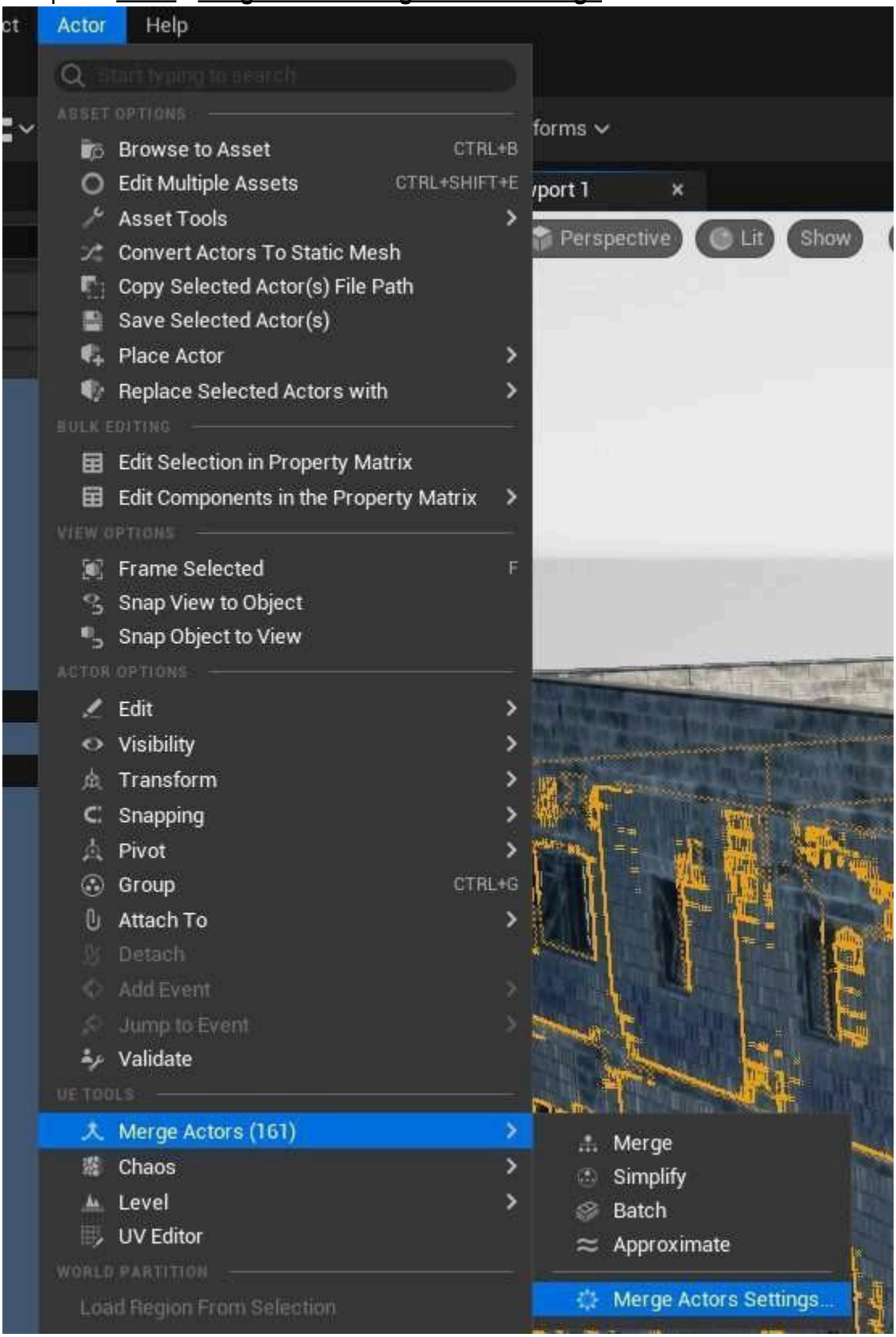
Yeah! Vertex paint is back!
Do not rush; we need to bake all wall paint/tiles/decor in a single static mesh and save it somewhere in the Content Browser in order to make it work when we return to our scene. Currently it is an illusion, because we temporarily left level with the actor palette and loaded level we created before (**LI_MyBuilding**). That is basically what happens when you press edit

on Packed level actor under the hood to give the toy back all nice controls with selection, gizmos, etc.

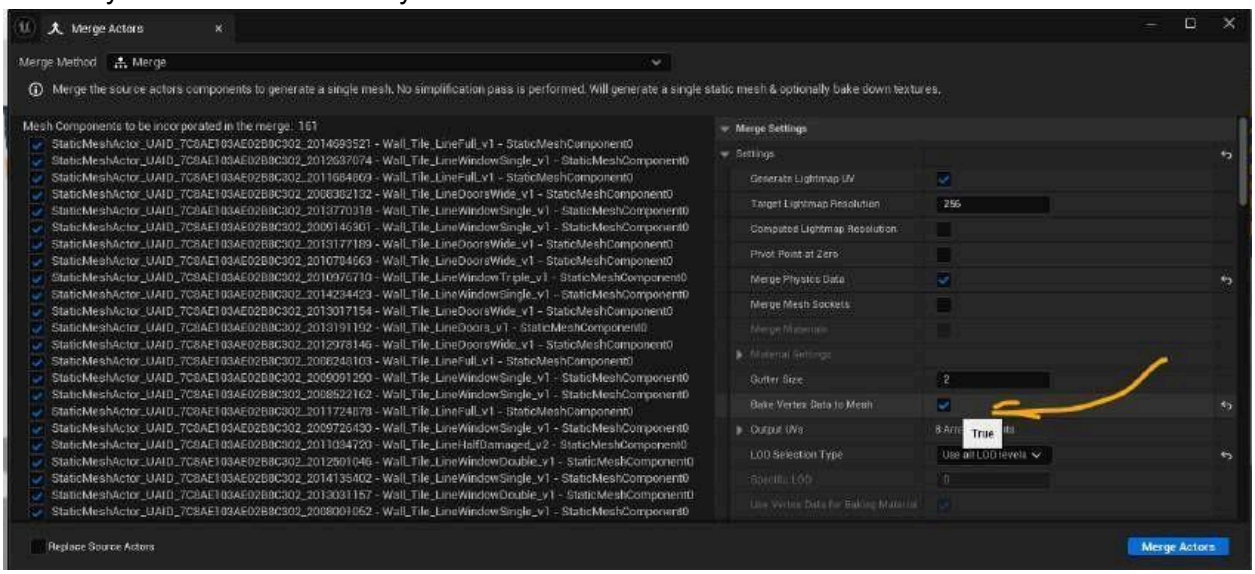
So, select all meshes with paint,tiles, wall decor you have made your vertex paint on. The quickest way is to use **Select All With Same Material**. Please note that it will select only stuff inside this level, so do not worry, it will not affect other buildings.



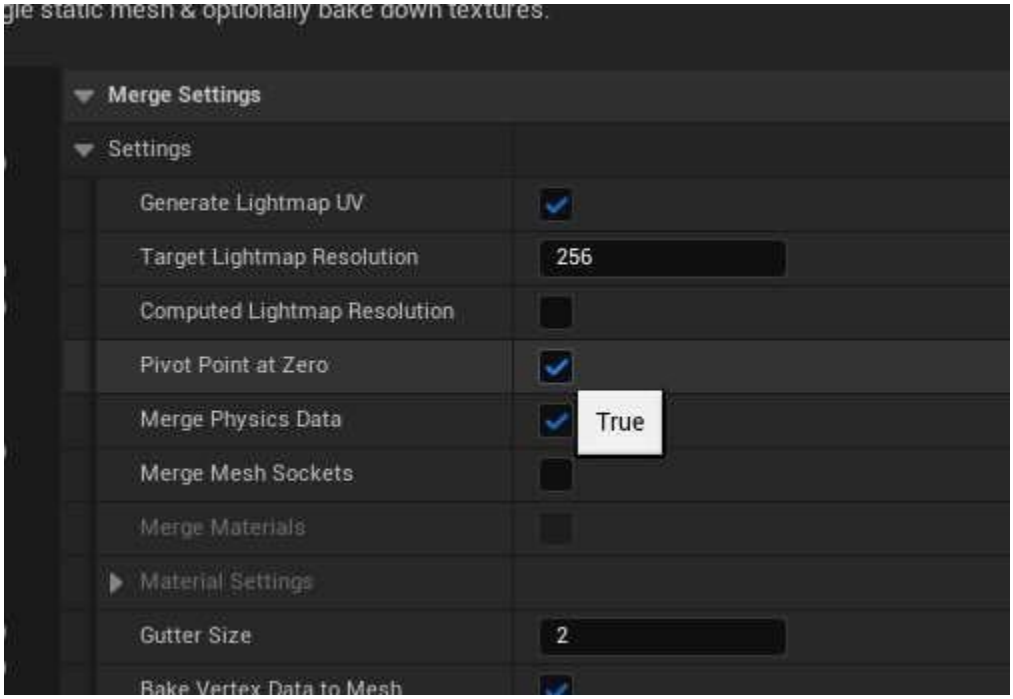
And press **Actor - Merge Actors-MergeActors Settings**.



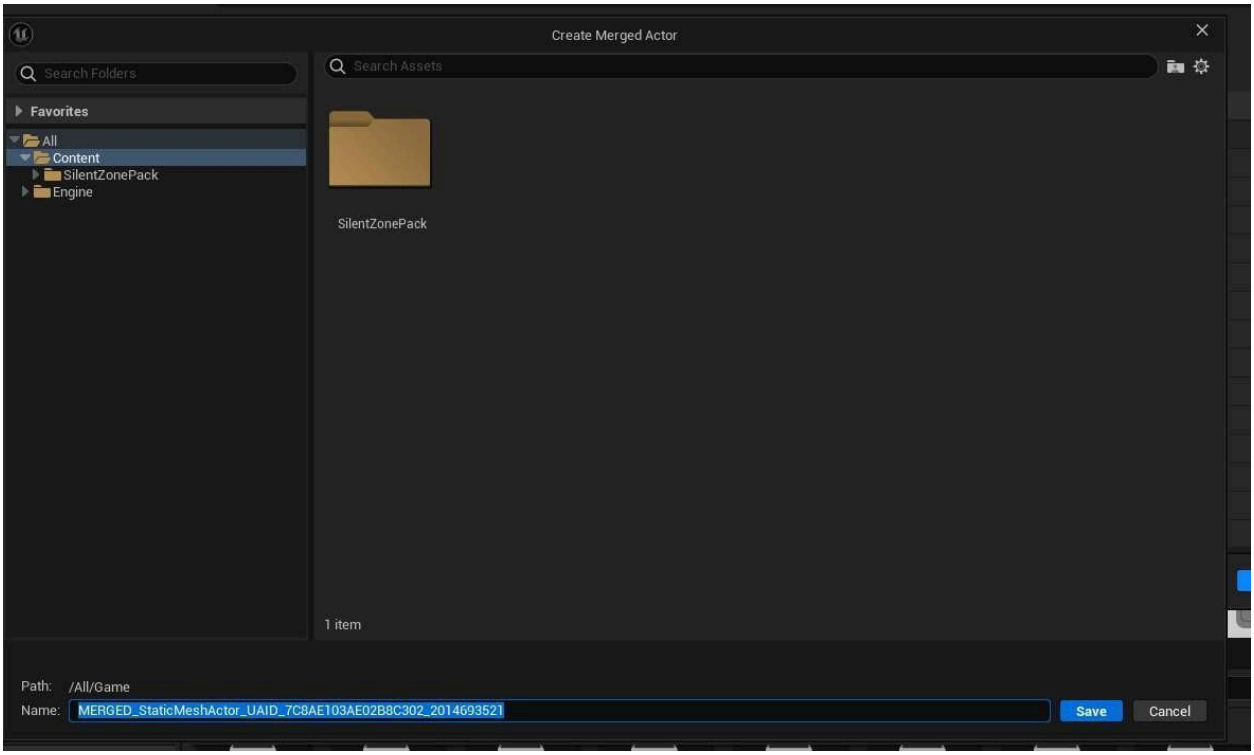
Make sure **Bake vertex Data** to Mesh toggle is on. This way all vertex paint you have made manually will be stored in newly created mesh.



And also check **Pivot Point at Zero.** It will help you to position everything exactly where it was asap.



Now it will ask where to save newly created Static Mesh.



And press save.

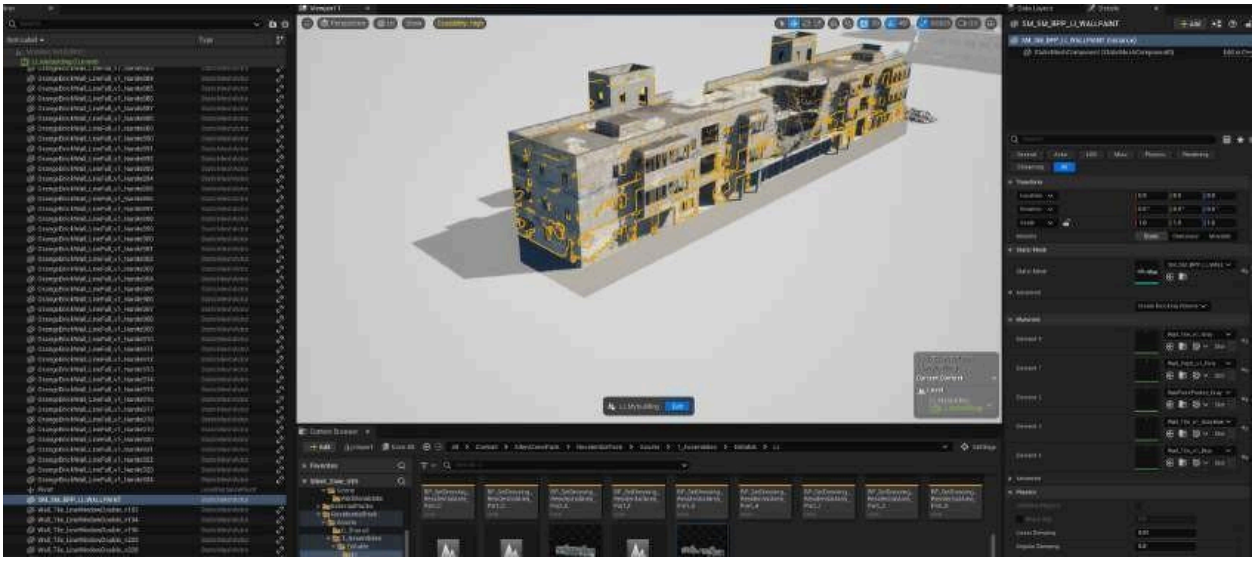
Now you have your mesh with vertex paint on!



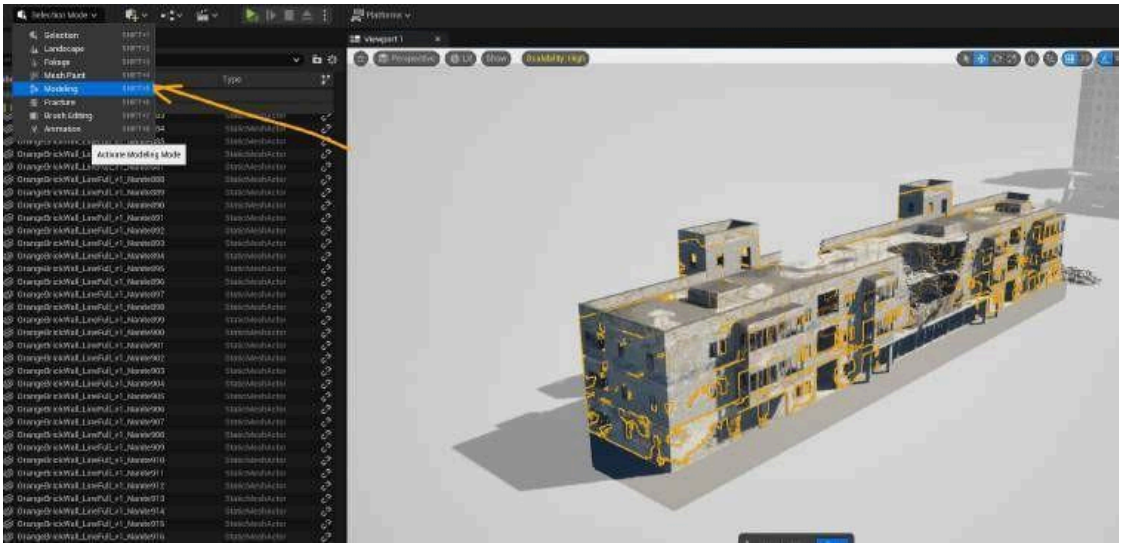
Now delete all static meshes we have selected to merge.



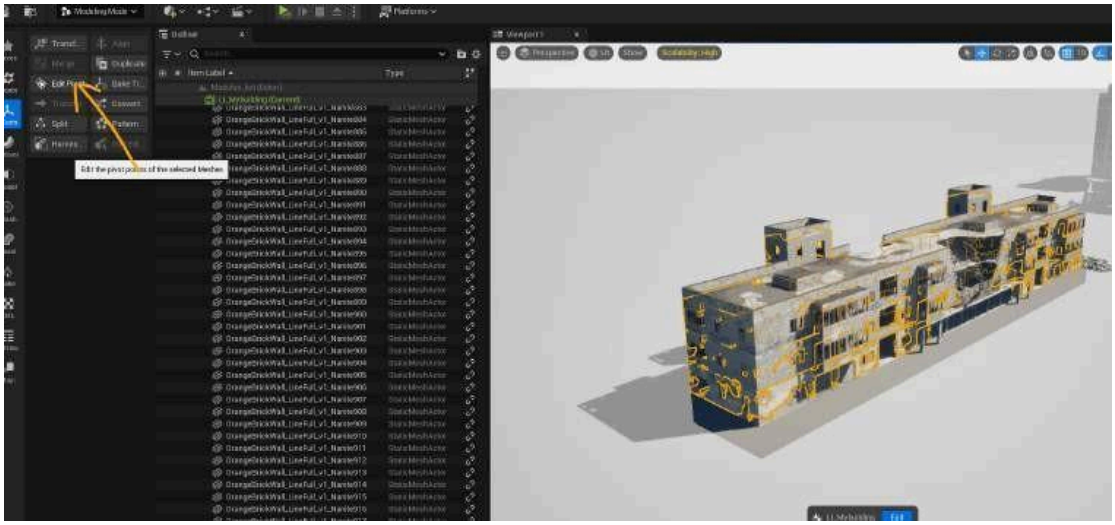
And place newly merged Static Mesh. Check so it has coordinates 0,0,0 to fit exactly as old paint meshes where.



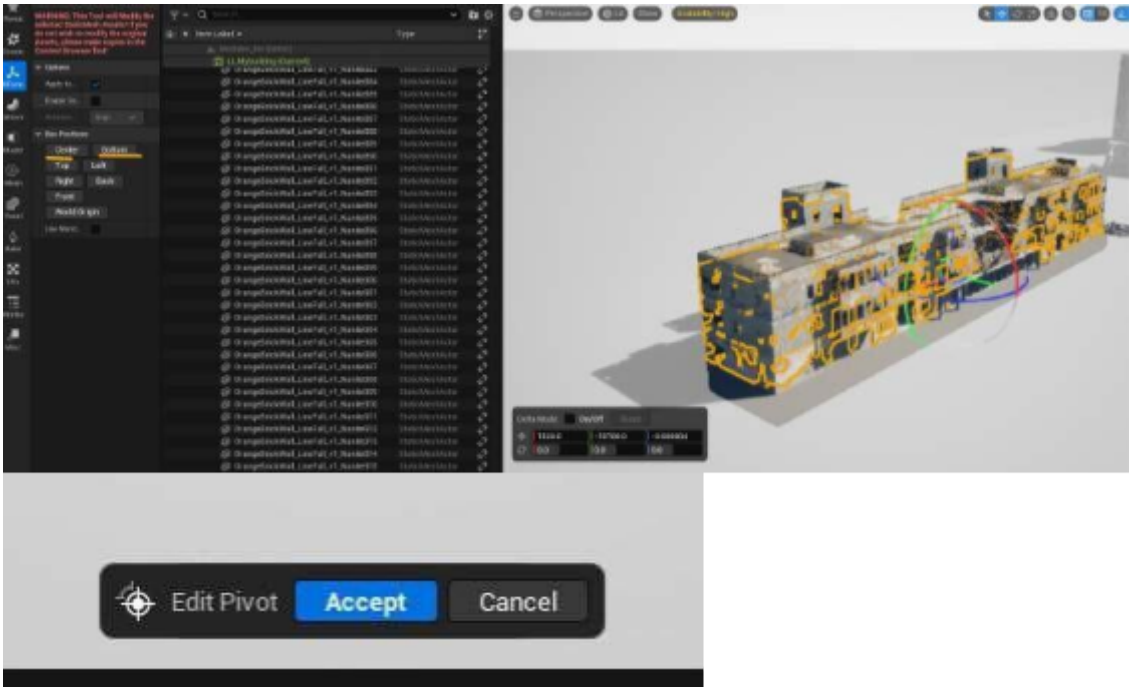
And the last step is to go to the modelling tab.



And find edit pivot mode.



Press Center Bottom and lastly Accept. This will ensure the correct pivot at the end.



And Press Exit



If for some reason your PLA looks weird.like this, just move it in a random direction and press Ctrl+Z.

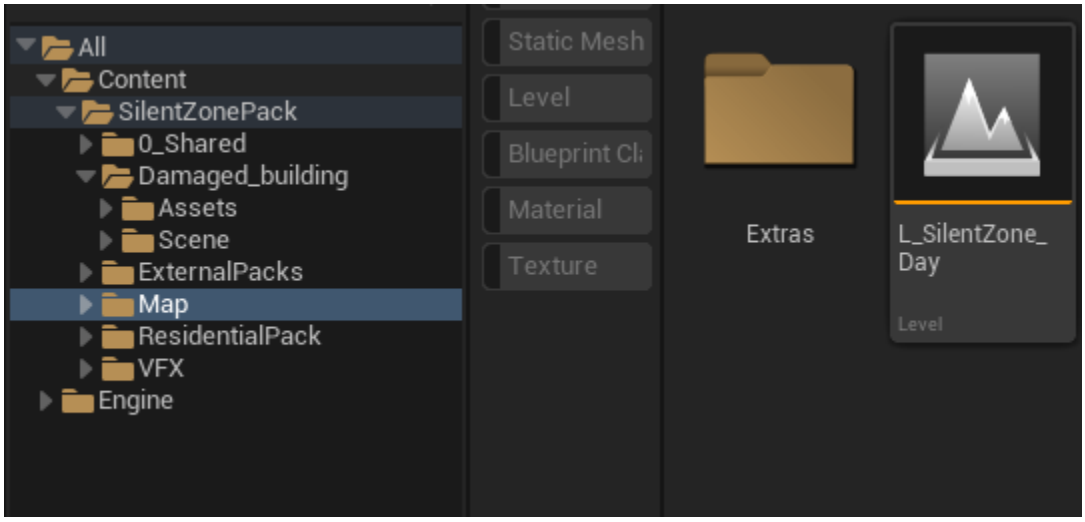


Congratulations! Now you finally have your building with vertex color applied!

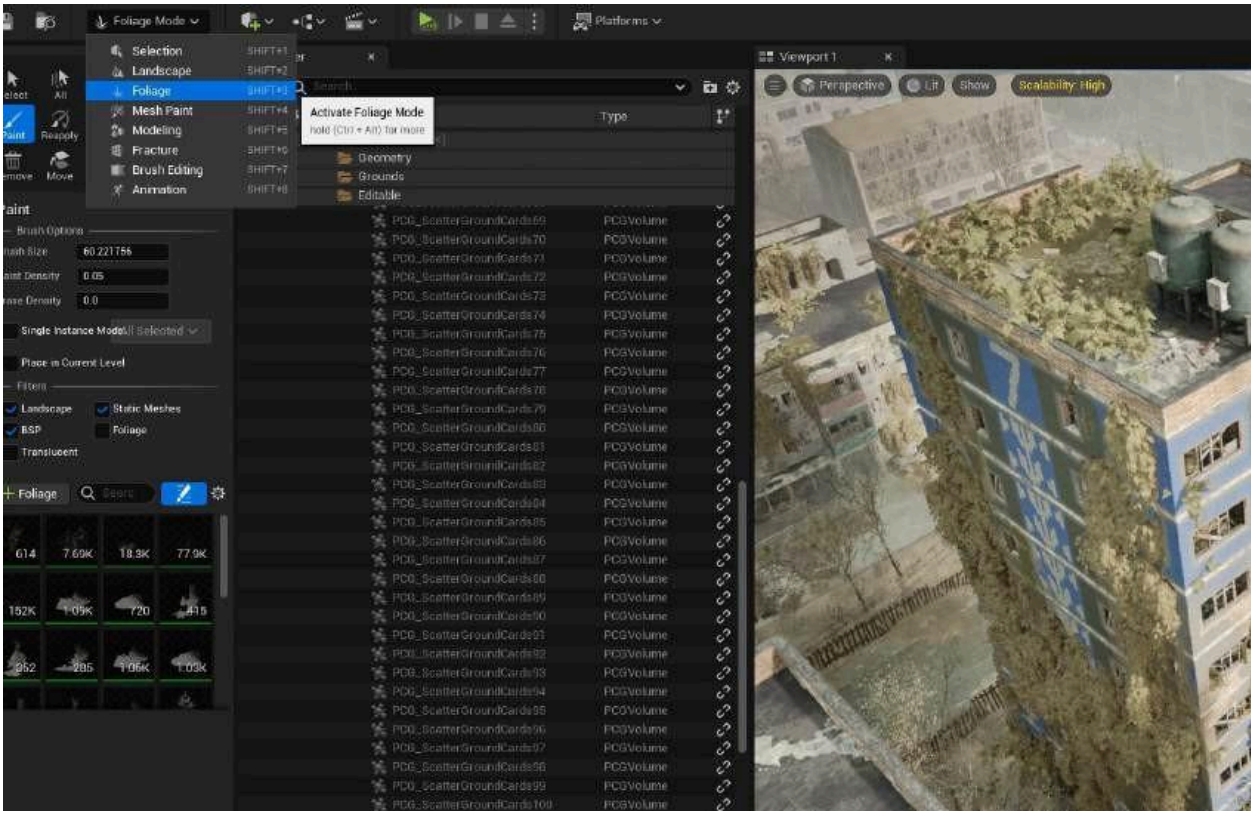


3.3. Adding foliage to buildings

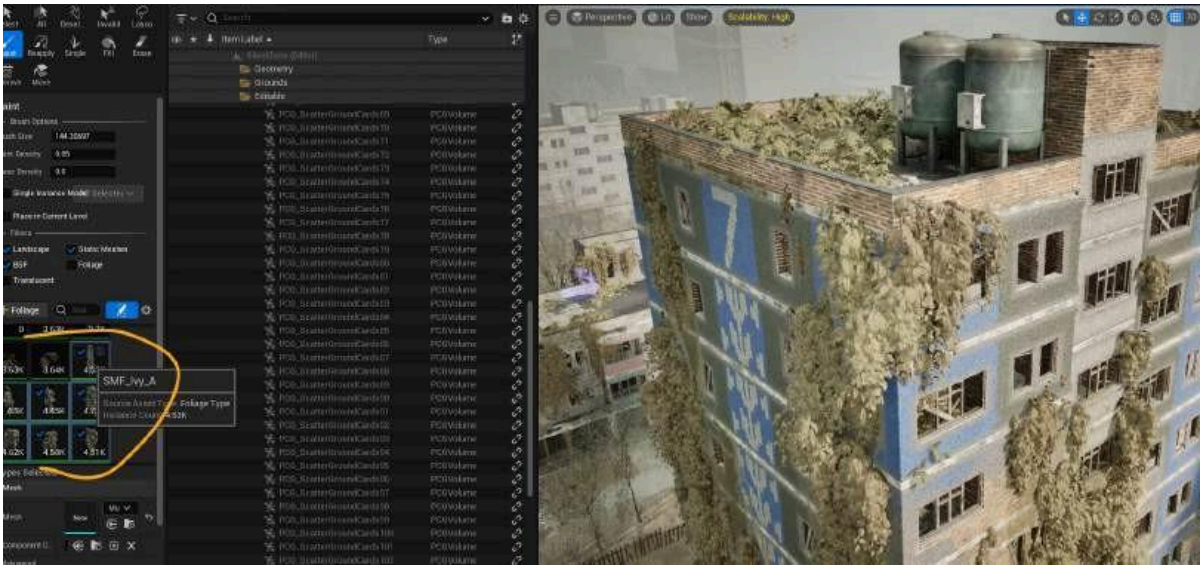
This one may be the most straightforward chapter.
Load Silent Zone level.



Go to the foliage tab.



Select the foliage you want. Please note that lots of small debris is also inside foliage.



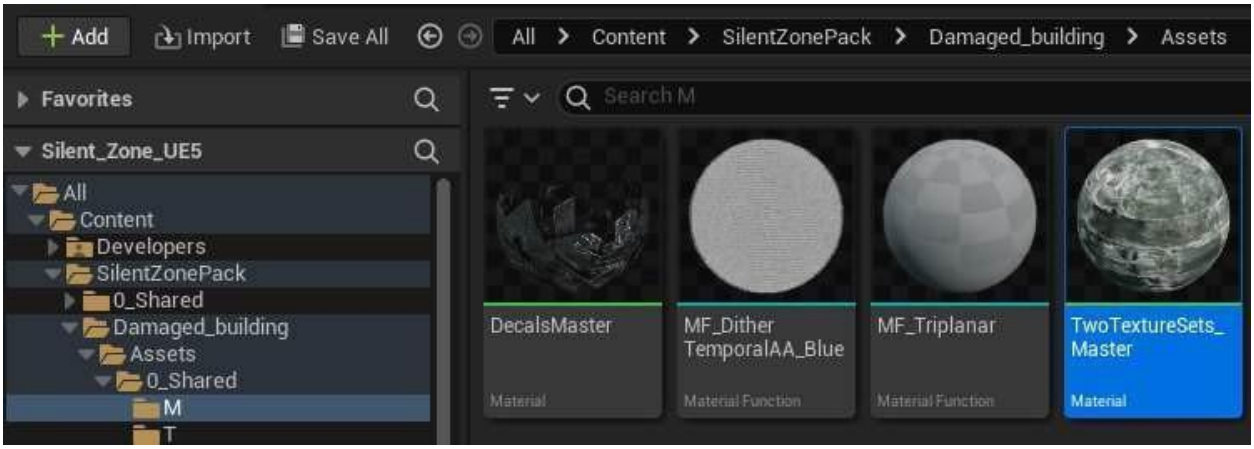
And enjoy making happy little accidents with the foliage brush!



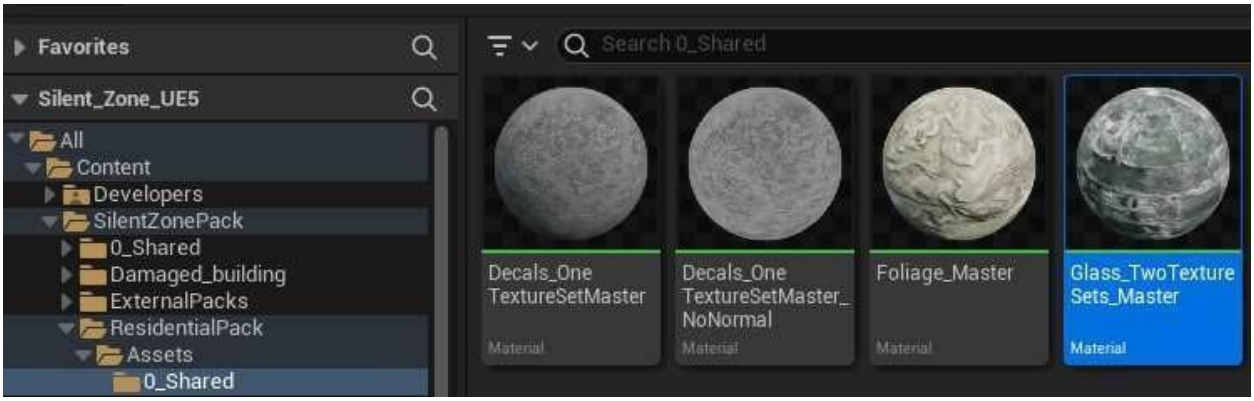
4.Shaders

4.1.Master Materials

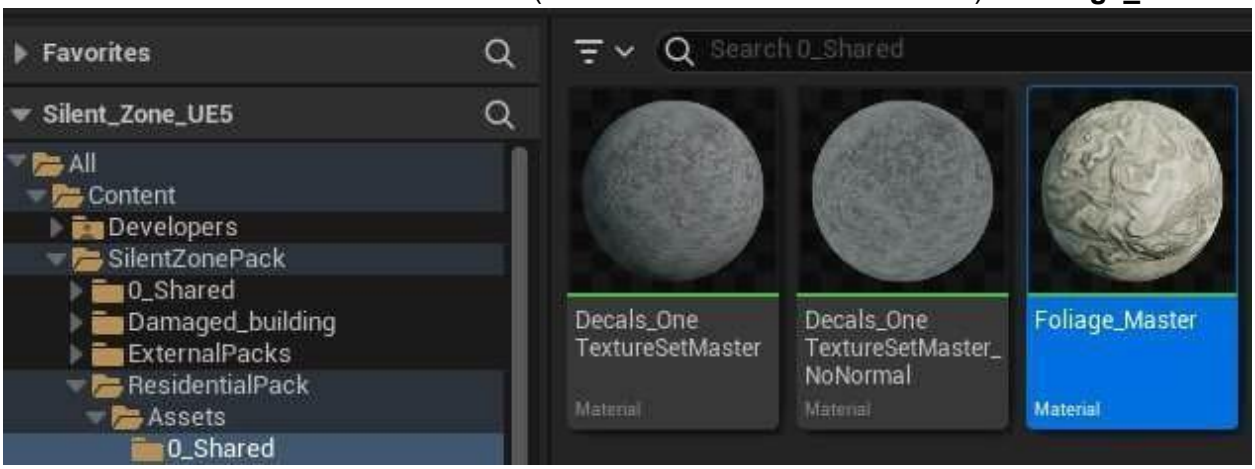
Main Master material for all opaque materials is **TwoTextureSets_Master**



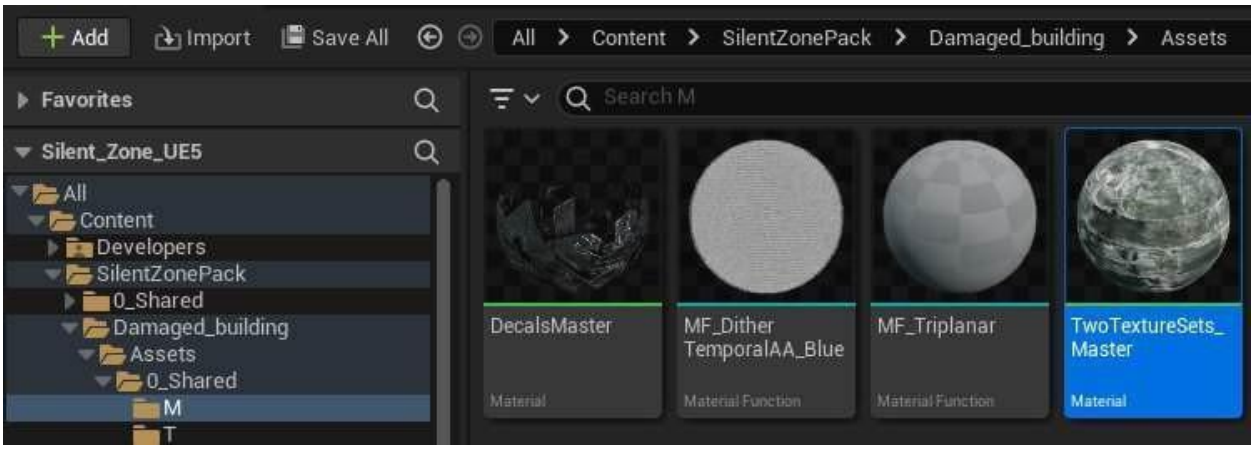
Main Master material for all transparent materials is **Glass_TwoTextureSets_Master.**



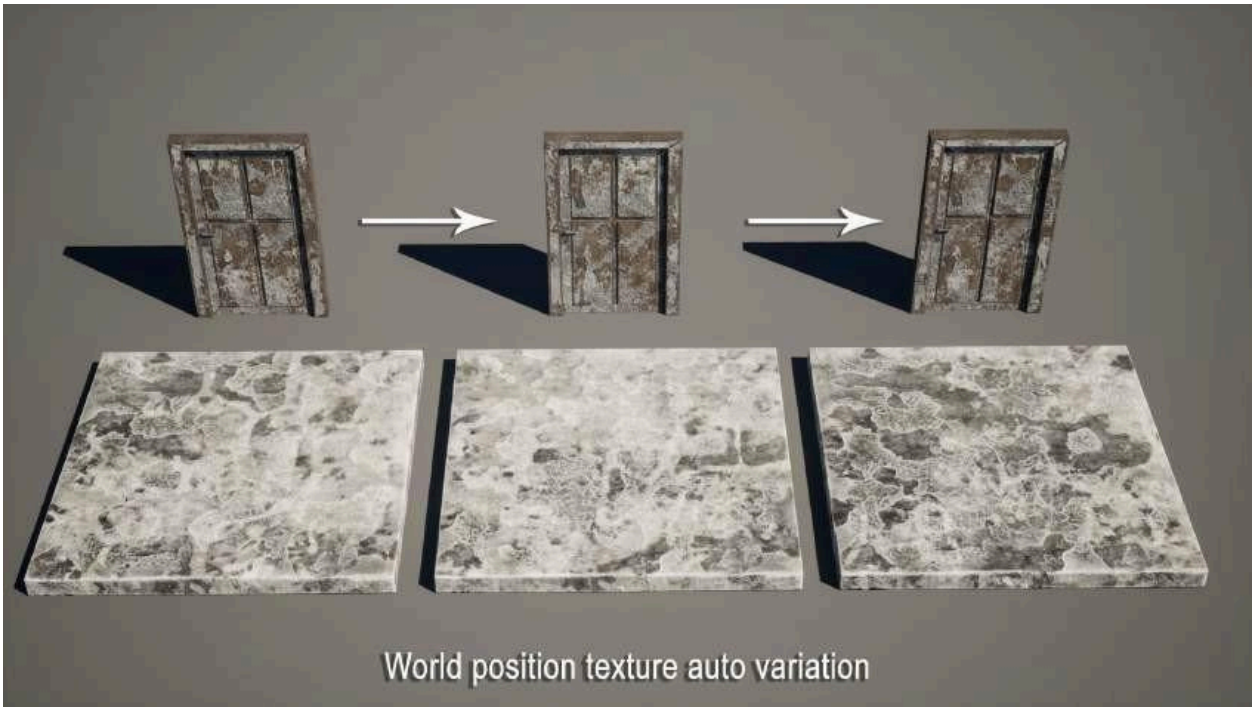
Main Master material for WPO materials (wind waved branches and leaves) is **Foliage_Master.**



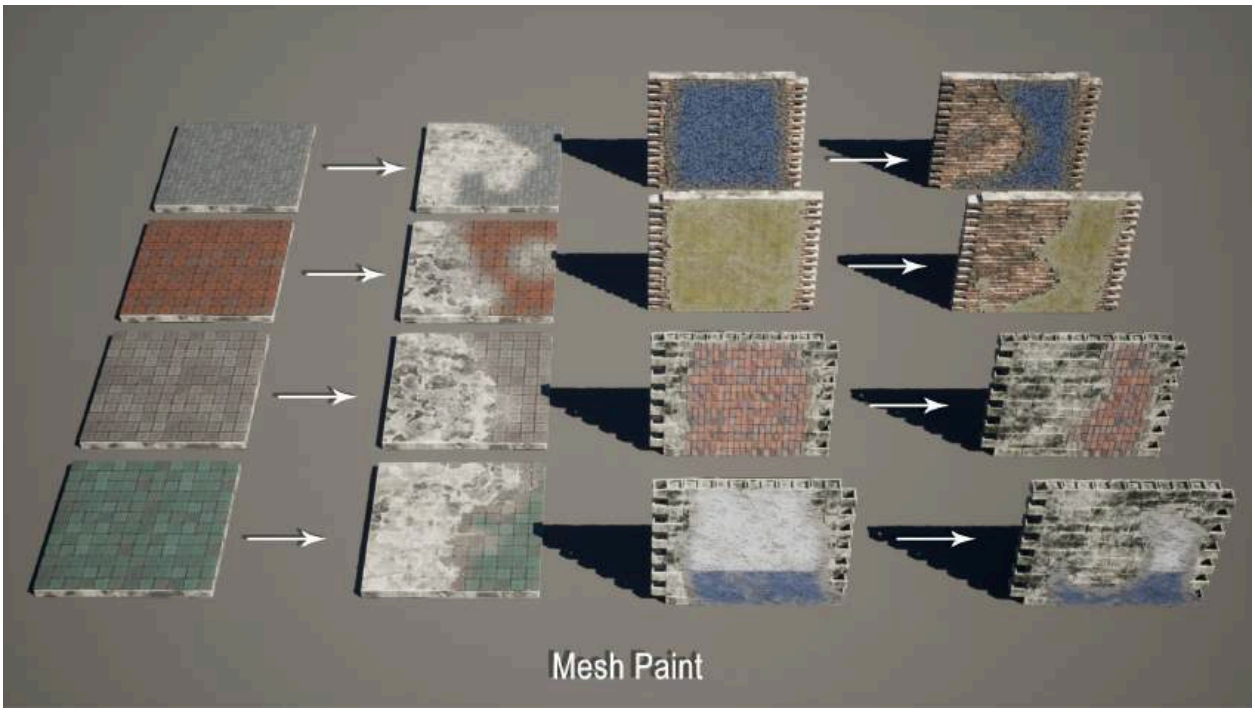
4.2. TwoTextureSets_Master



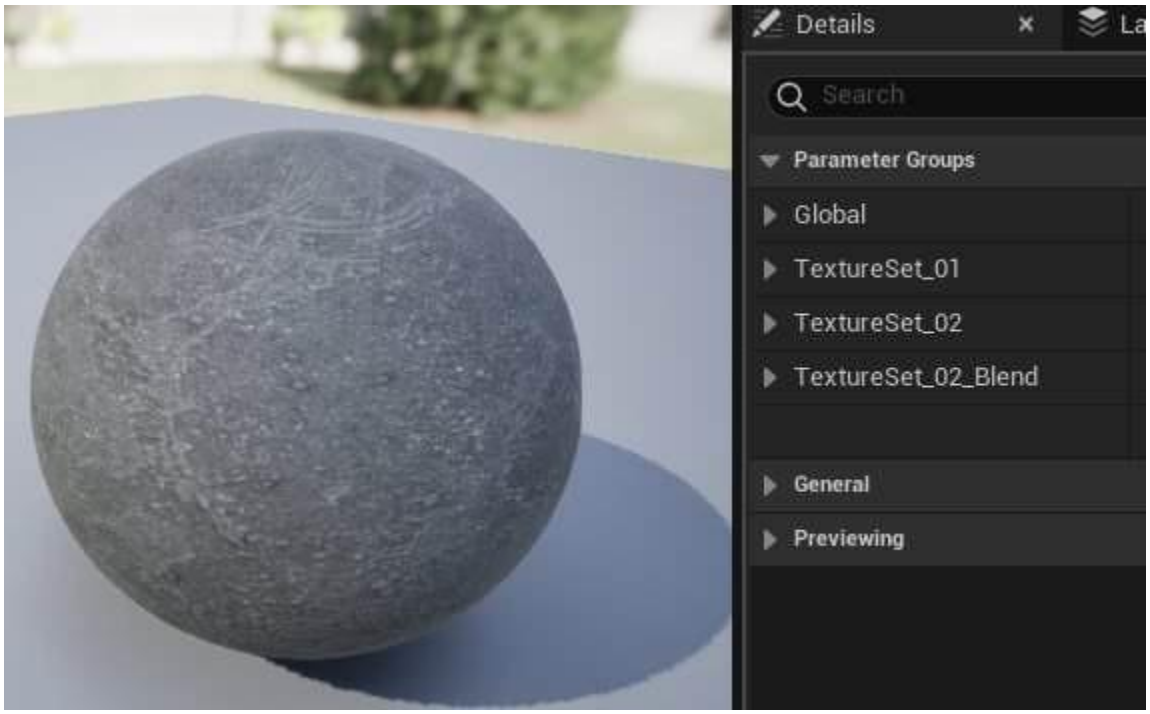
The basic idea is simple. Each of such assets using these shaders gets two different texture sets which are blended by noise texture in world space. This way we can



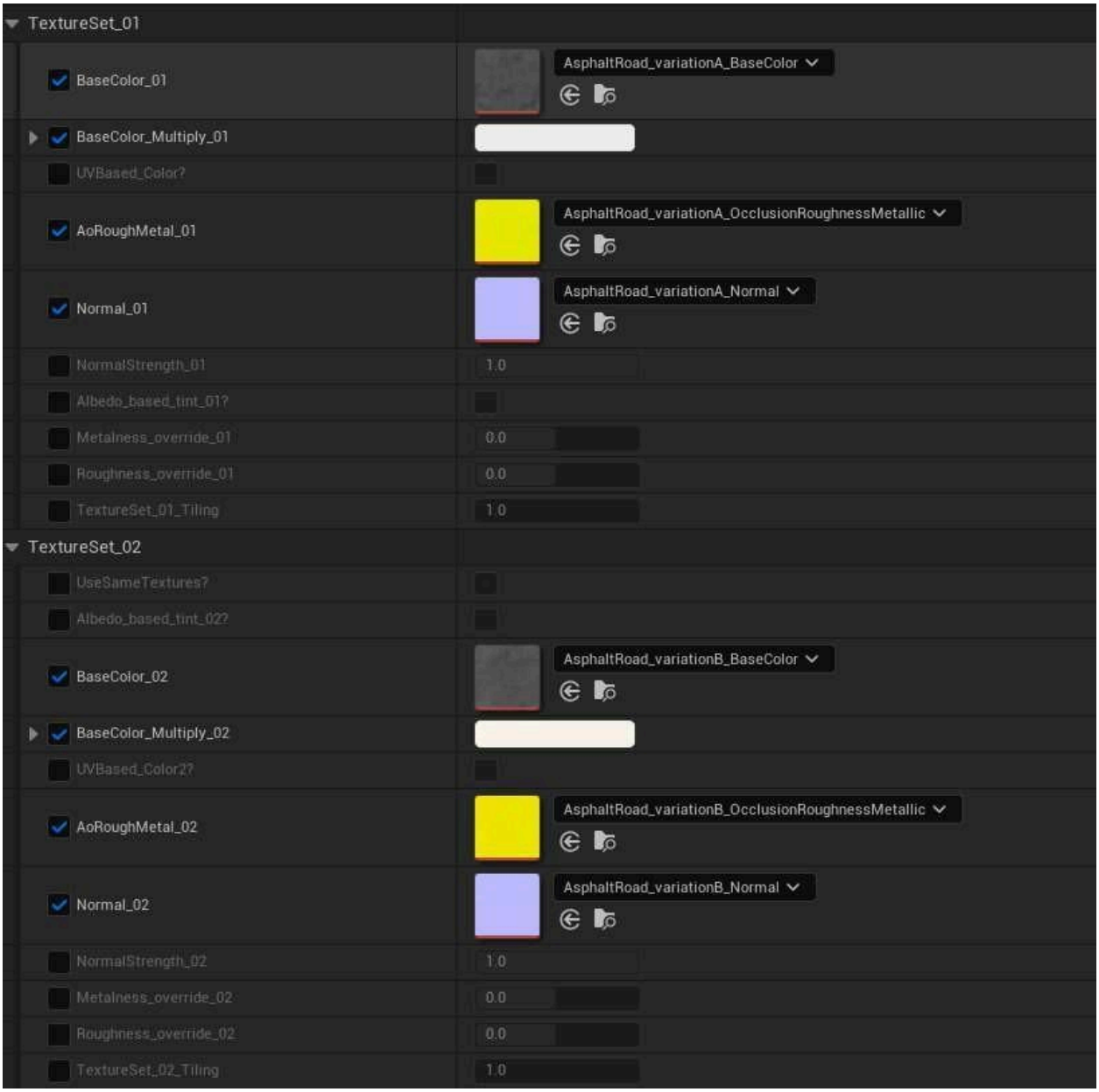
But what if you want to use some manual vertex paint instead of some random world space texture? This shader supports it too! Even a combination of both at same time.



Sounds exciting?Yeah! But it is probably very confusing to use…
Actually, no.
TextureSet_01 parameters group — everything related to the first texture set.
TextureSet_02 — to the second texture set.
TextureSet_02_Blend — is how they will blend together.



That's it!
But if you want more details. Sure here they are.
Most parameters for both **TextureSet_01** and **TextureSet_02** are identical. So I will focus on **TextureSet_01.**



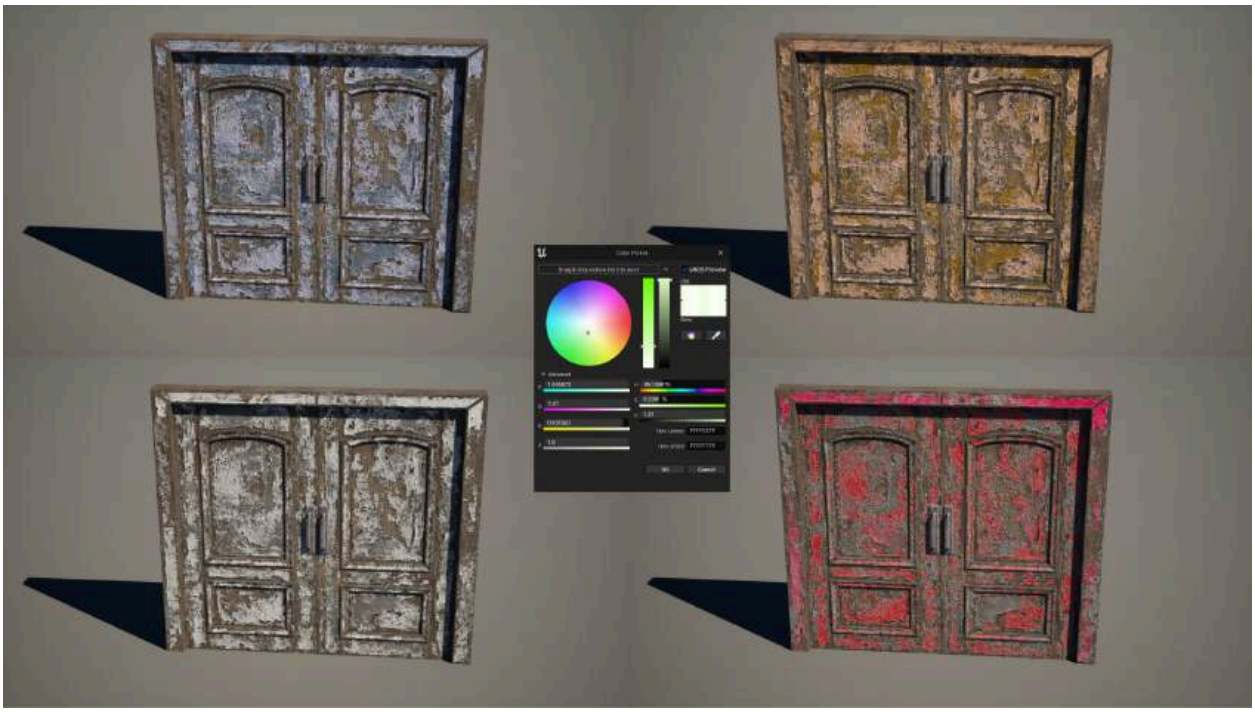
•**BaseColor_0**1 — Texture slot for sRGB Albedo Texture.
•**BaseColorMultiply_01** — good old fashion tint on everything. Just grabs albedo texture and multiplies by constant. Please note that you can go higher than 1 in luminosity making albedo brighter.



•**UVBasedColor?** — in case you want to have a gradient based on UV.
•**AoRoughMetal_01** — RGB texture input. Please note that textures for this input must have sRGB turned off in order to have correct gamma.

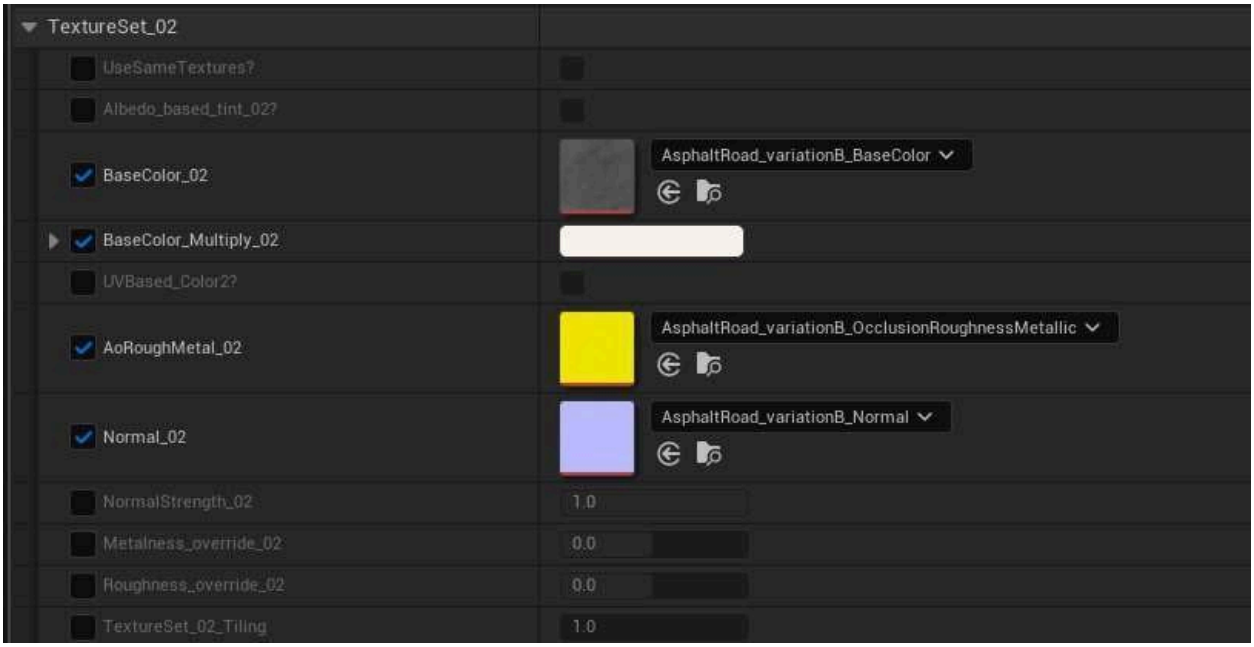


R - Ambien Occlusion
G - Roughness
B - Metalness
•**Normal_01** — Normal map input. UE will automatically use proper settings during import.
•**NormalStrength_01** — Strength of Normal map. Work as named.
•**Albedo_based_tint_01?** — This one is tricky. If you want only some part of albedo to be tinted. For example you made wooden material and there is light gray paint on top. And you want only light gray paint to become red, blue or some other color without affecting the wooden part of albedo. Please note that you have to have some part of the albedo map be significantly brighter than the rest of the map in order to work with this setting. You can set a threshold where tint will start.



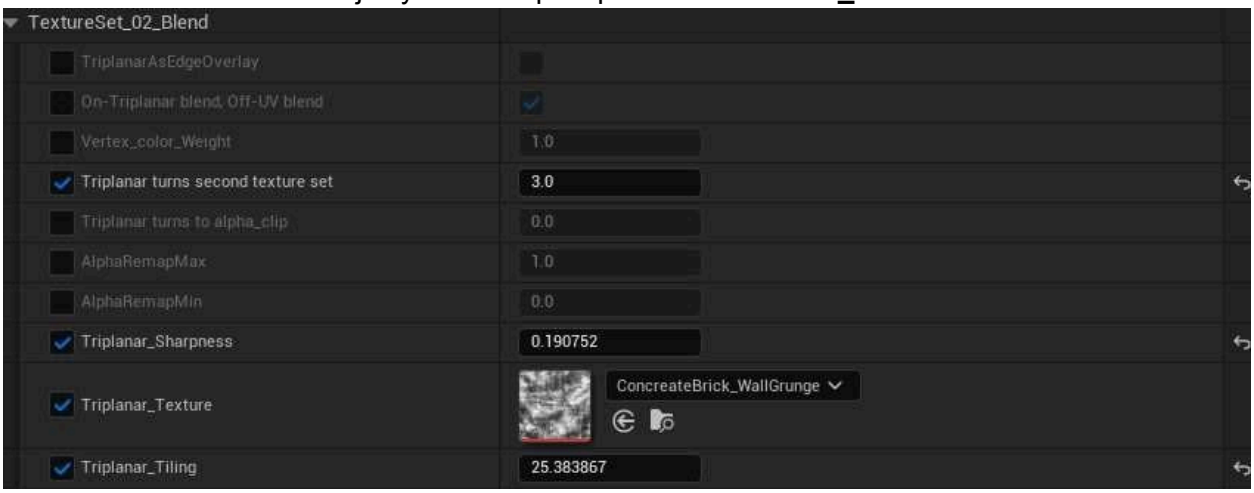
•**Metalness_override_01** — Adds this value to metalness
•**Roughness_override_01** — Adds this value to roughness
•**TextureSet_01_Tiling** — In case this is tileable material you can set tiling here.

•**All Settings for TextureSet02** — work the same way as setting for Texture Set01.



The only difference is **UseSameTextures?**
When toggled it will use the same textures as Texture Set 01. It may sound stupid like isn`t the whole point of this shader to mix two different texture sets?
But hear me out here. It can be useful if you just want to have just some extra world space noise based tint without sampling the same textures twice. Or you want to have different roughness on world space based noise. Or metalness. Or different colors of paint on wood. Way more flexibility and best of all reducing texture samplers two times. Kinda make it 2x faster. Now that may make sense.

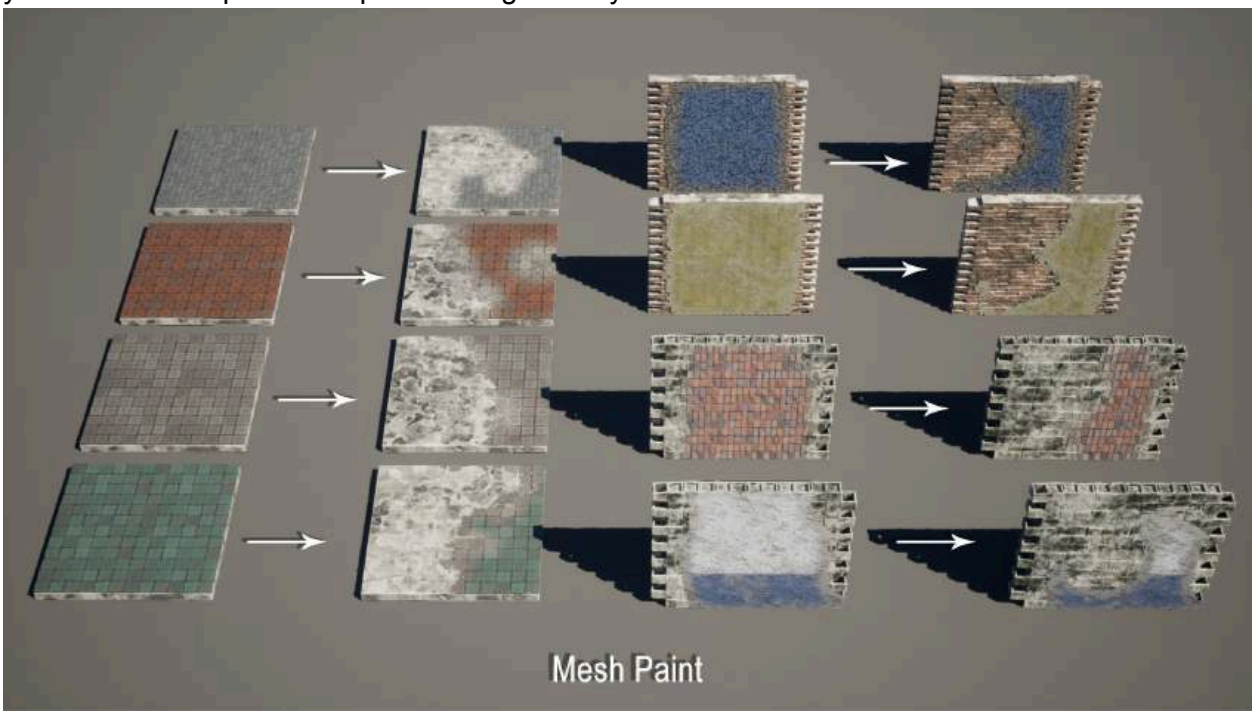
Now lets move to the most juicy and complex part **TextureSet02_Blend**.



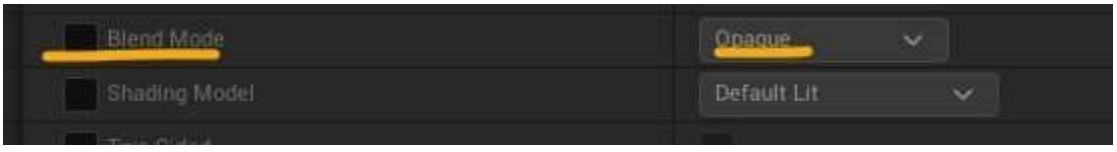
These settings can sound confusing. So lets go 1 by 1.

First, this triplanar still uses a single texture sampler. So it is cheap. Though it requires temporal AA to be enabled in the project, so keep that in mind. I was very proud of this discovery way back in 2023. And didn`t avoid any opportunity to tell everybody how smart that was. Now in 2026 Epic added this as the new default node in UE 5.7 material editor. So it is no longer as cool as it was. But thanks Epic for adding this!
So back to Shader.

•**Triplanar_Tiling** — Tiling of noise.
•**Triplanar_Texture** — Grayscale texture input for noise that will blend Texture Set 01 and 02.
•**Triplanar_Sharpness** — How sharp transition will be between X Y and Z projections. Most visible at 45 degree angles.
•**Triplanar_Texture** — Grayscale texture input for noise that will blend Texture Set 01 and 02.
•**Triplanar turns the second texture set** — How quickly noise will switch from Set 01 to Set 02. Kinda contrast setting. How smooth the transition will be.
•**Triplanar turns to alpha_clip** — Will turn the second texture set into a transparent clip. Useful when you want to have some extra layer of material with vertex paint on top of mesh. Like when you want tiles or paint on top of bricks geometry.



But in this case you will need to change Blend Mode to Masked in this particular Material Instance.



•**AlphaRemapMax and AlphaRemapMin** — Allows to change start, end, change contrast and event revert noise map which will control Alpha clip.
•**Vertex_color_Weight** — As I mentioned this transition between texture set 01 and texture set 02 is not governed only by some random world space noise texture map. You can use vertex paint to control it! This setting allows you to set how much more important vertex paint will be in comparison with the world space noise map. So it can only be world space noise. Or only vertex paint. Or some mix of both at same time!
•**On-Triplanar blend, Off-UV blend** — So maybe you do not want this noise to be world space based, but instead UV space based. You can do it here!
•**TriplanarAsEdgeOverlay** — And lastly maybe you want to use only vertex paint for blending and not really happy about blocky transition between vertex color due to low polycount. And you want a noise map to make transitions more detailed. Like use overlay photoshop mode on transition between black and white part of vertex paint. And overlay it with a noise map. Here is what this toggle is for!

So that was almost it.

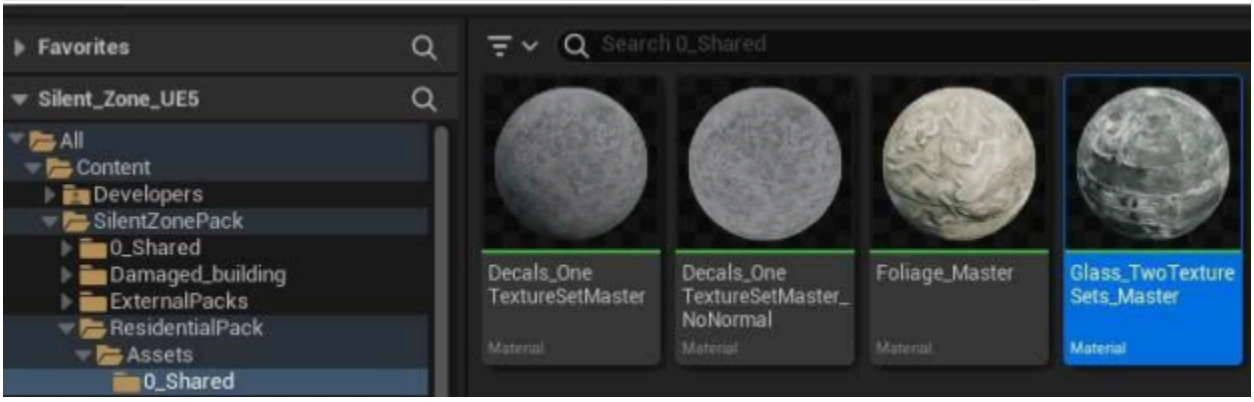
•**VerticalTint?** — Maybe all this was too confusing and you just wanted a gradient on static mesh from bottom to top. Like dirt from the ground on the prop. It can be done here!

So now that`s it!



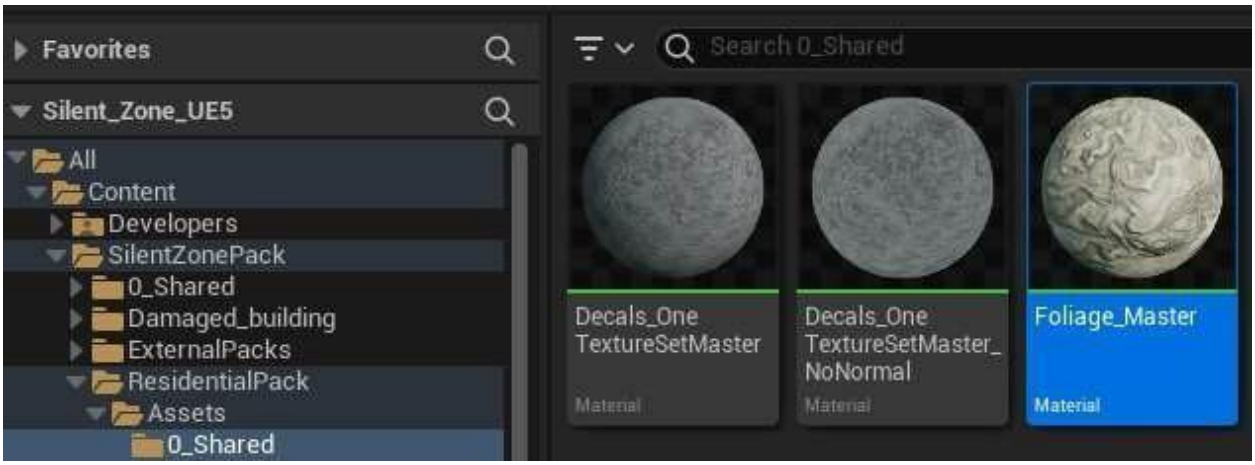
4.3. Glass_TwoTextureSets_Master



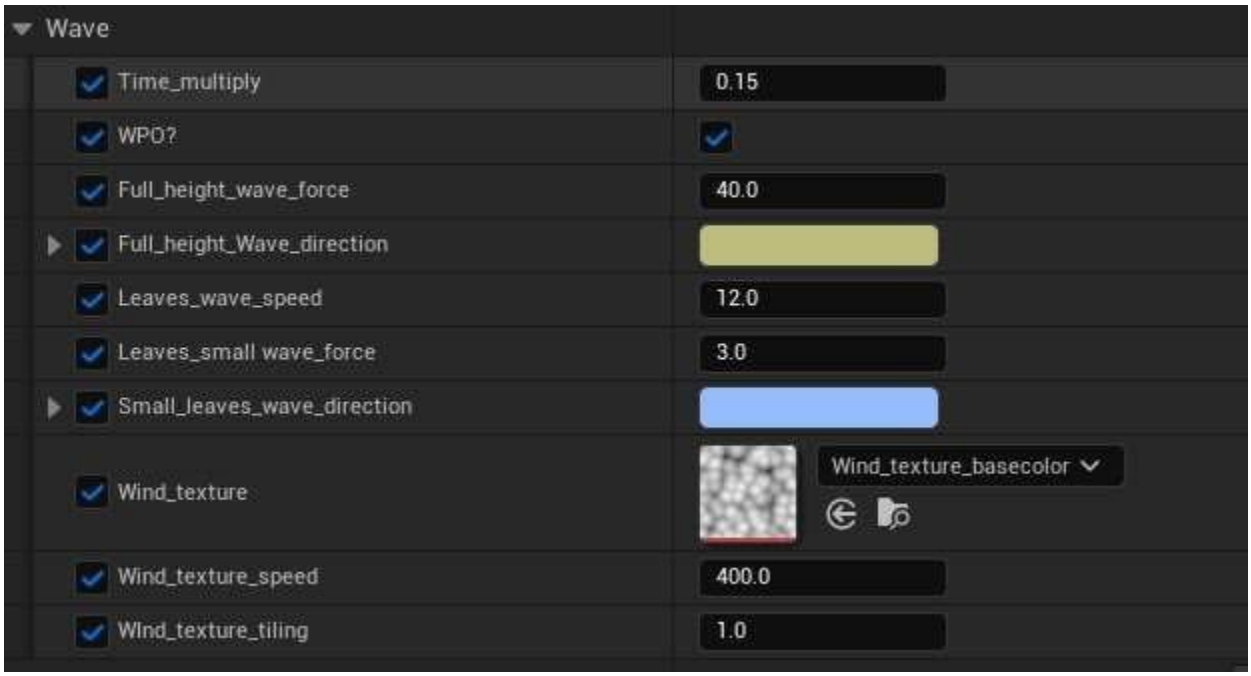


This one works totally the same as previous master material with only two key differences.
First. It supports partial transparency. It will grab it from **A** channel of Albedo.
Second. It works using TAA and Alpha Clip with marching Blue noise. In human words. It will work with dark post-apocalyptic glass, but will not work well with clean office like glass. Also it is cheap. A lot cheaper than Alpha Blend.

4.4. Glass_TwoTextureSets_Master



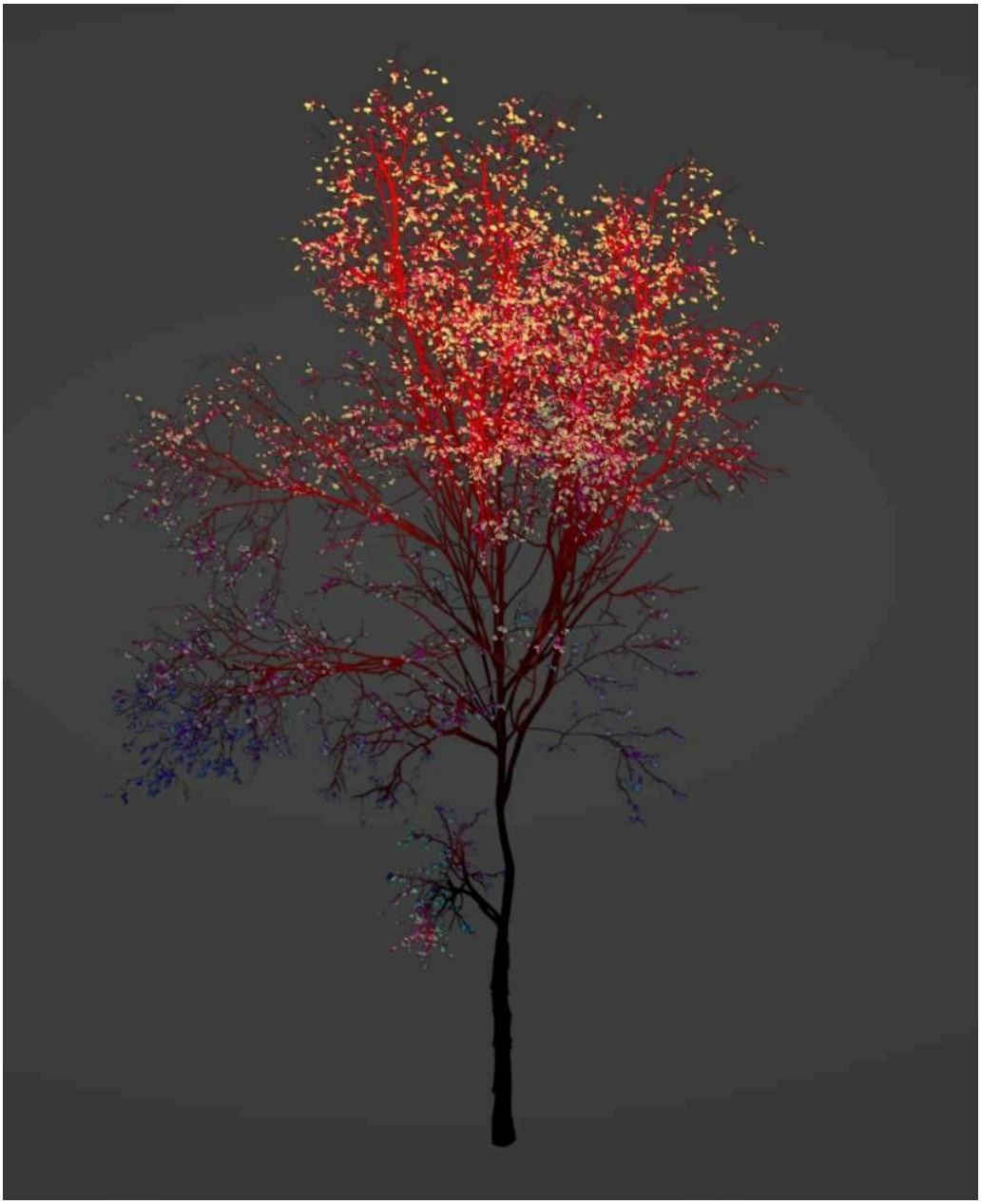
This one works totally the same as **TwoTextureSets_Master** with added Wind functionality on the World Position Offset slot.



•**WPO?** — Turn on/off wind
•**Time_multiply** — wind speed.
•**Full_height_wave_force** — amplitude of wind.
•**Full_height_Wave_direction** — 3D vector in direction of which mesh will be transformed as well as where texture will be scrolled.
•**Wind_texture** — Wind texture input.
•**Wind_texture_speed**— How quickly the texture will be scrolled..
•**Wind_texture_tiling** — World space tiling of wind texture.

•**Leaves_wave_speed** — speed of extra layer of wind for small leaves. It is controlled by **G** channel of foliage mesh vertex color
•**Leaves_wave_force** — Amplitude with which small leaves in the **G** channel will be transformed.
•**Small_leaves_wave_direction** — 3D vector in direction of which small leaves in **G** channel will be transformed.

Please note that leaves (**Mesh masked by Green Channel in Vertex Color**) are transformed in the direction of its normals, to improve its visual look.
The blue channel of vertex color will offset the phase of small leaves' wind to break the uniform look.
The red channel of vertex color will affect how much world space **Wind_texture** will affect mesh. Effectively making the top part of the tree bend much more than the lower part of the tree.



5. Final word

Wow, if you've made it from page 1 to page 50, you have our respect. This documentation took a long time to put together, and we truly appreciate you taking the time to go through it.

The Silent Zone pack is one of the most ambitious, detailed, and expansive projects we've created to date. Every asset, system, and workflow was designed with real production use in mind, to help you build faster and achieve high-quality results.

We want to sincerely thank you for your support. Your trust in our work is what makes projects like this possible, and it drives us to keep improving and delivering even better tools for creators.

Development on this pack began in July 2024 and was completed in July 2025, with ongoing refinements continuing into 2026. A lot has changed since then, but our commitment to quality and to our community remains the same.

Thank you for being part of it.