



UE4 Tips and Tricks

Chris Murphy

@HighlySpammable

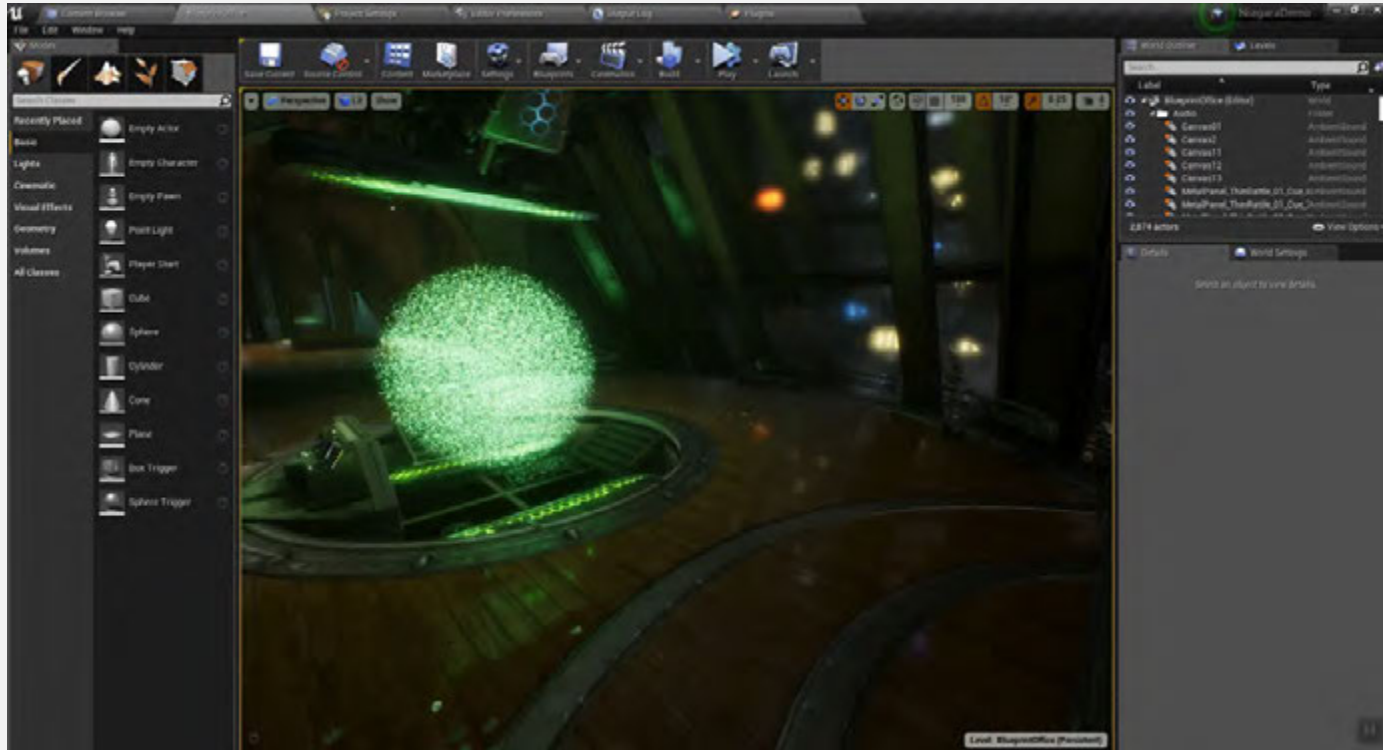
Unreal Engine Evangelist

A quick forward

My apologies to everyone as you will **need to take notes**:

- This presentation will be a little chaotic.
- Tips are going to be for all disciplines.
- I'm gonna go **quite quickly**.

Generate a list of Console Commands



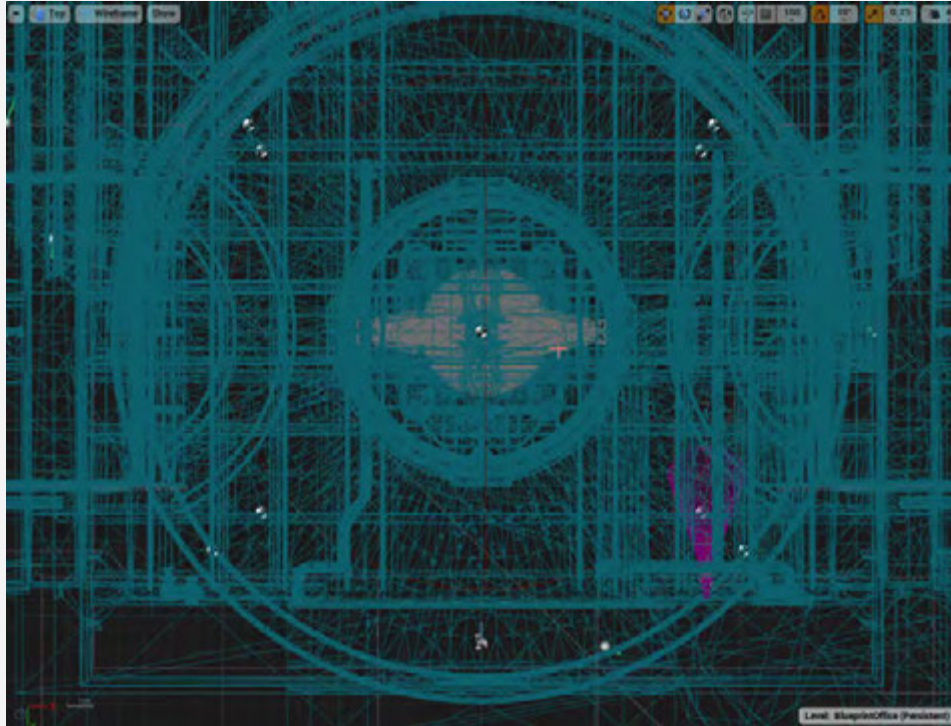
The Content Examples **Math Hall**



Blueprint helps different development disciplines **work together.**

The **best results** when developing with
Blueprint often come from **Engineer to
Artist mentorship**

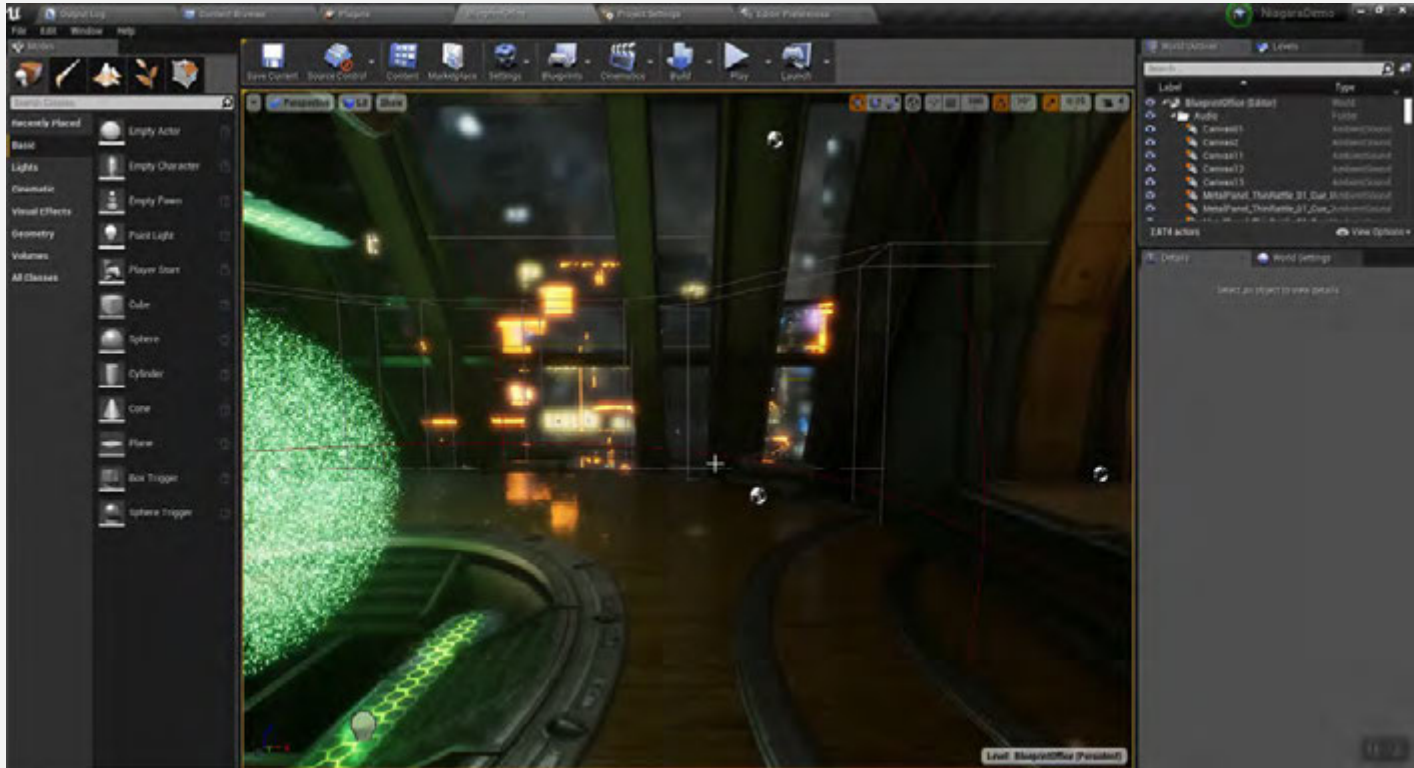
Middle Mouse Rules.



Alt+Middle Mouse to **Temporarily Pivot**



Ctrl+Middle Mouse To Change Views



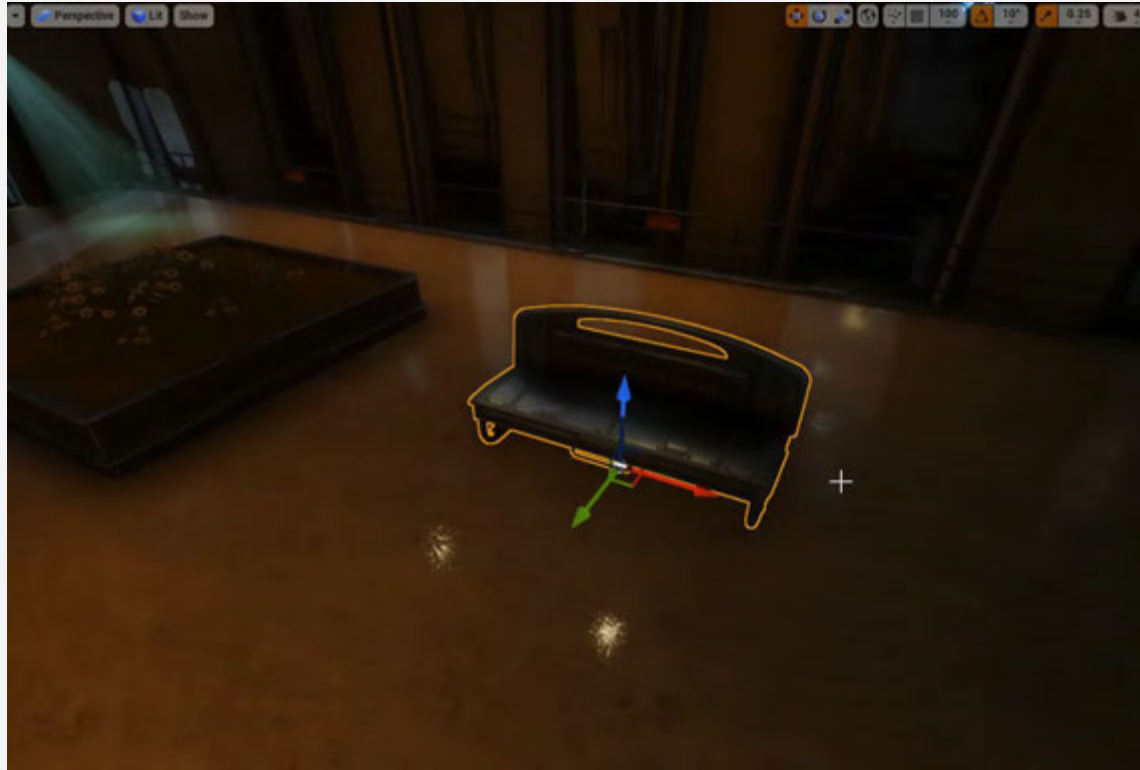
Shift+Drag to **Sync Camera Movement**



Ctrl+Mouse Left/Right/Both to **Transform**



Ctrl + ` to change between **Local** and **World**



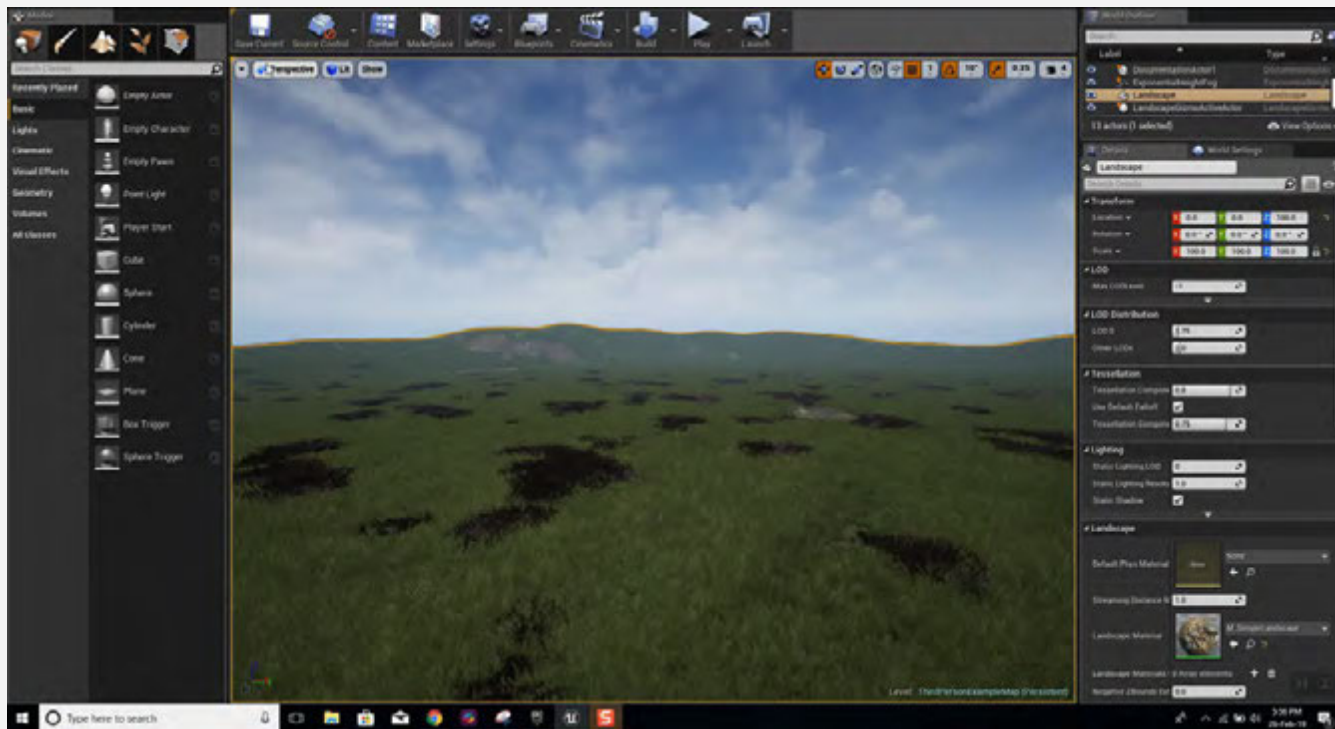
Scrolling while moving **alters your speed**



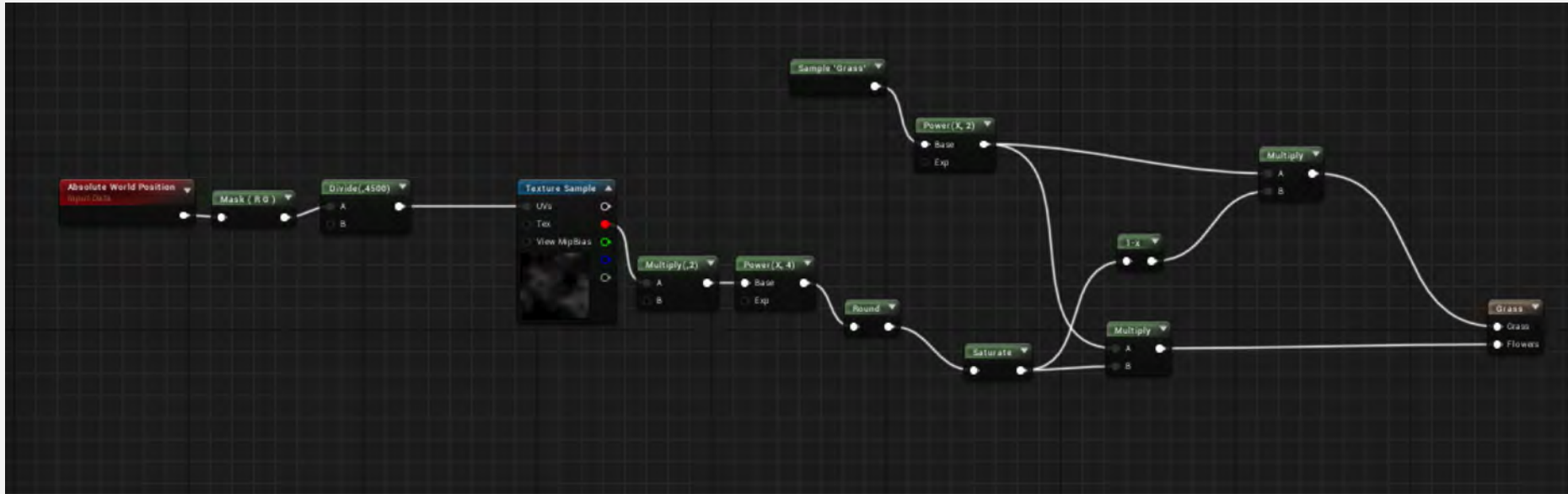
Press **End** to Snap Actors to the Floor



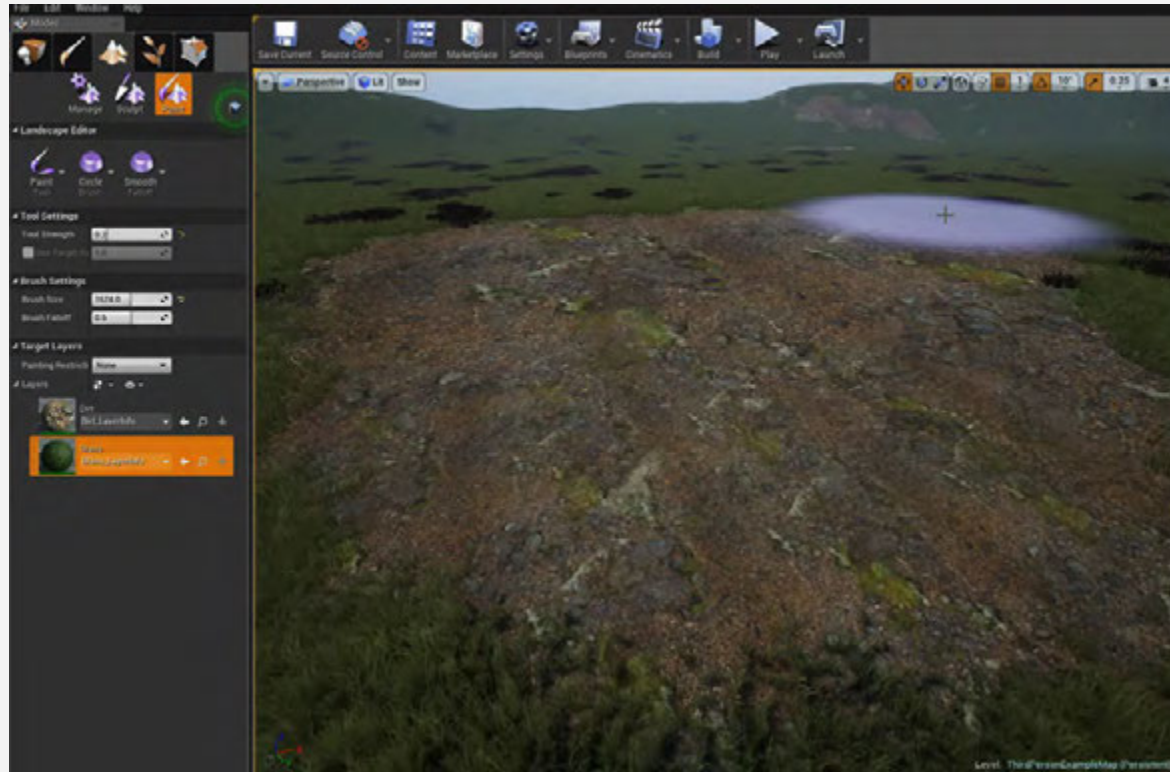
Split Perspective



Landscape Grass Node



Landscape Grass Node

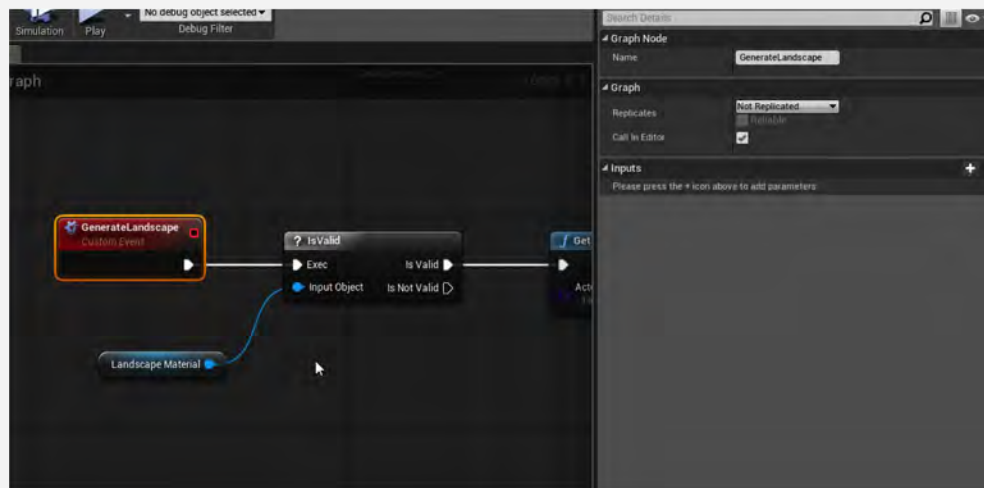


Use equations for value inputs

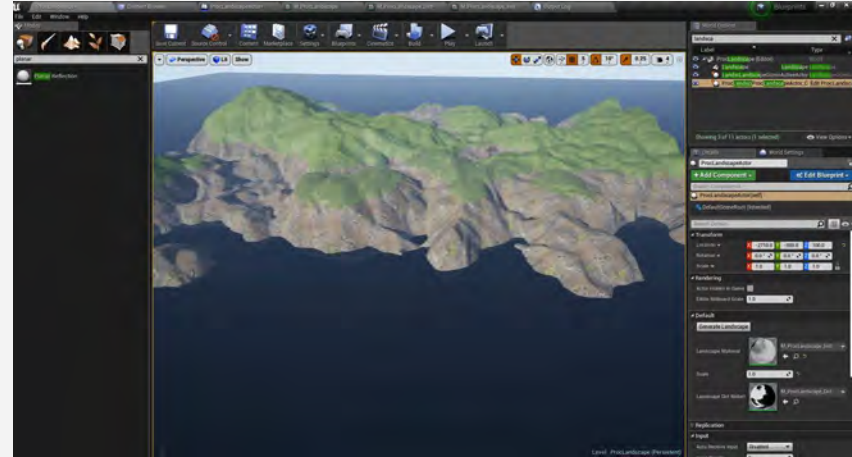
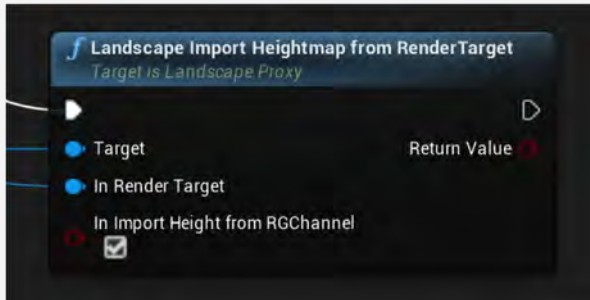
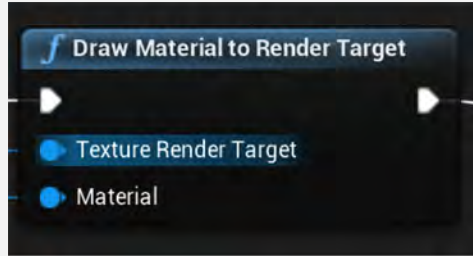


Add instant buttons with **Call in Editor**

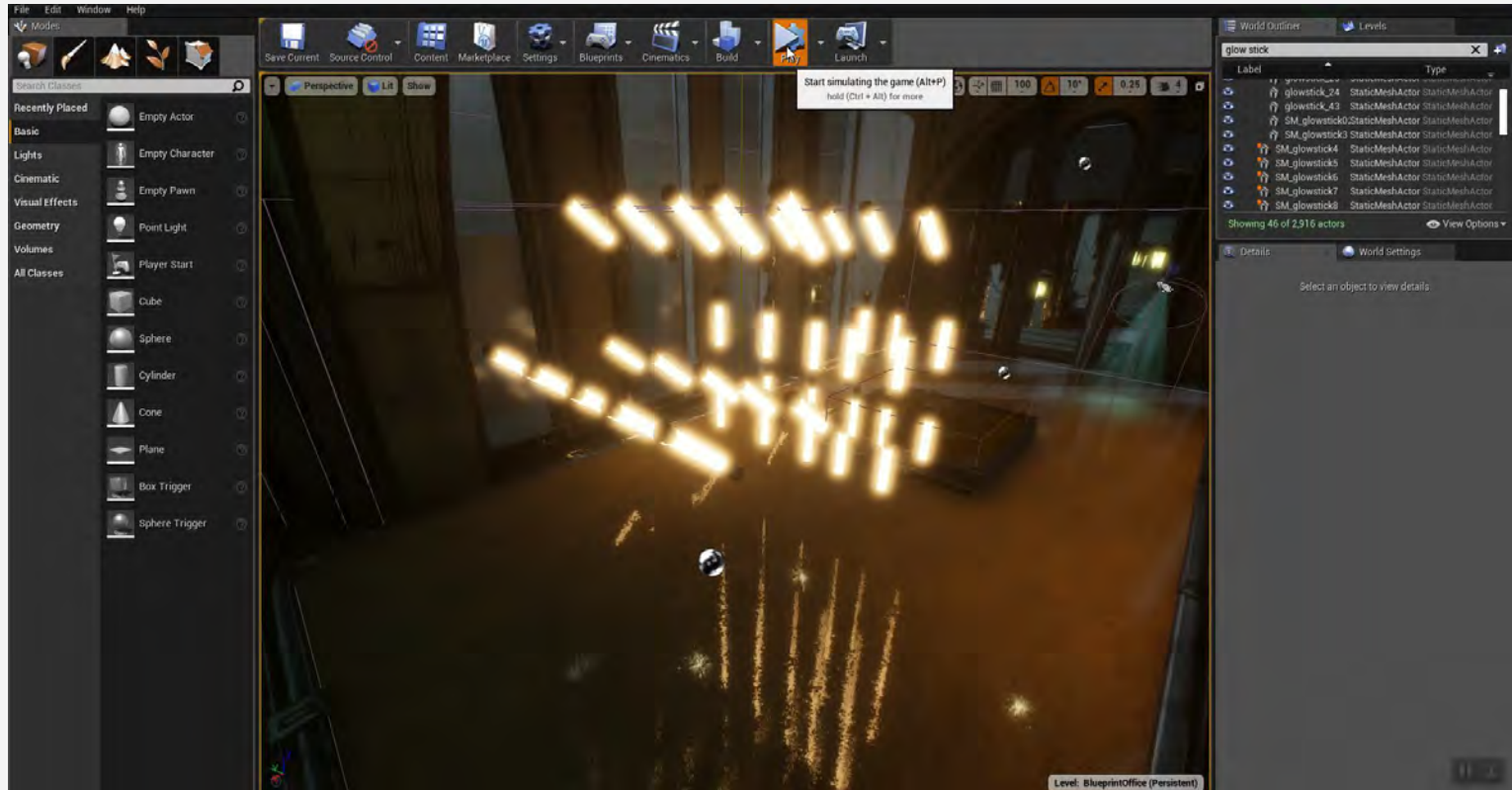
Selecting “Call in Editor” on an Event node in Blueprint will automatically turn it into a button.



Landscapes from Materials



Press K to Keep Simulation Changes



Content Browser Search Syntax

Advanced search terms are supported by the Content Browser! These include the ability to search Metadata such as:

- Number of Triangles
- Number of Collision Prims
- Material Attributes



*These terms are all searchable!
Just remove the spacing. For
example: Morph Target becomes
MorphTarget*

Use **Alt+Shift+O** to Quickly Open Assets

- Responds to the same commands as the content browser search field.
- Very quickly move between assets in a project.
- Searches any exposed folders, including Engine/Plugin folders.



Blutilities

The image displays the Unreal Engine 4 editor interface. The main window shows the Event Graph for 'TestBlutility', containing two custom event nodes: 'TestFunction' and 'OtherTestFunction'. The 'OtherTestFunction' node is highlighted with a yellow border. A 'BLUEPRINT' watermark is visible in the bottom left of the main editor area.

Overlaid on the right side is a 'Details' panel for the 'OtherTestFunction' class, showing the parent class as 'Global Editor Utility Base' and the name 'OtherTestFunction'. Below this, two utility cards are shown:

- Editor Scripting Utilities**: Helper functions to script your own UE4 editor functionalities with Blueprint or other scripting tools. It is Enabled.
- Python Editor Script Plugin**: Python integration for the UE4 editor. It is Enabled.

In the bottom right corner, a 'ProductViewer' window is open, showing 'Blutility Details' and a list of actions: 'Test Function' and 'Other Test Function'.

Blutilities

The image displays a Unreal Engine Blueprint workflow for replacing materials on actors using a data table. The workflow consists of the following steps:

- ReplaceMaterialWithDataTable** (Custom Event) - Triggered by a **Custom Event**.
- Get All Level Actors Of Class** - Returns an array of **Actor Class** objects.
- Get Data Table Row Names** - Returns an array of **Table** row names.
- ForEachLoop** - Iterates over the **Out Row Names** array. The **Loop Body** contains:
 - Get Data Table Row ReplaceMaterialTables** - Returns a **Data Table** row containing **Row Found** and **Row Not Found** outputs.
 - Break MaterialPair** - Outputs a **Material Pair**, **Key**, and **Value**.
- Replace Mesh Components Materials on Actors** - Replaces materials on the actors using the **Material Pair**, **Key**, and **Value** from the previous step.

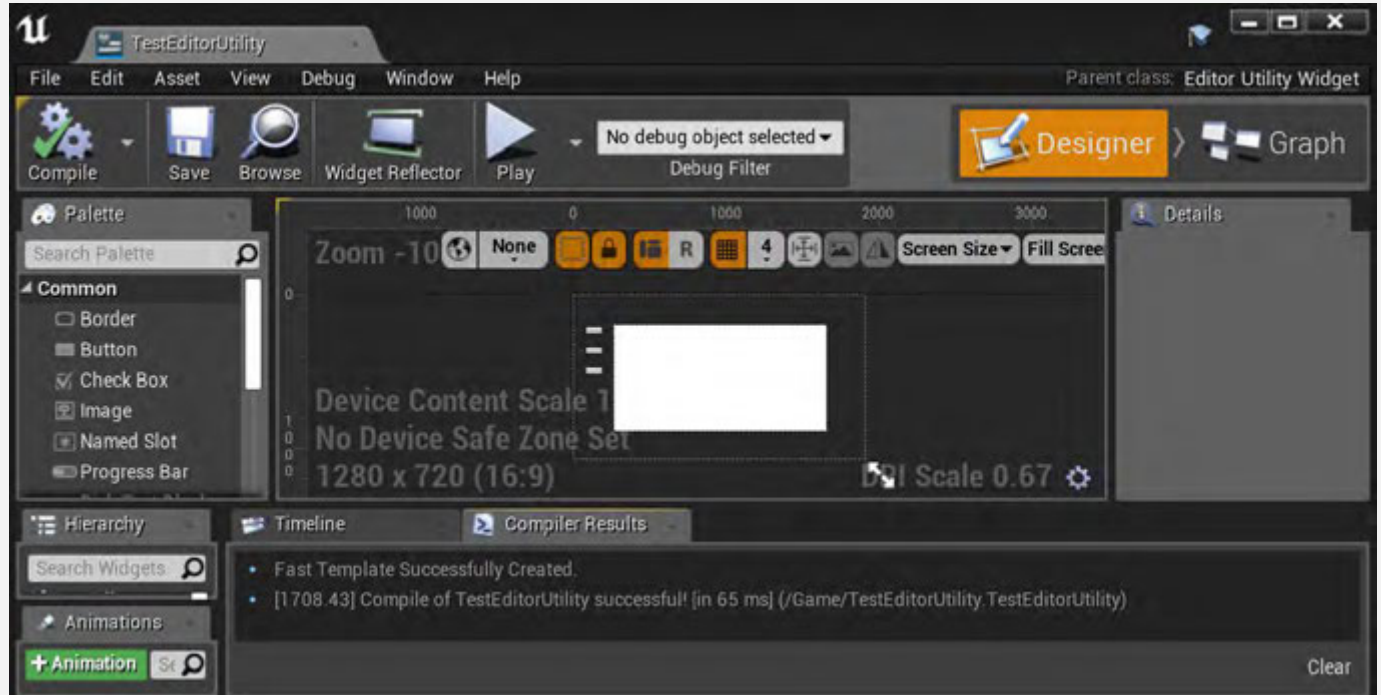
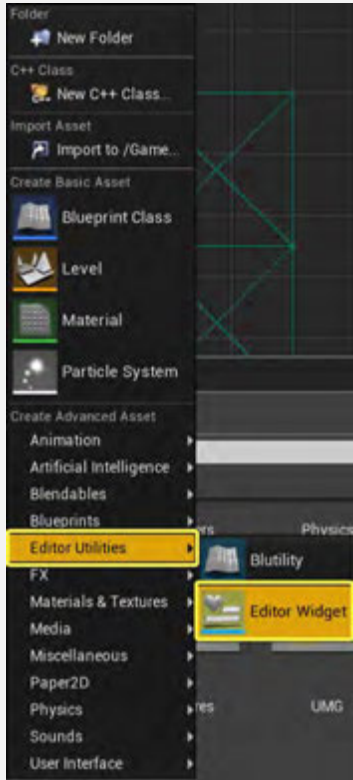
The inset window shows the **Data Table** editor with the following table:

Key	Value
Blue	Material /Engine/EditorMaterials/WidgetMaterial_Z, WidgetMaterial_Z; Material /Engine/EditorMaterials/Utilities/ImageReflectionPreview/Image
Red	Material /Engine/EditorMaterials/WidgetMaterial_X, WidgetMaterial_X; Material /Engine/EditorMaterials/Utilities/LinearColorPicker_MAT LinearC
Green	Material /Engine/EditorMaterials/WidgetMaterial_Y, WidgetMaterial_Y; MaterialInstanceConstant /Engine/EditorMaterials/Utilities/LinearColorP

The **Row Editor** for the **Blue** row shows the following configuration:

- Key:** WidgetMaterial_Z
- Value:** ImageReflectionPreview

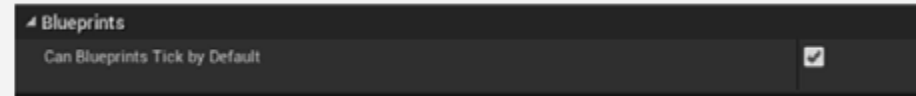
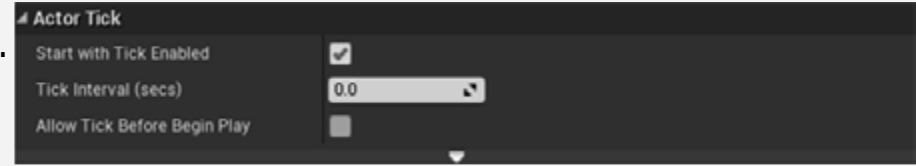
Editor Utility Widgets



Be careful what you Tick for..

Ticks are expensive. Think about why you're doing it. Consider disabling them by default. Alternatives include:

- Timelines
- Timers
- Manual toggling of Actor Tick on & off
- Polling vs Event driven systems (event dispatchers are often forgotten!)



```
[0003.6211985] ParticleSystemComponent /Temp/Autosaves/Game/Maps/UE4PCHighrise_Vista.Highrise_Vista:PersistentLevel.P_cars_straight_line_6.ParticleSystemComponent@ITickComponent). Enabled, ActualTickGroup: TG_DuringPhysics, Prerequisites: 0
[0003.6211985] ParticleSystemComponent /Temp/Autosaves/Game/Maps/UE4PCHighrise_Vista.Highrise_Vista:PersistentLevel.P_cars_straight_line_7.ParticleSystemComponent@ITickComponent). Enabled, ActualTickGroup: TG_DuringPhysics, Prerequisites: 0
[0003.6211985] ParticleSystemComponent /Temp/Autosaves/Game/Maps/UE4PCHighrise_Vista.Highrise_Vista:PersistentLevel.P_cars_straight_line_8.ParticleSystemComponent@ITickComponent). Enabled, ActualTickGroup: TG_DuringPhysics, Prerequisites: 0
[0003.6211985] ParticleSystemComponent /Temp/Autosaves/Game/Maps/UE4PCHighrise_Vista.Highrise_Vista:PersistentLevel.P_waterfall_mist_14.ParticleSystemComponent@ITickComponent). Enabled, ActualTickGroup: TG_DuringPhysics, Prerequisites: 0
[0003.6211985] ParticleSystemComponent /Temp/Autosaves/Game/Maps/UE4PCHighrise_Vista.Highrise_Vista:PersistentLevel.P_lightning_4.ParticleSystemComponent@ITickComponent). Enabled, ActualTickGroup: TG_DuringPhysics, Prerequisites: 0
[0003.6211985] ParticleSystemComponent /Temp/Autosaves/Game/Maps/UE4PCHighrise_Vista.Highrise_Vista:PersistentLevel.P_clouds_highrise_sky_4.ParticleSystemComponent@ITickComponent). Enabled, ActualTickGroup: TG_DuringPhysics, Prerequisites: 0
[0003.6211985] ParticleSystemComponent /Temp/Autosaves/Game/Maps/UE4PCHighrise_Vista.Highrise_Vista:PersistentLevel.P_clouds_highrise_sky_5.ParticleSystemComponent@ITickComponent). Enabled, ActualTickGroup: TG_DuringPhysics, Prerequisites: 0
[0003.6311985] ParticleSystemComponent /Temp/Autosaves/Game/Maps/UE4PCHighrise_Vista.Highrise_Vista:PersistentLevel.P_lightning_2.ParticleSystemComponent@ITickComponent). Enabled, ActualTickGroup: TG_DuringPhysics, Prerequisites: 0
[0003.6311985] ParticleSystemComponent /Temp/Autosaves/Game/Maps/UE4PCHighrise_Vista.Highrise_Vista:PersistentLevel.P_lightning_3.ParticleSystemComponent@ITickComponent). Enabled, ActualTickGroup: TG_DuringPhysics, Prerequisites: 0
[0003.6311985]
[0003.6311985] Total registered tick functions: 87, enabled: 77, disabled: 10.
[0003.6311985]
```

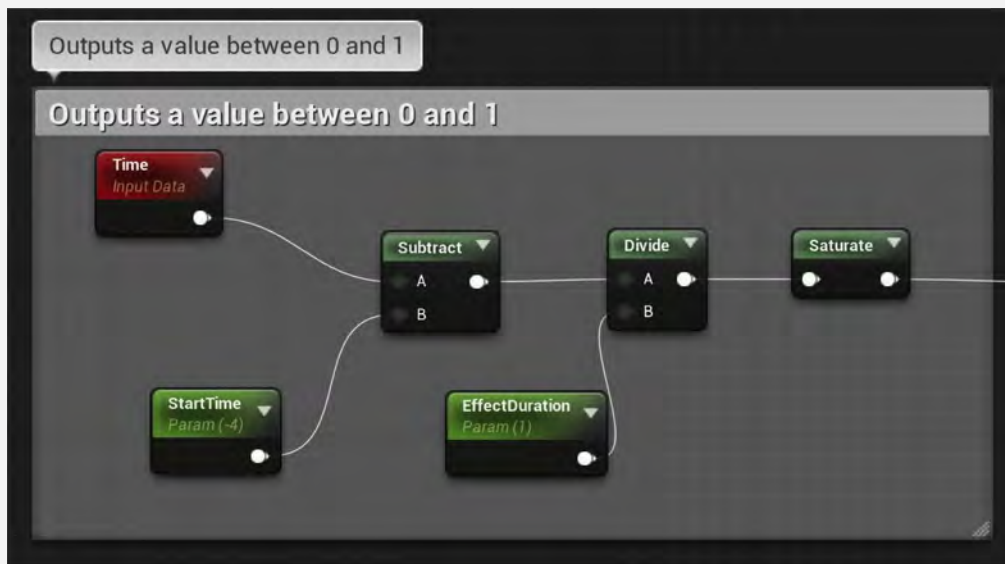
Why Tick when you can.. Not Tick?

By reading in a **Start Time** and **Duration** an effect can be fired-and-forgotten from within a material instead of ticked via a Blueprint.

One less thing to Tick.

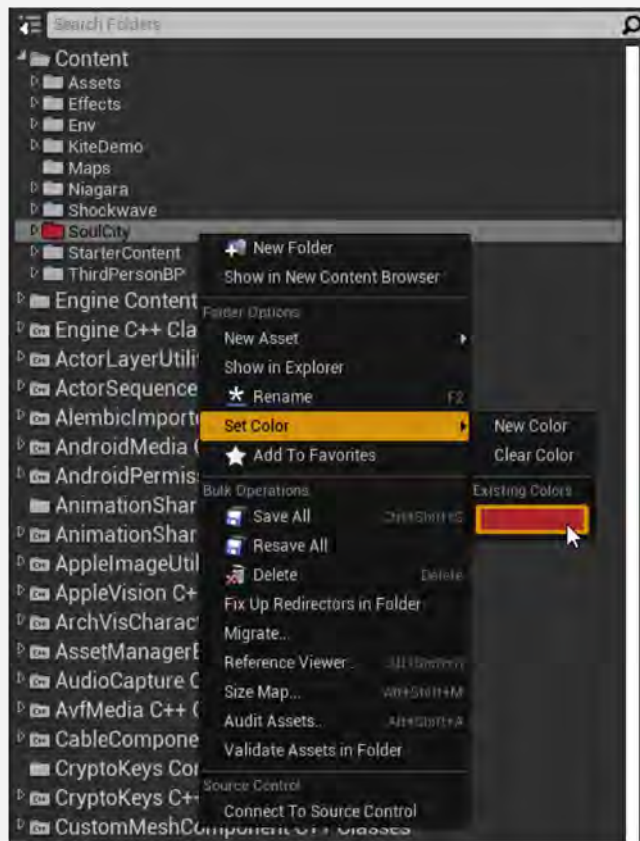
Easy to override.

Useful for short/contained effects like Damage & Health.



Color Code Folders

- Folders can be color coded for organisation
- Previous folder colors are cached to easily match types

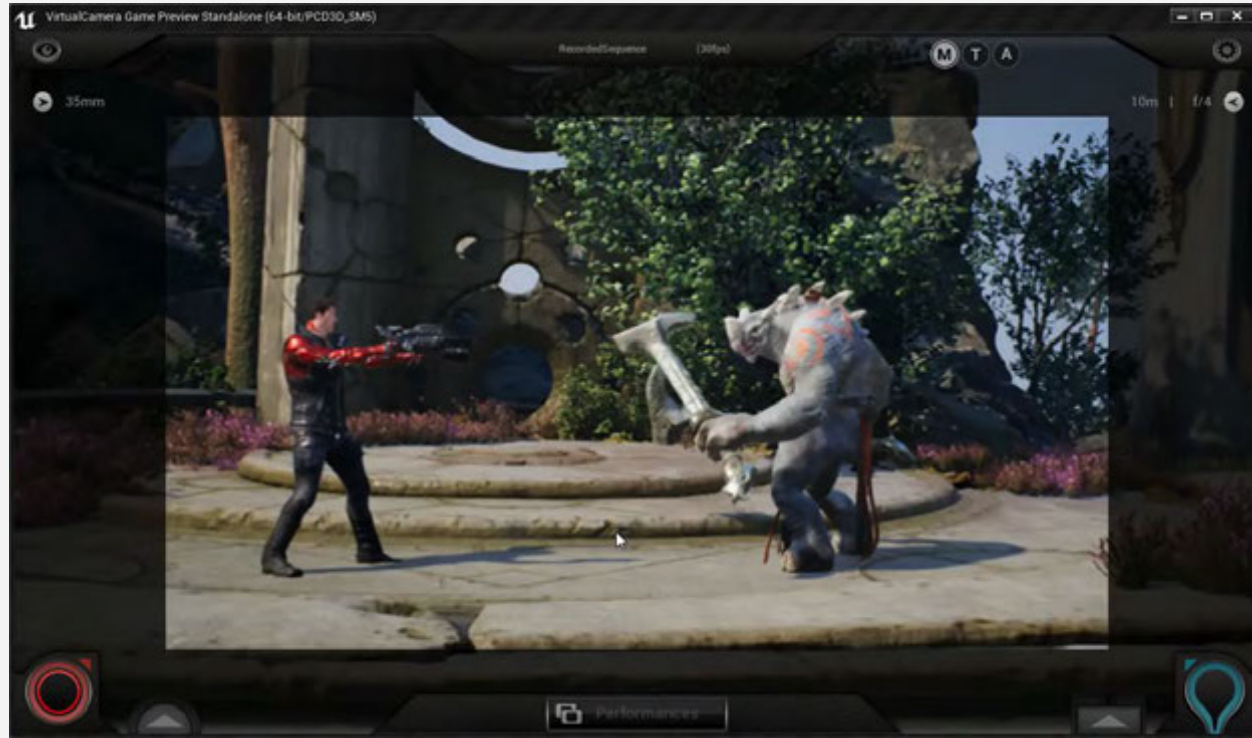


Color Picker

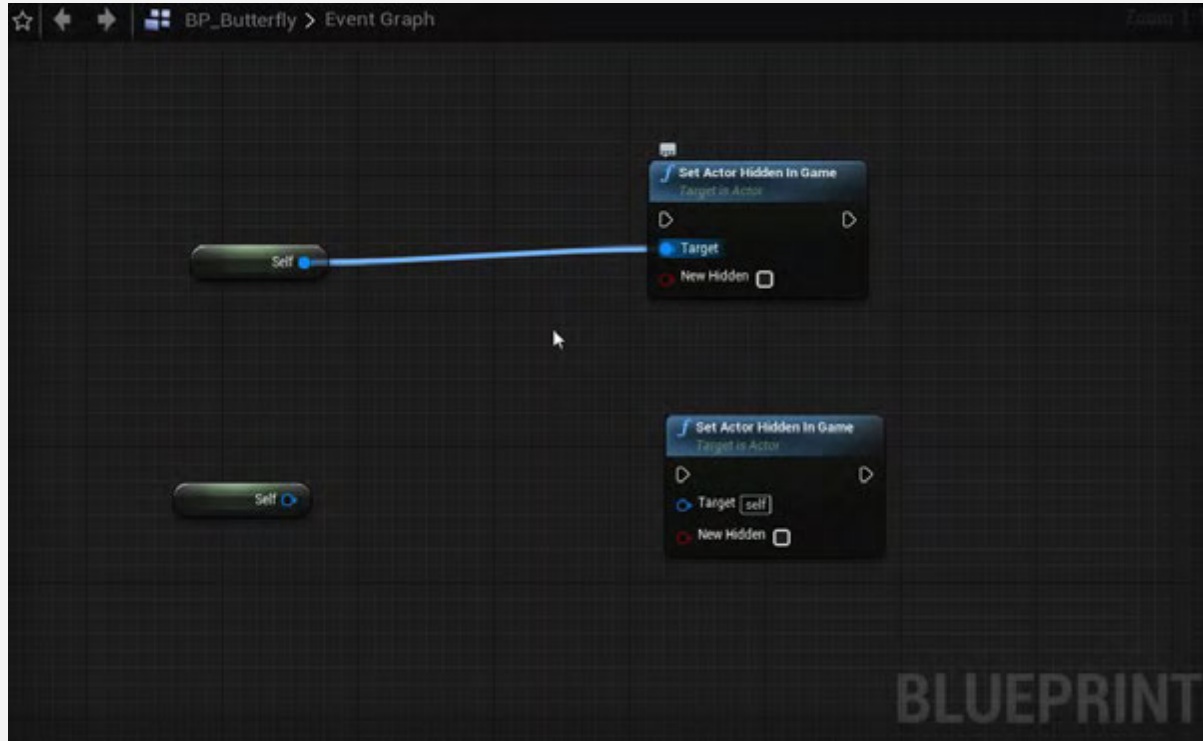
- Often overlooked as a feature
- Easily create and switch between themes



Virtual Camera



Use **Ctrl** to move Blueprint connectors



Promote to Variable



Gauntlet - Automation Framework

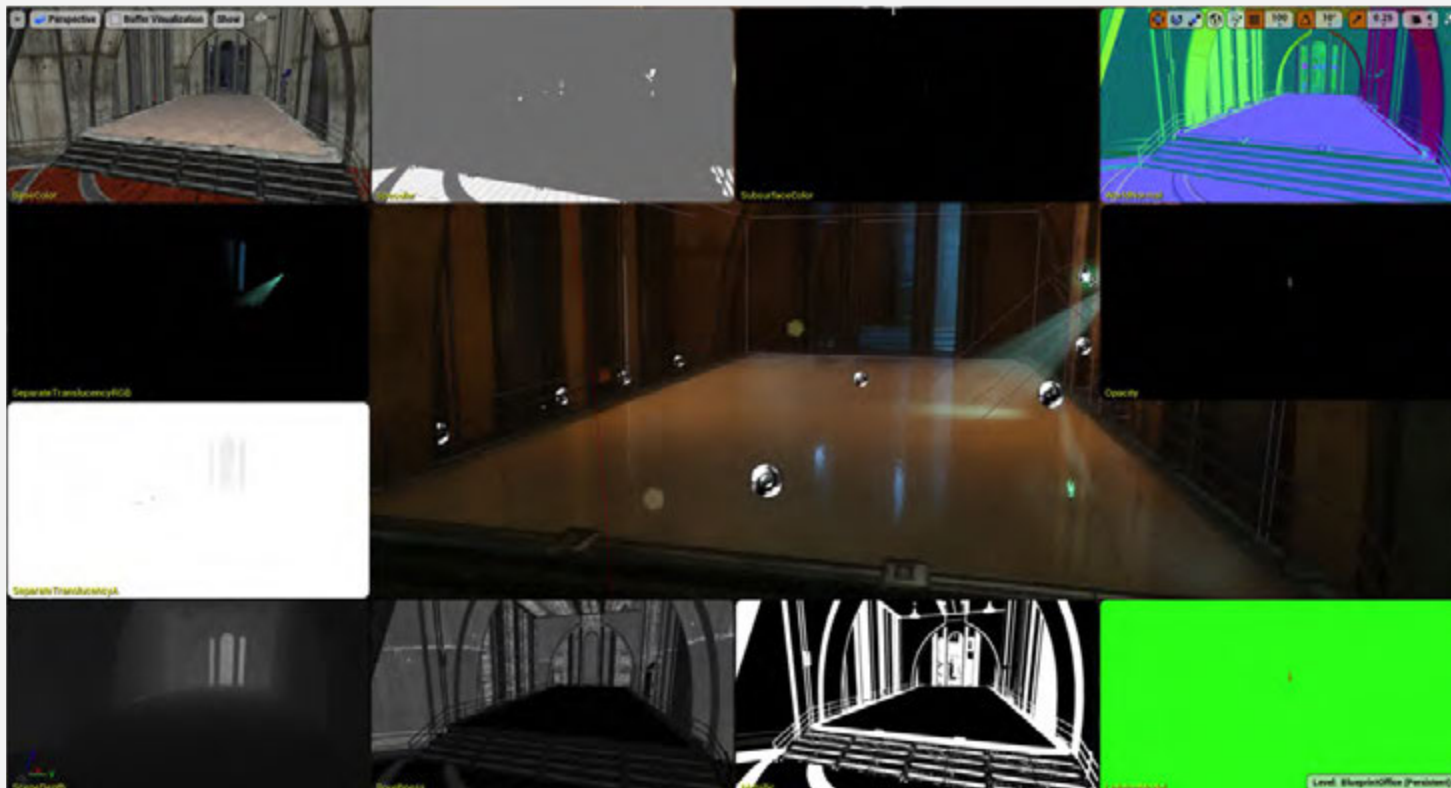
- Internal tool in-use on Paragon and Fortnite.
- Framework for **Deploying Builds** to devices, Running tests, collecting and reporting results.
- Supports PC, Mac, PS4, XboxOne, Switch, Android
- Functional testing and game-specific tests

Procedural Foliage Tool

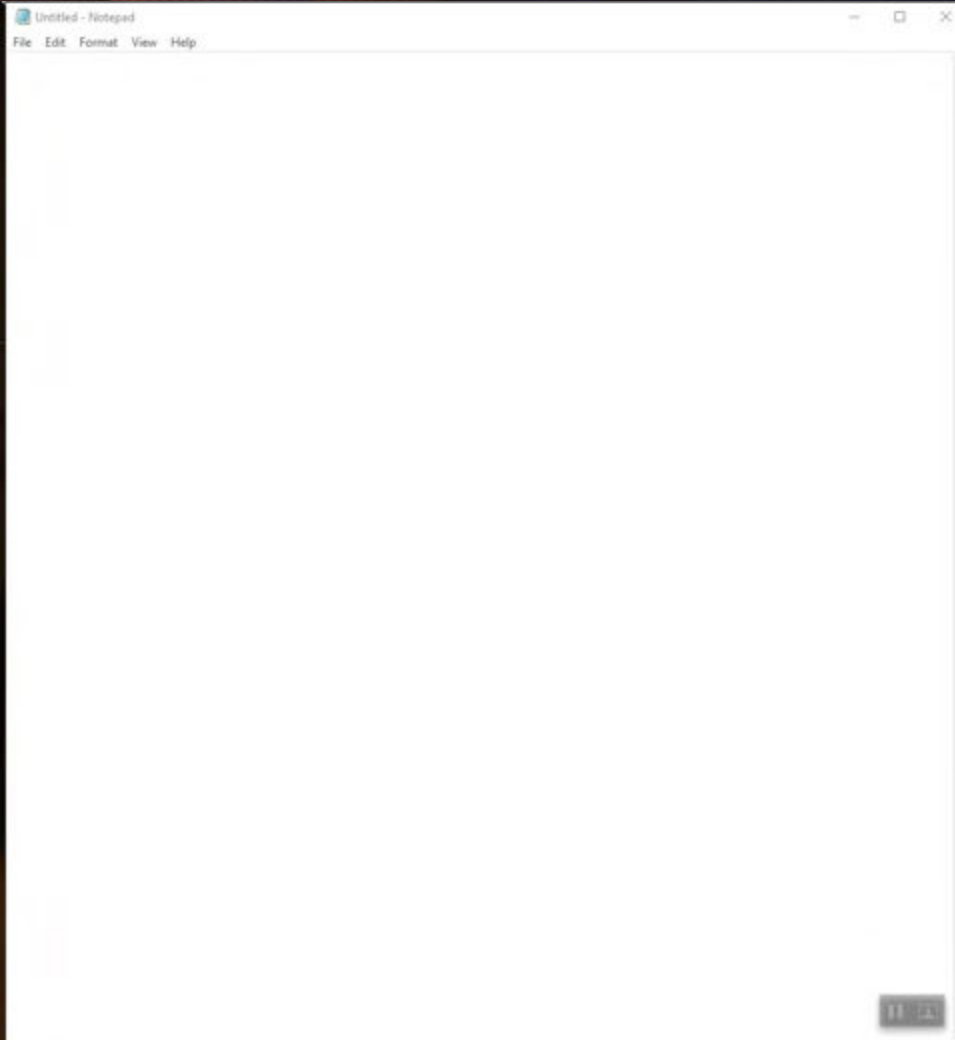
Procedurally place foliage that account for collisions, shade and number of generations of growth!



Visualisation Modes



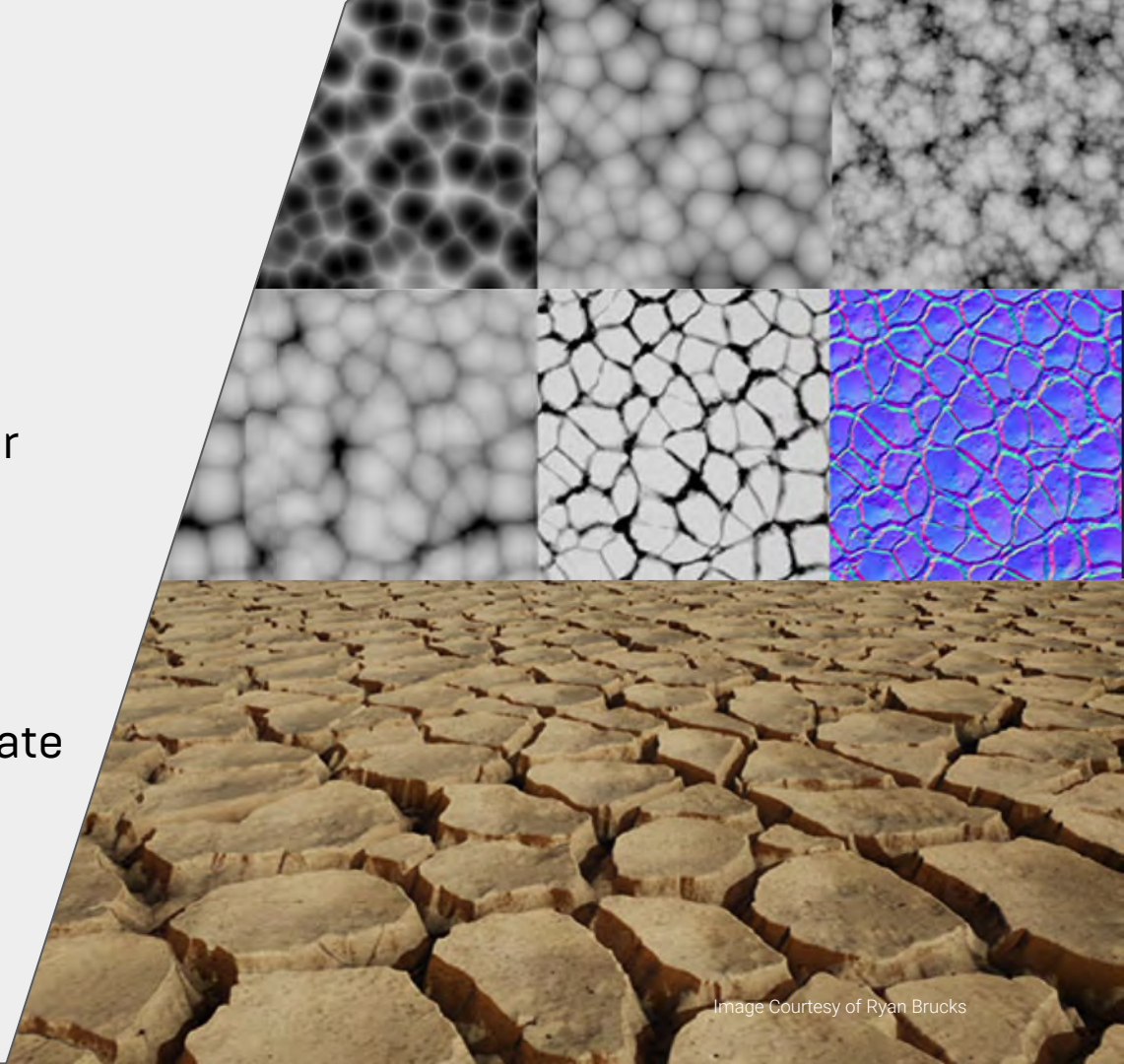
Copy-Paste to Notepad (and back!)



Render Noise Functions to Textures

Can be generated dynamically or rendered out to a texture and cached or saved.

File size too big? Why not generate on first load?



Use **Noise** to Generate **3D Textures** and Effects

Textures don't have to be 2D.
Why not take 3D approaches
to textures too?



Use **Noise** to Generate **3D Textures** and Effects

Textures don't have to be 2D.
Why not take 3D approaches
to textures too?



Free Automotive Materials

Free on the marketplace! Includes:

- Car Paint
- Carbon Fiber
- Caliper
- Rim
- Brake Rotor
- Exhaust
- Reflector
- Tire
- Metal
- Leather
- Alcantara
- Fabric
- Plastic
- Glass

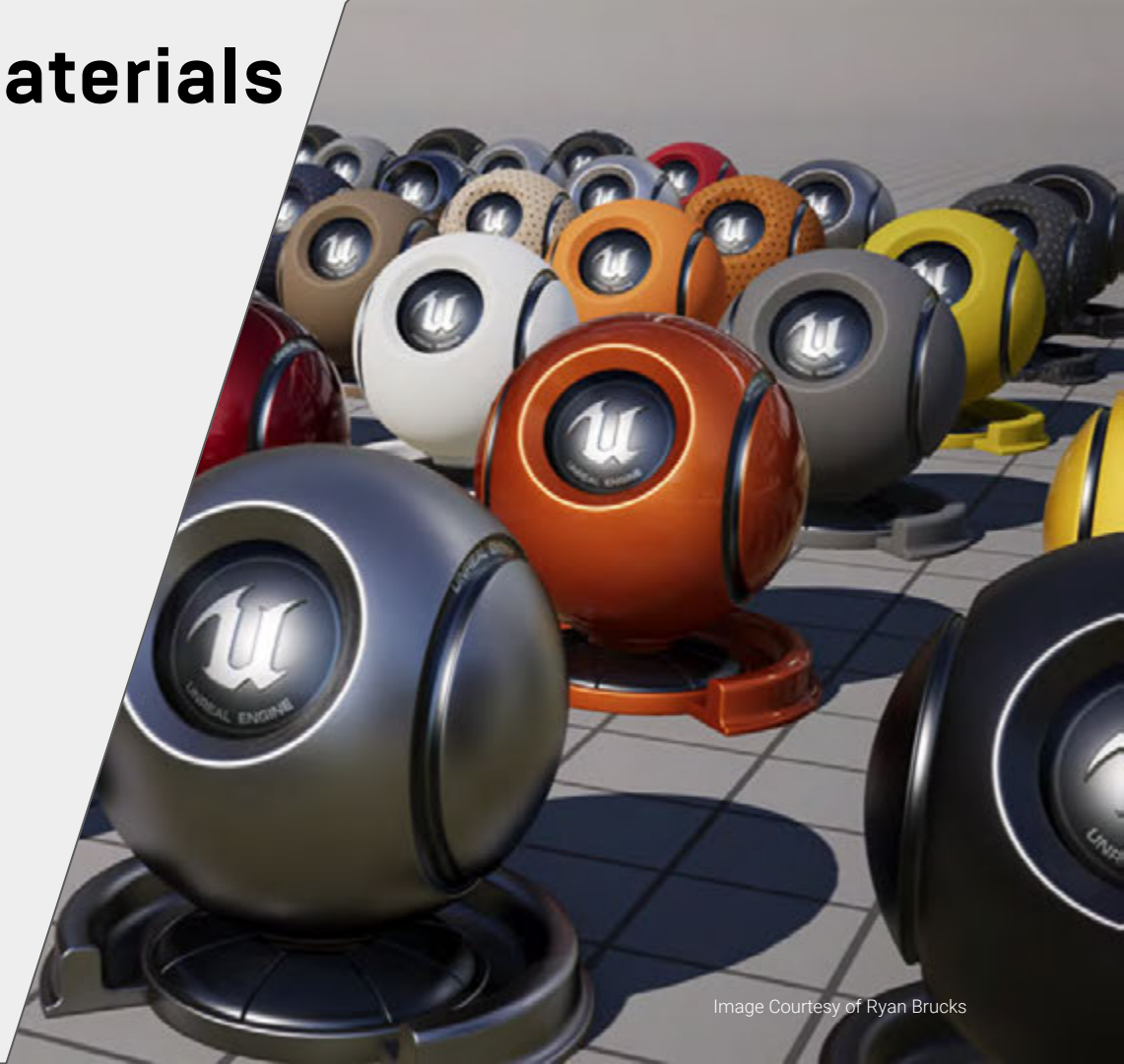


Image Courtesy of Ryan Brucks

Hold V to Enable **Vertex Snapping**



Get Creative with World Position Offset



Use Pivot Painter 2 for rad VFX



Use Pivot Painter 2 for rad VFX

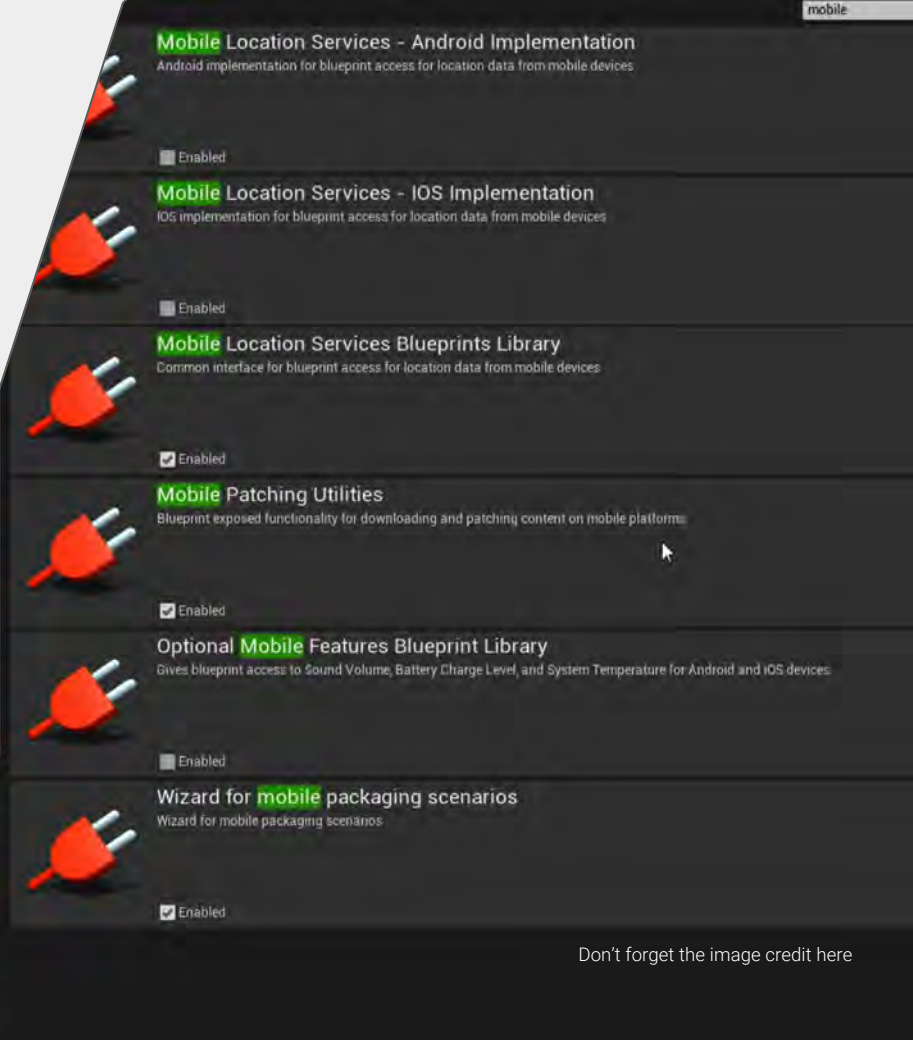


Additional mobile functionality

Additional functionality is **disabled by default** that can be enabled from within the **plugins** menu.

These include accessing:

- Sound volume
- Battery Level
- System Temperature
- Location Services

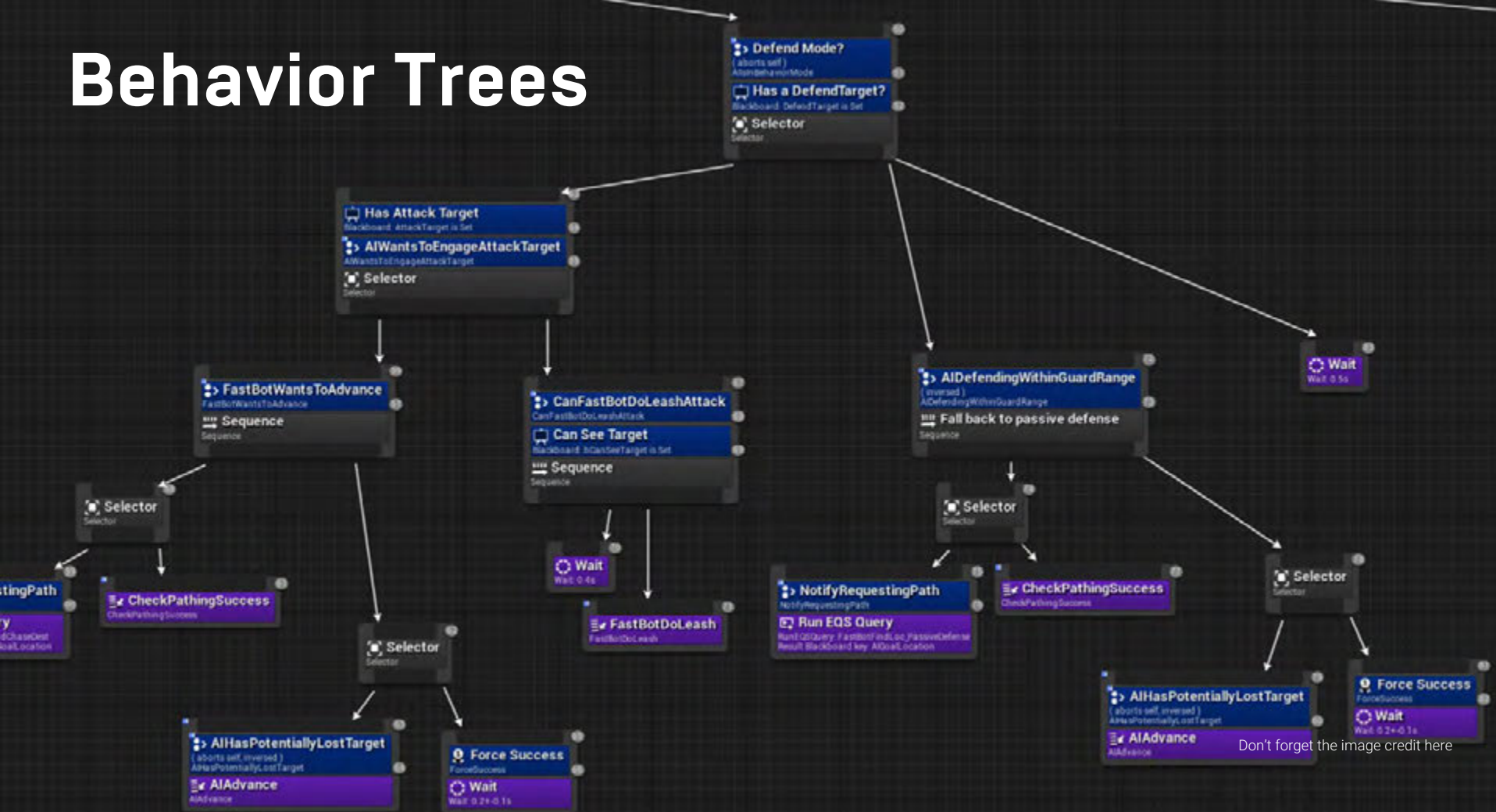


Don't forget the image credit here

R O B O
RECALL

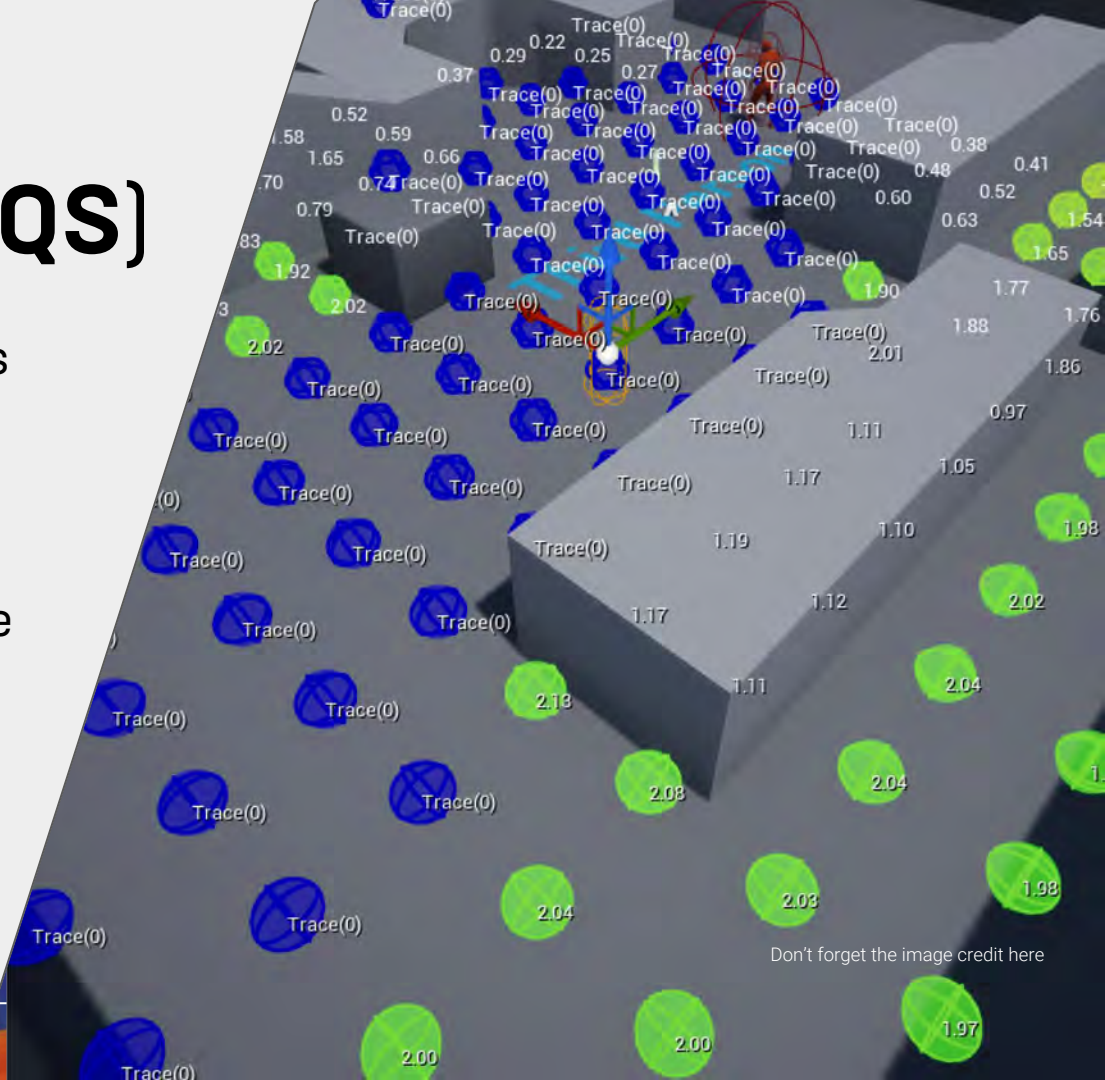


Behavior Trees



Environmental Query System (EQS)

The Environment Query System is a feature of the Artificial Intelligence system in Unreal Engine 4 for collecting data on the environment, asking questions of the data through Tests, then returning then one Item that best fits the questions asked.



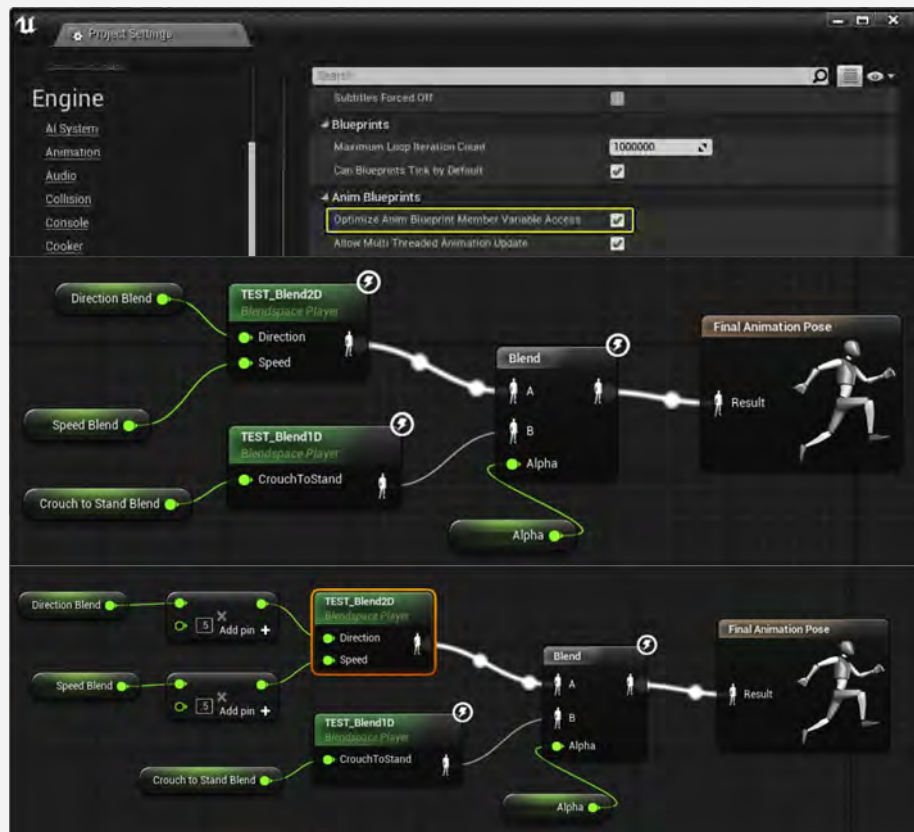
Splitscreen Support



Animation **Fast Path**

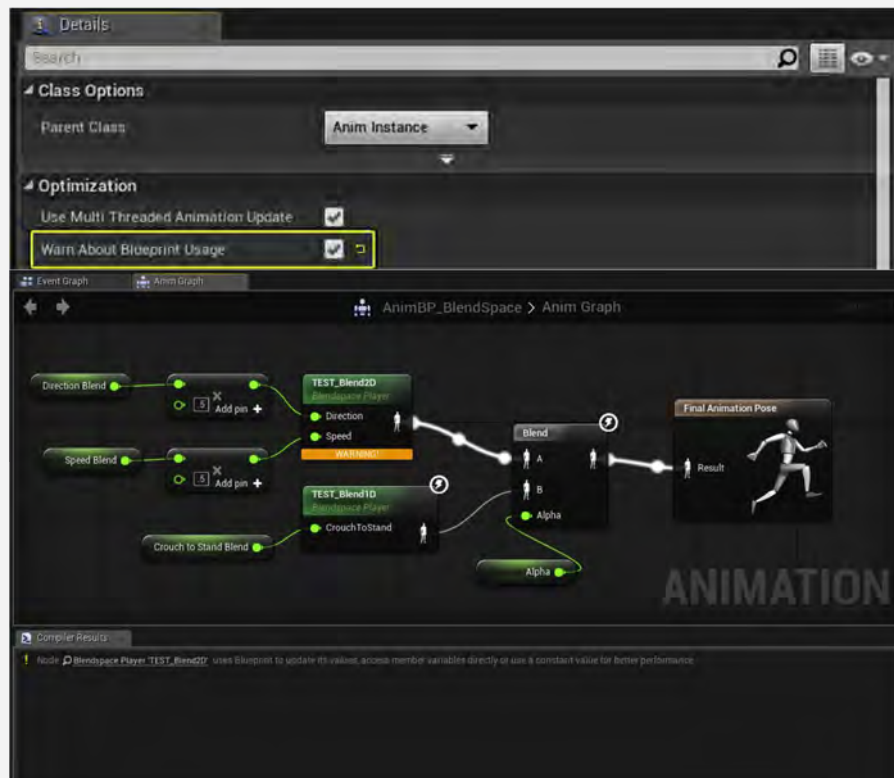
Animation Fast Path provides a way to optimize variable access inside the AnimGraph update.

This enables the engine to copy parameters internally rather than executing Blueprint code (which involves making calls into the Blueprint Virtual Machine).



Animation **Blueprint Warnings**

To ensure that your Animation Blueprints are using **Fast Path**, you can enable the **Warn About Blueprint Usage** option which will cause the compiler to emit warnings to the Compiler Results log whenever a call into the Blueprint Virtual Machine is made from the AnimGraph.

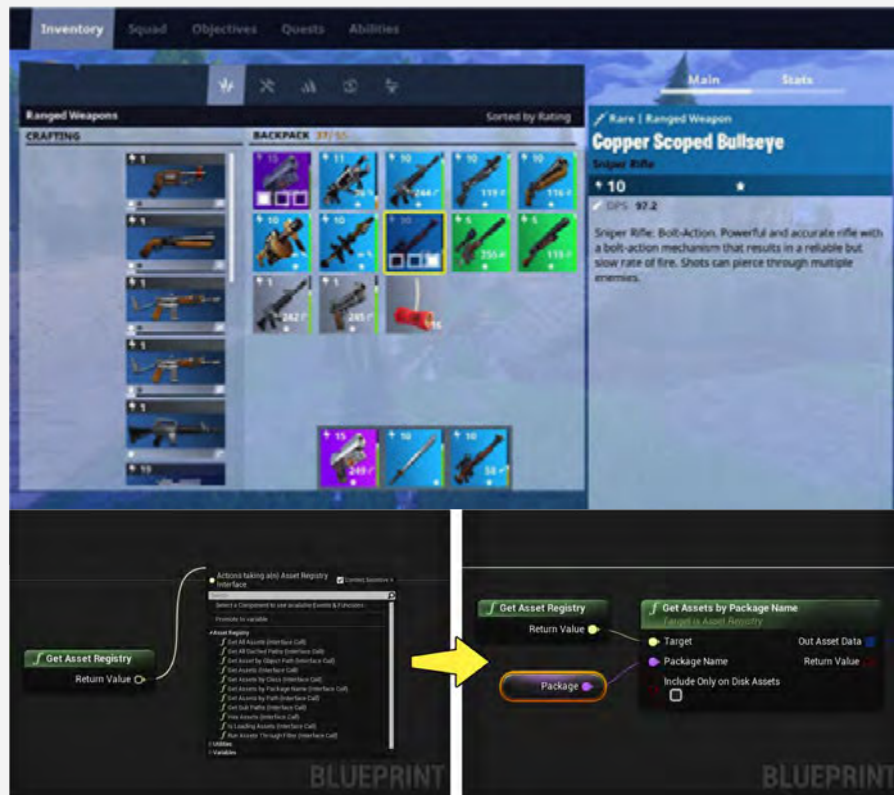


Asset Management Framework

First added in 4.16, the **Asset Manager** is a global object that can be used to discover, load and audit content in the editor or at runtime.

Allows you to asynchronously load assets to prevent hitches for easier management of quests, weapons, characters, etc.

Helps remove hard references between Blueprints.



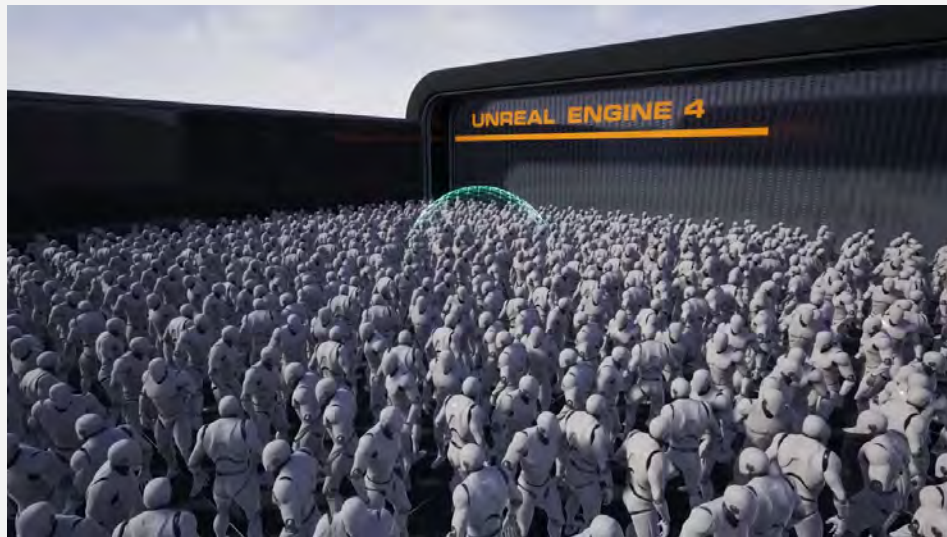
Consider the **Forward** Renderer

- Most people think of the Forward Renderer as the VR renderer.
- Faster for certain kinds of projects: especially when you have limited use of Dynamic Lights.
- Think about it in the context of your project!



Lightweight Rigid Body Simulation

- Create large numbers of physically-simulated characters with the **lightweight rigid body character simulation**
- You can now simulate a Physics Asset inside your Animation Blueprint using a new high-performance **immediate mode** PhysX API.
- Characters using this simulation can also generate collision with static geometry in the world.



THAT'S 50!



World Composition Tool

- System for developing extremely large environments.
- Automatically splits environment up in to tiles.
- Will not work with level streaming volumes but will work with Blueprint streaming.
- Origin Shifting will not currently work on a listen server and requires a dedicated server.



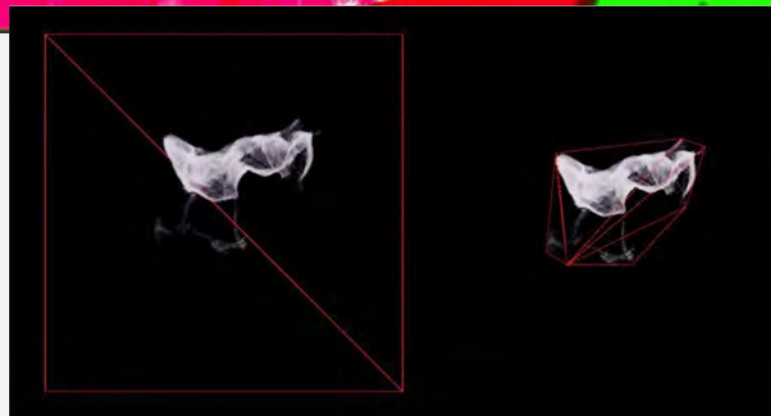
Be aware of **Overdraw**

Minimize the geometry area for overdraw

Adding vertices is almost always cheaper than relying on overdraw

Make use of **Particle Cutout** property!

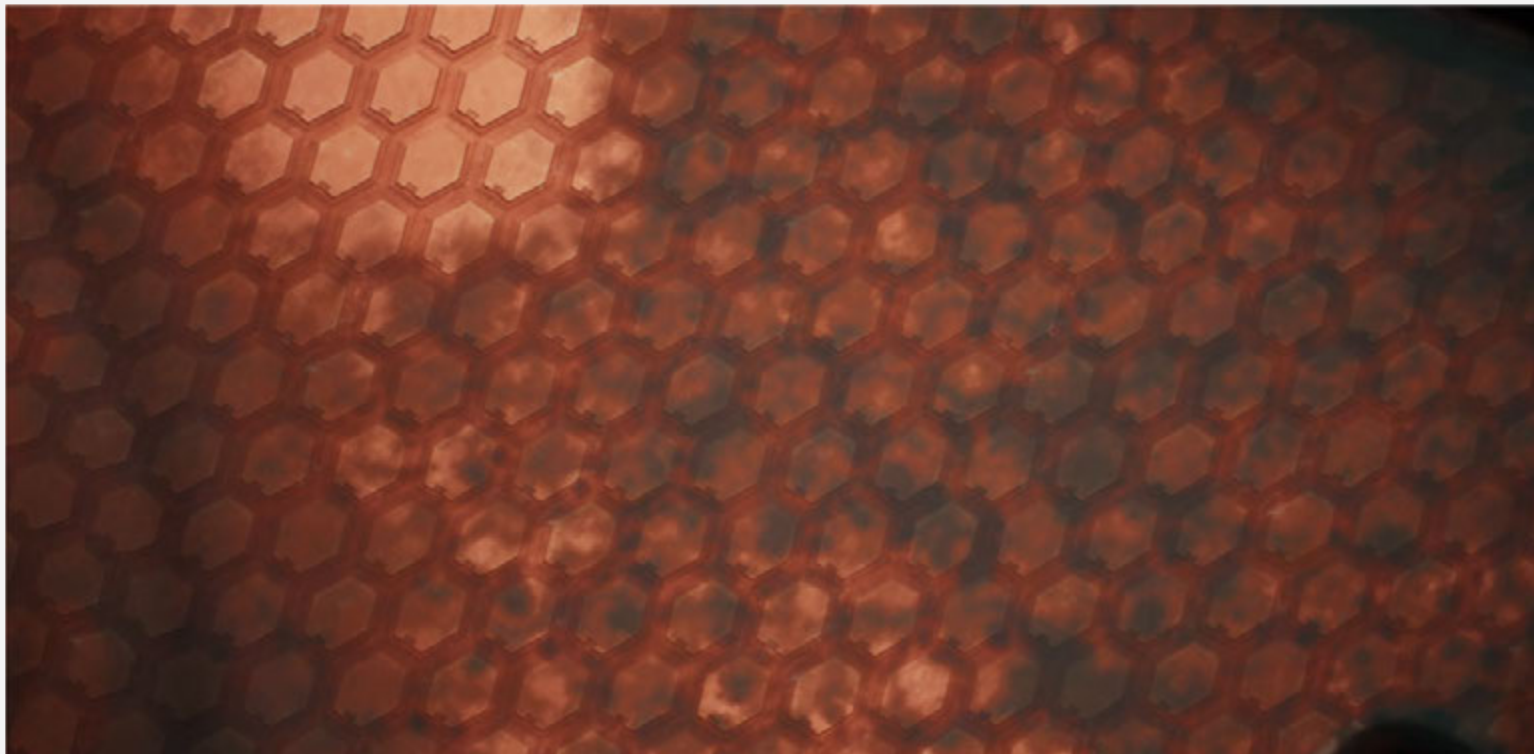
- This is found under the Cascade Required Module.
- Also works on subUVs, with a different cutout for every frame!



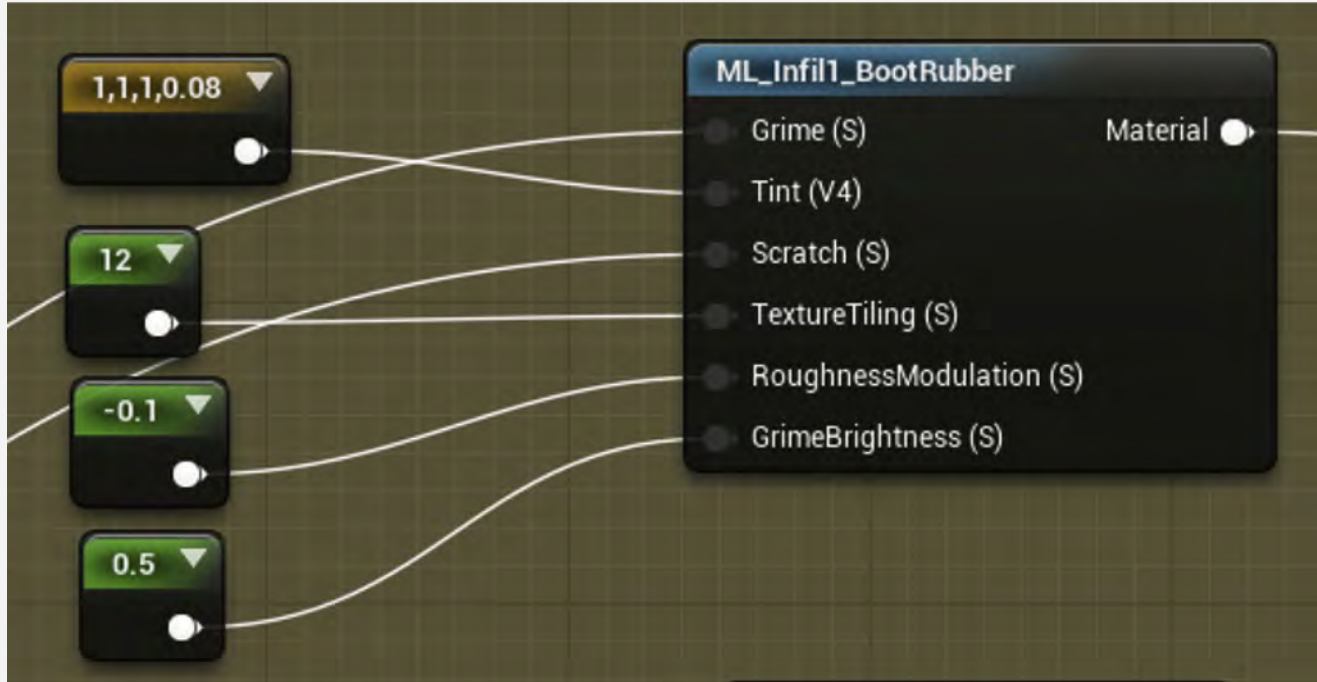
Material Functions as **Layers**



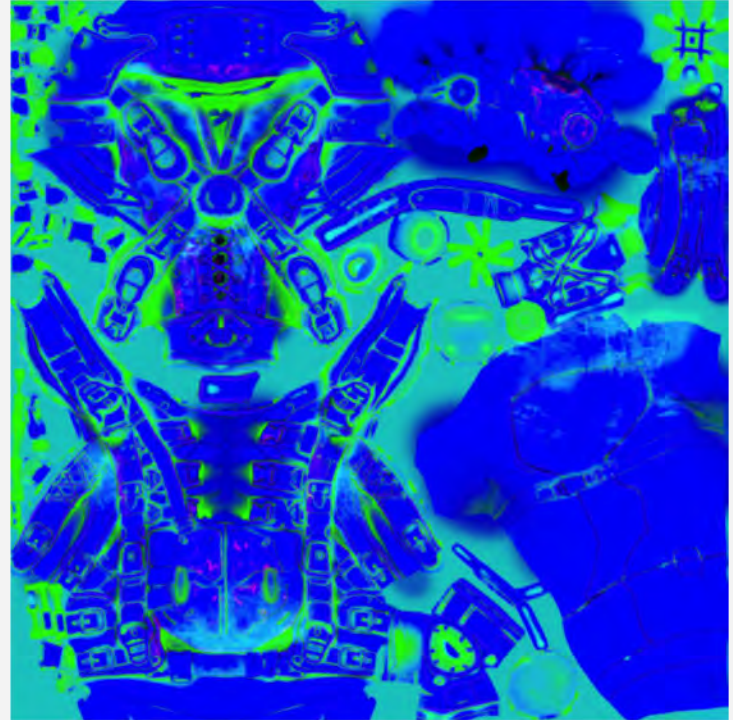
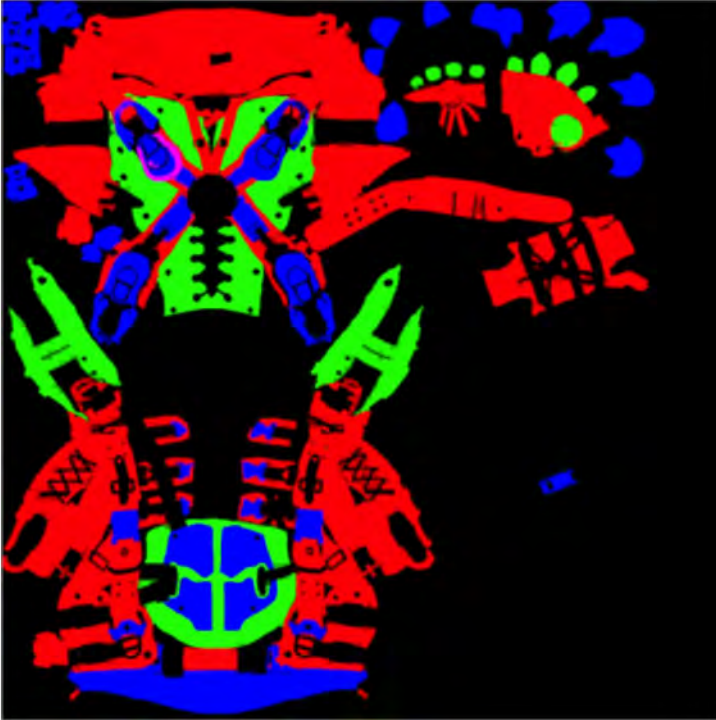
Material Functions as **Layers**



Material Functions as **Layers**



Material Functions as **Layers**

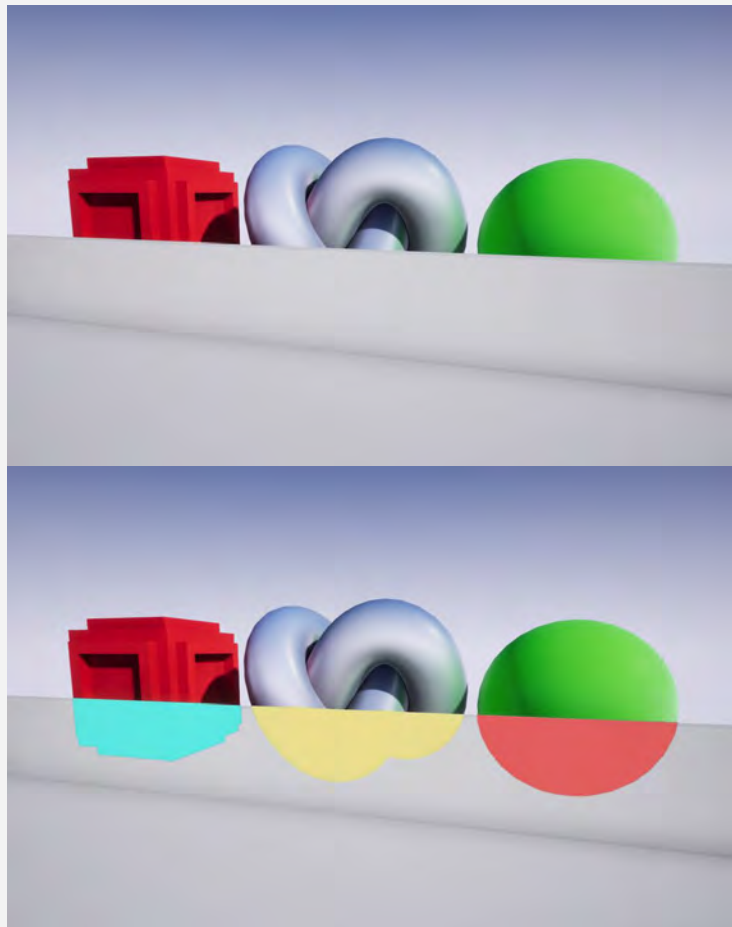


Material Functions as **Layers**



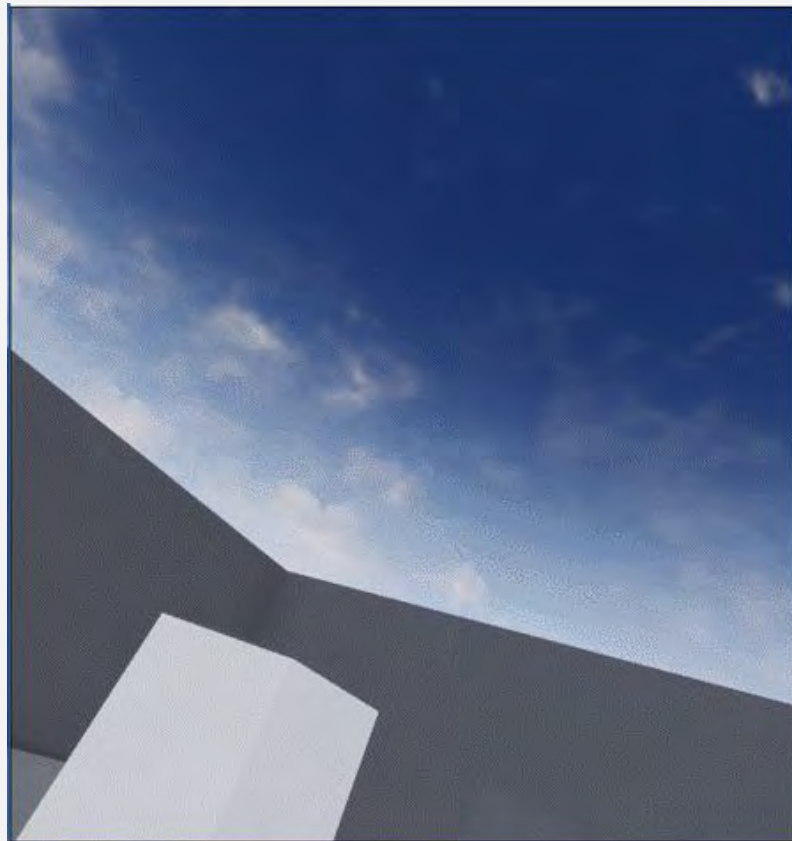
Stencil Buffer

The Stencil Buffer allows assets to be rendered to a separate buffer. In this image the buffer is being used to render the actor through the portal as it emerges.



Stencil Buffer

The Stencil Buffer allows assets to be rendered to a separate buffer. In this image the buffer is being used to render the actor through the portal as it emerges.



Pixel Depth Offset

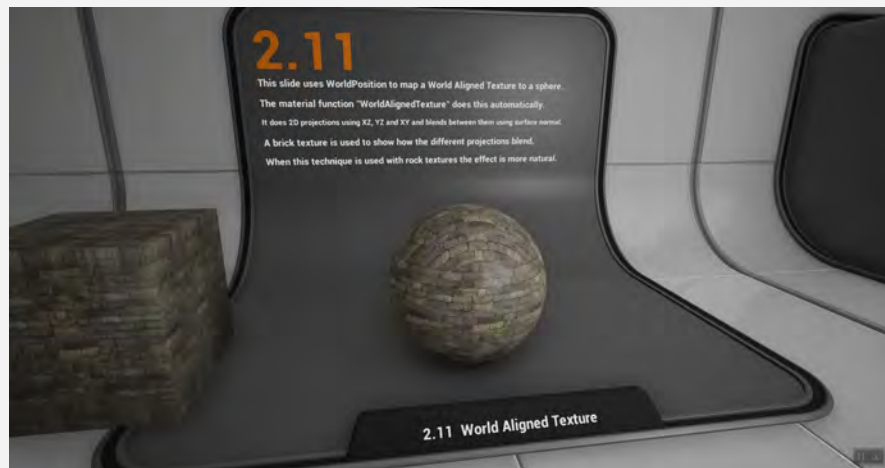
Pixel Depth Offsets allow for manipulating the occlusion tests within a renderer. In this instance random pixels close to the ground are instead placed under the surface.

Adds some overhead!



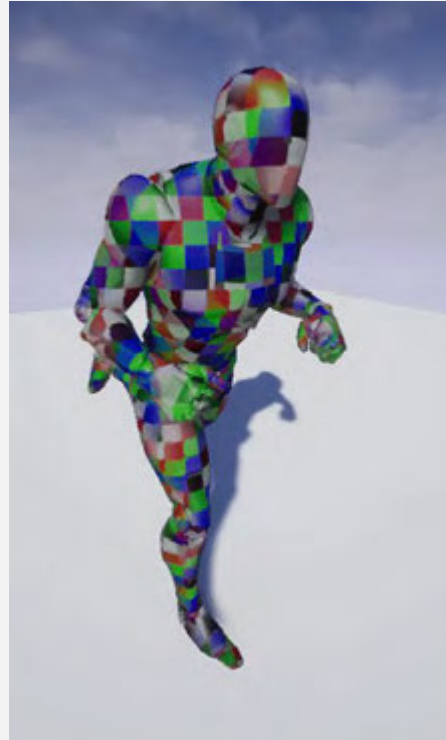
World Aligned Materials

- Also called tri-planar mapping
- World projected textures ignore base UVs
- Higher performance cost but simpler scene setup
- Allows separate meshes that are touching to appear to share the same surface (such as continuous brick)
- World Position can be transformed into local space or scaled independently



Pre-Skinned Local Position/Normal

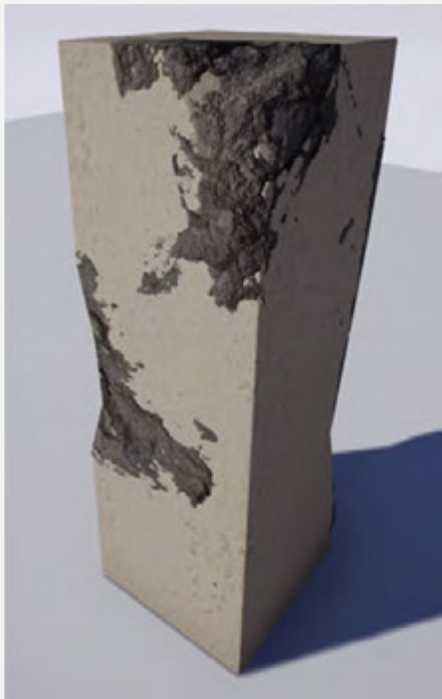
- Caches position from T-Pose
- Can be used for easy projections onto meshes for decals/camouflage without warping.



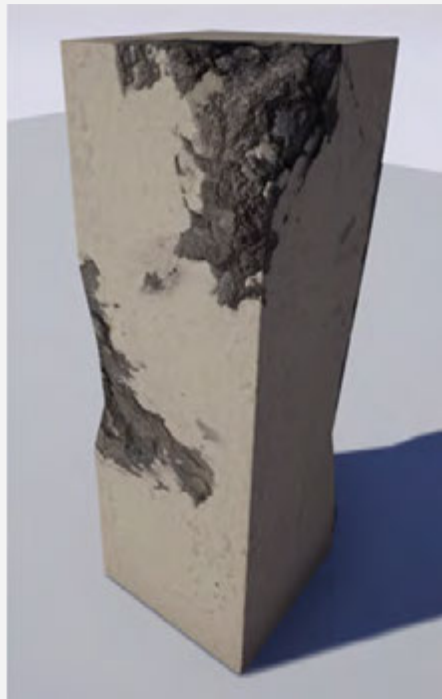
Sun Position Calculator Plugin



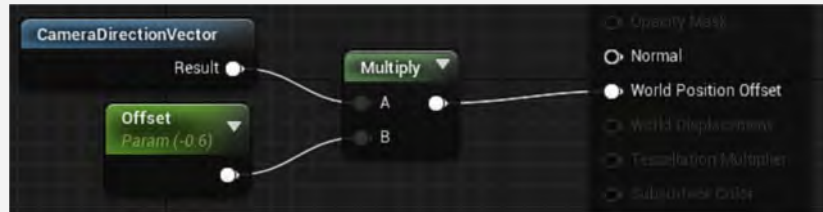
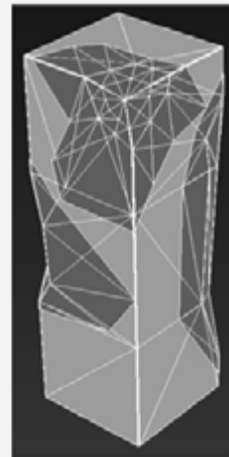
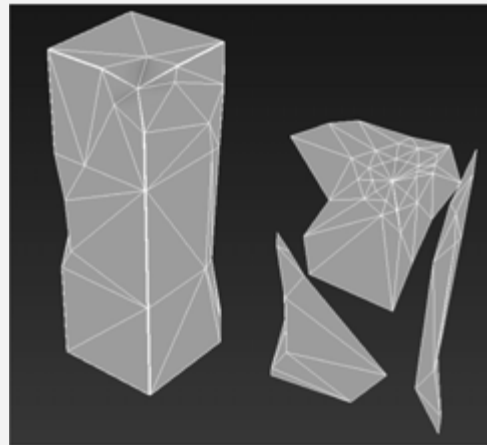
Mesh Decals are ridiculously overlooked



Masked



Mesh Decal



Auto LOD Generation with **LOD Groups**



LOD Generation with **Skeletal Meshes**



Low Latency Frame Syncing

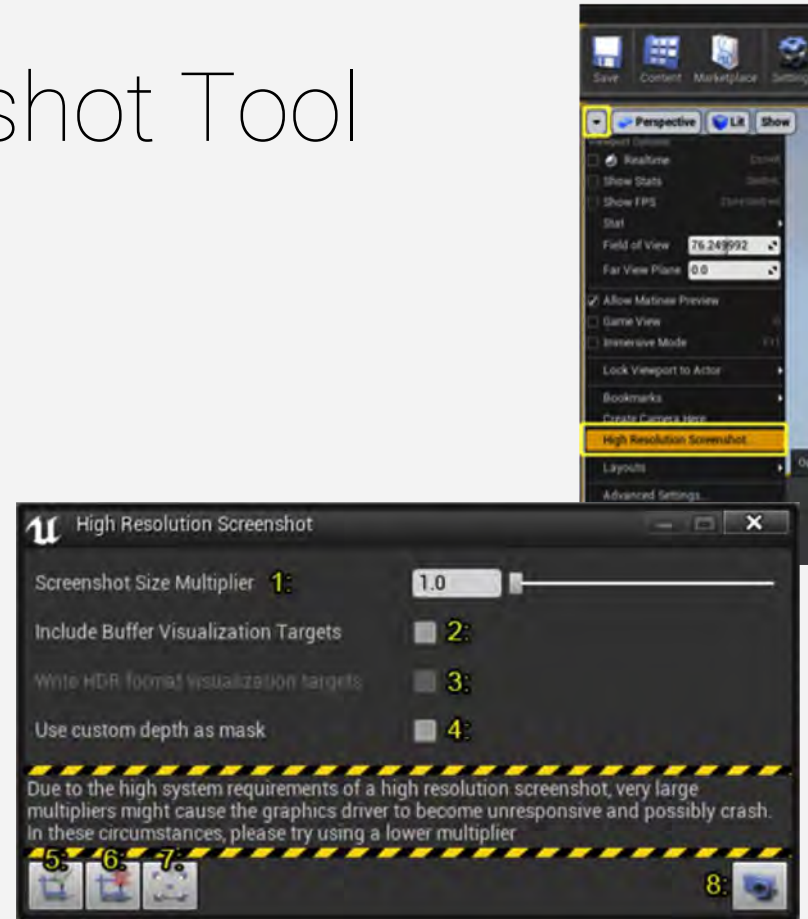
Low-latency frame syncing mode modifies the way thread syncing is performed between the game, rendering and RHI threads and the GPU to greatly reduce and control input latency. In previous engine releases, the game thread synced with the rendering thread at the end of the frame. When the **r.OneFrameThreadLag CVar** is **enabled** (as is the default), this syncing ensures that the game thread does not get more than one whole frame ahead of the rendering thread.

r.GTSyncType Value	Description
0	Game thread syncs with rendering thread (old behaviour, and default).
1	Game thread syncs to the RHI thread (equivalent to UE4 before parallel rendering)
2	Game thread syncs with the swap chain present +/- an offset in milliseconds.

High Resolution Screenshot Tool

High Resolution Screenshot Tool allows for developers to take images with a wide range of options:

- Multiplies the screenshot resolution size against this number.
- This will include each pass in the GBuffer as an image that gets exported.
- This will write all exported images into .EXR format.
- Uses the Custom Depth as a mask for objects in the scene. Must be enabled in the project settings first.
- This allows you to specify a region within the viewport that you wish to capture.
- Clicking on this will clear any previous capture regions that you might have created.
- Clicking on this will make the capture region the entire editor window.
- Clicking on this this will take the screenshot.



Depth Projected Textures

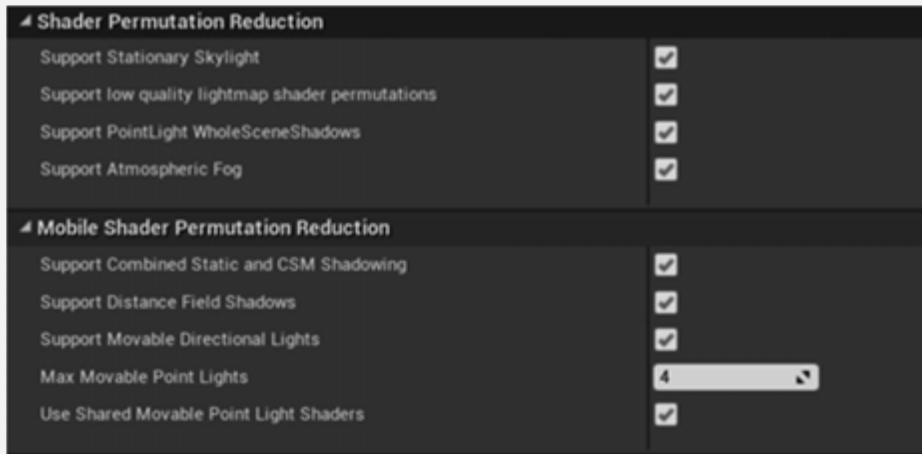
A Render to Texture Blueprint is available in:
Engine\ArtTools\RenderToTexture\ Blueprints.

- Gives Illusion of Shadows in Unlit Materials
- Useful in avoiding lightmap rebaking
- Allows consistent shadows during cutaways
- More performant on AR demos while still keeping realistic shadows
- The main computations are done in the vertex shader, and a good looking unlit material is cheap enough to run on mobile as evidenced by use on Zen demo.
- Higher content creation cost, though it can be triggered/Updated by a Blutility



Consider disabling **Shader Permutations**

If shader compilation is taking time consider disabling variations that are auto generated by default that won't be used in your project.



Pixels Aren't Always The Problem

Games process both Pixels and Vertices and sometimes it's necessary to identify whether a bottleneck is from pixels or vertices in the scene. By using the **r.SetRes** and **r.ScreenPercentage** commands the number of pixels being rendered can be scaled up or down.

If changing the resolution does not decrease the GPU time by much then it suggests that the issue is vertex related. Typical causes include:

- Too many vertices. (Use Level of Detail meshes)
- Too many objects in a scene (high number of draw calls)
- Complex World Position Offset / Displacement Material using Textures with poor MIP mapping. (adjust the Material)
- Tessellation (Avoid if possible or lower the tessellation factor if necessary)
- Many UV or Normal seams that result in more vertices.
- Too many vertex attributes. (extra UV channels)
- Verify the vertex count is reasonable, some importer code might not have welded the vertices. (combine vertices that have same position, UV and normal)

Pixels Aren't Always The Problem

Less often, you are bound by something else. That could be:

- Object cost (more likely a CPU e.g. cost but there might be some GPU cost)
- Triangle setup cost (very high poly meshes with a cheap vertex shader e.g. shadow map static meshes, rarely the issue)
- Use level of detail (LOD) meshes
- View cost (e.g. HZB occlusion culling)
- Scene cost (e.g. GPU particle simulation)

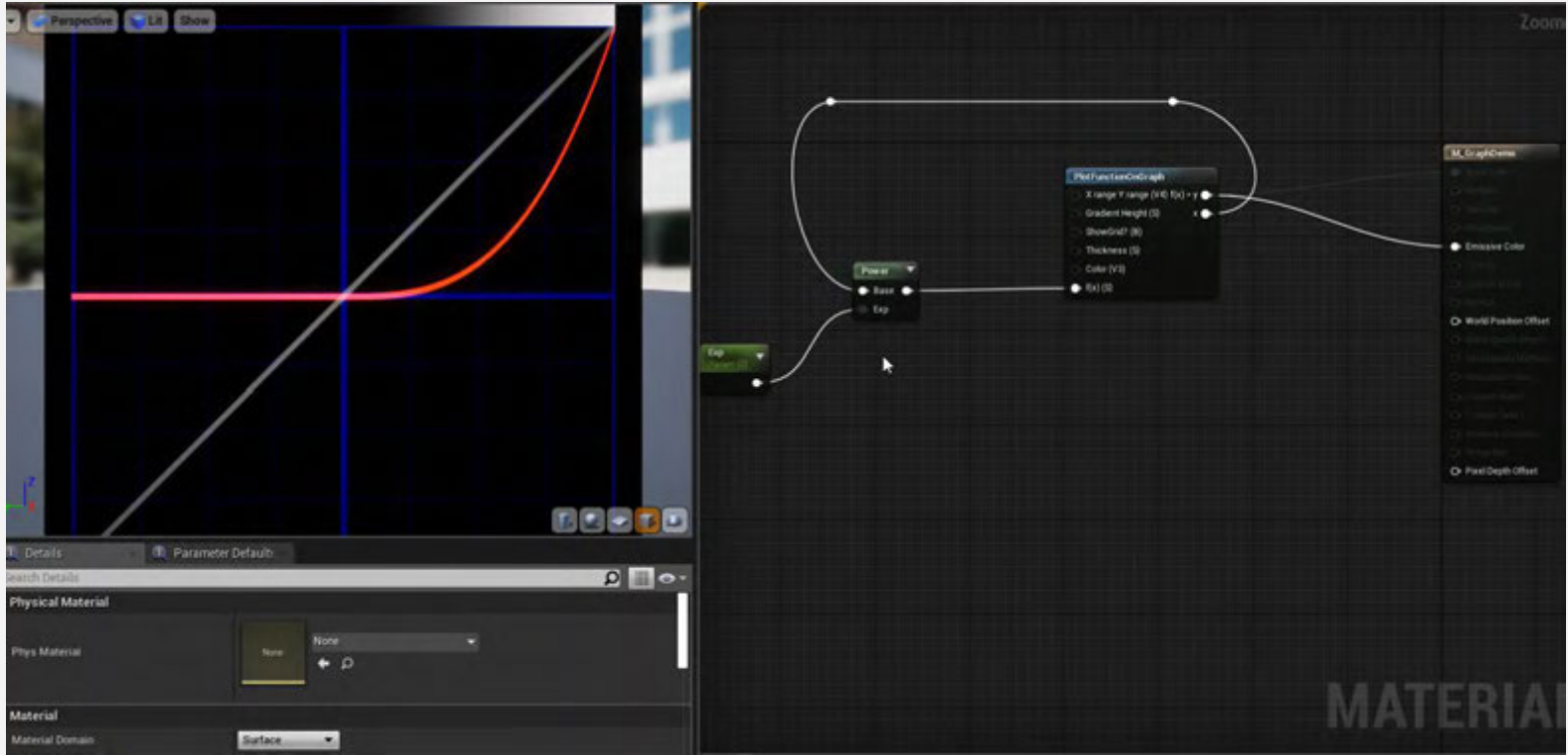
Avoid Hard References in Blueprint

When a blueprint references another blueprint it must load it into memory. If the blueprint that has been loaded in also references blueprints then these must also be loaded. This can get out of hand.

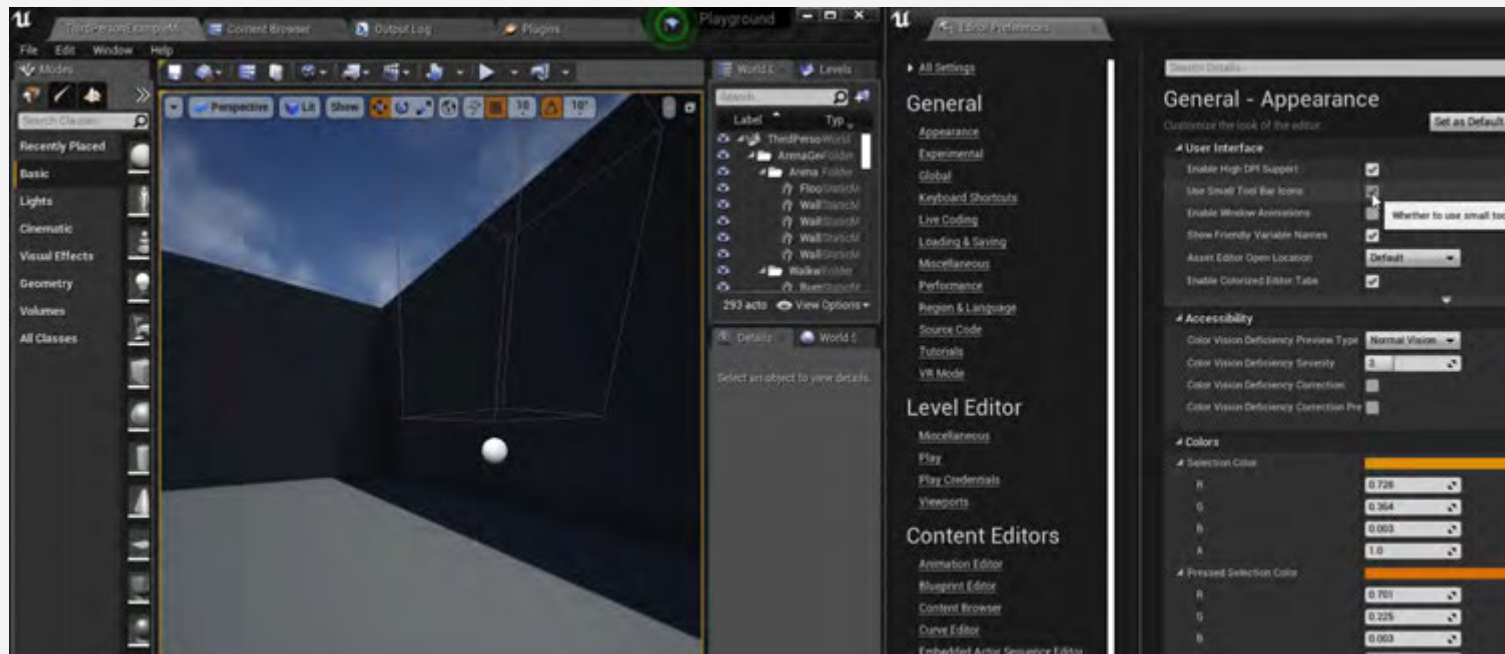
While this *won't* slow down in-game performance it can eat away at memory and load times.

Interfaces allow for confirmation of functionality without casts and should often be used instead.

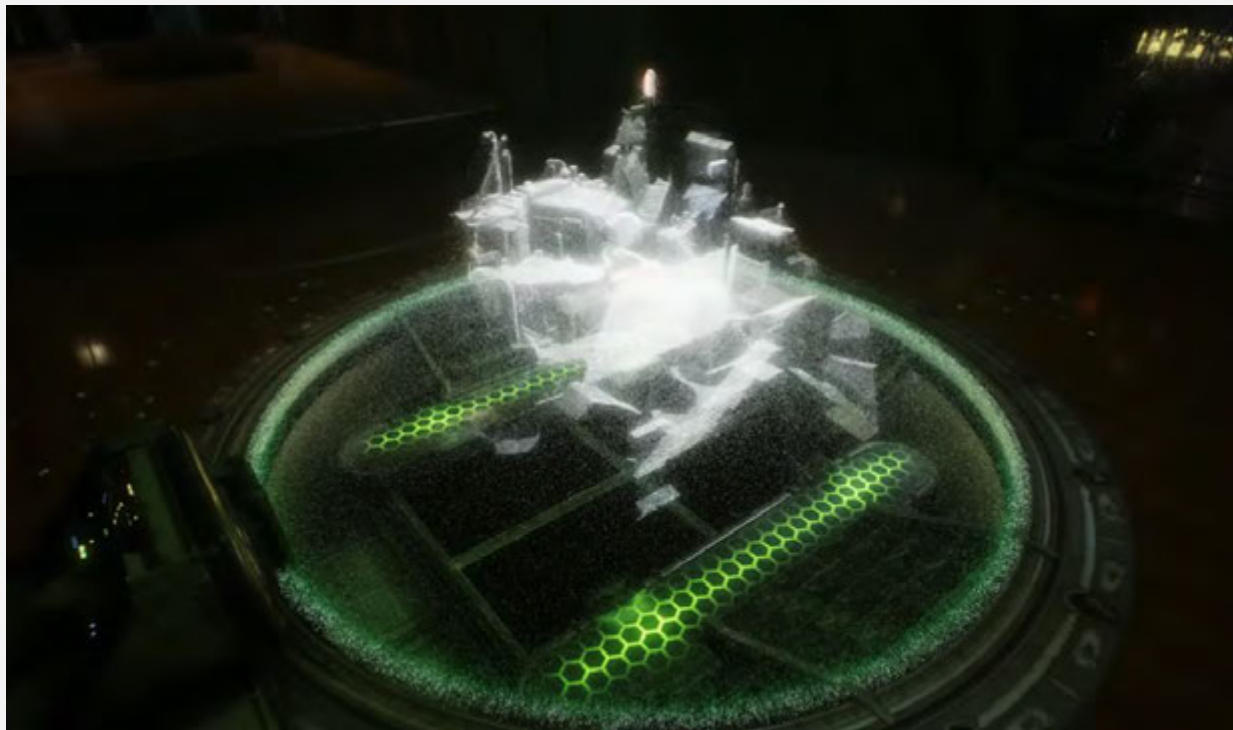
PlotFunctionOnGraph Material Function



Small Toolbars



Disabling Interpolated Spawning in Niagara



VR Editor

VR Mode enables you to design and build worlds in a virtual reality environment using the full capabilities of the Unreal Editor toolset combined with interaction models designed specifically for Virtual Reality world building.



Simple AR Setup

- Actor configurations can be organized into levels instead of over-sized Blueprints
- Levels can be loaded, and instanced, with a given transform
- Instead of Spawning an Actor at a location a level can be created at a location
- VR World to Scale can be used to create real-world scale assets that can be then scaled and placed on a table.



Simple AR Setup

Advantages

- Cleaner for separate content
- Allows for Sequencer and multiple Blueprints and Level Blueprint setups to be streamed in
- Can be easily rescaled and have content moved between viewing types: VR can be run at regular scale while AR can be rescaled accordingly

Disadvantages

- Your line traces are still in game world space and will fall short of where you expect them to end. Consider multiplying them against world scale.





UE4 Tips and Tricks

Chris Murphy

@HighlySpammable

Chris.Murphy@epicgames.com

Unreal Engine Evangelist