

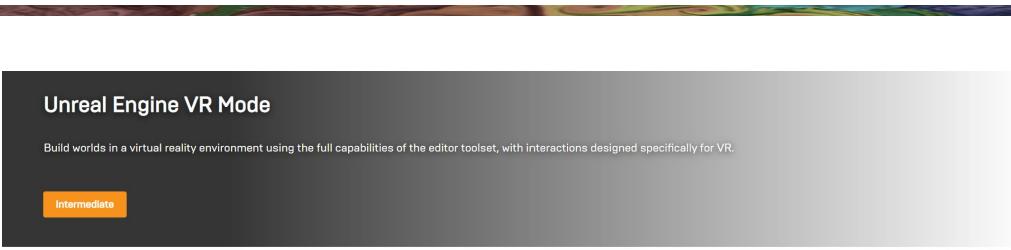
# Virtual Reality Unreal Best Practices

## CS 415: Game Development

Professor Eric Shaffer

Some slides courtesy of Luis Cataldi

# Technical Resources for VR Development in Unreal



**Unreal Engine VR Mode**

Build worlds in a virtual reality environment using the full capabilities of the editor toolset, with interactions designed specifically for VR.

Intermediate

Use VR Mode 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 (VR) world building.

Working directly in VR provides the proper sense of scale necessary to create realistic, believable worlds, while the use of motion controllers means you can build environments with natural motions and interactions.



Unreal Editor can run in VR...

Can speed up design of VR worlds

- <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/VRMode/>

# Technical Resources for VR Development in UE4

Unreal Engine 4.27 Documentation ▾ > Sharing and Releasing Projects > XR Development > Virtual Reality Development

Search Documentation...

Filter pages...

+ Setting Up Your Production Pipeline  
+ Testing and Optimizing Your Conte...  
- Sharing and Releasing Projects  
AutoSDK Reference  
Building Unreal Engine as a Lib...  
+ Containers  
+ HTML5 Game Development  
+ Linux Game Development  
+ Mobile Game Development  
+ Patching and DLC  
+ Pixel Streaming  
- XR Development  
+ OpenXR  
- Virtual Reality Development  
+ VR Platforms  
+ VR How-To's  
Virtual Reality Best Pract...  
+ VR Performance and Pro...  
VR Cheat Sheet  
+ Augmented Reality Develop...  
Setting Device Profiles  
Low Latency Frame Syncing

## Virtual Reality Development

Virtual reality (VR) refers to an interactive experience where the user's real-world environment is replaced by a virtual environment through a wearable device.

Virtual reality (VR) refers to an interactive experience where the user's real-world environment is replaced by a virtual environment through a wearable device. The Unreal Engine VR framework provides a rich, unified framework for building virtual reality apps using the Unreal Engine.

### Platforms

**Developing for Oculus VR**  
Information on using UE4 to develop for Oculus VR.

**Developing for SteamVR**  
Information on using UE4 to develop for SteamVR.

**Windows Mixed Reality**  
Information on using UE4 to develop for Windows Mixed Reality platforms.

**Developing for Samsung Gear VR**  
Information on using UE4 to develop for Samsung Gear VR.

**Developing for Google VR**  
Information on using UE4 to develop for Google VR.

#### ON THIS PAGE

Platforms

Guides

References

<https://docs.unrealengine.com/4.27/en-US/SharingAndReleasing/XRDevelopment/VR/>

**I** ILLINOIS

# Technical Resources for VR Development in UE5

The screenshot shows the Unreal Engine 5.0 Documentation website. The top navigation bar includes links for "Unreal Engine 5.0 Documentation", "Sharing and Releasing Projects", and "XR Development". A search bar and a three-dot menu are also present. The left sidebar contains a tree view of documentation categories, with "XR Development" currently selected. The main content area features the Unreal Engine logo and a banner image of a city skyline. The "XR Development" section header is visible, along with a brief description of developing for augmented reality, virtual reality, and mixed reality devices. Below this, there's a list of experiences: Augmented reality (sensory information overlays real world), Virtual reality (virtual environment replaces user's view), and Mixed reality (blend between AR and VR). A note states that Unreal Engine supports both developing for XR platforms and using XR devices in your content creation pipeline. The "ON THIS PAGE" sidebar on the right lists various topics related to XR development, such as "Developing for Head-Mounted XR", "Experiences with OpenXR", and "Creating UI for XR Experiences".

<https://docs.unrealengine.com/5.0/en-US/developing-for-xr-experiences-in-unreal-engine/>

# VR Development Best Practices

Substantially different platform the mobile or console/PC

- Different performance demands
- Different user interface principles needed

The screenshot shows the 'Virtual Reality Best Practices' page from the Unreal Engine 4.27 Documentation. The page title is 'Virtual Reality Best Practices' and it describes best practices for creating and optimizing content for VR projects. It includes sections for 'VR Project Settings' and 'Project Settings'. A note states that VR projects can be Blueprint or C++. A sidebar on the left lists various documentation categories, and a sidebar on the right lists 'ON THIS PAGE' topics such as VR Project Settings, VR.ini Settings, VR Frame Rate Optimization, VR World Scale, VR and Simulation Sickness, VR Camera Setup, VR Character Settings, Character Height and Width, Movement Speed, Camera Location, VR Content Considerations, Known Limitations, and Normal Mapping Issues.

<https://docs.unrealengine.com/4.27/en-US/SharingAndReleasing/XRDevelopment/VR/VRBestPractices/>



# Scale Your World Carefully

Getting the scale of your world correct is important things for a good user experience possible on VR platforms.

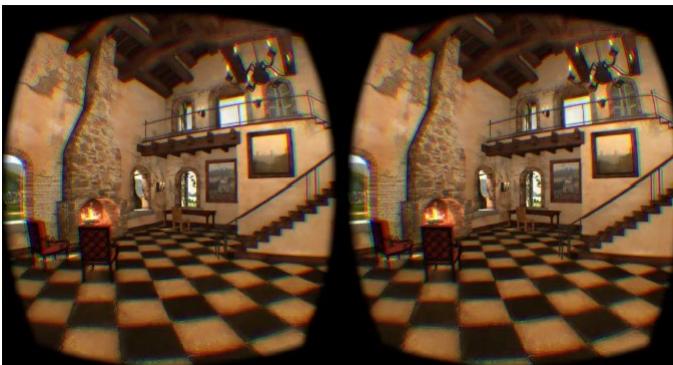
Having the wrong scale can lead to all kind of sensory issues for users and could even result in Simulation Sickness.

- Make sure camera is at the correct height for an average person
- Make sure distance between pupils matches real-world distance

Objects are most easily viewed in VR when they are in a range of 0.75 to 3.5 Meters from the player's camera.

Inside of UE4, 1 Unreal Unit (UU) is equal to 1 Centimeter (CM).

Objects inside Unreal are best viewed when they are 75 UU to 350 UU away when using VR.



In the Tuscany demo from Oculus VR, there are not enough familiar objects to precisely resolve depth and size. Have you ever been to a villa like this? Are the floor tiles a familiar size? Is the desk too low?

# VR Character Settings

The setup for a character using a VR headset is slightly different than for a standard character. Height, Width, Speed, and Camera Location all need to be slightly modified to accommodate a VR character.

Property	UE4 Default	Recommended VR
Height:	192 CM	176 CM
Width:	84 CM	68 CM

Property	UE4 Default	Recommended VR
Base Eye Height:	180 CM	160 CM

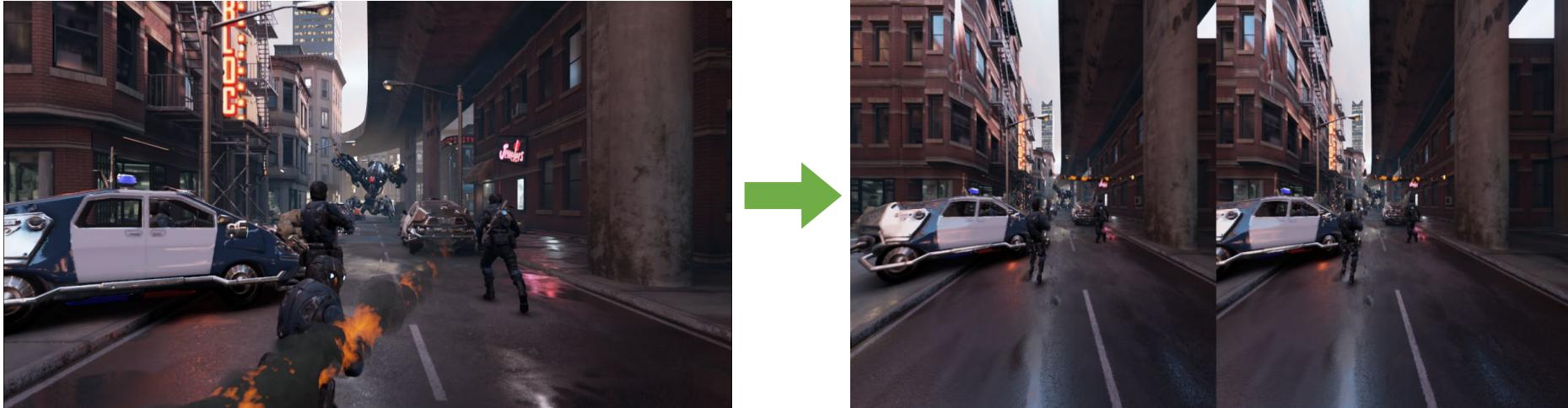
VR movement speed is a difficult property to recommend a setting for because the movement speed that you choose will mainly be determined by the type of experience that you are trying to achieve. In the Elemental VR demo for example, the movement speed was cut to about 1/4 normal speed.

Property	UE4 Default	Recommended VR
Movement Speed:	60 M/S	24 M/S

# Frame Rate is Hugely Important

Need low latency, high framerate to respond to head movement...so people don't get sick

The frame is rendered twice, once per eye



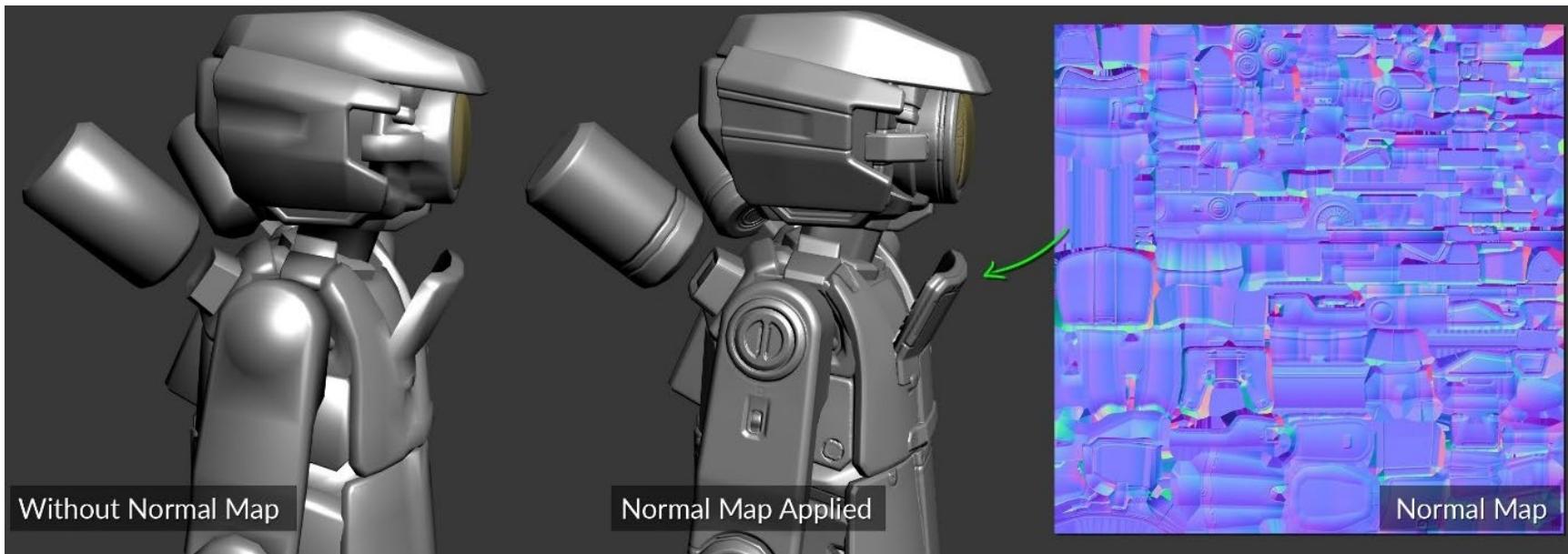
Maintain a very simplistic approach to making your content.

- Minimize complex shaders as best possible.
- Add detail to the mesh within reason in lieu of relying of complex shaders for surface details.

# More Tips...

- LODs and aggressive culling are a must
- Use real geometry instead of:
  - Normal maps (e.g. surface detail)
  - Sprites (e.g. particle effects)
  - Billboards (e.g. foliage)
  - Flipbook movies (e.g. explosions)
- Anti-aliasing:
  - Use None, FXAA or hardware MSAA
- Minimize number of drawcalls
  - VR doubles the drawcall count
- Minimize post-process effects
  - High frame rate is critical

# Normal Map Issues

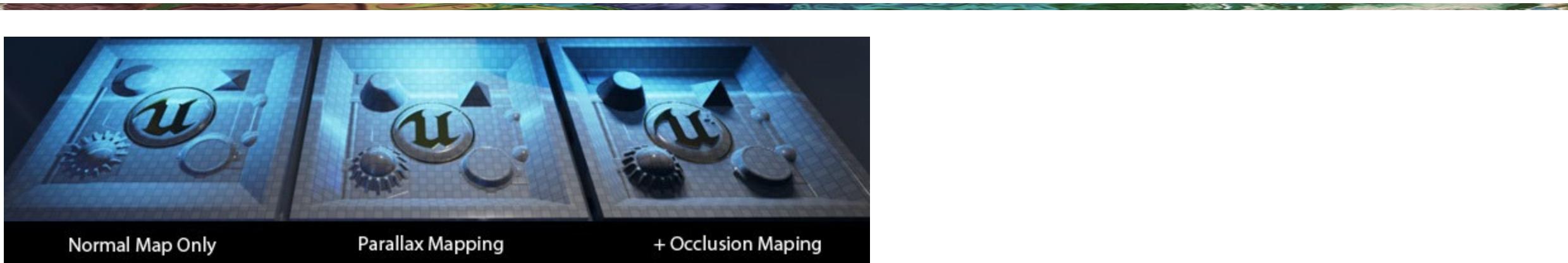


When viewing Normal maps on objects in VR, they do not have the same visual impact as on desktop/console. This is because normal mapping does not account for a binocular display or motion parallax.

Because of this, Normal maps will come out looking flat when viewed with a VR device.

To get around this, you can do one of two things.

# Use Parallax Mapping



Parallax mapping takes Normal mapping to the next level by accounting for depth cues

Normal mapping does not.

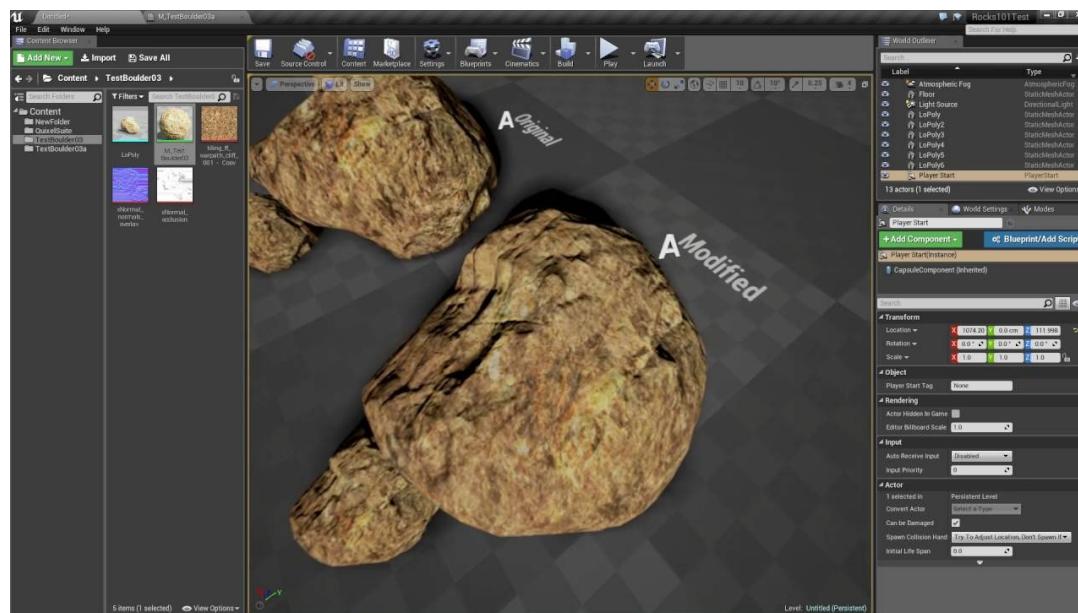
A Parallax mapping shader can better display depth information,  
making objects appear to have more detail than they do.

This is because no matter what angle you look at,  
a Parallax map will always correct itself to show you the correct depth information from that view point.

The best use of a Parallax map would be for cobblestone pathways and fine detail on surfaces.

# Use Tessellation Shader Displacement

- Tessellation Shader Displacement will displace 3D Geometry in real time
- Adding details that are not modeled into the object.
- Tessellation shaders create more vertices and displace them in 3D Space.



# More Performance Considerations

For the VR experience to feel smooth, your game needs to run at 75 hz to 90 hz

To see the current framerate type in “stat fps” or “stat unit” in your console when running the game.

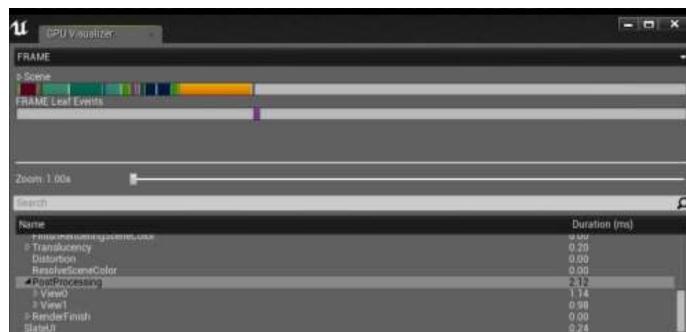
## For GPU Profiling:

To capture a single frame with GPU timings press Ctrl+Shift+, or type in “profilegpu” in the console.

This command dumps accurate timings of the GPU.

You will find that certain processes are a heavy burden on the framerate when using VR

Ambient Occlusion is one common example



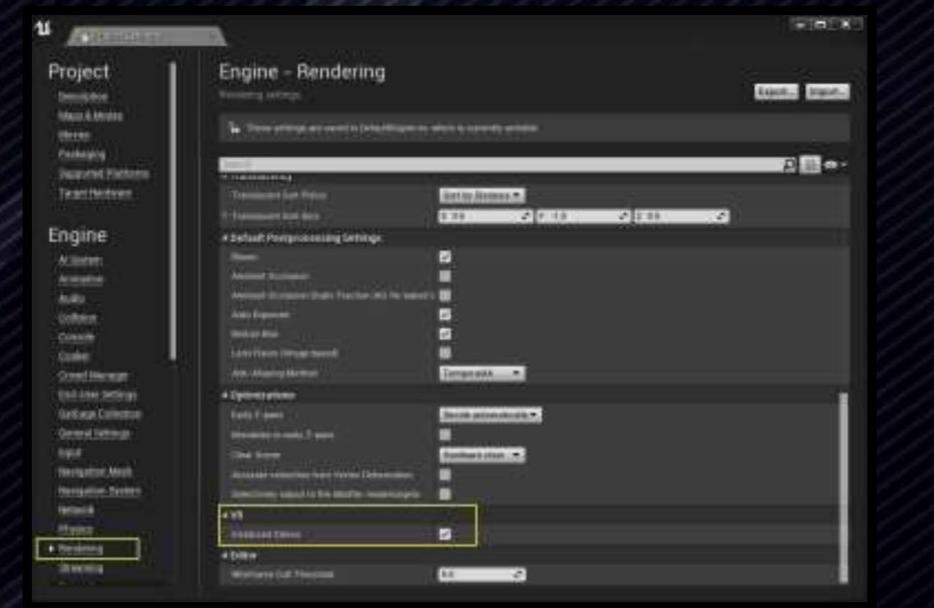
<https://docs.unrealengine.com/4.27/en-US/TestingAndOptimization/PerformanceAndProfiling/GPU/>

# Consider Instanced Stereo

The latest 4.11 release introduces Instanced Stereo Rendering, check the video below for a comparison video of how that works.

*"Basically, we're utilizing hardware instancing to draw both eyes simultaneously with a single draw call and pass through the render loop. This cuts down render thread CPU time significantly and also improves GPU performance. Bullet Train was seeing ~15 – 20% CPU improvement on the render thread and ~7 – 10% improvement on the GPU." – Ryan Vance.*

To enable this feature in 4.11 and above, go to your Project Settings and look for "Instanced Stereo" under the Rendering category.



# Disable Heavy Post-Processors

- Add a Post Process(PP) volume to your level if there is not already one there.
- Select the PP volume and in the Post Process Volume section enable the **Unbound** option so that the settings in the PP volume will be applied to the entire level.



- Expand the Settings of the Post Process Volume and then go through each section and **disable any active PP** settings by enabling that property by clicking on it and then set the value from the default, usually 1.0, to 0 to disable the feature.



- When doing this you will not need to hit every section and **set all the properties to 0**. Instead first disable the really heavy hitting features like Lens Flares, Screen Space reflections, Temporal AA, SSAO, and anything else that might have an impact on performance.
- While a lot of the features are disabled by setting things in your .INI this ensures that nothing will happen to performance if someone deletes the .INI by mistake.

# VFX



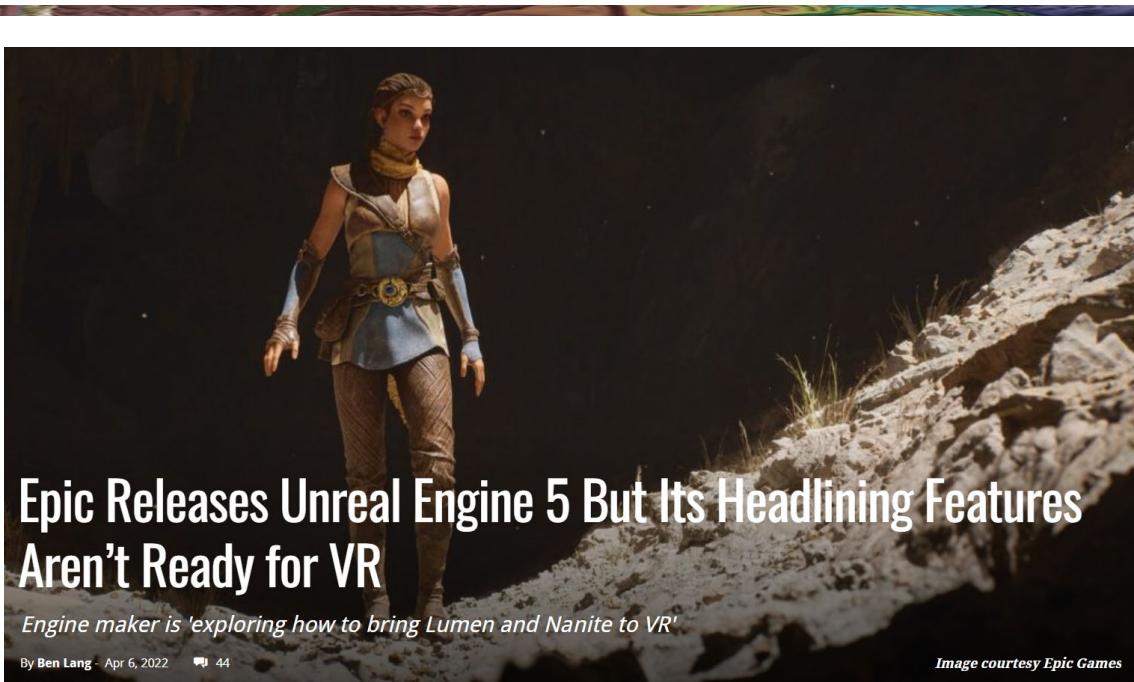
Some VFX techniques like using SubUV Textures to simulate fire or smoke do not hold up well when viewed in VR. In many cases it can be more desirable to use static meshes emitters instead of 2D sprite particles to simulate VFX's like explosions or smoke trails. Near field effects, or effects that happen very close to the camera can work well in VR but only when the effects are made up of Static Meshes particles.

# Lighting and VFX

You should always use Static lighting and lightmaps when making a VR project as this is the cheapest option to render. If you need to use dynamic lighting make sure to limit the number of dynamic lights to as few as possible and make sure that they never touch one another.

If you have an outdoor scene set your directional light to dynamic instead of stationary and then turn on Cascading Shadow Maps and adjust the settings to be as low as possible while still giving you shadows. This is going to take a lot of trial and error to get correct.

# VR in UE 5...but not with all of UE5



Epic Releases Unreal Engine 5 But Its Headlining Features Aren't Ready for VR

*Engine maker is 'exploring how to bring Lumen and Nanite to VR'*

By Ben Lang · Apr 6, 2022 · 44

Image courtesy Epic Games

Also...

## Featured Comment



**kontis** • 20 days ago • edited

Last year they had a Lumen-dedicated stream and they said it wasn't planned, at all, for VR.

Not surprising considering it's the main reason Matrix Awakens runs below 30 FPS at 1080p on PS5.

Lumen also requires deferred renderer (for valid technical reasons), so the VR optimal forward renderer (which Oculus and Valve pushed for) that supports MSAA will never have. This means even if Lumen was available for VR using it would result in blurry vision.

"Although the advantages [of Nanite] can be game-changing, there are practical limits that still remain. For example, instance counts, triangles per mesh, material complexity, output resolution, and performance should be carefully measured for any combination of content and hardware," the company warns developers. "Nanite will continue to expand its capabilities and improve performance in future releases of Unreal Engine."

Lumen, meanwhile, is only designed to target 60 FPS for large outdoor scenes and 30 FPS for indoor scenes on the very latest console hardware. That's a far cry from the 90 FPS minimum for most PC VR headsets. And with Quest 2 being significantly less powerful than the latest consoles, there's just no way it'll be able to handle those kinds of demands. Which may mean that the ultimate limitation in bringing these features to VR is simply performance (or lack thereof).

<https://www.roadtovr.com/unreal-engine-5-release-nanite-lumen-vr/>