Interior Cubemap System

User Guide for Unreal Engine 5.1+

Interior Cubemap System is a production-oriented material solution for creating convincing, view-dependent rooms behind building windows without modeling complete interiors. It is designed for architectural visualization, modular buildings, hero facades, urban environments, and large ISM/HISM city scenes where conventional interior geometry would be too expensive.

The system combines procedural room parallax, Texture2D Array-based interior selection, stable per-window variation, optional facade masking, day/night lighting, configurable interior card layers, and selective wall visibility in one masked master material.

> **Recommended workflow: Duplicate one of the supplied Presets, or create a child Material Instance from a Preset, and edit that copy for your building. Do not create Material Instances directly from MM_InteriorCubemap unless you are creating a new preset, and do not modify the master material or its functions for normal asset setup.**

wParallax compatibility and discount

Expand your room library with compatible wParallax collections, sold separately. These additional libraries are optional.

- Browse wParallax Library
- 20% discount code: **WPAF20**

Table of contents

Feature summary

- View-dependent fake room depth generated entirely in the material.
- Texture2D Array support for compact management of multiple room variants.
- Automatic, stable room selection across facade UV tiles.
- Additional variation between duplicated actors and ISM/HISM instances.
- Optional manual slice selection for art direction and diagnostics.
- Paired unlit/lit slice workflow for day/night changes.
- Sun-vector lighting and externally driven time-of-day lighting workflows.
- Retail and office lighting behaviors.
- Stable per-room maximum emissive variation, optional saturation variation, and optional color-temperature variation.
- Procedural room tiling, aspect, depth, wall-divider, and slab controls.
- Four optional atlas-card layers with independent placement, scale, depth, and opacity.
- Optional texture mask with selectable channel and independent mask tiling.
- Individual floor, ceiling, left-wall, right-wall, and back-wall visibility controls.
- Debug overlay for inspecting the selected cubemap slice.
- Masked blend mode and a reusable Interior Opacity Mask function output.

Package contents

The production materials are located under /Game/InteriorCubemapSystem/Materials in the Content Browser. Ready-to-use Material Instances are in the Presets subfolder.

Main assets

Asset	Type	Purpose
MM_InteriorCubemap	Master Material	Ready-to-use masked facade material containing the complete system.
MF_InteriorCubemaps	Material Function	Main reusable interior system and parameter interface.
MF_ProceduralParallax	Material Function	Builds room parallax coordinates, procedural divisions, depth, and surface masks.
MF_InteriorCubemapLayer	Material Function	Samples and composes an interior cubemap/card layer.
MF_InteriorCardUVPlacement	Material Function	Places and scales optional atlas cards.
MF_InteriorAtlasTiling	Material Function	Provides shared atlas tiling behavior.
MF_InteriorCubemapSelectionSeed	Material Function	Generates stable automatic cubemap selection.
MF_InteriorVariationCore	Material Function	Combines selection and variation inputs.
MF_InteriorWorldPositionRandomness	Material Function	Supplies stable actor/world-position variation.
MF_RandomNoise	Material Function	Produces procedural RGB noise from input UVs using a hashed sine function, used as a randomness source by other interior functions.
MPC_InteriorCubemapSystem	Material Parameter Collection	Exposes the shared TimeOfDay value.
CA_InteriorSystem	Curve Atlas	Day/night response data used by the lighting system.
CA_InteriorSystemFallback	Curve Atlas	Sun-elevation fallback response data.
CURVE_Interior_DayNightEmission	Curve	Default time-of-day emission response.
CURVE_Interior_ElevationFallback	Curve	Default sun-elevation response.

Examples

- Presets include MI_ICS_Rooms_S and small, medium, and large office, lobby, and retail configurations.
- Texture2D Arrays are included under the cubemap texture folders.
- The example level demonstrates the material on prepared facade geometry.

How the system works

The material does not create real room geometry. Instead, it evaluates the camera direction against a virtual room volume behind the facade and calculates which virtual surface the view ray reaches first. The resulting coordinates sample an interior Texture2D Array, producing parallax that changes as the camera moves.

The system divides facade UV space into procedural room cells. Each cell can receive a stable cubemap selection and lighting variation. Optional atlas cards add foreground or midground silhouettes at configurable virtual depths. Finally, the function composites the visible room surfaces and provides both an RGB result and an opacity mask to the master material.

This technique is best understood as a high-quality facade illusion. It provides depth and variation at a fraction of the geometry, draw-call, lighting, and authoring cost of complete interiors, but it is not a replacement for geometry when the camera must enter a room or interact with interior objects.

Requirements and preparation

Engine version

This package supports Unreal Engine 5.1 and newer.

Recommended source assets

Prepare the following before configuring a new building:

1. A facade or window mesh with clean UVs.
2. A Texture2D Array containing the room imagery.
3. The exact number of slices in that array.
4. An optional facade/window mask texture if the material is assigned to surfaces that should not all show interiors.
5. Optional paired unlit/lit slices if automatic day/night appearance is required.

Quick start

1. In the Content Browser, open InteriorCubemapSystem/Materials/Presets and select MI_ICS_Rooms_S as a starting point.
2. Duplicate the preset, give the copy a descriptive name, and use it for your building.
3. Open the new Material Instance.
4. Keep the supplied Cubemap Array for the first setup, or assign a compatible Texture2D Array.
5. Set Cubemap Array Slice Count to the array's exact number of slices.
6. Enable Lit Cubemap Variants Only only if the array follows the paired unlit/lit convention described below.
7. Apply the Material Instance to the facade or window surface.
8. Set Cubemap Grid Columns and Cubemap Grid Rows so the procedural room grid matches the facade.
9. Set Interior Aspect Ratio (W/H) and Interior Depth Scale to match the intended room proportions.
10. Adjust Cubemap U/V Offset only if the sampled imagery is misaligned.
11. Configure the preferred lighting workflow.

> **Before shipping: disable all debug options, including Show Cubemap Slice Index (Debug), so no diagnostic overlay is left visible in the final build.**

For a fresh instance instead of a themed preset, duplicate the closest supplied Preset (or create a child Material Instance from it) and reset any values you do not need, then configure the array, slice count, and room layout as above. Avoid creating Material Instances directly from MM_InteriorCubemap unless you are creating a new preset.

First-pass validation

Before detailed tuning, confirm that:

- A room is visible through the intended facade area.
- Camera movement produces convincing parallax.
- Adjacent windows can select different room slices.
- The configured slice count matches the Texture2D Array.
- Lighting changes through the selected time-of-day workflow.
- No interior is visible on facade regions that should remain opaque.

Mesh and UV requirements

Surface requirements

The material should be applied to planar or near-planar window/facade surfaces. It can cover a full facade when a texture mask identifies the windows, or it can be assigned directly to dedicated window polygons.

The system uses the surface as the opening into the virtual room. Extreme curvature, heavily distorted topology, or rapidly changing surface orientation can weaken the illusion.

UV requirements

- Use clean, non-degenerate UVs.
- Keep window regions consistently oriented.
- Avoid unintended mirroring unless the mirrored interior is acceptable.
- Align UV room cells predictably across modular pieces.
- Check seams where separate facade meshes meet.

The procedural grid is derived from facade UV space. If windows do not line up with the room cells, adjust the interior tiling/aspect controls or revise the mesh UVs. A texture mask can limit visibility, but it cannot correct fundamentally inconsistent UV orientation.

Scale and transforms

Apply sensible mesh scale before final tuning. Non-uniform actor scaling can change the visual relationship between the facade and virtual room. When using world-position-based actor variation, moving an actor between variation cells can intentionally change its selected rooms.

Cubemap array preparation

Slice indexing

Texture2D Array slices use zero-based indexing:

- A 1-slice array has valid index 0.
- A 10-slice array has valid indices 0 through 9.
- A 32-slice array has valid indices 0 through 31.

Always set Cubemap Array Slice Count to the exact number of slices. The system uses this number to keep automatic and manual selection inside the valid range.

Paired unlit/lit convention

When each room has an unlit and lit version, arrange slices in adjacent pairs:

Slice	Content
0	Room 01 - unlit/day
1	Room 01 - lit/night
2	Room 02 - unlit/day
3	Room 02 - lit/night
4	Room 03 - unlit/day
5	Room 03 - lit/night

With Lit Cubemap Variants Only enabled, automatic selection is constrained to odd lit entries (1, 3, 5...). Use this option only with a compatible array layout.

Texture authoring recommendations

- Keep exposure and white balance reasonably consistent across slices.
- Avoid visible seams at projection boundaries.
- Reserve sufficient dark value range for emissive lighting control.
- Test the imagery at the final window size and viewing distance.
- Use the lowest resolution that preserves the desired result.
- Keep paired slices visually registered so day/night changes do not appear to jump.
- Check mipmaps at grazing angles and at the longest expected camera distance.

Material parameter reference

Parameter visibility can depend on static switches. Enable the relevant feature before looking for its dependent controls.

Core cubemap and selection controls

Cubemap Array

The Texture2D Array containing the interior room imagery. This is the main content input for the system.

Cubemap Array Slice Count

The total number of slices in Cubemap Array. This value must exactly match the assigned array. An incorrect value can produce repeated, unexpected, or invalid selections.

Cubemap Grid Columns / Cubemap Grid Rows

Defines the grid structure used by the cubemap content where applicable. Keep these values consistent with the authored layout.

Lit Cubemap Variants Only

Restricts automatic selection to lit variants in a paired unlit/lit array. Enable only when lit variants occupy odd indices.

Use Manual Cubemap Slice

Overrides automatic selection with Manual Cubemap Slice Index. This is useful for art direction, array inspection, and troubleshooting.

Manual Cubemap Slice Index

Zero-based slice index used when manual selection is enabled. Enter a whole number between 0 and Cubemap Array Slice Count - 1.

Cubemap Variation Seed

Changes automatic cubemap selection without changing room geometry or UV layout. Use it to create alternate facade patterns while retaining deterministic results.

Actor Variation Cell Size (cm)

Controls the size of the world-space cell used for actor-position variation. The supplied default is 300 cm. A duplicated actor generally must move into a different cell before this component of the selection changes.

ICS | 02 Interior Layout

Interior Aspect Ratio (W/H)

Matches the virtual interior to the window width-to-height ratio:

- 1.0 represents a square opening.

- 2.0 represents an opening twice as wide as it is tall.
- 0.5 represents an opening twice as tall as it is wide.

Interior Depth Scale

Controls how far the virtual room appears to extend behind the facade. 1.0 preserves the authored baseline. Lower values make the room shallower; higher values make it deeper. Excessive depth can stretch the imagery and expose the limits of the illusion at oblique angles.

Interior Grid Wall Thickness

Controls the gap between adjacent room columns. This is a relative, unitless control, not a distance in centimeters. Zero removes the gap; increasing the value narrows the room opening and produces black emissive dividers. The parallax calculation accounts for this thickness.

Interior Grid Floor/Ceiling Slab Thickness

Controls the gap between room rows. This is a relative, unitless control. Zero removes the gap; increasing the value shortens the room opening and produces black emissive slab regions. Thickness is included in the parallax calculation, including the interior card paths.

ICS | 07 Mapping & Mask

Cubemap U Offset / Cubemap V Offset

Moves the cubemap mapping horizontally or vertically. Use small adjustments after the room grid has been aligned.

Use Cubemap Mask

Compile-time switch that enables texture-mask sampling. Leave it disabled when the material is assigned only to dedicated window polygons. Disabling the switch removes the unused mask branch from that shader variant.

Cubemap Mask

Optional texture controlling where the interior effect is visible.

Cubemap Mask Channel

Selects the texture channel used as the mask. Choose the channel containing the intended window/interior coverage.

Cubemap Mask U Tiling / Cubemap Mask V Tiling

Controls repetition of the mask independently from the interior grid. This is useful when the facade mask and room layout use different texture scales.

ICS | 09 Lighting

Use Sun Vector Fallback

Uses directional-light sun elevation to drive the interior lighting state. This is the simplest automatic workflow and does not require an external TimeOfDay update.

Sun Vector Twilight Bias

Offsets the sun-elevation response. Higher values make lights turn on earlier at dusk and remain on longer at dawn.

Use Manual Time of Day

Uses Manual Time of Day instead of the automatic/shared lighting source. This is useful for look development, screenshots, and diagnostics.

Manual Time of Day

Manually controlled normalized time value used when Use Manual Time of Day is enabled.

Interior Lighting Start Time / Interior Lighting End Time

Defines the normalized time range during which interior lights are active in the time-of-day workflow. If the intended interval crosses midnight, test both sides of the wrap to confirm the desired behavior.

Min Interior Emissive / Max Interior Emissive

Sets the emissive intensity range. The lighting response interpolates between these values as the interior light state changes.

Max Interior Emissive Variation

Varies the maximum emissive intensity between rooms using the existing room-selection seed. Use values from 0 to 1. At 0, Max Interior Emissive is unchanged. At 0.2, rooms receive 80-100% of the configured maximum; at 0.5, they receive 50-100%. The supplied default is 0.2. Min Interior Emissive is unchanged. Changing the selection seed also changes the brightness pattern.

Randomize Light Saturation

Enables per-room desaturation of the final interior color. Disabled preserves the original saturation. This changes color saturation, not hue, and is independent of the color-temperature switch.

Max Interior Saturation Variation

Sets the maximum desaturation when Randomize Light Saturation is enabled. Use values from 0 to 1. At 0, original color is preserved; at 0.2, rooms can lose up to 20% of their saturation; at 1, full grayscale is possible. The supplied default is 0.2. Because this affects the final interior color, it also affects visible card layers.

Retail Style Lighting?

Forces interiors to remain lit independently of the normal day/night interval. Use for shops, restaurants, lobbies, and other spaces expected to remain illuminated.

Office Style Lighting?

Keeps lighting more consistent across an entire floor, reducing per-room variation. Use for office towers or facades where floor-level patterns are more believable than fully independent windows.

Randomize Light Temperature?

Enables variation of interior light color temperature.

Min Light Temperature / Max Light Temperature

Defines the random color-temperature range in Kelvin. Use a narrow range for controlled corporate interiors and a broader range for varied residential or mixed-use facades.

Floor Height (floors light randomization)

World-space floor height used to organize floor-based lighting variation. Set this to the actual vertical spacing between floors for coherent results.

ICS | 10 Wall Visibility

Enable Wall Visibility Controls

Enables individual visibility controls for the virtual room surfaces. Leave disabled unless a specific facade angle or art-direction requirement needs surfaces removed.

Hide Floor? / Hide Ceiling?

Removes the virtual floor or ceiling from the composited interior.

Hide Left Wall? / Hide Right Wall?

Removes the corresponding side wall.

Hide Back Wall?

Removes the rear plane of the virtual room.

Wall visibility controls alter which virtual interior surface contributes to the material result; they do not create physical openings, collision, or real geometry. Hiding a side wall does not automatically connect neighboring interiors or detect building corners.

ICS | 90 Advanced & Debug

Time-of-Day Lighting Curve (Advanced)

Selects the curve-atlas row that controls the time-of-day lighting fade. Leave the supplied value unchanged unless authoring a compatible custom curve atlas.

Sun Elevation Lighting Curve (Advanced)

Selects the curve-atlas row used by the sun-vector workflow. Leave the supplied value unchanged unless deliberately replacing the sun-elevation response.

Show Cubemap Slice Index (Debug)

Displays the selected array index as an on-surface diagnostic. Disable this option in final materials.

Debug Index Digits / Debug Text Color

Controls the diagnostic index display. These parameters do not affect the final interior appearance when debug display is disabled.

Texture Mip Bias

Offsets the mip level used by the interior texture sampling:

- 0 uses the normal calculated mip.
- Positive values select blurrier mips and can reduce distant shimmer.
- Negative values select sharper mips but can increase aliasing and temporal instability.

Test at the final window size, viewing distance, output resolution, and anti-aliasing configuration.

Cubemap selection and variation

Automatic selection is deterministic and combines several sources:

1. UV tile position creates variation between procedural room cells on the same mesh.
2. Object world position creates variation between duplicated actors.
3. Per Instance Random creates variation between ISM/HISM instances.
4. Variation seed allows controlled reshuffling without moving geometry.

The result is intentionally stable: rooms should not change every frame or shimmer as the camera moves. Moving an actor far enough to enter a new world-position variation cell can change its assignment by design.

Choosing between automatic and manual selection

Use automatic selection for production facades and instanced city environments. Use manual selection when:

- A specific room must appear in a hero window.
- A Texture2D Array slice needs inspection.
- An invalid slice-count configuration is being diagnosed.
- Multiple Material Instances are being authored as curated variants.

Lighting workflows

Workflow A: Sun-vector fallback

Recommended when the level has a directional sun and a simple automatic dusk/dawn response is sufficient.

1. Enable Use Sun Vector Fallback.
2. Disable Use Manual Time of Day.
3. Adjust Sun Vector Twilight Bias to control when lights begin to appear.
4. Set the minimum and maximum emissive intensity.
5. Verify the result at noon, sunset, night, and sunrise.

The normalized sun-elevation response is designed to remain independent of directional-light yaw.

Workflow B: MPC / Blueprint time of day

Recommended when the project already has a clock or weather/sky system and multiple materials need a synchronized time value.

1. Disable Use Sun Vector Fallback.
2. Disable Use Manual Time of Day.
3. Update TimeOfDay in MPC_InteriorCubemapSystem through the supplied Blueprint or your own system.
4. Configure Interior Lighting Start Time and Interior Lighting End Time.
5. Verify the normalized time convention used by the external system.

Workflow C: Manual look development

1. Enable Use Manual Time of Day.
2. Adjust Manual Time of Day directly in the Material Instance.
3. Use this mode to tune emissive range, temperature, and timing.
4. Disable manual mode before returning control to the sun or MPC workflow.

Tuning room brightness and saturation

First set Min Interior Emissive and Max Interior Emissive for the desired lighting response. Then adjust Max Interior Emissive Variation for differences between rooms. Enable Randomize Light Saturation only if color variation is wanted, and increase its maximum variation gradually. Use a fixed camera, exposure, and lighting state when comparing settings.

Integrating a third-party sky system

Do not make the master material depend directly on a specific sky plugin. Read the third-party system's time value in Blueprint, normalize it to the convention expected by this package, and write it to MPC_InteriorCubemapSystem.TimeOfDay. This keeps the material portable and centralizes integration logic.

Optional interior card layers

The system provides four optional atlas-card slots:

- Layer TL - top-left atlas card.
- Layer TR - top-right atlas card.
- Layer BL - bottom-left atlas card.
- Layer BR - bottom-right atlas card.

Each card has a compile-time Use Layer? switch. Disable unused cards so their card-specific shader branches are omitted from that Material Instance's compiled variant.

Layer parameters

Each layer exposes:

- Depth - places the card between the opening and back wall. 0 is nearest the opening; 1 is at the rear wall.
- Opacity - 0 hides the card and 1 makes it fully opaque within its authored mask.
- U Offset / V Offset - moves the card inside its atlas slot.
- U Scale / V Scale - scales the card inside its atlas slot; 1 preserves the authored size.

The compositor performs depth-aware ordering and masks supported free-depth cards where a nearer room surface should occlude them. For believable results, keep card depth consistent with the visual perspective baked into the source imagery.

Layer tuning order

1. Enable one card at a time.
2. Set its depth.
3. Correct scale.
4. Correct offset.
5. Set opacity.
6. Re-enable the remaining cards and check overlap from several camera angles.

Masking and wall removal

Facade/window masking

Use Use Cubemap Mask when one material covers both window and non-window regions. White/active mask regions show the interior; black/inactive regions are discarded by the final masked material, subject to the master material's clip threshold.

For dedicated window geometry, leaving the mask feature disabled is usually simpler and cheaper.

Black wall and slab gaps

Interior Grid Wall Thickness and Interior Grid Floor/Ceiling Slab Thickness create separation between room openings. These gap regions output black emissive. This color mask does not itself discard pixels or change the facade opacity mask. The existing facade mask and wall-visibility settings still determine opacity.

Virtual wall removal

The Wall Visibility controls remove selected virtual room surfaces from the function output. This can open the floor, ceiling, left, right, or back side of the fake room for special facade layouts and viewing conditions.

Because the master material uses Masked blend mode, the final opacity behavior is binary. Pixels that fail the opacity-mask test are discarded; surviving pixels render as part of the masked surface.

Important limitation: glass on the same surface

> **Important: True translucent glass and binary opacity-masked wall removal cannot be rendered correctly in one standard Unreal material on one surface. If you need both, use a separate glass surface/material pass over the masked facade rather than combining them in a single material.**

Blend mode is a material-level setting:

- Masked supports efficient binary cutouts and the supplied Interior Opacity Mask, but it cannot produce physically translucent glass in the same pass.
- Translucent can produce glass-like partial opacity, but the entire material is handled as translucent and no longer behaves like a normal masked/opaque facade. Depth writing, sorting, shadows, lighting, and performance characteristics change accordingly.

The professional production solution is a separate glass surface/material pass over the masked interior/facade surface. If one surface and one material are mandatory, glass can only be approximated as an opaque or dithered visual effect; it will not have the correctness of dedicated translucent glass.

Blueprint and MPC integration

MPC_InteriorCubemapSystem contains the shared scalar parameter TimeOfDay. Drive it from your own project clock, weather system, or sky plugin using the steps below.

When implementing a custom driver:

1. Obtain the authoritative project time.
2. Convert it to the normalized range expected by the material.
3. Set the TimeOfDay scalar on MPC_InteriorCubemapSystem.
4. Update only when needed; avoid redundant per-actor updates because an MPC is global.
5. Test editor preview, Play In Editor, standalone, and packaged behavior.

> **Important: Material Parameter Collections are shared globally. One MPC value cannot vary per actor or per building. If separate buildings require different clocks, use separate local controls or another data path instead of expecting MPC_InteriorCubemapSystem.TimeOfDay to differ between them.**

Performance and scalability

The system trades geometry and interior draw calls for shader work and texture sampling. It is well suited to repeated facades, but should still be profiled on the target platform.

Recommended optimization practices

- Use Material Instances instead of duplicating the master material.
- Disable Use Cubemap Mask when no mask is required.
- Disable all unused optional card layers.
- Disable wall visibility controls when every room surface is needed.
- Disable debug display in production.
- Use the smallest Texture2D Array resolution that holds up at the final viewing distance.
- Avoid unnecessarily large arrays containing near-duplicate rooms.
- Prefer the built-in variation system over stacking additional random-noise functions.
- Use ISM/HISM for repeated building modules where appropriate.
- Tune positive Texture Mip Bias cautiously for distant facade stability.
- Profile shader complexity, texture memory, overdraw, and frame time in the actual scene.

Masked-material considerations

Masked materials can create overdraw and may cost more than fully opaque materials, especially when a large portion of the surface is discarded. Keep facade geometry and masks reasonably tight when targeting large scenes or constrained hardware.

Scalability strategy

For large cities, consider separate quality tiers:

- Hero: higher-resolution arrays, optional card layers, close viewing.
- Mid-range: reduced array resolution and fewer enabled layer features.
- Distant: simplified emissive window material or baked facade treatment.

Transition distances should be chosen by screen size and tested for visible popping.

Troubleshooting

The material is gray, black, or blank

- Confirm Cubemap Array is assigned.
- Confirm the Texture2D Array imported successfully.
- Confirm Cubemap Array Slice Count is greater than zero and exactly correct.
- Disable manual selection or choose a valid manual index.
- Temporarily disable the cubemap mask and wall visibility controls.
- Check the Output Log and material compiler for errors.

The wrong cubemap slice appears

- Verify zero-based indexing.
- Verify the configured slice count.
- Check Lit Cubemap Variants Only against the array's ordering.
- Enable Use Manual Cubemap Slice and inspect each slice.
- Use Show Cubemap Slice Index (Debug) to inspect automatic selection.

Only odd slices appear

Lit Cubemap Variants Only is enabled. This is expected for paired arrays where lit variants occupy odd indices. Disable it for unrestricted automatic selection.

Adjacent windows all show the same room

- Confirm the windows occupy different procedural UV cells.
- Increase interior U/V tiling if the full facade currently resolves to one room cell.
- Change Cubemap Variation Seed.
- Verify the array contains more than one eligible slice.

Duplicated actors look identical

- Move actors farther than Actor Variation Cell Size (cm).
- Change the variation seed on one Material Instance.
- For large repeated sets, consider ISM/HISM so Per Instance Random contributes automatically.

ISM/HISM instances do not vary as expected

- Confirm the instances actually use an instanced component.
- Confirm all instances use the intended Material Instance.

- Verify the automatic-selection path is active.
- Check that the array has enough valid eligible slices.

The room grid does not match the windows

- Correct Cubemap Grid Columns and Cubemap Grid Rows first.
- Set Interior Aspect Ratio (W/H) to match the opening.
- Adjust wall and slab thickness after aligning the room grid.
- Inspect the facade UVs for scaling, rotation, or mirroring errors.
- Use mask tiling only to align the mask, not to compensate for incorrect room-grid UVs.

Parallax looks inverted or moves in the wrong direction

- Check UV orientation and mirrored islands.
- Check surface normals and tangent basis.
- Confirm the material is applied to the intended outward-facing surface.
- Test on a simple planar mesh to separate asset problems from material configuration.

Interiors look stretched at oblique angles

- Reduce Interior Depth Scale.
- Match aspect ratio more accurately.
- Check whether the facade is too curved for the intended illusion.
- Add real geometry for windows that must withstand extreme close-up or grazing views.

Black strips between rooms are too wide

Reduce Interior Grid Wall Thickness or Interior Grid Floor/Ceiling Slab Thickness. These are unitless relative controls, so do not enter centimeter measurements. Set both to 0 to remove the added gaps, then increase them gradually. Match the mesh/UV aspect ratio before tuning thickness.

The cubemap mask is misaligned

- Verify the selected mask channel.
- Reset mask tiling to a known baseline.
- Check whether the mask should use the same UV layout as the facade.
- Correct Cubemap Mask U/V Tiling independently from interior tiling.

Wall visibility controls appear to do nothing

- Enable Enable Wall Visibility Controls.
- Ensure the hidden surface is visible from the current camera angle.

- Temporarily hide one surface at a time.
- Remember that these controls affect virtual room surfaces, not facade geometry.

Removed walls do not reveal translucent glass

This is expected when everything uses one masked material and one surface. A discarded masked pixel cannot also render partially transparent glass in the same pass. Use separate glass geometry/material for correct translucency.

Lights do not change with time

- Check whether Use Manual Time of Day is enabled.
- Check whether Use Sun Vector Fallback is enabled.
- For MPC mode, verify MPC_InteriorCubemapSystem.TimeOfDay is changing at runtime.
- Confirm start/end values use the same normalized time convention.
- Check whether Retail Style Lighting? intentionally forces lights on.

Lights switch too early or too late in sun-vector mode

Adjust Sun Vector Twilight Bias. If a substantially different response is required, author and assign a compatible custom sun-elevation curve rather than forcing extreme bias values.

Light colors are too varied

- Disable Randomize Light Temperature?, or
- Narrow the difference between Min Light Temperature and Max Light Temperature.

Some rooms are too dim

Set Max Interior Emissive Variation to 0 to compare against the unvaried maximum. Then adjust Max Interior Emissive and the lighting state. Room imagery can also contain different baked brightness levels.

Interior colors look washed out

Disable Randomize Light Saturation or reduce Max Interior Saturation Variation. If the color cast is the issue, check Randomize Light Temperature and its Kelvin range separately.

Texture Mip Bias seems to have no effect

Its effect is easiest to see at distance and grazing angles. Temporarily test +3 for visibly blurrier sampling or -3 for sharper sampling, then return to a restrained production value. Negative bias can increase shimmer.

Optional cards intersect or sort incorrectly

- Enable and tune one card at a time.
- Verify each card's depth is in the intended front-to-back order.

- Keep offsets and scales within the authored atlas region.
- Check the result from both left and right camera angles.
- Avoid source art whose baked perspective contradicts the configured virtual depth.

FAQ

Is this real geometry?

No. The rooms are generated through view-dependent material projection. They do not provide collision, navigation, physical interior shadows, or interactive objects.

Can the player enter the rooms?

No. Use modeled geometry for enterable spaces. A common production approach uses real geometry for accessible floors and this system for inaccessible windows.

Can I use one material over an entire facade?

Yes. Enable Use Cubemap Mask and provide a window mask. Dedicated window polygons can be more efficient and easier to control in some projects.

Can I remove individual walls?

Yes. Enable Enable Wall Visibility Controls, then hide the floor, ceiling, left wall, right wall, or back wall independently.

Can I remove walls and also render real glass in the same material on the same surface?

Not correctly with the supplied masked pipeline. Unreal assigns one blend mode to the material. Masked pixels are binary, while glass requires translucency. Use a second glass surface/material for production-quality results.

Can I convert the master material to Translucent?

Technically yes, but it changes the rendering model for the entire material. Expect different depth, sorting, shadow, lighting, and performance behavior. This is a custom modification and is not the recommended supported setup.

Why use a Texture2D Array instead of individual textures?

An array lets the shader select among many consistently formatted room images through an index while keeping one material interface. Every slice must follow compatible dimensions and format requirements.

How many cubemap slices can I use?

The practical limit depends on texture dimensions, format, memory budget, target hardware, and the engine's Texture2D Array limits. Use only as many slices as provide meaningful visible variety, then profile memory and rendering cost on the target platform.

Are manual slice indices zero-based?

Yes. For N slices, valid manual indices are 0 through N - 1.

Why are only slices 1, 3, 5, and so on selected?

Lit Cubemap Variants Only is enabled. Those odd entries are expected to be lit variants in the paired array convention.

Will automatic variation change while the camera moves?

No. Selection is designed to remain stable during camera movement. Moving the actor into another world-position variation cell or changing its seed can intentionally produce a new pattern.

Does Nanite improve this material?

Nanite addresses mesh rendering rather than the interior projection itself. Whether it benefits the facade depends on the facade geometry and the project's Nanite/material constraints. Validate the complete asset on the target engine version and platform.

Does the system work with Lumen?

The facade material can be used in a Lumen project, but the virtual rooms are shader imagery rather than real emissive interior geometry. Do not assume they will contribute lighting exactly like modeled rooms. Evaluate reflections, indirect lighting, and emissive behavior in the final scene.

Can the interior cast accurate shadows inside the room?

No. Shadows visible in the cubemap are baked into the imagery. The virtual furniture and walls are not independent scene geometry.

Can I use my own day/night system?

Yes. Normalize its time value and write it to MPC_InteriorCubemapSystem.TimeOfDay, or adapt the integration Blueprint. Keep project-specific logic outside the master material when possible.

Can different buildings use different times of day?

Not through one shared MPC value alone. An MPC is global. Use per-instance/local parameters or separate control paths if buildings need independent clocks.

Should I edit the material function directly?

Not for normal use. Configure Material Instances. Duplicate and modify the master/function only when intentionally creating an unsupported custom branch and keep the original assets unchanged for comparison and updates.

Can I add more than four card layers?

That requires modifying the material functions and compositor. Every additional layer increases complexity and sampling cost, so profile any extension carefully.

What should be disabled before shipping?

Disable the cubemap-index debug display, unused mask branches, unused card layers, unnecessary wall controls, and manual look-development overrides.

Support checklist

Include the following when reporting an issue:

- Unreal Engine version and target platform.
- Name of the affected Material Instance.
- Screenshot of all overridden Material Instance parameters.
- Assigned Texture2D Array and exact slice count.
- Whether Lit Cubemap Variants Only is enabled.
- Whether selection is automatic or manual.
- Mesh type: Static Mesh, ISM, or HISM.
- Screenshot of the mesh UVs when alignment is involved.
- Screenshot or short capture of the problem from the affected camera angle.
- Active lighting workflow: sun vector, MPC/Blueprint, or manual.
- Whether the issue occurs in Editor, PIE, standalone, or packaged build.
- Any material compile errors or relevant Output Log messages.

Third-party content, licensing, and credits

This package includes sample cubemap textures from wParallax for demonstration of the Interior Cubemap System.

wParallax sample cubemaps

The textures and related layout/naming convention are included with permission from the wParallax creator. Copyright and ownership of the original imagery remain with their respective rights holder. For additional information about wParallax products, visit wparallax.com.