



**Unreal Fest 2024 Seoul**

# **Advanced Rendering & Debugging Tips for Unreal Engine 5**

**Noriaki Shinoyama**  
Developer Relations, Lead  
Epic Games Japan

**Original Author**  
**Yutaro Sawada**  
Software Engineer, Developer Relations  
Epic Games Japan

# Theme



Introduction

# Who made this presentation?

Yutaro Sawada

Epic Games Japan,  
Developer Relations Engineer

Graphics Specialist



Introduction

# Who I am?

## Noriaki Shinoyama

Epic Games Japan,  
Developer Relations, Lead  
[nori.shinoyama@epicgames.com](mailto:nori.shinoyama@epicgames.com)

International Sake Sommelier

X(Twitter): [@tempkinder](https://twitter.com/tempkinder)



# Agneda

Nanite

Lumen

VSM

Other Development Tips

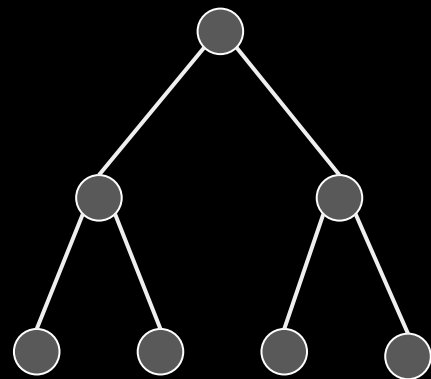
**Nanite**



# Nanite: Overview

## Virtualized Geometry

- Divide the mesh into **clusters** to build a hierarchical structure
- Switching drawing clusters like LOD on a per-cluster basis
- Streaming only the data that is actually drawn to memory



# Nanite : Overview

## Point

- Even super high-resolution meshes render down to one triangle of detail per pixel
- Software rasterizer of Nanite enables **multi-view rendering**



# Nanite supported a variety of materials...

But still **opaque rendering is preferred**, especially when aiming for 60 fps.  
Masked materials are computationally expensive

Trees in Epic's game are also drawn opaque with polygonal leaves.

<https://www.unrealengine.com/ja/tech-blog/bringing-nanite-to-fortnite-battle-royale-in-chapter-4>

※Trees in the Electric Dream environment demo shown at GDC2023 are Masked Material

Not entirely useless, but you should do profiling carefully.



# Example: Trees generated by PCG



# SM\_EuropeanHornbeam\_Field\_01

Tree assets used in Electric Dream

- Masked Material
- Color changes depending on position and natural sway (WPO) already set up.

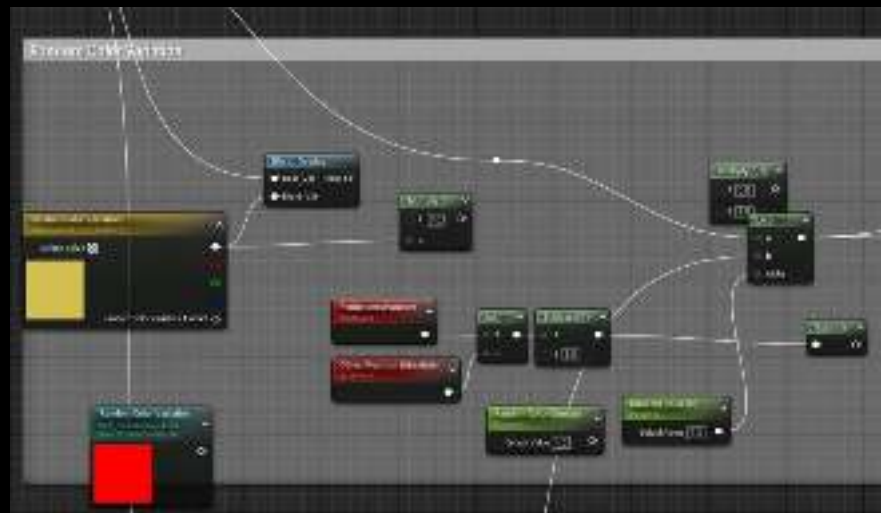
It's **Substrate material**.

Needs to be fixed when used in your projects

(Delete **Legacy Conversion node** in the parent material "M\_Foliage")



# MF\_WorldspaceColorVariation



# Nanite performance on the test scene

## Environment

- Masked material
- with WPO
- PC Development build
- 1080p



## PC Spec

CPU: AMD Ryzen Threadripper PRO 3995WX

GPU: NVIDIA GeForce RTX 3080

Memory: 256GB

# Removing WPO and Masked...

- has very large improvements on performance

**Original**

Frame: 25.83 ms  
Game: 13.88 ms  
Draw: 1.26 ms  
RHIT: 1.04 ms  
GPU Time: 24.42 ms  
DynRes: Unsupported  
Draws: 1762  
Prims: 23.5K

**Without WPO**

Frame: 15.65 ms  
Game: 11.08 ms  
Draw: 1.20 ms  
RHIT: 0.70 ms  
GPU Time: 14.34 ms  
DynRes: Unsupported  
Draws: 1614  
Prims: 21.4K

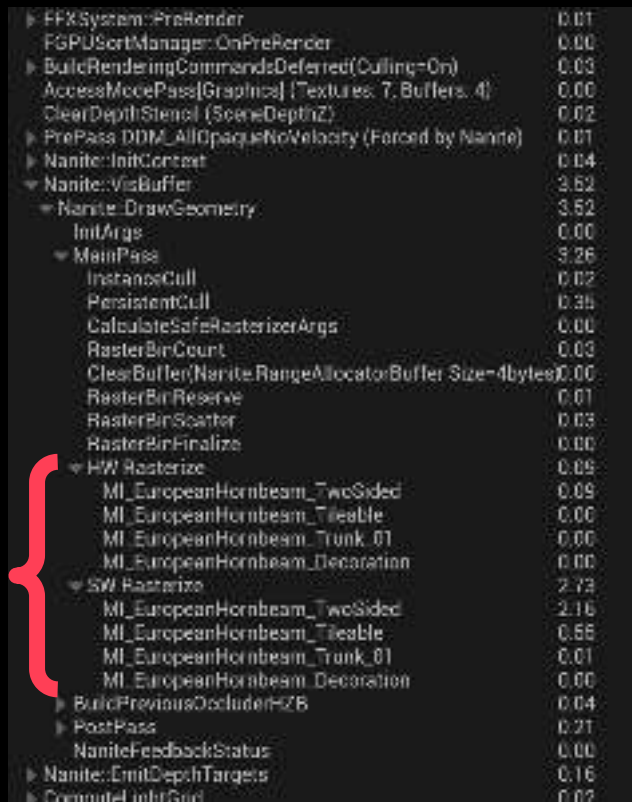
**Without Opaque/Masked**

Frame: 13.08 ms  
Game: 8.10 ms  
Draw: 1.23 ms  
RHIT: 7.85 ms  
GPU Time: 11.85 ms  
DynRes: Unsupported  
Draws: 1354  
Prims: 19.3K

# Nanite performance on the test scene

- Where Nanite affects a lot is...
  - VisBuffer
  - Shadow Depth
  - Lumen

Rasterization time per material in Nanite  
can be displayed  
by activating the following console command  
*r.Nanite.ShowMeshDrawEvents 1*



FFXSystem:PreRender	0.01
FGPUSortManager:OnPreRender	0.00
BuildRenderingCommandsDeferred(Culling=On)	0.03
AccessModePass(Graphics) [Textures: 7, Buffers: 4]	0.00
ClearDepthStencil (SceneDepthZ)	0.02
PrePass DDM_AllOpaqueNoVelocity (Forced by Nanite)	0.01
Nanite:InitContext	0.04
Nanite:VisBuffer	3.52
Nanite:DrawGeometry	3.52
InitArgs	0.00
MainPass	3.26
InstanceCull	0.02
PersistentCull	0.35
CalculateSafeRasterizerArgs	0.00
RasterBinCount	0.03
ClearBuffer(Nanite.RangeAllocator:Buffer Size=4bytes)	0.00
RasterBinReserve	0.01
RasterBinScatter	0.03
RasterBinFinalize	0.00
HW Rasterize	0.09
MI_EuropeanHornbeam_TwoSided	0.09
MI_EuropeanHornbeam_Tileable	0.00
MI_EuropeanHornbeam_Trunk_01	0.00
MI_EuropeanHornbeam_Decoration	0.00
SW Rasterize	2.73
MI_EuropeanHornbeam_TwoSided	2.16
MI_EuropeanHornbeam_Tileable	0.56
MI_EuropeanHornbeam_Trunk_01	0.01
MI_EuropeanHornbeam_Decoration	0.00
BuildPreviousOccluderHZB	0.04
PostPass	0.21
NaniteFeedbackStatus	0.00
Nanite:EmitDepthTargets	0.16
GenerateLightGrid	0.02

# About Nanite's Rasterizer

- **Software** and **Hardware** Rasterizer

GPU HW rasterization is slow for very tiny polygons,  
SW rasterizer was made for such tiny polygons.

(HW and SW are divided according to the following settings)

*r.Nanite.MinPixelsPerEdgeHW 32*

- Software rasterizer is turned off when the following setting is 0

*r.Nanite.ComputeRasterization*

✳ Nanite uses Async Compute on supported platforms

You can turn off it by the following settings to 0

*r.Nanite.AsyncRasterization*

HW Rasterize	0.00
MI_EuropeanHombeam_TwoSided	0.00
MI_EuropeanHombeam_Tileable	0.00
MI_EuropeanHombeam_Trunk_01	0.00
MI_EuropeanHombeam_Decoration	0.00
SW Rasterize	2.73
MI_EuropeanHombeam_TwoSided	2.16
MI_EuropeanHombeam_Tileable	0.55
MI_EuropeanHombeam_Trunk_01	0.01
MI_EuropeanHombeam_Decoration	0.00

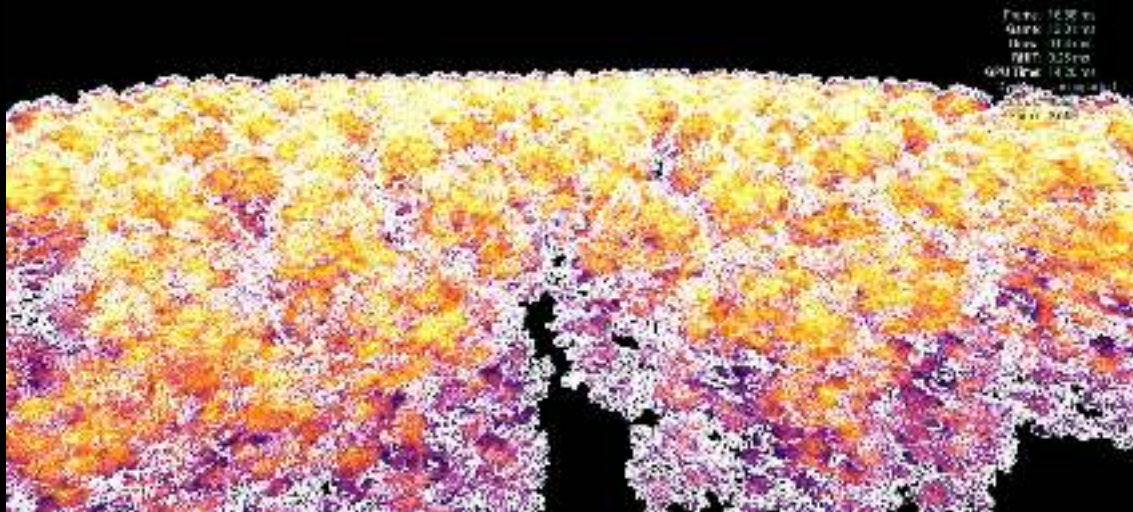
# Nanite: Optimization General Guidelines

- Reduce overdraw
- Minimize the use of Programmable Rasterizer such as Masked



# Comparison Opaque and Masked

- Nanite Visualization -> **Overdraw**  
Rasterization of the trees causes overdraw



# WPO disabled by distance

You can disable WPO for each distant StaticMeshComponent

Nanite Visualization -> **Evaluate WPO**

It visualize the meshes whose WPO is evaluated / isn't evaluated



Evaluate World Position Offset	<input checked="" type="checkbox"/>
World Position Offset Writes Velocity	<input checked="" type="checkbox"/>
World Position Offset Disable Distance	3000

# When Nanite is not rendered correctly...

Tons of Nanite objects might not be rendered correctly.

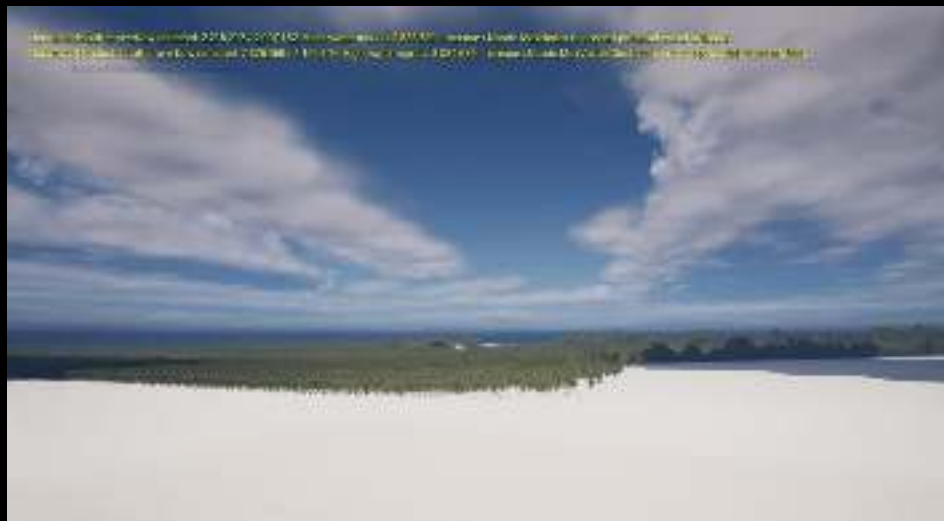
Adjust not to exceed the following parameters

*r.Nanite.MaxNodes*

*r.Nanite.MaxVisibleClusters*

You might need to check Nanite PoolSize

*r.Nanite.Streaming.StreamingPoolSize*



# When Nanite is not rendered correctly...

In some cases, the silhouette of the Nanite meshes can be broken.

Try to increase the level of detail per pixel with

*r.Nanite.MaxPixelsPerEdge*

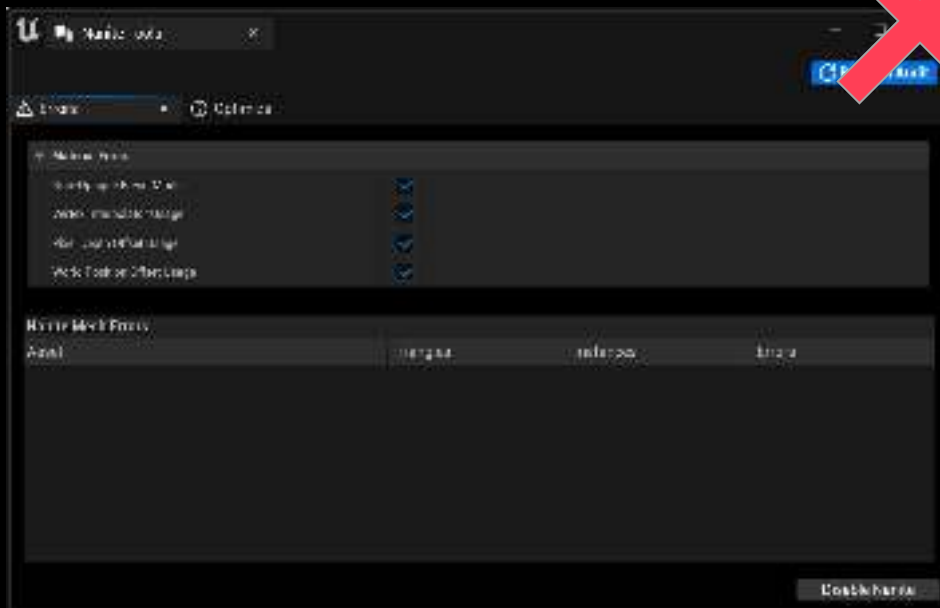
It specifies the size of Nanite triangles in pixels

Smaller values (e.g., 0.1) display higher res clusters



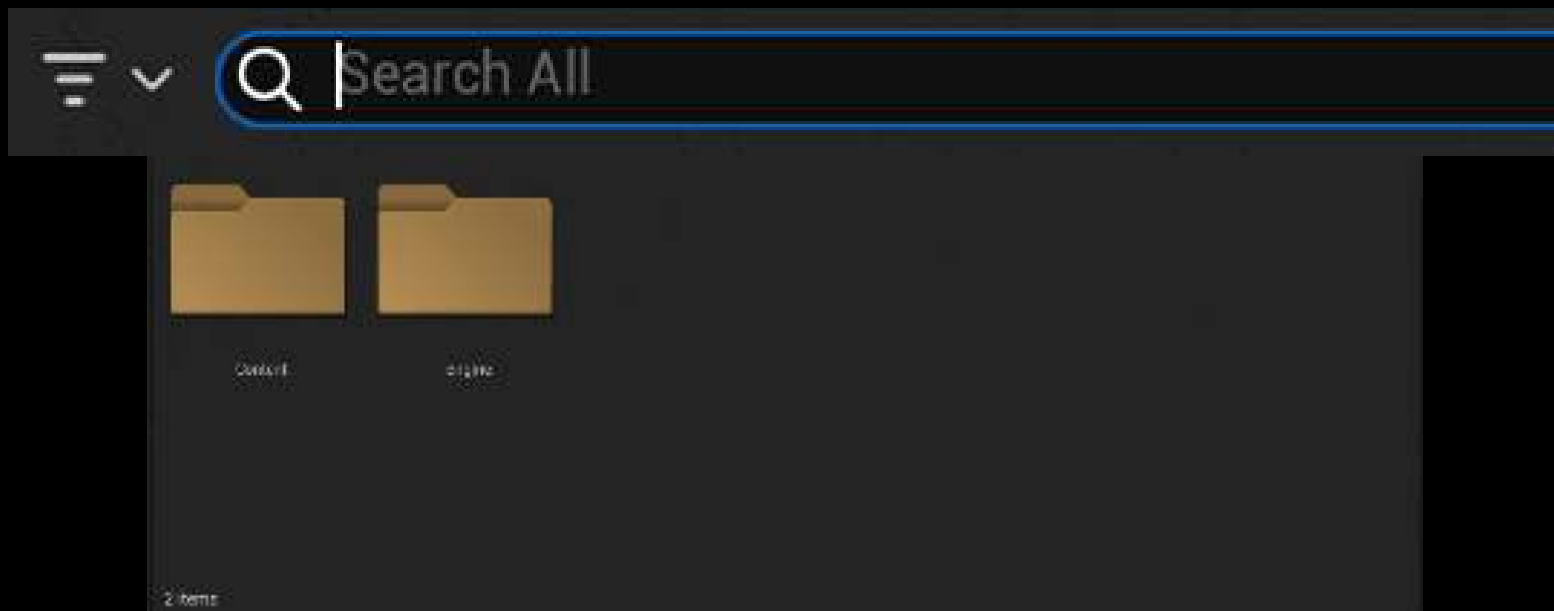
# Note: Nanite Tools

Nanite Tools was discontinued because it could be replaced by Content Browser.



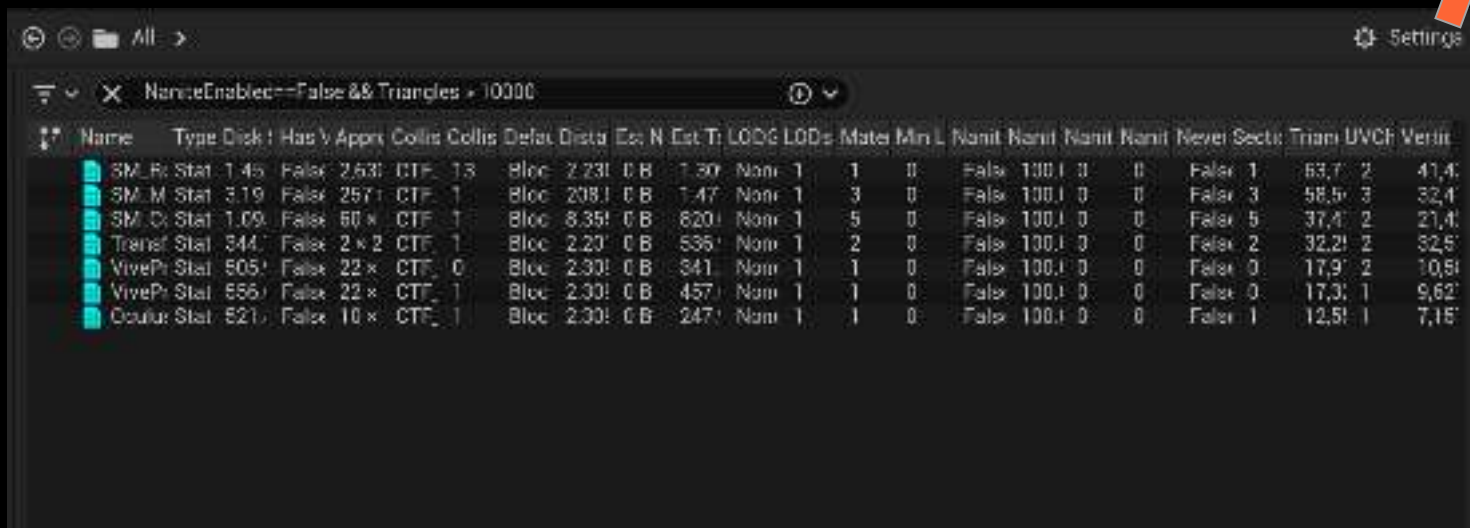
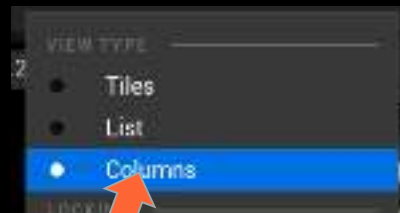
# Filter Methods for Non Nanite Meshes

Filtering by property value on the search bar of Content Browser



# Filter Methods for Non Nanite Meshes

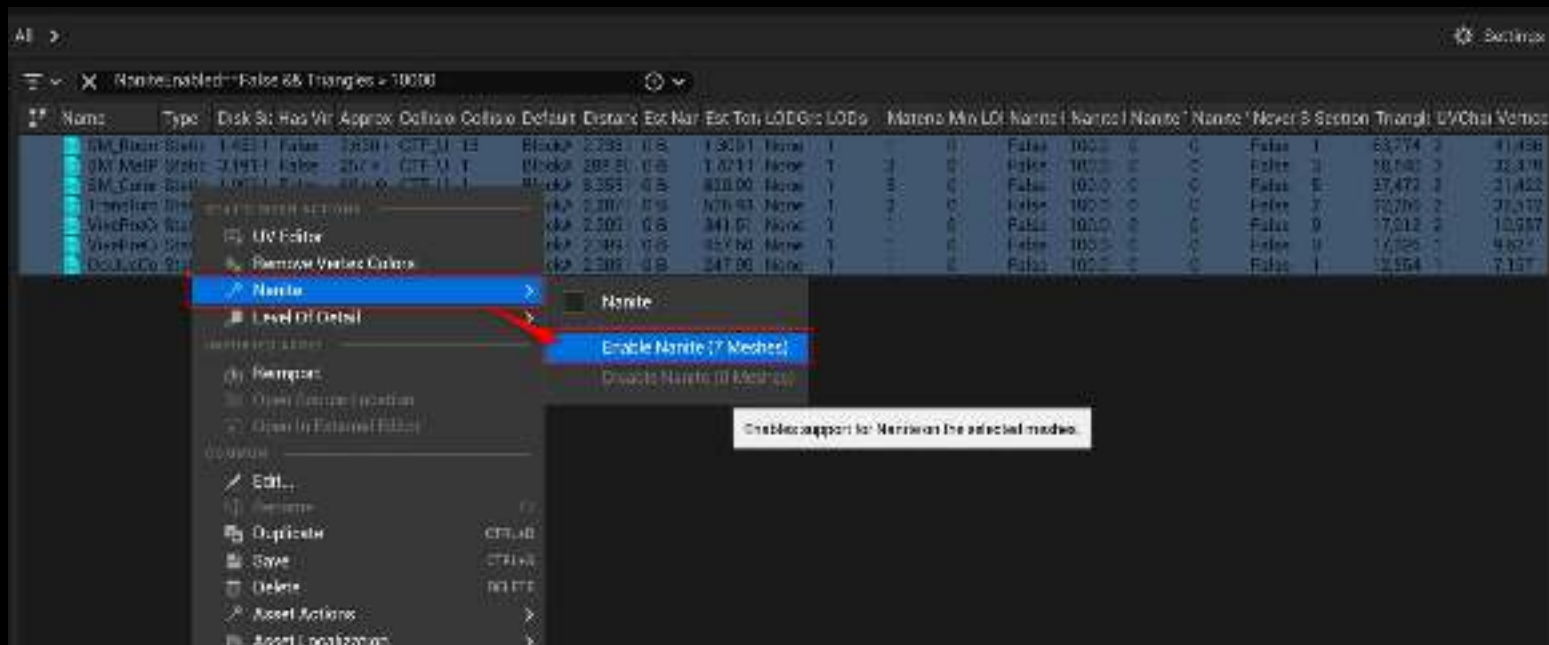
Columns view allows sorting, etc.

A screenshot of the Unreal Engine interface showing a table of mesh data. The table is titled 'NaniteEnabled==False && Triangles > 10000'. The table has 23 columns and 7 rows of data. The columns are: Name, Type, Disk I, Has V, Appri, Collis, Collis, Defat, Dista, Est, N, Est, T, LODG, LODs, Mate, Min, L, Nanit, Nanit, Nanit, Nanit, Nerve, Sects, Trian, UVCH, Verit. The rows represent different mesh assets with their respective statistics.

Name	Type	Disk I	Has V	Appri	Collis	Collis	Defat	Dista	Est	N	Est	T	LODG	LODs	Mate	Min	L	Nanit	Nanit	Nanit	Nanit	Nerve	Sects	Trian	UVCH	Verit
SM_R: Stat	1.45	Falax	2.63	CTF	13	Bloc	2.23	0 B	1.30	Nanr	1	1	0	Fals	100.0	0	0	Fals	100.0	0	0	Fals	1	63.7	2	41.4
SM_M: Stat	3.19	Falax	257	CTF	1	Bloc	208.1	0 B	1.47	Nanr	1	3	0	Fals	100.0	0	0	Fals	100.0	0	0	Fals	3	58.5	3	32.4
SM_D: Stat	1.09	Falax	60 x	CTF	1	Bloc	8.35	0 B	620	Nanr	1	5	0	Fals	100.0	0	0	Fals	100.0	0	0	Fals	5	37.4	2	21.4
Transf: Stat	344	Falax	2 x 2	CTF	1	Bloc	2.20	0 B	536	Nanr	1	2	0	Fals	100.0	0	0	Fals	100.0	0	0	Fals	2	32.2	2	32.5
VivePr: Stat	505	Falax	22 x	CTF	0	Bloc	2.30	0 B	341	Nanr	1	1	0	Fals	100.0	0	0	Fals	100.0	0	0	Fals	0	17.9	2	10.5
VivePr: Stat	556	Falax	22 x	CTF	1	Bloc	2.30	0 B	457	Nanr	1	1	0	Fals	100.0	0	0	Fals	100.0	0	0	Fals	0	17.3	1	9.62
Contu: Stat	521	Falax	10 x	CTF	1	Bloc	2.30	0 B	247	Nanr	1	1	0	Fals	100.0	0	0	Fals	100.0	0	0	Fals	1	12.5	1	7.15

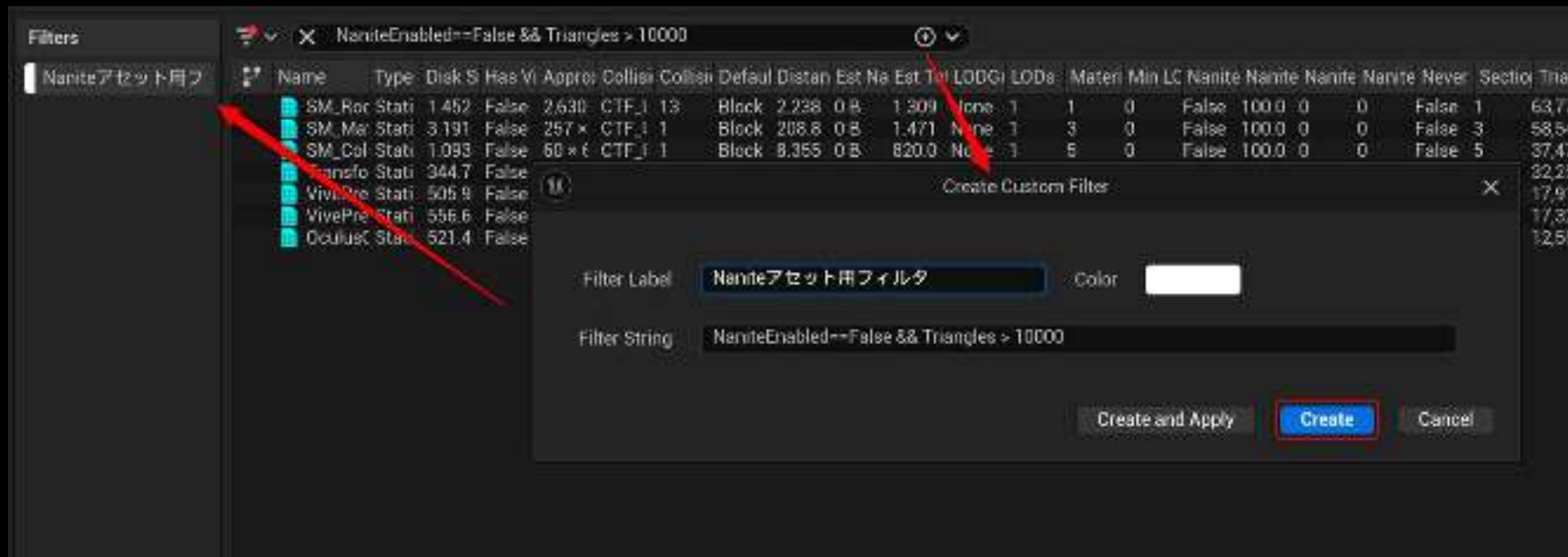
# Filter Methods for Non Nanite Meshes

Make multiple assets Nanite from the right-click menu



# Saving Filters

Filters can be saved from the + icon to the right of the search window



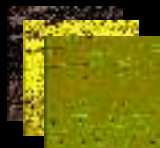
**Lumen**



# Lumen Algorithm Overview



Lumen Card  
Generation



Construction of Lumen Scene  
and Lighting on Lumen Scene

On-screen  
Low resolution  
GI calculations

Upsampling  
on Lumen

Advance Preparation  
(Editor)

Runtime

# Lumen Cards

In order to calculate GI,  
Lumen creates Lumen cards covering each object to calculate GI

You can visualize Lumen cards by *r.Lumen.Visualize.CardPlacement 1*



# Lumen card resolution



Mirror material assigned to floor / screen tracing turned off

# FLumenSurfaceCacheUpdateMeshCardsTask

The resolution of a Lumen Card can dynamically change (max 1024)

- Cvars useful to adjust

Black if required resolution is less than *CardMinResolution*

*r.LumenScene.SurfaceCache.CardMinResolution*

Scale value according to the size of the card's OBB (maximum X or Y) and distance

*r.LumenScene.SurfaceCache.CardTexelDensityScale*

Coefficient to decide the max value according to the size of the OBB of the card (maximum X or Y)

*r.LumenScene.SurfaceCache.CardMaxTexelDensity*



# Lumen: LumenScene Update

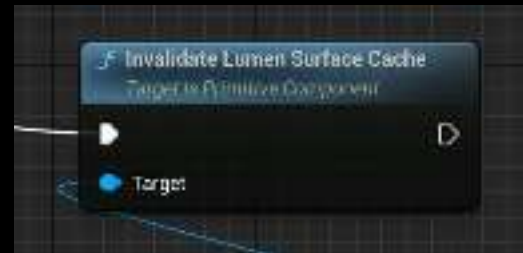
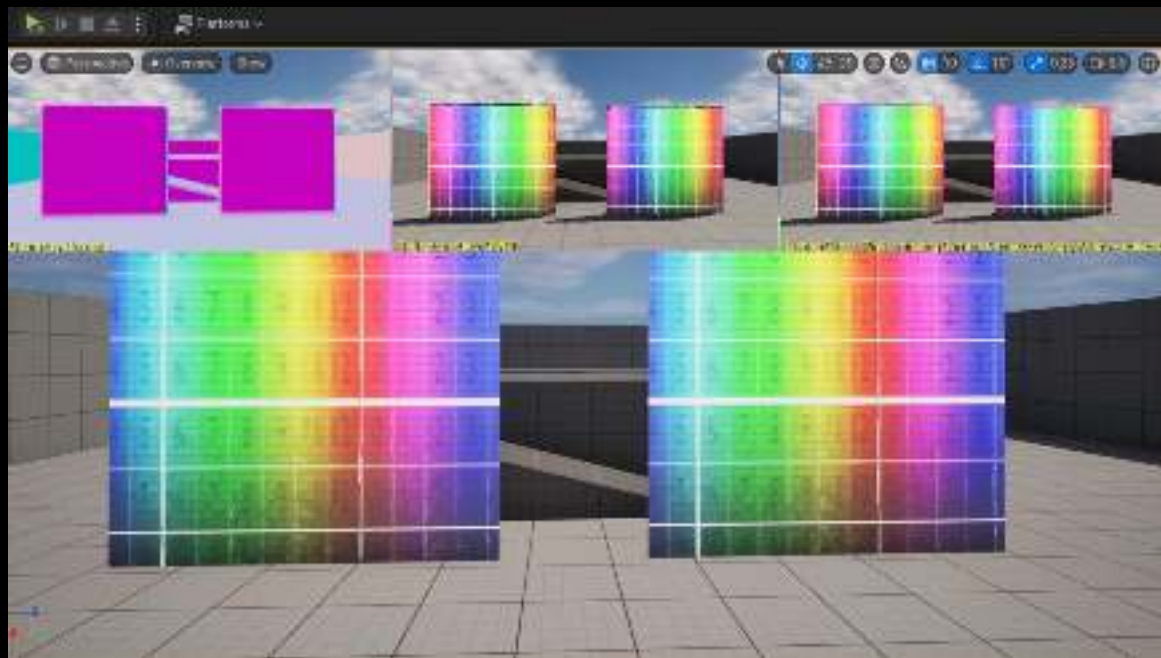


# Lumen: LumenScene Update



*r.LumenScene.SurfaceCache.CardCaptureRefreshFraction (default: 0.125)*

# Lumen: LumenScene Update



*InvalidateLumenSurfaceCache* (C++/BP)

# Lumen: Optimization

The first move is to set up *scalability*

Possible actions on the asset side ...

- Merge Lumen cards
- Adjustment to lower the resolution of the Lumen card
- Use Nanite meshes as much as possible

Scalability Groups					
	Low	Medium	High	Epic	Cinematic
View Distance	Near	Medium	Far	Epic	Cinematic
Anti-Aliasing (TSR)	Low	Medium	High	Epic	Cinematic
Post Processing	Low	Medium	High	Epic	Cinematic
Shadow	Low	Medium	High	Epic	Cinematic
Global Illumination	Low	Medium	High	Epic	Cinematic
Reflections	Low	Medium	High	Epic	Cinematic
Textures	Low	Medium	High	Epic	Cinematic
Effects	Low	Medium	High	Epic	Cinematic
Foliage	Low	Medium	High	Epic	Cinematic
Shading	Low	Medium	High	Epic	Cinematic

# Virtual Shadow Maps

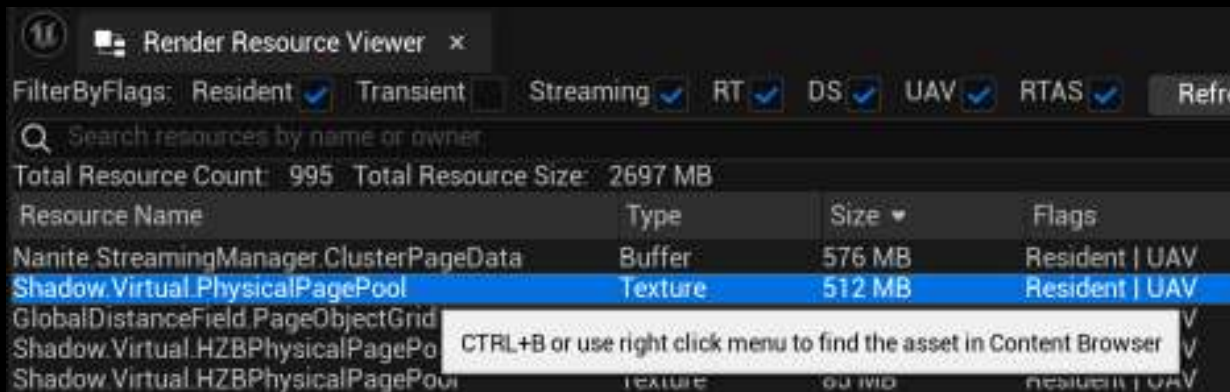


# Virtual Shadow Maps

The default setting of VSM has high memory consumption and load, such as **PoolSize of 512MB**.

Since we've heard from several devs that they have given up on using VSM,

Let me introduce the basic behavior of the VSM to understand how to adjust and set up the VSM



# VSM Algorithm

**Shadow Map  
Generation**

+

**Shadow  
Rednering**

# VSM Algorithm

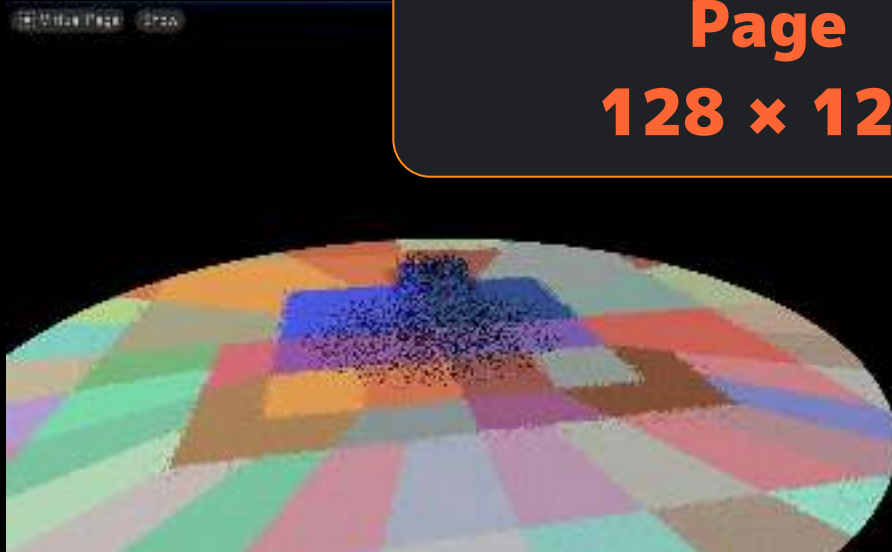
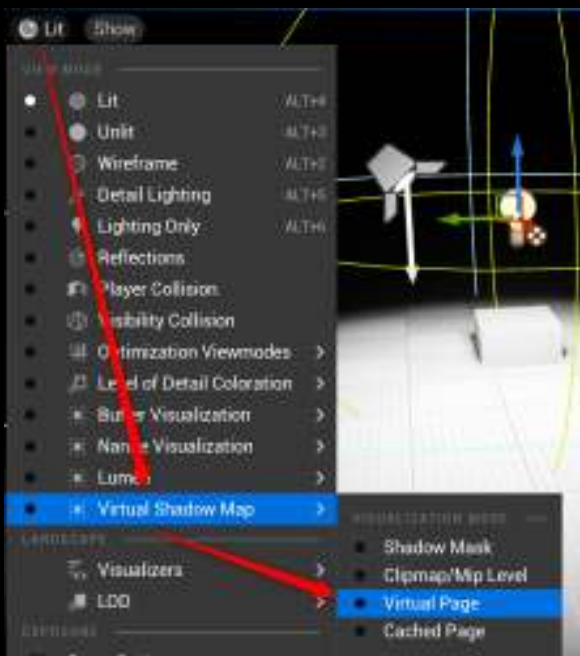
**Shadow Map  
Generation**

+

**Shadow  
Rednering**

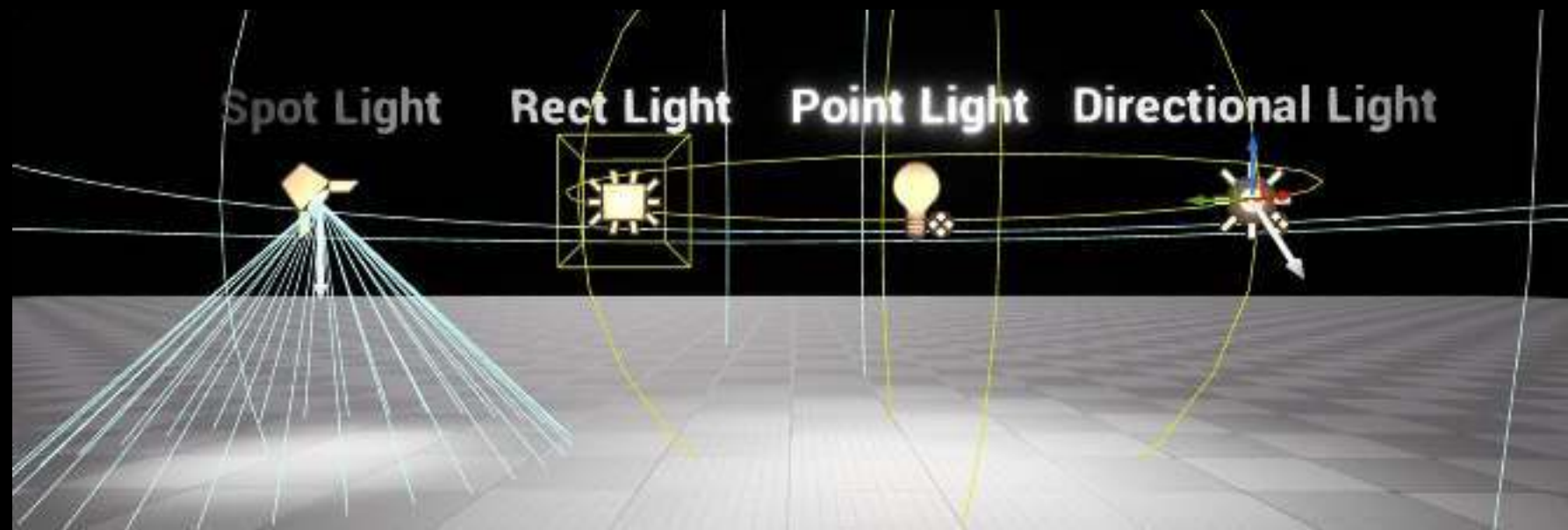
# Virtual Shadow Maps: Feature

Decompose a shadow map into tiles and set the required resolution for each tile  
It enables high-resolution Shadow Map



**Page**  
**128 × 128**

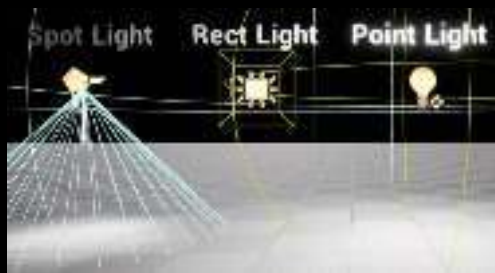
# Various lights in UE



# Light Classification by VSM

## Local Light

- Point Light
- Spot Light
- Rect Light



## Directional Light

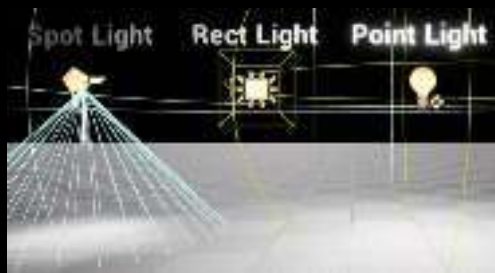
- Directional Light



# Light Classification by VSM

## Local Light

- Point Light
- Spot Light
- Rect Light



## Directional Light

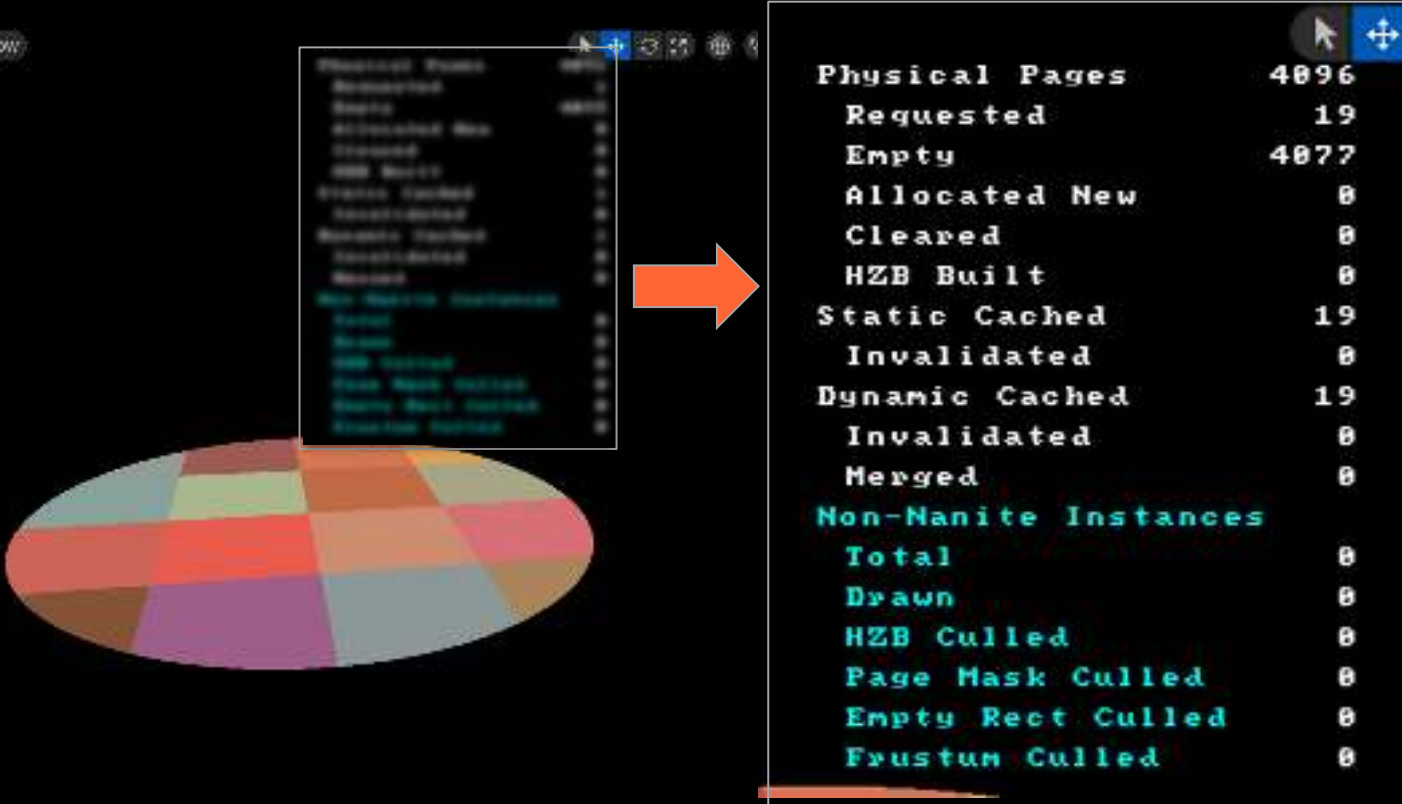
- Directional Light





# r.shadow.virtual.showstats 1

Perspective Virtual Page Show



Physical Pages	4896
Requested	19
Empty	4877
Allocated New	0
Cleared	0
HZB Built	0
Static Cached	19
Invalidated	0
Dynamic Cached	19
Invalidated	0
Merged	0
Non-Manite Instances	
Total	0
Drawn	0
HZB Culled	0
Page Mask Culled	0
Empty Rect Culled	0
Frustum Culled	0

Physical Pages	4896
Requested	19
Empty	4877
Allocated New	0
Cleared	0
HZB Built	0
Static Cached	19
Invalidated	0
Dynamic Cached	19
Invalidated	0
Merged	0
Non-Manite Instances	
Total	0
Drawn	0
HZB Culled	0
Page Mask Culled	0
Empty Rect Culled	0
Frustum Culled	0

# Spot Light - Shadow Depth

Virtual Page Show

```
Physical Pages 4096  
Requested 1  
Empty 4095  
Allocated New 0  
Cleared 0  
HZB Built 0  
Static Cached 1  
Invalidated 0  
  
New Memory Instances  
Total  
Empty  
HZB Built  
Free Mem Cached  
Empty Mem Cached  
Reserved Mem
```

Navigation icons: back, forward, search, etc.

Physical Pages	4096
Requested	1
Empty	4095
Allocated New	0
Cleared	0
HZB Built	0
Static Cached	1
Invalidated	0

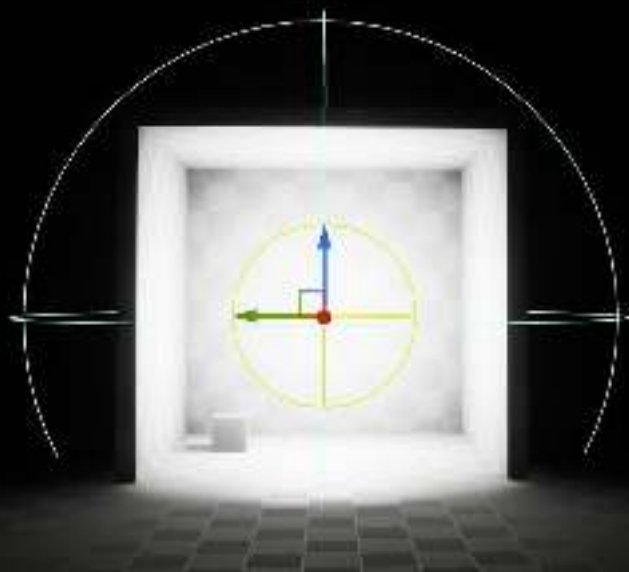


# Spot Light



Physical Pages	4096
Requested	1
Empty	4095
Allocated New	0
Cleared	0
HZB Built	0
Static Cached	1
Invalidated	0
Dynamic Cached	1
Invalidated	0
Merged	0

# Point Light & Rect Light



# Point Light & Rect Light - Cube map

Show



Physical Pages	4096
Requested	6
Empty	4090
Allocated New	0
Cleared	0
HZB Built	0
Static Cached	6
Invalidated	0
Dynamic Cached	6
Invalidated	0
Merged	0

# Point Light & Rect Light

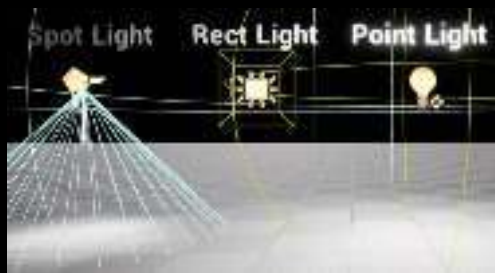


Physical Pages	4896
Requested	6
Empty	4896
Allocated New	8
Cleared	8
HZB Built	8
Static Cached	6
Invalidated	8
Dynamic Cached	6
Invalidated	8
Merged	8

# Light Classification by VSM

## Local Light

- Point Light
- Spot Light
- Rect Light



## Directional Light

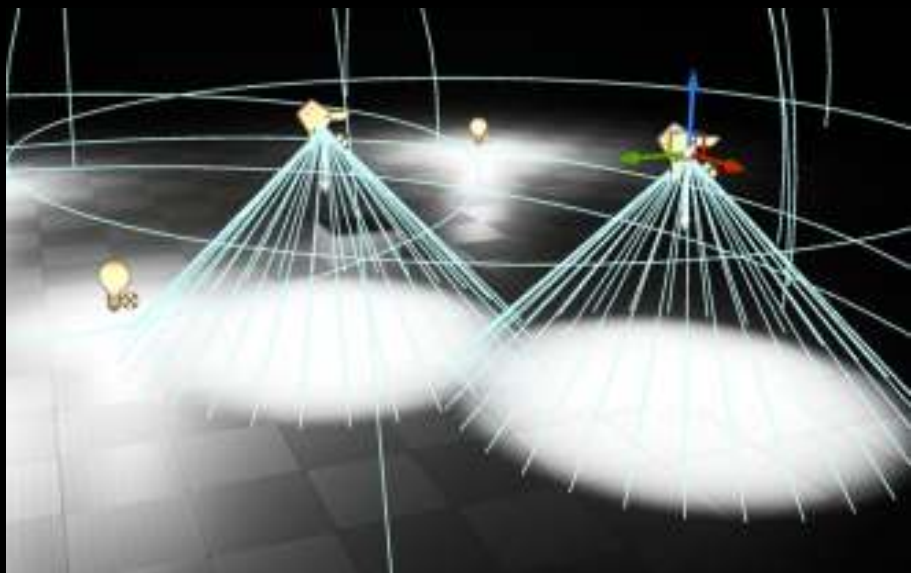
- Directional Light



# Multiple lights

All visible pages (shadow map) of the lights are requested

→ The more lights, the more pages and updates required!



Physical Pages	4096
<b>Requested</b>	<b>162</b>
Empty	3940
Allocated New	6
Cleared	14
HZB Built	7
Static Cached	155
Invalidated	1
Dynamic Cached	155
Invalidated	0
Merged	0

# Two important functions of the VSM

**Cache  
System**

**Page  
Management**

# Two important functions of the VSM

**Cache  
System**

**Page  
Management**

# Cache System

VSM try to cache pages as much as possible if the pages aren't updated across frames  
(Default: *r.Shadow.Virtual.Cache 1*)



Physical Pages	4096
Requested	1
Empty	4095
Allocated New	0
Cleared	0
HZB Built	0
Static Cached	1
Invalidated	0
Dynamic Cached	1
Invalidated	0
Merged	0

# Separate Cache

*r.Shadow.Virtual.Cache.StaticSeparate 1*

Enabled by default

To reduce cache invalidation as much as possible

VSM can have two caches

- for static objects
- for dynamic objects

Physical Pages	4096
Requested	1
Empty	4095
Allocated New	0
Cleared	0
HZB Built	0
Static Cached	1
Invalidated	0
Dynamic Cached	1
Invalidated	0
Merged	0

# Cache Visualization



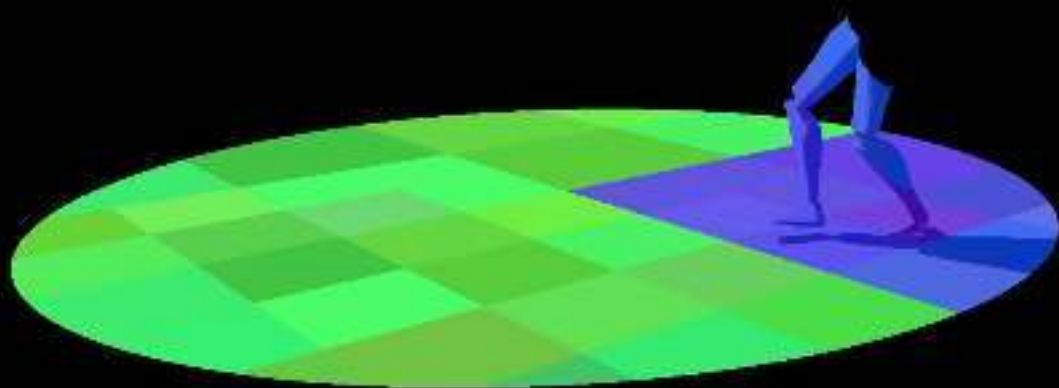
Physical Pages 4036  
Requested 64  
Exits 4032  
Allocated New 0  
Cleared 0  
H2D Duff 0  
Static Cached 64  
Invalidated 0  
Dynamic Cached 64  
Invalidated 0  
Maxed 0

Virtual Shadow Map  
Shadow Mask  
Clipmap/Mip Level  
Virtual Page  
Cached Page

Cached pages are tinted green, uncached are red. Pages where only the static page is cached (dynamic uncached) are blue.

# Cache Visualization

The blue area is the updated Pages (Invalidated)



Physical Pages	4096
Requested	40
Empty	4056
Allocated New	0
Cleared	11
HZB Built	0
Static Cached	40
Invalidated	0
Dynamic Cached	29
Invalidated	11
Merged	0

# Two important functions of the VSM

**Cache  
System**

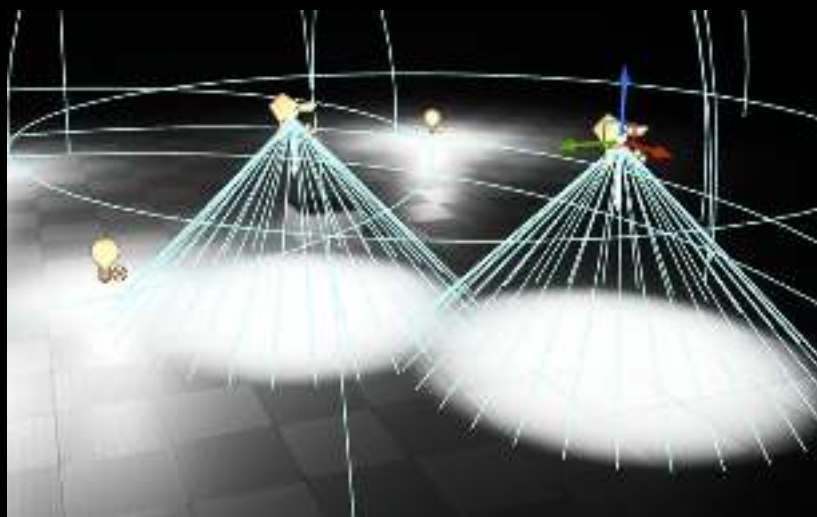
**Page  
Management**

# VSM page management

The more lights you place, the more shadow maps in VSM

*Max Physical Pages* limits the number of shadow maps in the entire VSM

*r.Shadow.Virtual.MaxPhysicalPages* (Default: 4096)

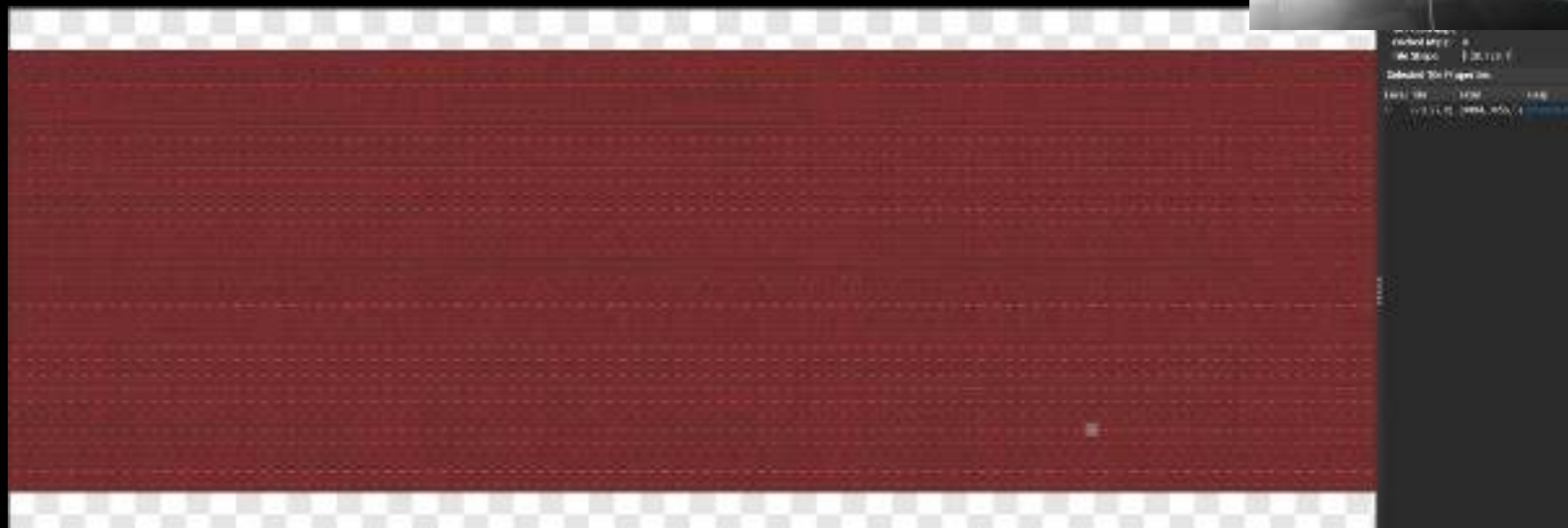
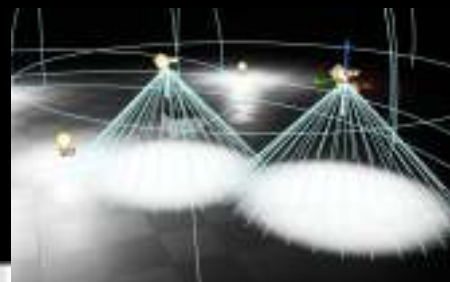


Physical Pages	4096
Requested	162
Empty	3940
Allocated New	6
Cleared	14
HZB Built	7
Static Cached	155
Invalidated	1
Dynamic Cached	155
Invalidated	0
Merged	0

# VSM page management

Shadow maps of all lights are managed in one pool

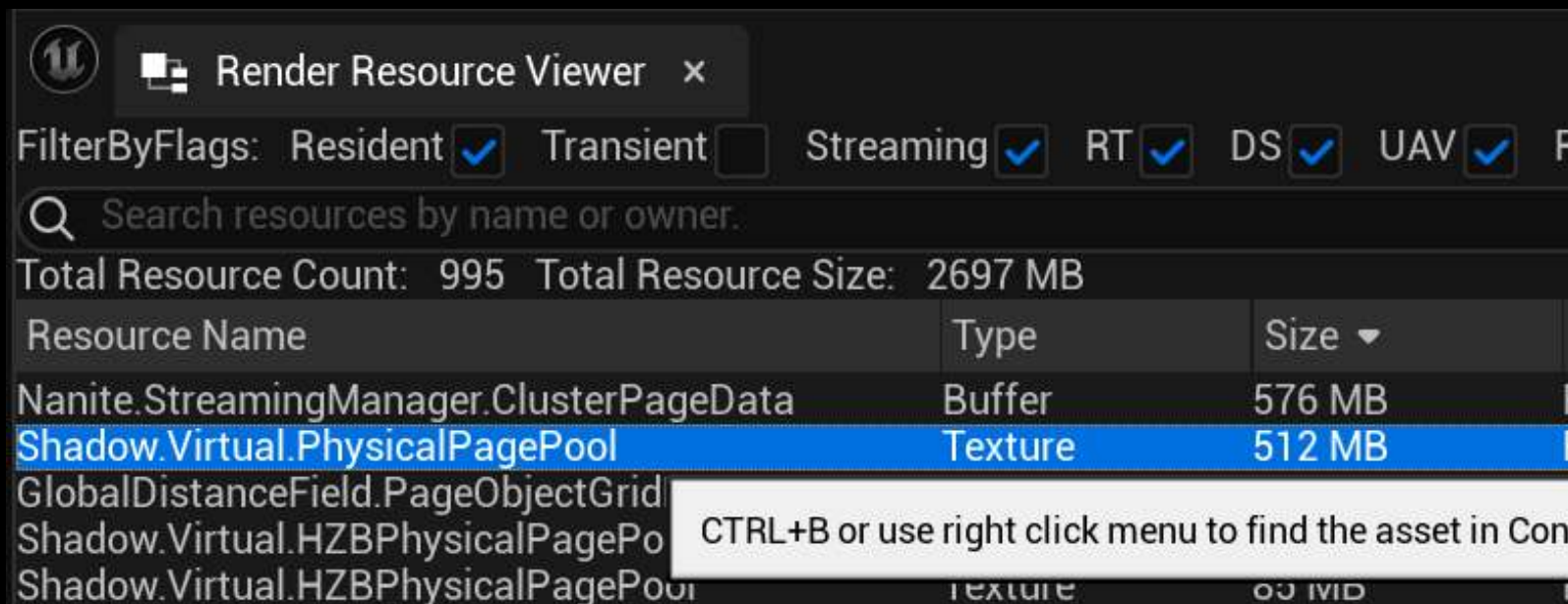
※Separate Static option -> 2 pools



# VSM Memory Consumption

Pool Size = (R32 buffer) \* 128 \* 128 \* (MaxPhysicalPages) \* (Separate Pool)

**512MB** = 4 Bytes \* 128 \* 128 \* 4096 \* 2



The screenshot shows the 'Render Resource Viewer' window in Unreal Engine. At the top, there are filter checkboxes for Resident (checked), Transient (unchecked), Streaming (checked), RT (checked), DS (checked), and UAV (checked). Below the filters is a search bar with the text 'Search resources by name or owner.' The summary shows 'Total Resource Count: 995' and 'Total Resource Size: 2697 MB'. A table lists resources with columns for 'Resource Name', 'Type', and 'Size'. The resource 'Shadow.Virtual.PhysicalPagePool' is highlighted in blue and has a size of 512 MB. A tooltip is visible over the table, containing the text 'CTRL+B or use right click menu to find the asset in Con'.

Resource Name	Type	Size
Nanite.StreamingManager.ClusterPageData	Buffer	576 MB
<b>Shadow.Virtual.PhysicalPagePool</b>	<b>Texture</b>	<b>512 MB</b>
GlobalDistanceField.PageObjectGrid		
Shadow.Virtual.HZBPhysicalPagePo		
Shadow.Virtual.HZBPhysicalPagePoo	Texture	63 MB

# To Reduce VSM Memory Consumption

## 1. Reduce Max Physical Pages

※ But put lots lights that exceed the page number will break shadows

## 2. If memory size is a priority

**Separate Static option should be turned off**

(Memory usage cut in half)

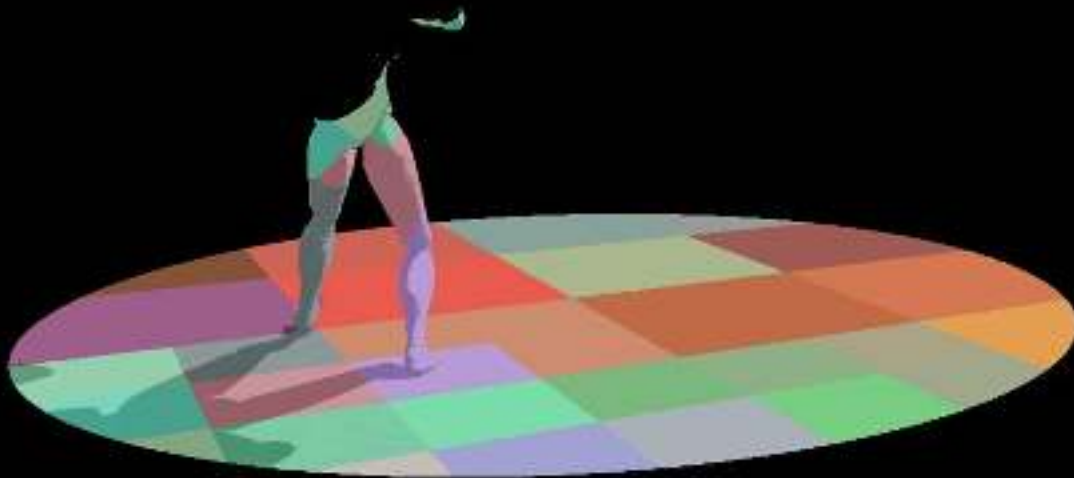
## 3. **bias adjustment** by console command



# ResolutionLodBiasLocal

*r.Shadow.Virtual.ResolutionLodBiasLocal*

- Set to 1 or less for finer detail, 1 or more for rougher detail



Physical Pages	4896
Requested	32
Empty	4864
Allocated New	0
Cleared	17
HZB Built	0
Static Cached	32
Invalidated	0
Dynamic Cached	15
Invalidated	17
Merged	0

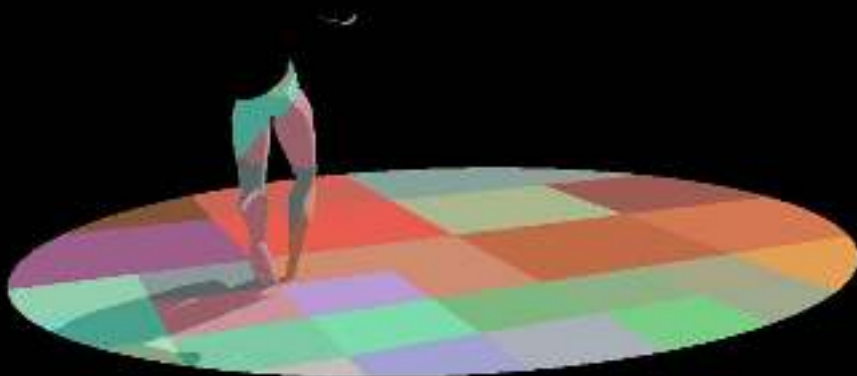
# ResolutionLodBiasLocalMoving

## *r.Shadow.Virtual.ResolutionLodBiasLocalMoving*

Apply bias ONLY when lights are moving

This bias can reduce the number of pages updated

Because all cache of the light must be invalidated when the light is moving.

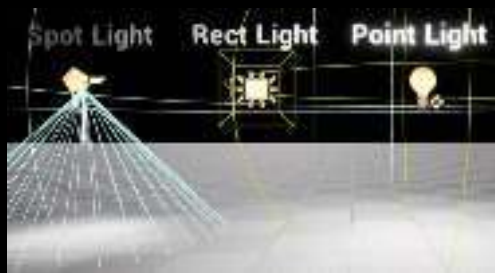


Physical Pages	4096
Requested	32
Empty	4064
Allocated New	0
Cleared	15
HZB Built	0
Static Cached	32
Invalidated	0
Dynamic Cached	17
Invalidated	15
Merged	0

# Light Classification by VSM

## Local Light

- Point Light
- Spot Light
- Rect Light



## Directional Light

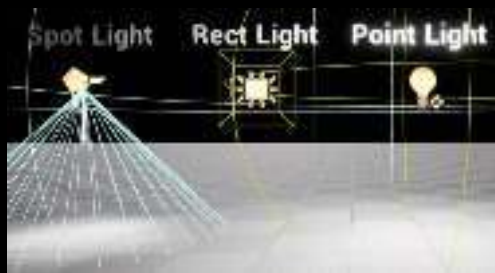
- Directional Light



# Light Classification by VSM

## Local Light

- Point Light
- Spot Light
- Rect Light

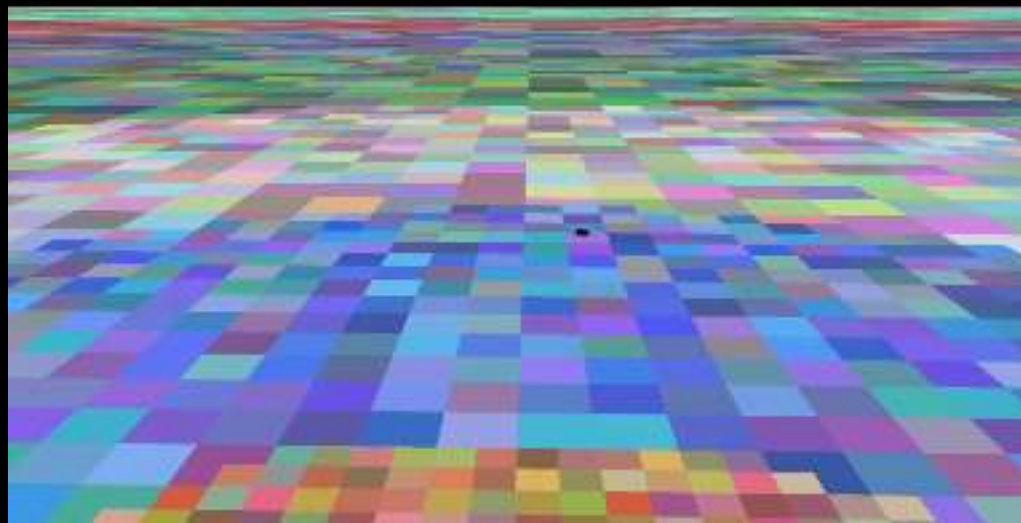


## Directional Light

- Directional Light



# Directional Light



Physical Pages	4096
Requested	2098
Empty	2187
Allocated New	189
Cleared	378
HZB Built	189
Static Cached	1909
Invalidated	0
Dynamic Cached	1909
Invalidated	0
Merged	0

# Directional Light

Lighting the entire screen

-> the closer a page is to the camera,  
it needs to be more detailed



Basically, the same mechanism as Local Light,

As an additional function,

there is a clip map that changes LOD according to the distance from the camera.

CVars

*r.Shadow.Virtual.ResolutionLodBiasDirectional*

*r.Shadow.Virtual.ResolutionLodBiasDirectionalMoving*

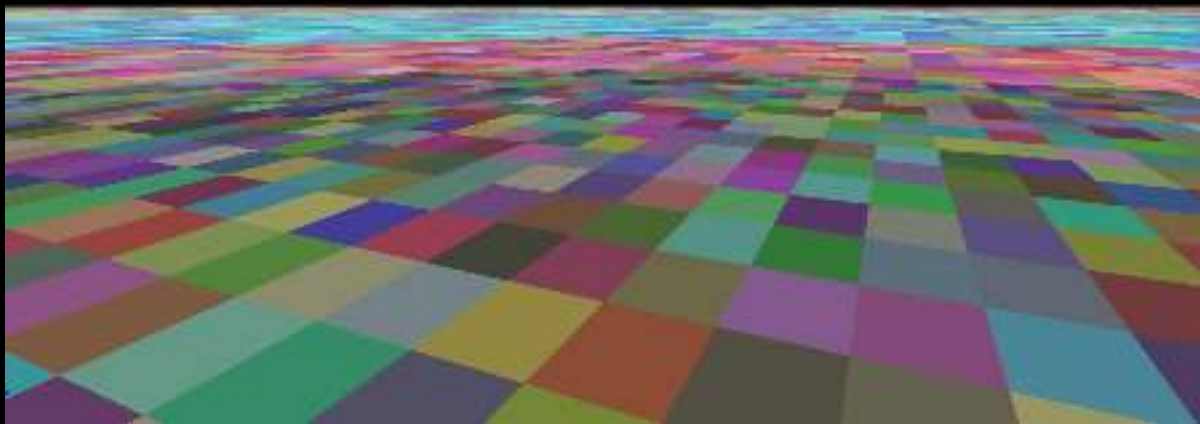
# Directional Light – Adjusting Clip Map

## *r.Shadow.Virtual.Clipmap.FirstLevel*

Use the *FirstLevel* page size for levels below the *FirstLevel*  
(Default: 6) - 64 cm ( $2^6$  cm)

## *r.Shadow.Virtual.Clipmap.LastLevel*

Do not render pages above the *LastLevel* (Default: 22) - about 40 km ( $2^{22}$  cm)



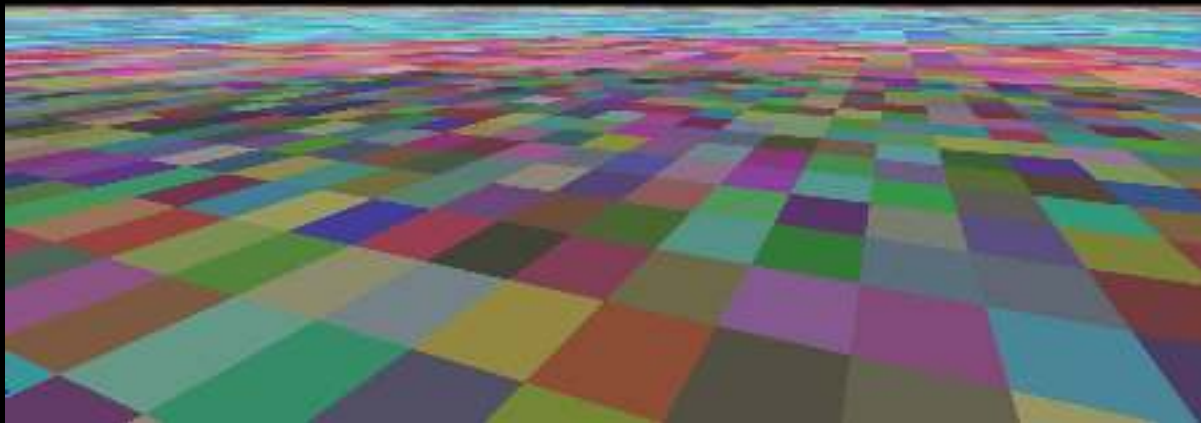
# Directional Light – Adjusting Clip Map

## *r.Shadow.Virtual.Clipmap.FirstLevel*

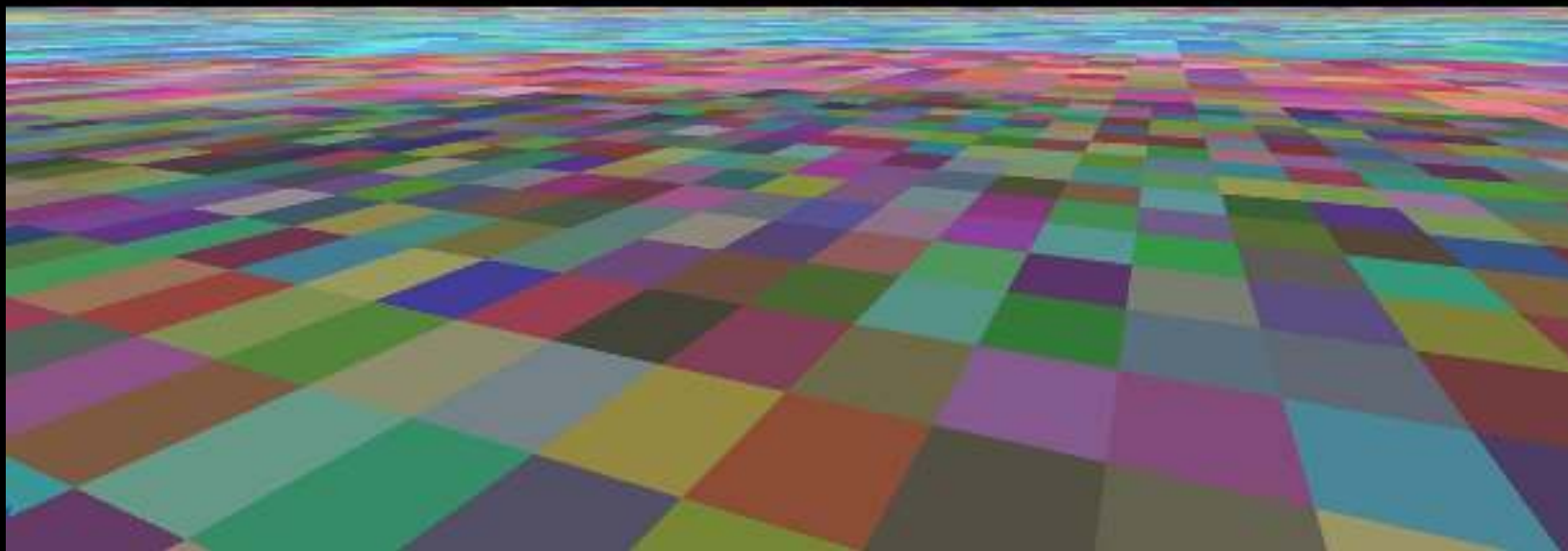
Use the *FirstLevel* page size for levels below the *FirstLevel*  
(Default: 6) - 64 cm ( $2^6$  cm)

## *r.Shadow.Virtual.Clipmap.LastLevel*

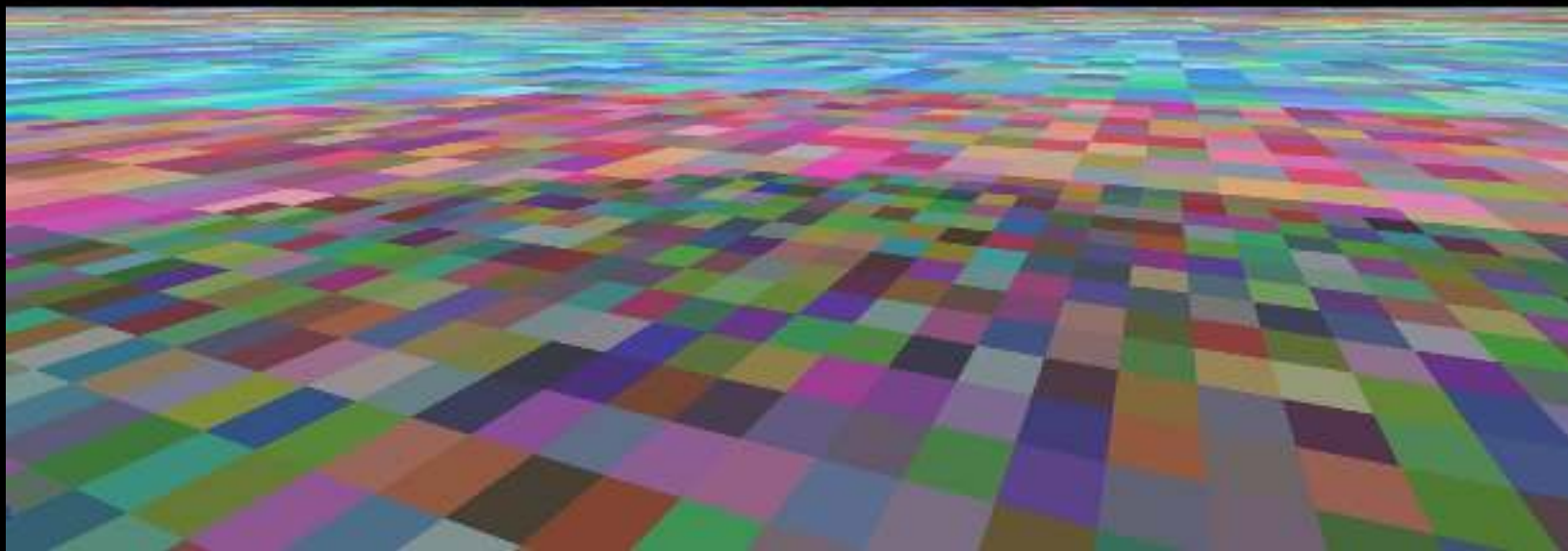
Do not draw pages above the *LastLevel* (Default: 22) - about 40 km ( $2^{22}$  cm)



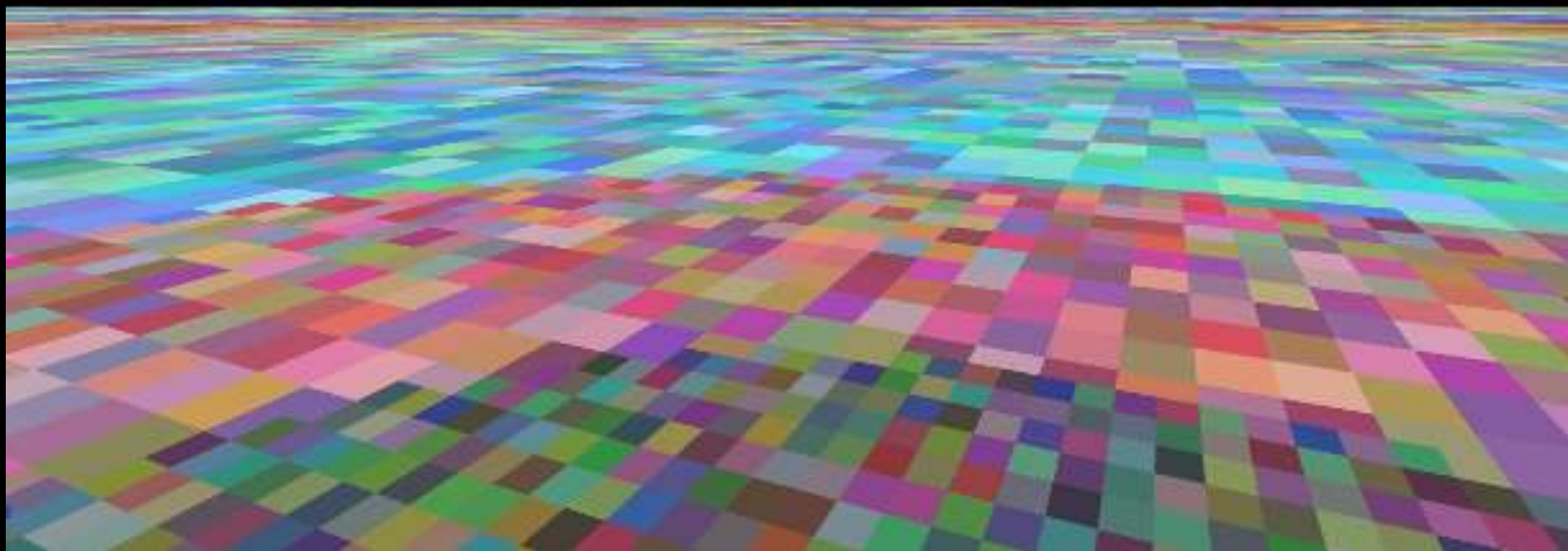
# r.Shadow.Virtual.Clipmap.FirstLevel 6



# r.Shadow.Virtual.Clipmap.FirstLevel 5



# r.Shadow.Virtual.Clipmap.FirstLevel 4



# Directional Light – Adjusting Clip Map

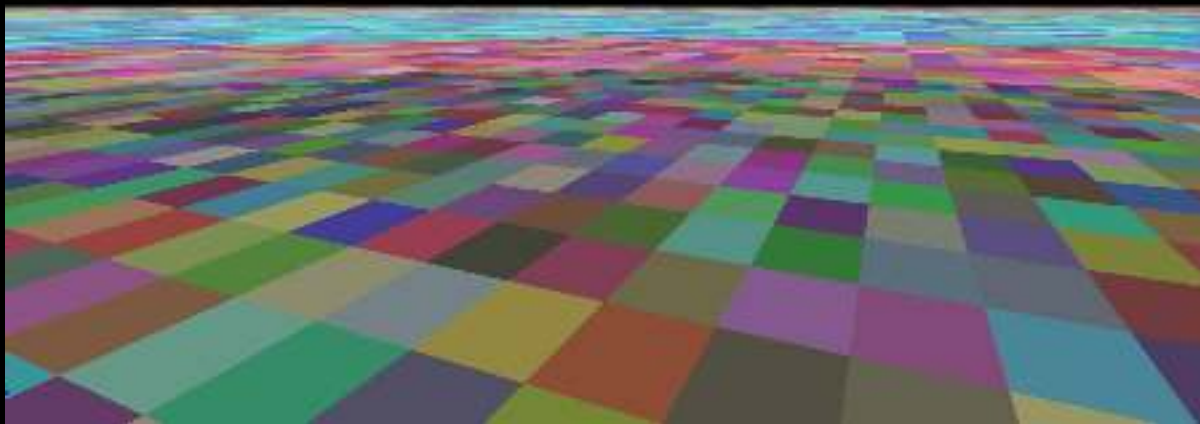
*r.Shadow.Virtual.Clipmap.FirstLevel*

Use the *FirstLevel* page size for levels below the *FirstLevel*

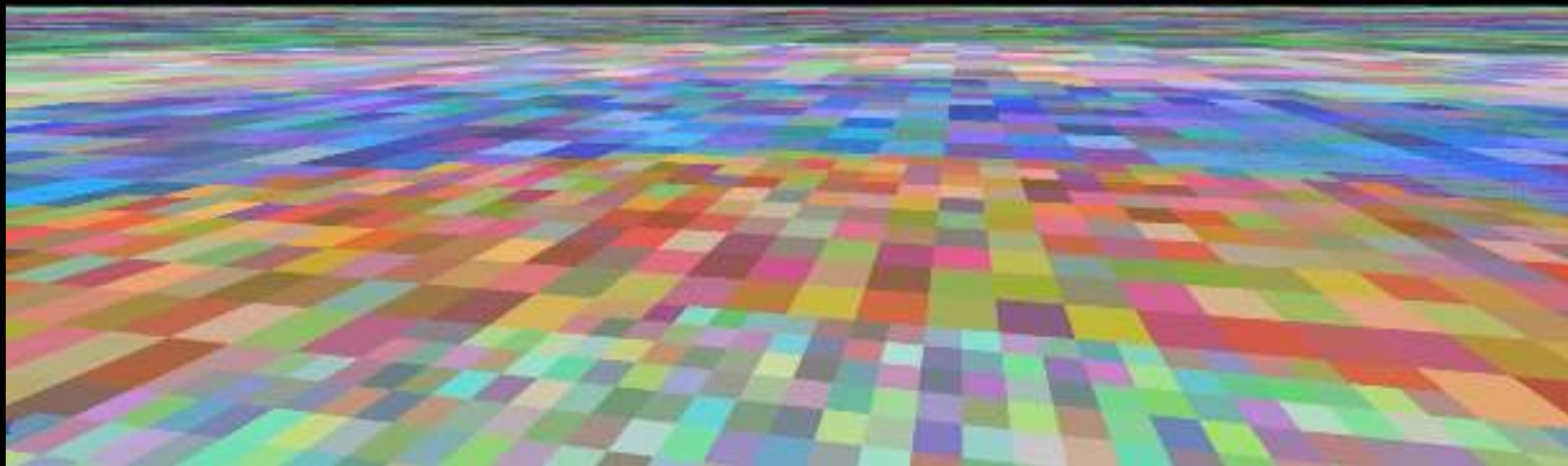
(Default: 6) - 64 cm ( $2^6$  cm)

*r.Shadow.Virtual.Clipmap.LastLevel*

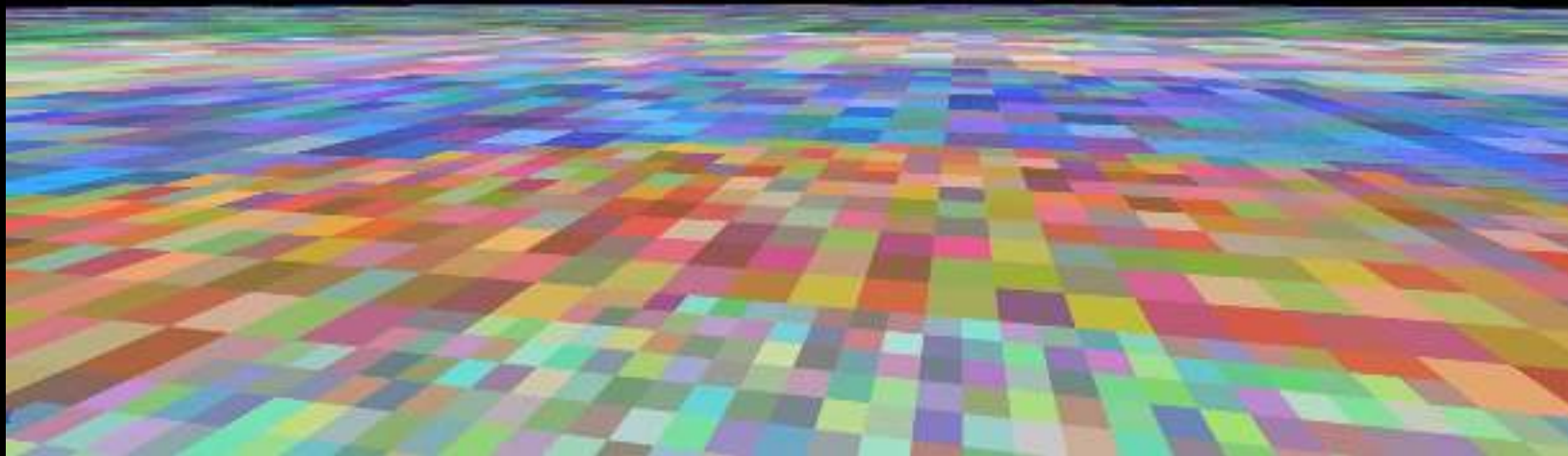
Do not draw pages above the *LastLevel* (Default: 22) - about 40 km ( $2^{22}$  cm)



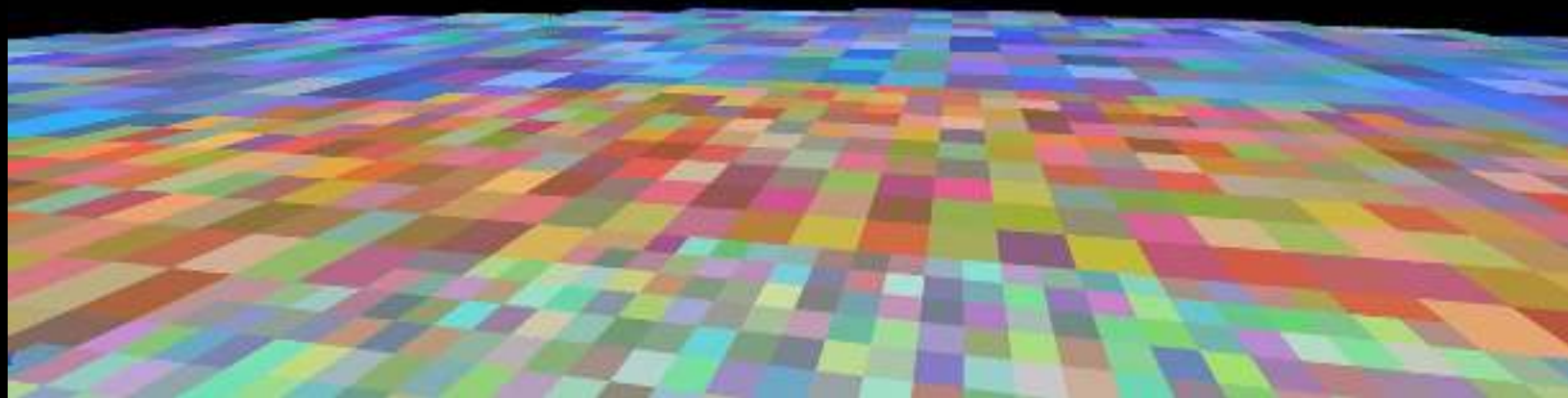
# r.Shadow.Virtual.Clipmap.LastLevel 22



# r.Shadow.Virtual.Clipmap.LastLevel 13



# r.Shadow.Virtual.Clipmap.LastLevel 10



# Directional Light setting for VSM

- Adjust LOD with *LodBias*

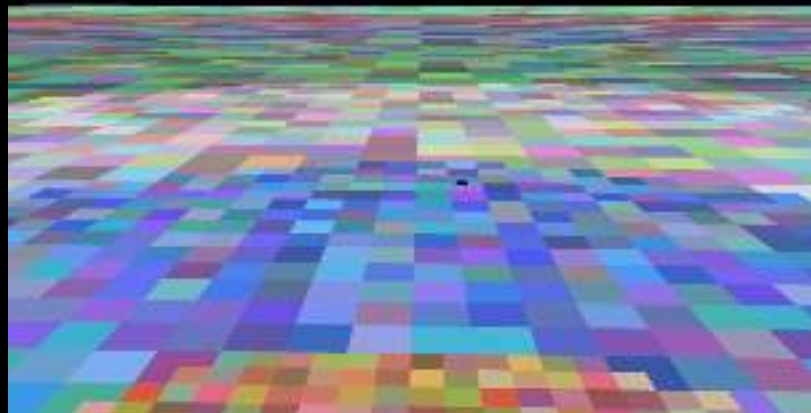
*r.Shadow.Virtual.ResolutionLodBiasDirectional*

*r.Shadow.Virtual.ResolutionLodBiasDirectionalMoving*

- Adjust range with *ClipmapLevel*

*r.Shadow.Virtual.Clipmap.FirstLevel*

*r.Shadow.Virtual.Clipmap.LastLevel*



# VSM Algorithm

**Shadow Map  
Generation**

+

**Shadow  
Rendering**

# VSM Algorithm

**Shadow Map  
Generation**

+

**Shadow  
Rendering**

# Shadow Map Ray Tracing (SMRT)

VSM does ray tracing on the shadow map for high quality shadow rendering. Shadow rays are distributed with the following parameters of the light

- Local Light: **Source Radius**
- Directional Light: **Source Angle**



# SMRT - RayCount

*SMRT.RayCount* directly affect shadow quality and performance

※Default:8 (If set to 0, single-sample = hard shadow)

*r.Shadow.Virtual.SMRT.RayCountLocal*

*r.Shadow.Virtual.SMRT.RayCountDirectional*

Test scene->

Source Radius is fixed to 1.5

Changing *RayCountDirectional*



# SMRT - RayCountLocal

More local lights has a greater impact

*r.Shadow.Virtual.SMRT.RayCountLocal*

Test scene ->

Meshes and Lights placed with PCG

Source Radius = 100

Adjust *RayCountLocal*



# VSM: Optmization

Cache as much as possible → Reduce Objects to invalidate cache

Consider using only Contact Shadow in situations where the cache cannot be utilized in large areas such as grasslands



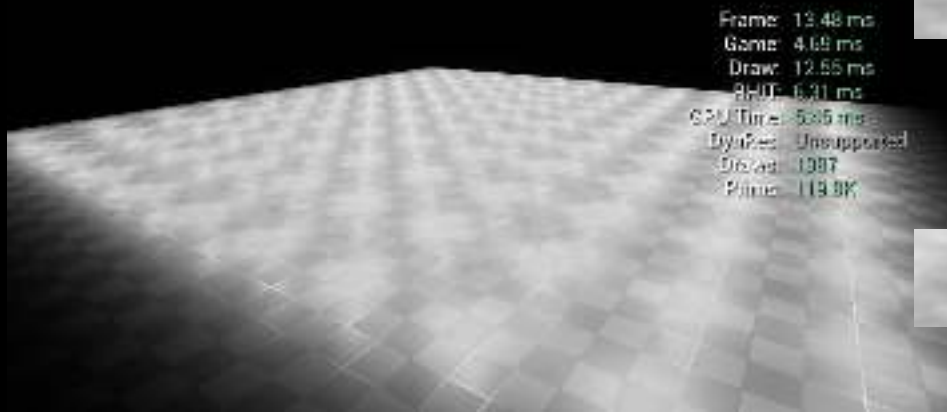
Set Contact Shadow Length  
in Directional Light settings



In the Static Mesh settings  
Turn off Dynamic Shadow  
Enable Contact Shadow only

# VSM: Optmization

- **Distant Lights** and **Onepass Projection** are enabled by default



GPU Time: 6.46 ms



**Enabled both**

GPU Time: 4.32 ms

# of point lights:100  
With RTX4090

# Distant Lights and Onepass Projection

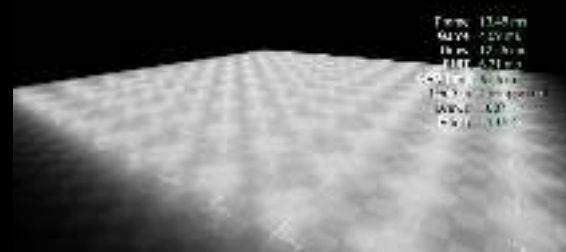
## • Distant Lights

the function to reduce the frequency of updating distant lights that have a small area of influence

*r.shadow.virtual.distantlightmode*

Maximum number of distant lights to update each frame

*r.Shadow.Virtual.MaxDistantUpdatePerFrame 1*



## • Onepass Projection

Projects all local light virtual shadow maps in a single pass

*r.Shadow.Virtual.OnePassProjection 1*

※Some restrictions are described in the official VSM document

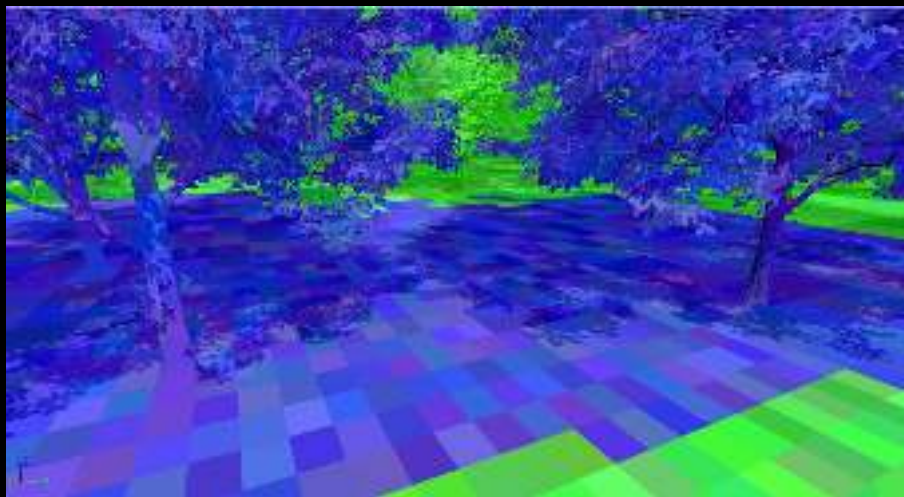
<https://dev.epicgames.com/documentation/en-us/unreal-engine/virtual-shadow-maps-in-unreal-engine>

# VSM: Optimization - 5.4~

Dynamically change page resolution to match Pool Size

*r.Shadow.Virtual.DynamicRes.MaxPagePoolLoadFactor* Default: 0.85

Adjust Page's LOD when exceeding 85% of pool size



Example: Limit PagePoolSize to 4096->100 (memory reduced from 512MB to 16MB)

# Other Tips

- **Some debug commands**
- **Render Resource View**
- **RDG Insights**

# Shader Print

Some Stats/Debug info are drawn by *ShaderPrint*  
(Compute Shader converts various resources into data for display)

*r.Nanite.ShowStats*

*r.Shadow.Virtual.ShowStats*

etc.

Some settings to adjust the ShaderPrint view

*r.ShaderPrint.FontSize*

*r.ShaderPrint.FontSpacingX*

*r.ShaderPrint.FontSpacingY*



# Shader Print

Need to be implemented on CPP side and CS Shader respectively

**RadianceCacheUpdateStats** can be a good simple example

```
MaxBucket 13
MaxTracesFromMaxUpdateBucket 681
Out
TraceCost 1289
TraceCostFromMaxUpdateBucket 2827
Probes 513
MaxProbesInAtlas 16384
ProbesInAtlas 3491
Allocator 12079
FreeList 9483
```

*r.Lumen.ScreenProbeGather.RadianceCache.Stats*

```
Newline(Context);
Print(Context, TEXT("MaxBucket "));
Print(Context, MaxUpdateBucketIndex);

Newline(Context);
Print(Context, TEXT("MaxTracesFromMaxUpdateBucket "));
Print(Context, LocalMaxTracesFromMaxUpdateBucket);

Newline(Context);
Print(Context, TEXT("Out "));

Newline(Context);
Print(Context, TEXT(" TraceCost "));
Print(Context, ProbesToUpdateTraceCost[0]);
```

*LumenRadianceCacheDebug.usf*

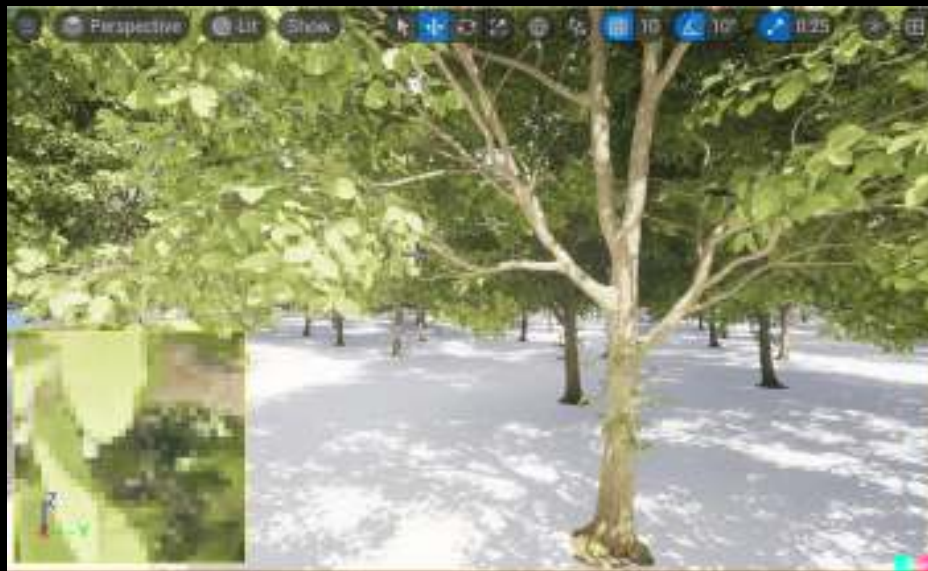
If you want to check other examples in UE,  
You can see some implementations by searching “../ShaderPrint.ush”

# r.ShaderPrint.Zoom

You can zoom in on the position of the mouse cursor in the viewport.

*r.ShaderPrint 1* and  
*r.ShaderPrint.Zoom 1*

- Change the debug view location  
*r.ShaderPrint.Zoom.Corner*
- Number of pixels to be zoomed  
*r.ShaderPrint.Zoom.Pixel*
- Zooming magnification  
*r.ShaderPrint.Zoom.Factor*





# Vis - Pool

*Vis Pool* : Show resource list only

*Vis Pool BYNAME* : Sort by name

*Vis Pool BYSIZE* : Sort by size

```
Cmd: Vis pool BYSIZE
LogConsoleResponse: 5 - 524288 KB (2D[2] 16384x4096 PF_R32_UINT UAV) Shadow.Virtual.PhysicalPagePool
LogConsoleResponse: 113 - 524288 KB (2D[2] 16384x4096 PF_R32_UINT UAV) Shadow.Virtual.PhysicalPagePool
LogConsoleResponse: 4 - 87424 KB (2D 8192x2048 PF_R32_FLOAT UAV) Shadow.Virtual.HZBPhysicalPagePool
LogConsoleResponse: 114 - 87424 KB (2D 8192x2048 PF_R32_FLOAT UAV) Shadow.Virtual.HZBPhysicalPagePool
LogConsoleResponse: 33 - 87424 KB (2D 8192x2048 PF_R32_FLOAT UAV) Shadow.Virtual.HZBPhysicalPagePool
LogConsoleResponse: 16 - 87424 KB (2D 8192x2048 PF_R32_FLOAT UAV) Shadow.Virtual.HZBPhysicalPagePool
LogConsoleResponse: 131 - 73984 KB (2D 4352x4352 PF_FloatRGB UAV) Lumen.RadianceCache.FinalRadianceAtlas
LogConsoleResponse: 10 - 73984 KB (2D 4352x4352 PF_FloatRGB UAV) Lumen.RadianceCache.FinalRadianceAtlas
LogConsoleResponse: 145 - 65536 KB (2D 4096x4096 PF_B8G8R8A8_RT) VisualizeTexture
LogConsoleResponse: 69 - 65536 KB (2D 4096x4096 PF_FloatR11G11B10_RT UAV) Lumen.SceneDirectLighting
LogConsoleResponse: 151 - 65536 KB (2D 4096x4096 PF_FloatR11G11B10_RT UAV) Lumen.SceneDirectLighting
LogConsoleResponse: 154 - 65536 KB (2D 4096x4096 PF_FloatR11G11B10_RT UAV) Lumen.SceneFinalLighting
LogConsoleResponse: 57 - 65536 KB (2D 4096x4096 PF_FloatR11G11B10_RT UAV) Lumen.SceneFinalLighting
LogConsoleResponse: 70 - 65536 KB (2D 4096x4096 PF_FloatR11G11B10_RT UAV) Lumen.SceneIndirectLighting
LogConsoleResponse: 152 - 65536 KB (2D 4096x4096 PF_FloatR11G11B10_RT UAV) Lumen.SceneIndirectLighting
LogConsoleResponse: 9 - 65536 KB (2D 4096x4096 PF_FloatRGB UAV) Lumen.RadianceCache.RadianceProbeAtlasTextureSource
LogConsoleResponse: 130 - 65536 KB (2D 4096x4096 PF_FloatRGB UAV) Lumen.RadianceCache.RadianceProbeAtlasTextureSource
LogConsoleResponse: 157 - 65536 KB (2D 4096x4096 PF_FloatRGB UAV) Lumen.Radiosity.TraceRadianceAtlas
LogConsoleResponse: 54 - 65536 KB (2D 4096x4096 PF_FloatRGB UAV) Lumen.Radiosity.TraceRadianceAtlas
LogConsoleResponse: 116 - 57344 KB (3D 1024x1024x56 PF_R8 UAV) GlobalDistanceField.PageAtlas
LogConsoleResponse: 35 - 57344 KB (3D 1024x1024x56 PF_R8 UAV) GlobalDistanceField.PageAtlas
```

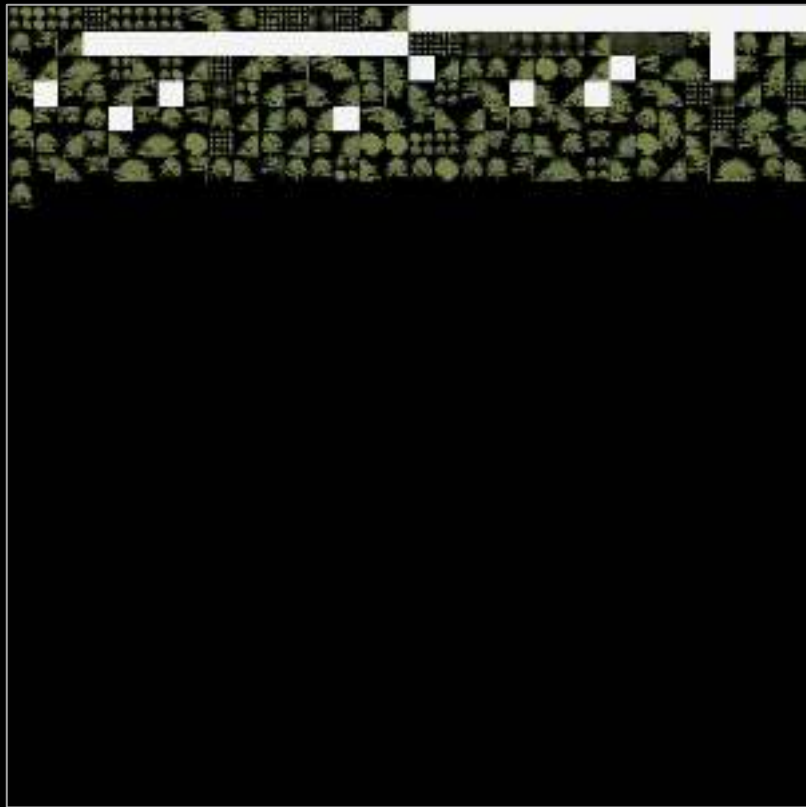
# Vis - BMP

- Can output images in original size with BMP command

*Vis [ResourceName] BMP*

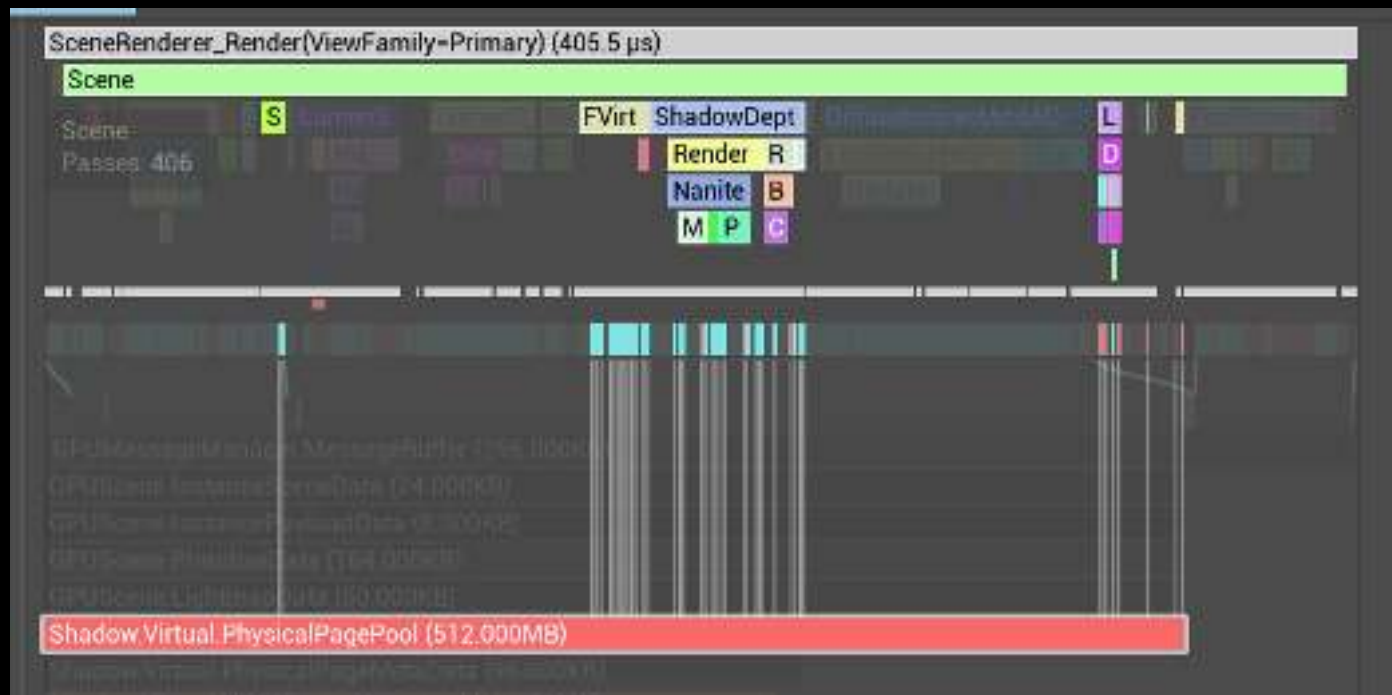
- Example

*vis Lumen.SceneAlbedo BMP*





# RDG Insights

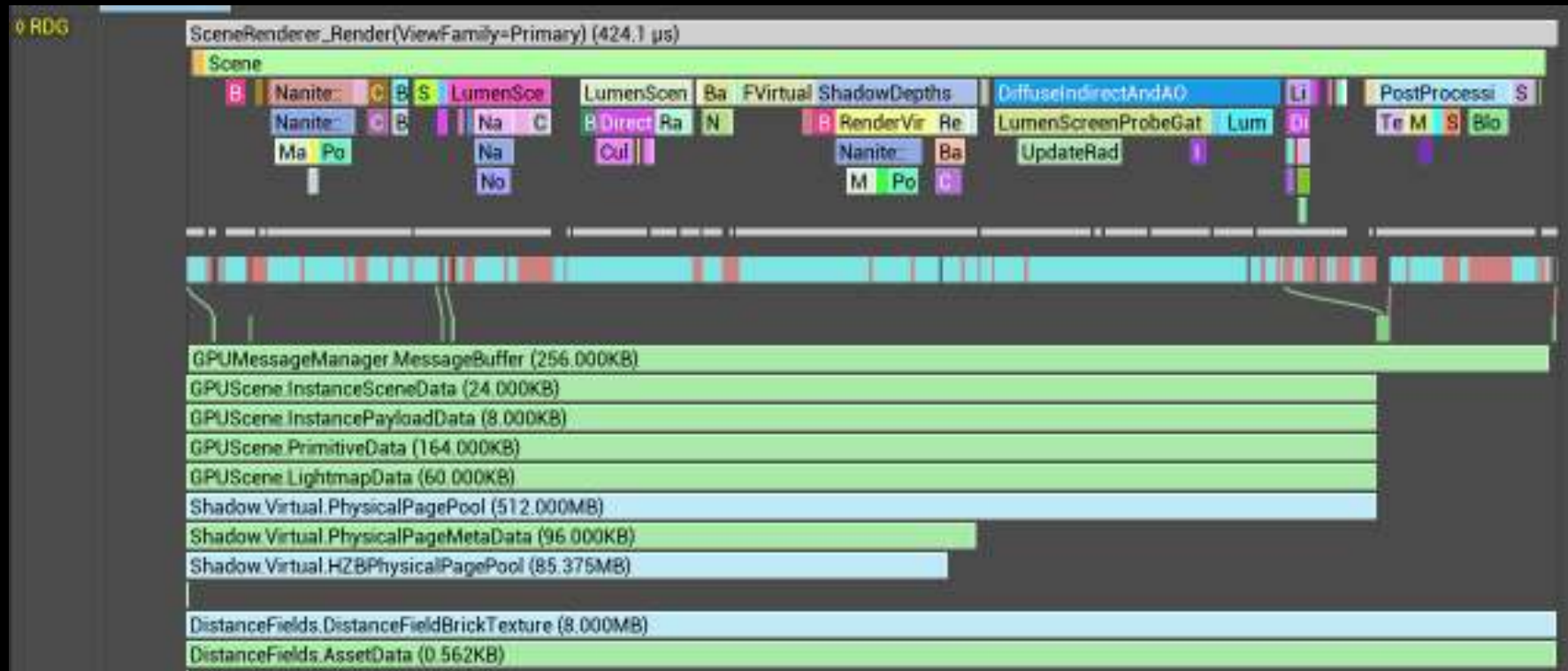




# RDG Insights - RDG is at the bottom of the trace



# RDG Insights - 3 categories



# RDG Insights - 3 categories





# RDG Insights - Pass

Blue is Compute 、 Red is Raster、  
Green at the bottom of the row is Async Compute



# RDG Insights - 3 categories

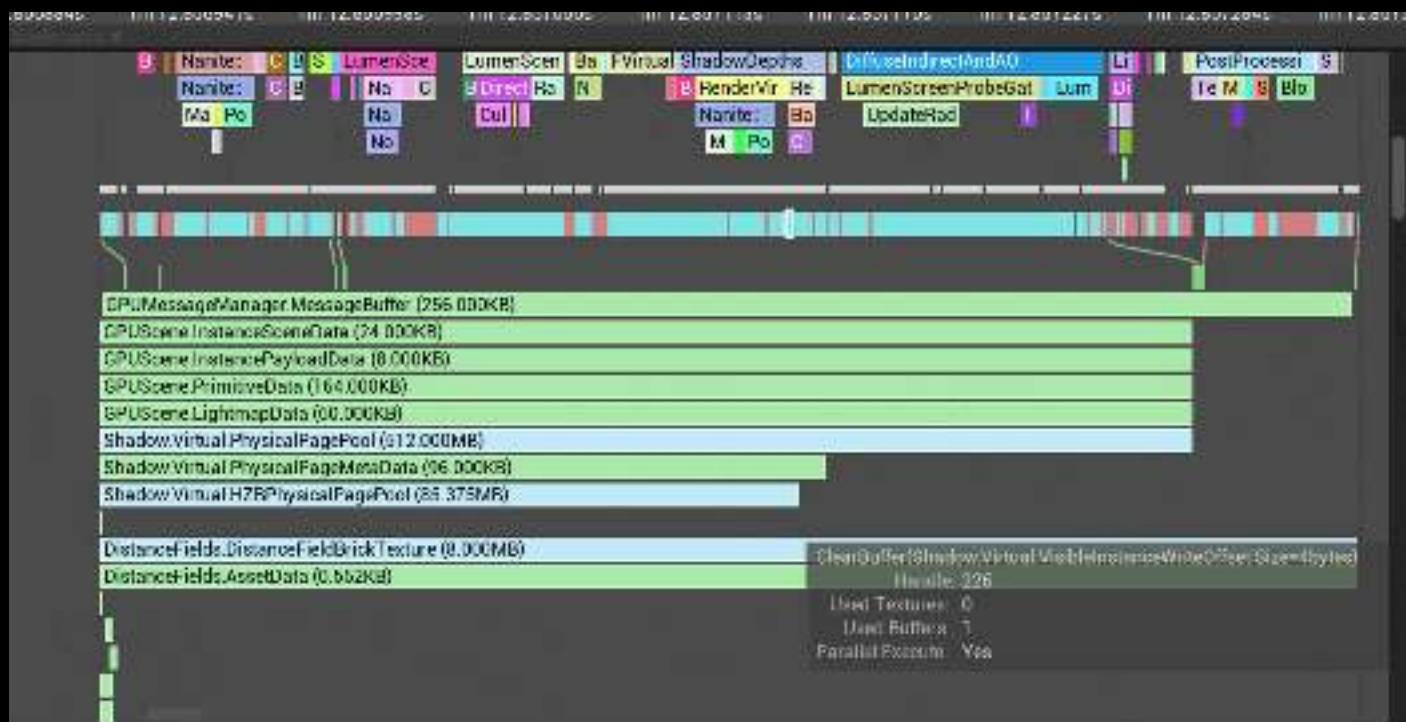
The screenshot shows a table with columns for Name, Location, Size, and other details. The table lists various resource categories and their sizes.

Name	Location	Size	Virtual	Physical	Pool	Pool
GPUMessageManager MessageBuffer		256.000KB				
GPUScene InstanceSceneData		24.000KB				
GPUScene InstancePayloadData		8.000KB				
GPUScene PrimitiveData		164.000KB				
GPUScene LightmapData		60.000KB				
Shadow Virtual PhysicalPagePool		512.000KB				
Shadow Virtual PhysicalPageMetaData		96.000KB				
Shadow Virtual HZBPhysicalPagePool		85.375MB				
DistanceFields DistanceFieldBrickTexture		8.000MB				
DistanceFields AssetData		0.562KB				

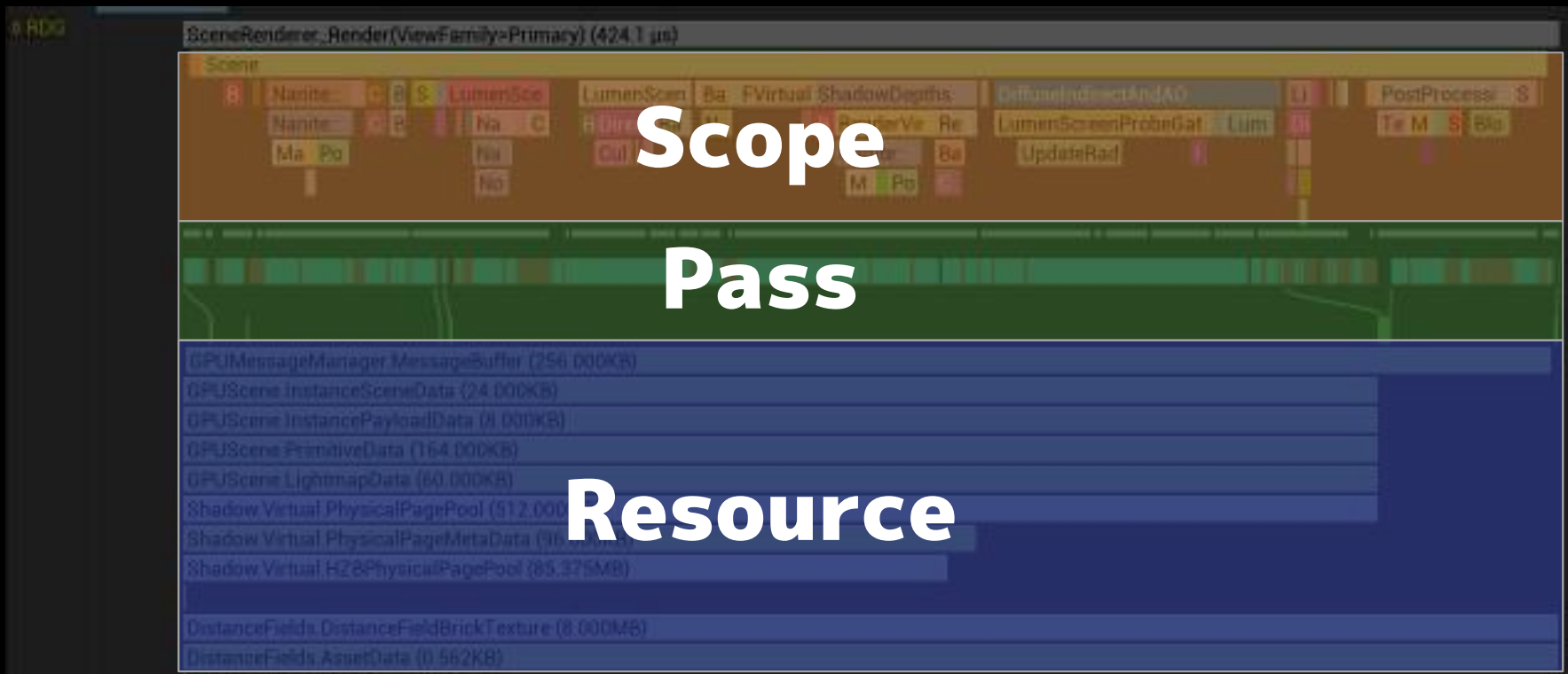


Resource

# RDG Insights - Resource

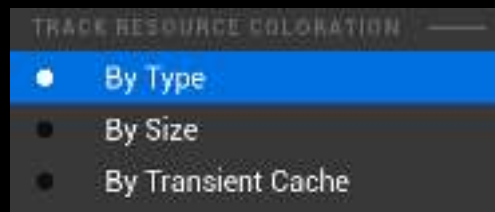
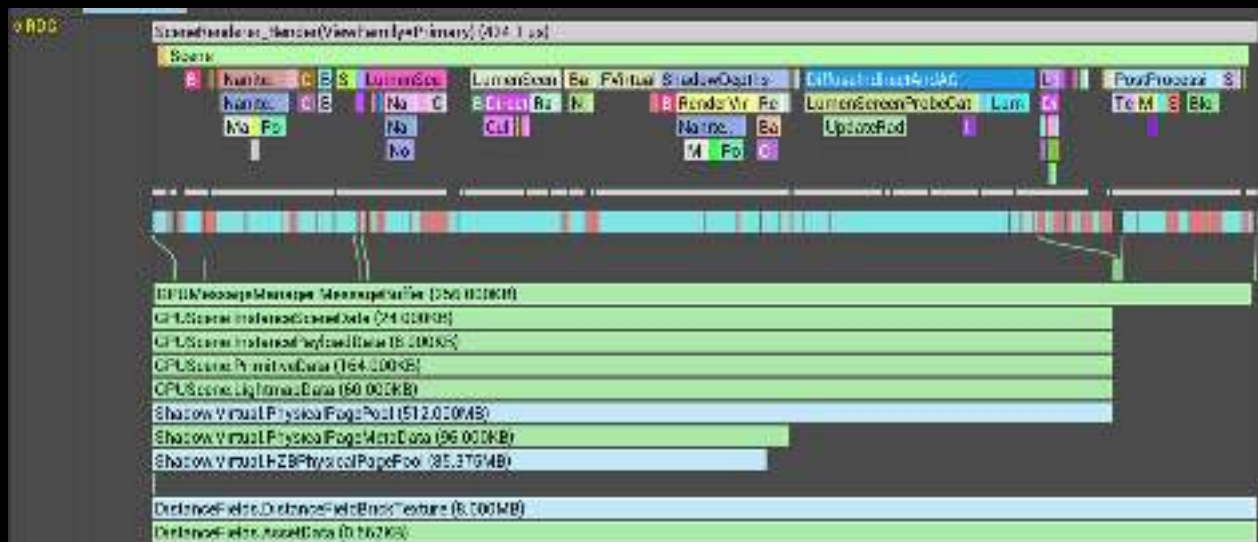


# RDG Insights - 3 categories



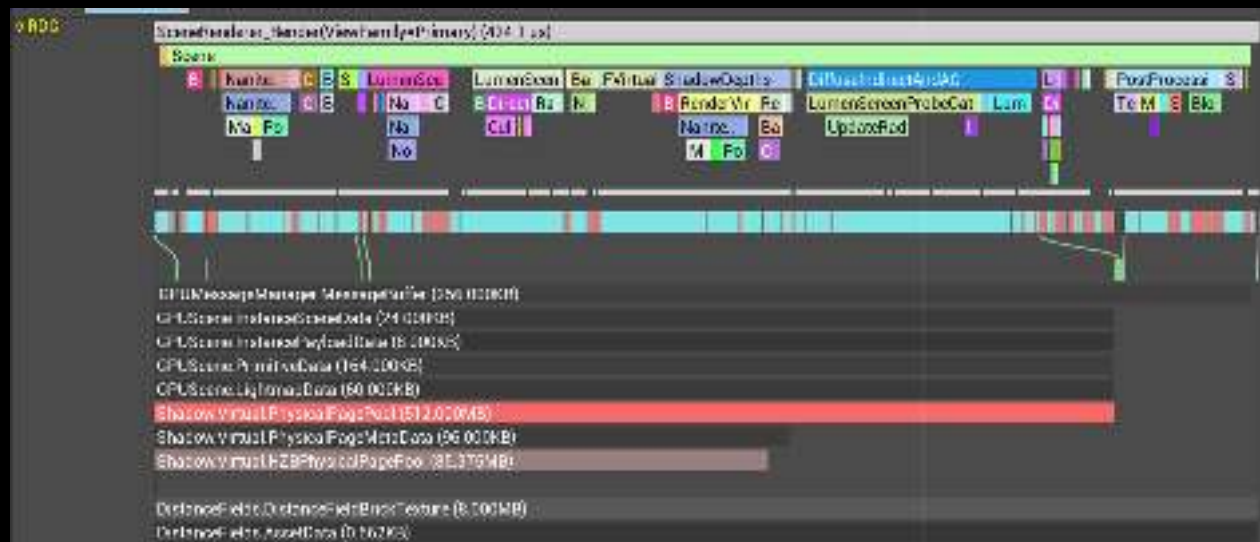
# RDG Insights – Resource color change

- The default setting is “By Type”  
Green is Buffer, Blue is Texture

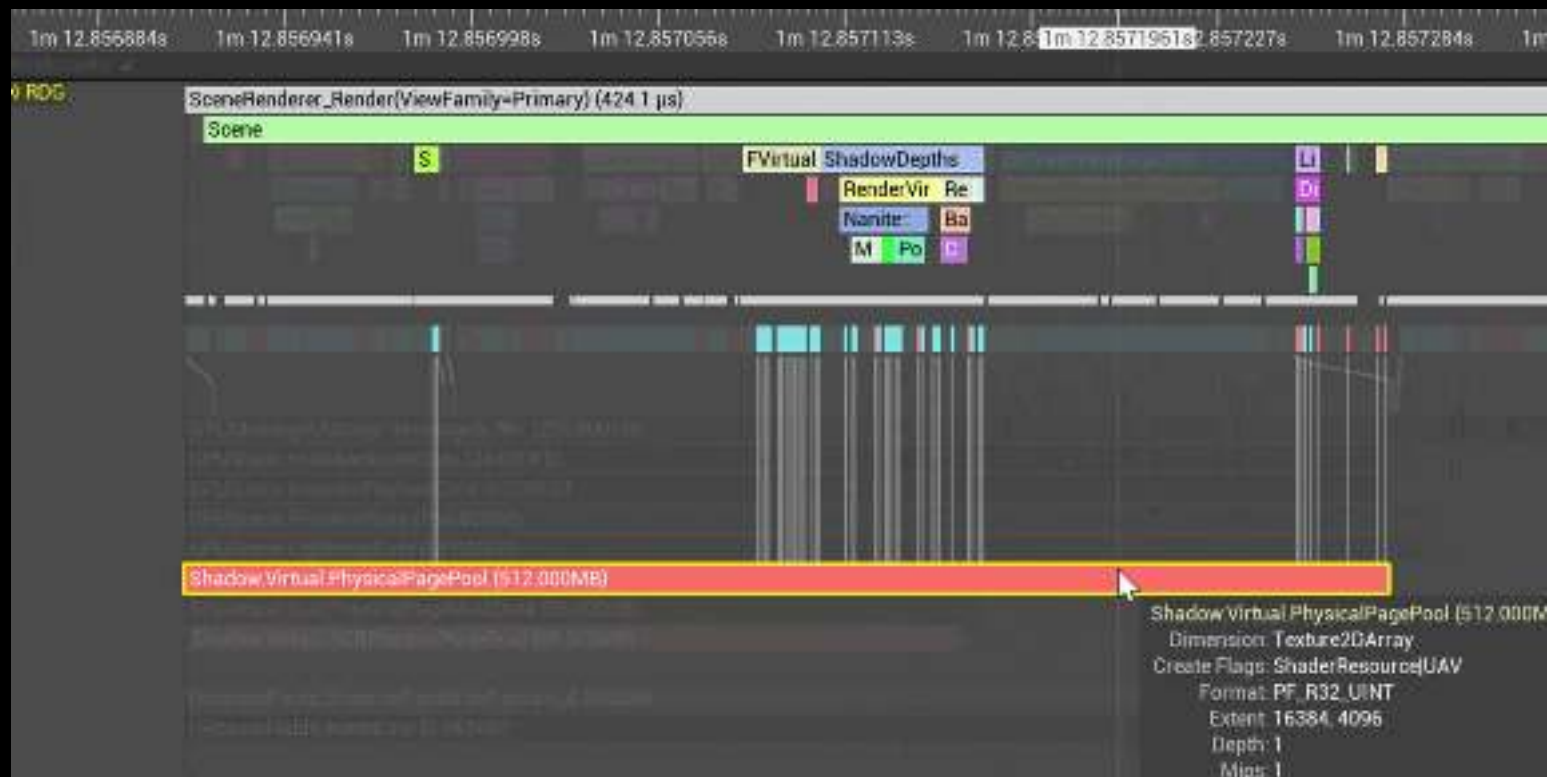


# RDG Insights – Coloration by Size

- Color can be changed on the right-click menu
- by Size = Larger resources are highlighted in red



# RDG Insights



A portrait of a person with short dark hair, wearing a dark blazer over a dark t-shirt with some text on it. The person is looking slightly to the right of the camera. The background is a dark, solid color.

# Agneda

Nanite

Lumen

VSM

Other Development Tips



**Thank you!**



