Unreal Engine
Best practices

MENDELU
Faculty
of Business
and Economics

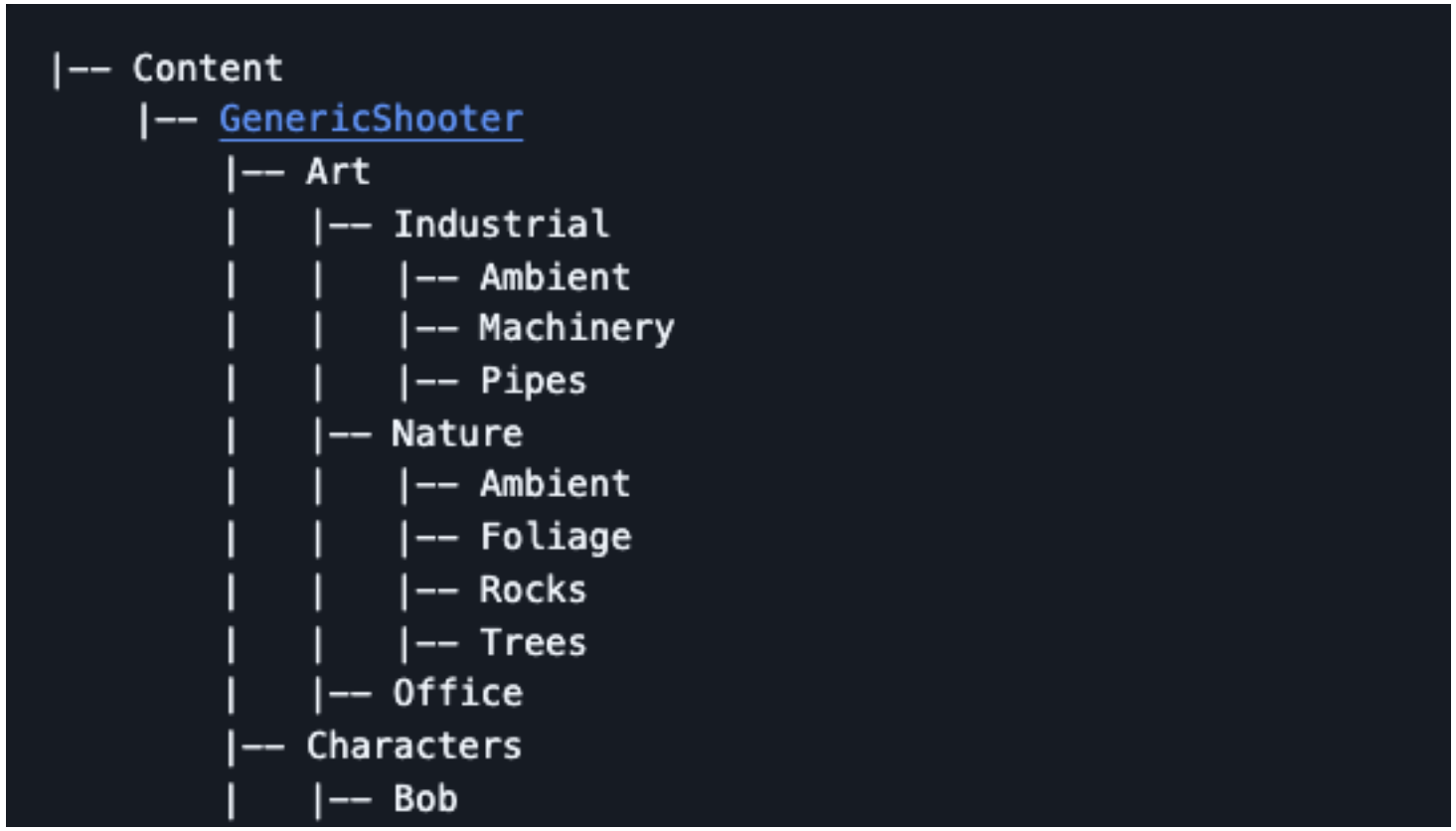
Table of Contents

1. Project Organization
2. Naming Conventions
3. Blueprint Best Practices
4. What NOT to Do
5. Version Control
6. Collaboration on Project
7. Performance Optimization
8. Modern UE Technologies
9. Useful Resources

MENDELU
Faculty
of Business
and Economics

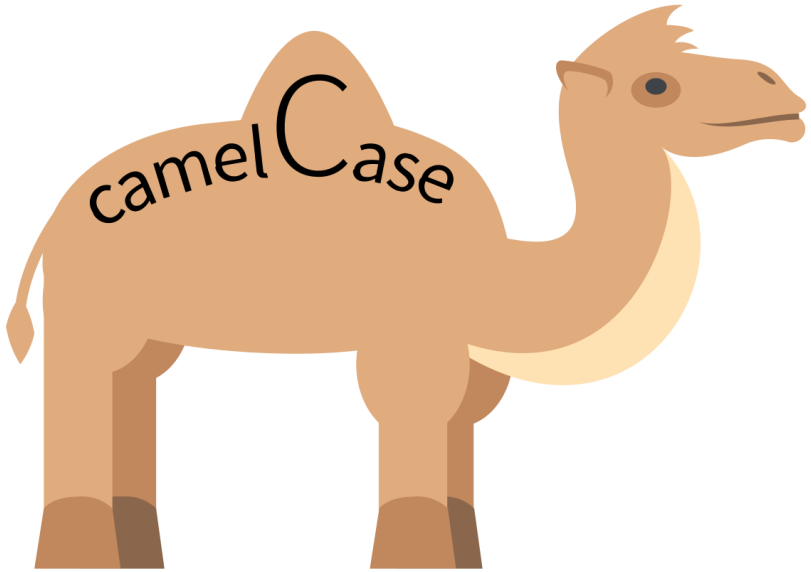
Project Organization

- Consistent folder structure
- Organize assets by logical groups or levels



Naming Conventions

- One naming convention in project



Naming Conventions

- Prefixes based on object type

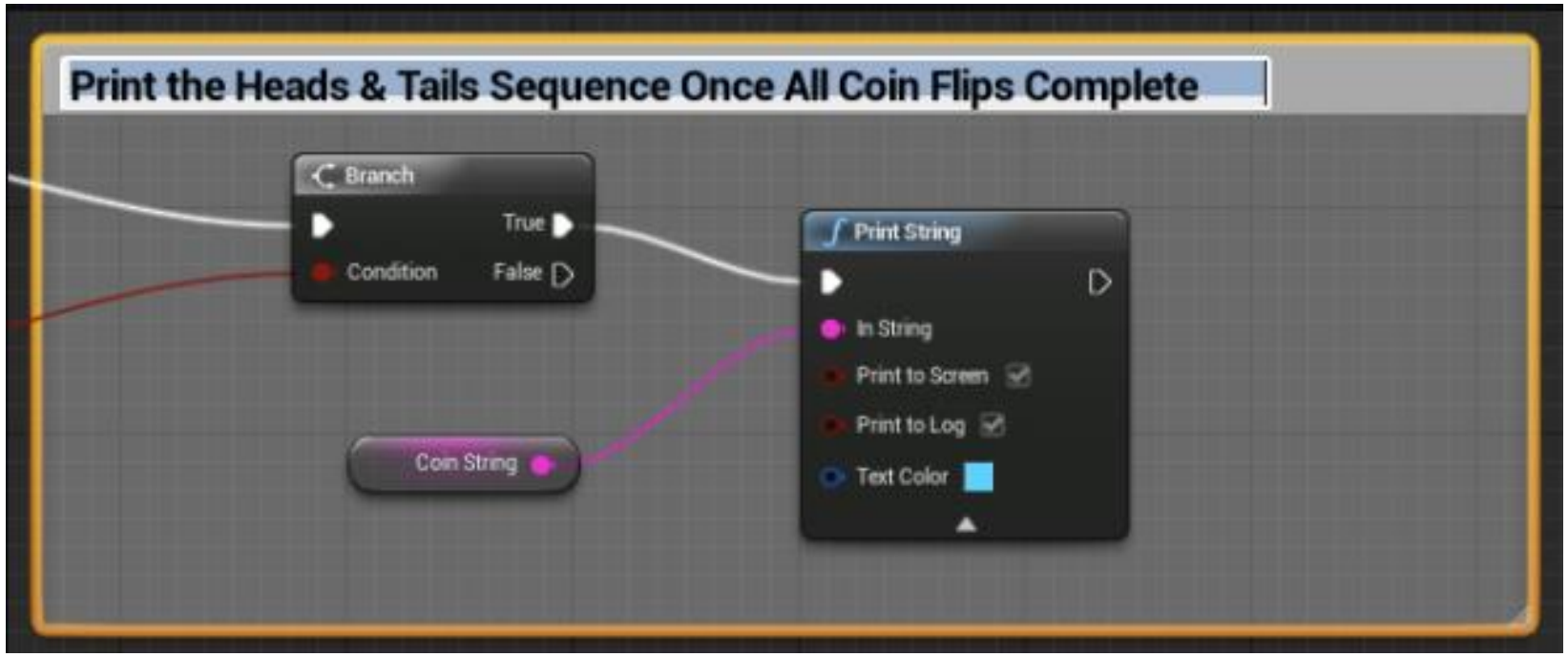
Asset Type	Prefix	Suffix
Blueprint	BP_	
Blueprint Component	BP_	Component
Blueprint Function Library	BPFL_	
Blueprint Interface	BPI_	
Blueprint Macro Library	BPML_	

Asset Type	Prefix	Suffix
Level / Map		
Level (Persistent)		_P
Level (Audio)		_Audio
Level (Lighting)		_Lighting
Level (Geometry)		_Geo
Level (Gameplay)		_Gameplay

MENDELU
Faculty
of Business
and Economics

Blueprint Best Practices

- Use comments and comment blocks
- Split logic into small, reusable functions
- Keep logic flow left-to-right, top-to-bottom
- Avoid “Blueprint spaghetti” (complex, tangled graphs)



What NOT to Do in UE

- Avoid using Tick unnecessarily – use Event Dispatchers or timers
- Avoid GetAllActorsOfClass in large scenes – save instances
- Don’t use simple Delay nodes – prefer timers
- Limit casting – use interfaces instead
- Don’t import whole asset packages

Version Control in UE

- Install Git LFS for large binary files
- Set up proper .gitignore and .gitattributes
- Close UE before commiting
- Consider saving plugins
- Merge is not possible

vs

PERFORCE

https://www.anchorpoint.app/blog/version-control-using-git-and-azure-devops-for-game-projects

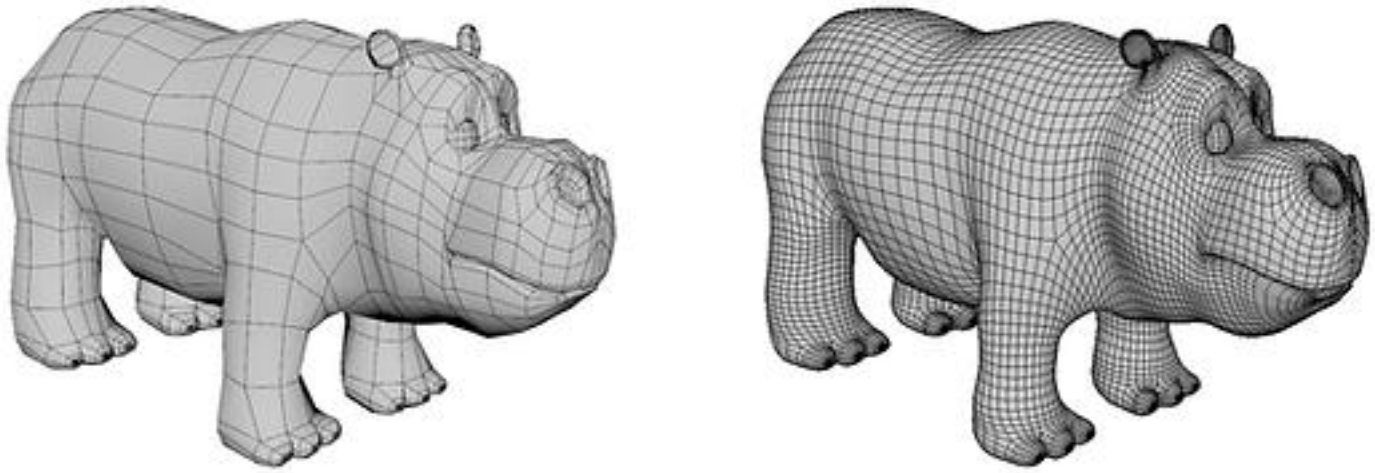
Collaboration in UE

- Define clear roles and responsibilities
- Communicate changes and updates regularly
- Plan work areas to minimize conflicts
- Lock files

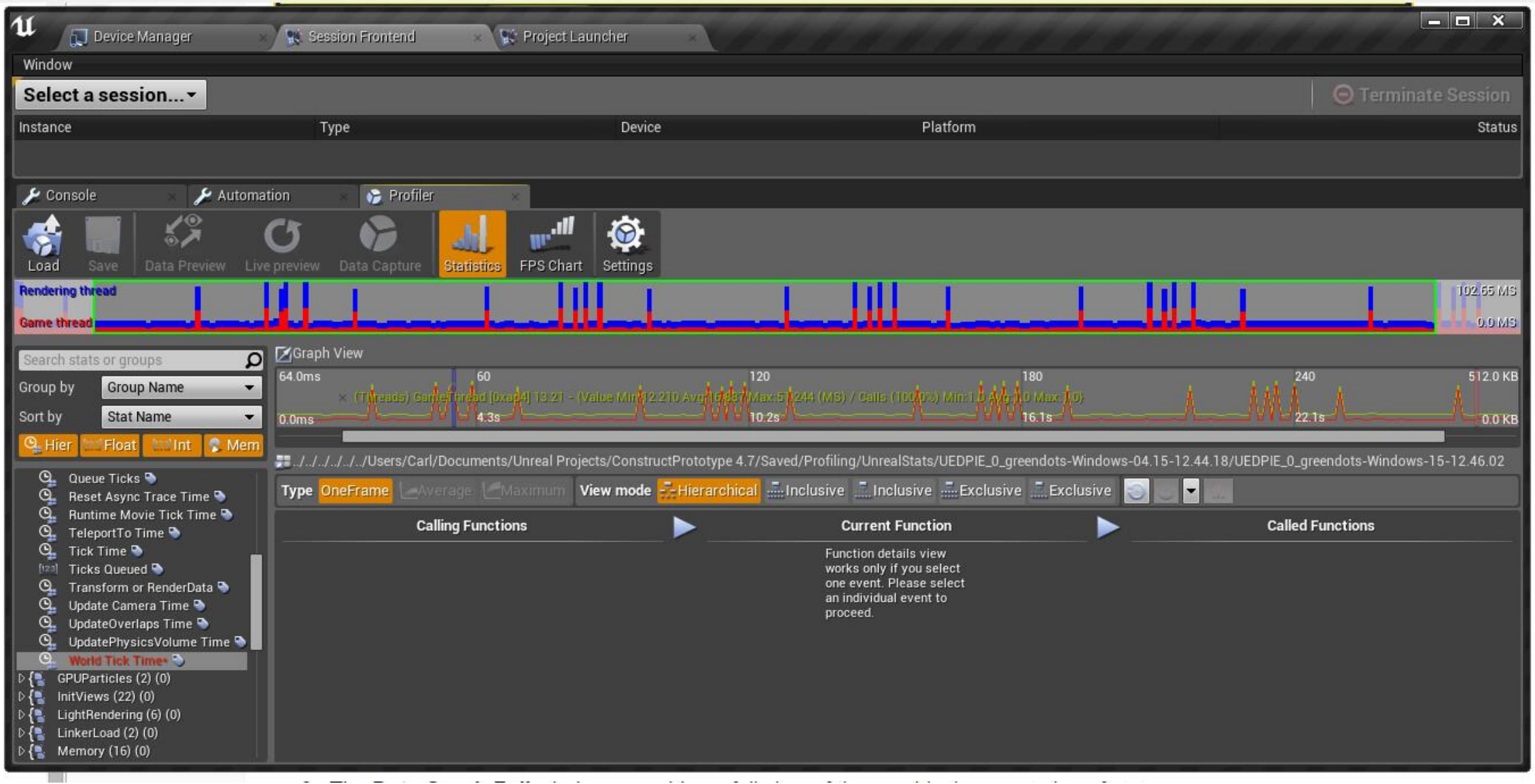
Performance Optimization

- Don’t use gravity and collision if not needed
- Use level of detail (LOD’s)
- LOD’s and calculation of collisions
- Use adequate texture quality

Low poly – High Poly



Performance Profiling



Modern UE Technologies - Nanite

- Virtualized geometry system
- Processes only perceived details
- Automatic LOD’s

High Poly Static Mesh

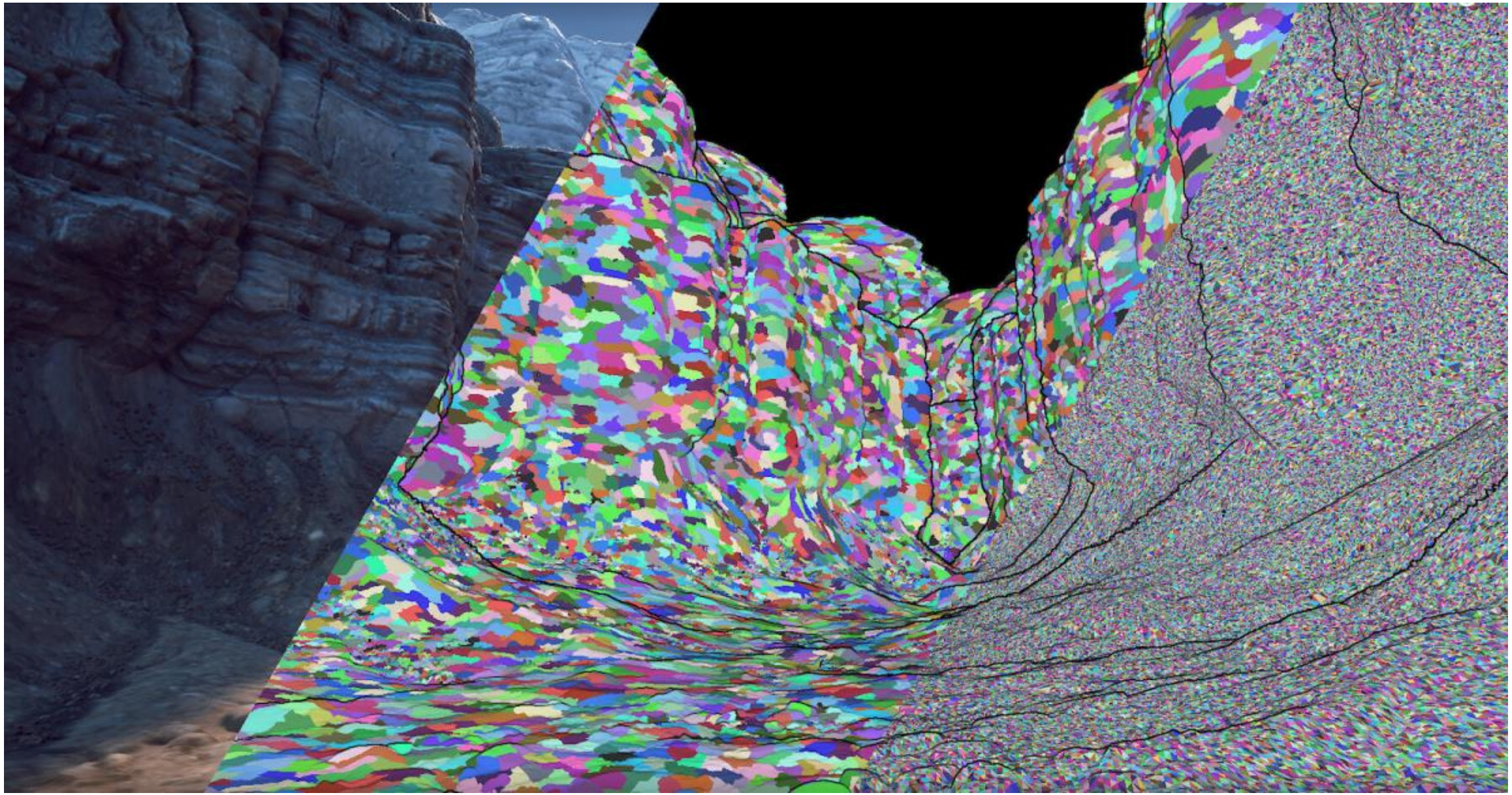
- Triangles: **1,545,338**
- Vertices: **793,330**
- Num LODs: **4**
- Nanite: **Disabled**

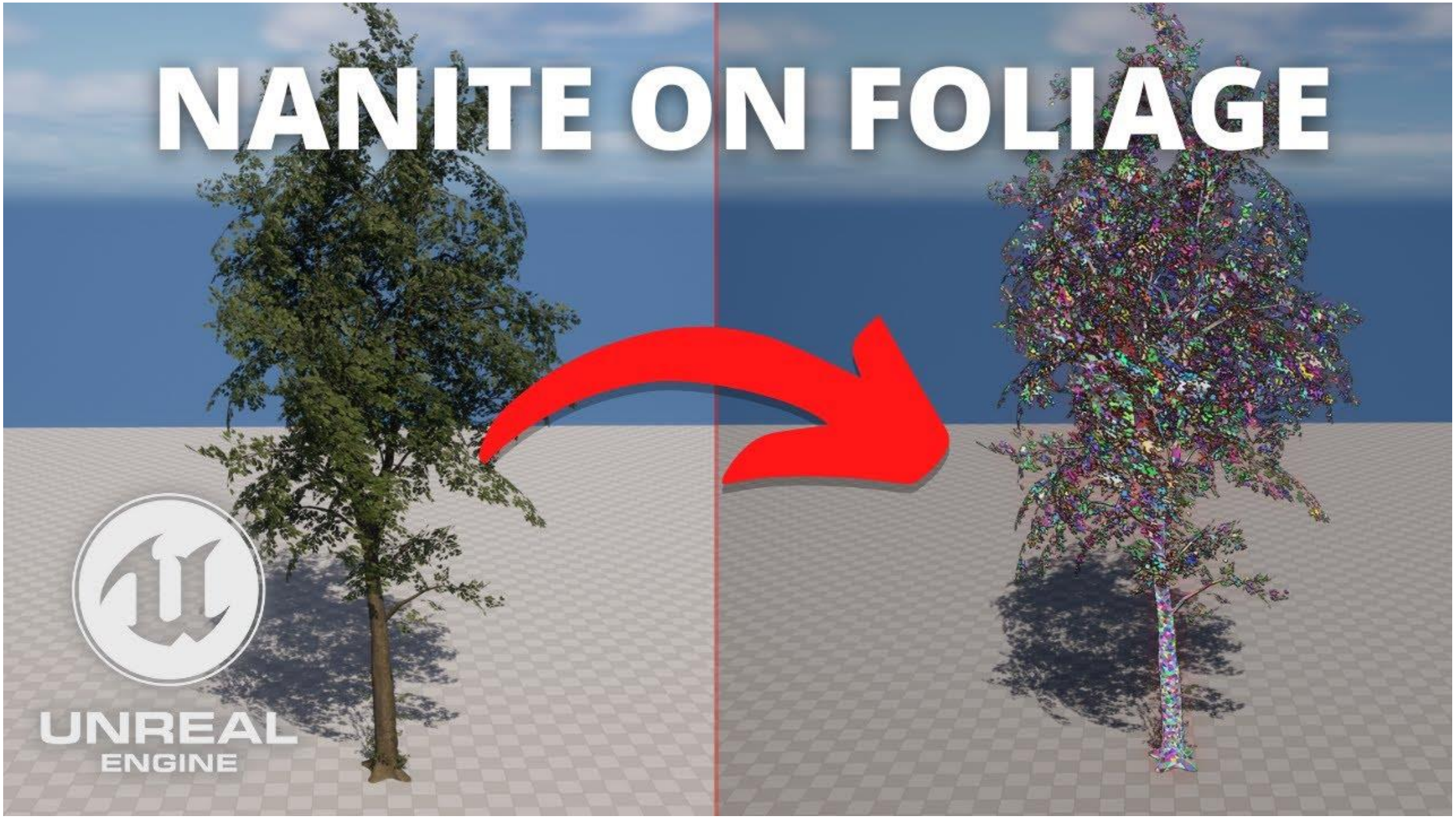
Static Mesh Compressed Packaged Size: **148.95MB**

Nanite Mesh

- Triangles: **1,545,338**
- Vertices: **793,330**
- Num LODs: n/a
- Nanite: **Enabled**

Static Mesh compressed package size: **19.64MB**

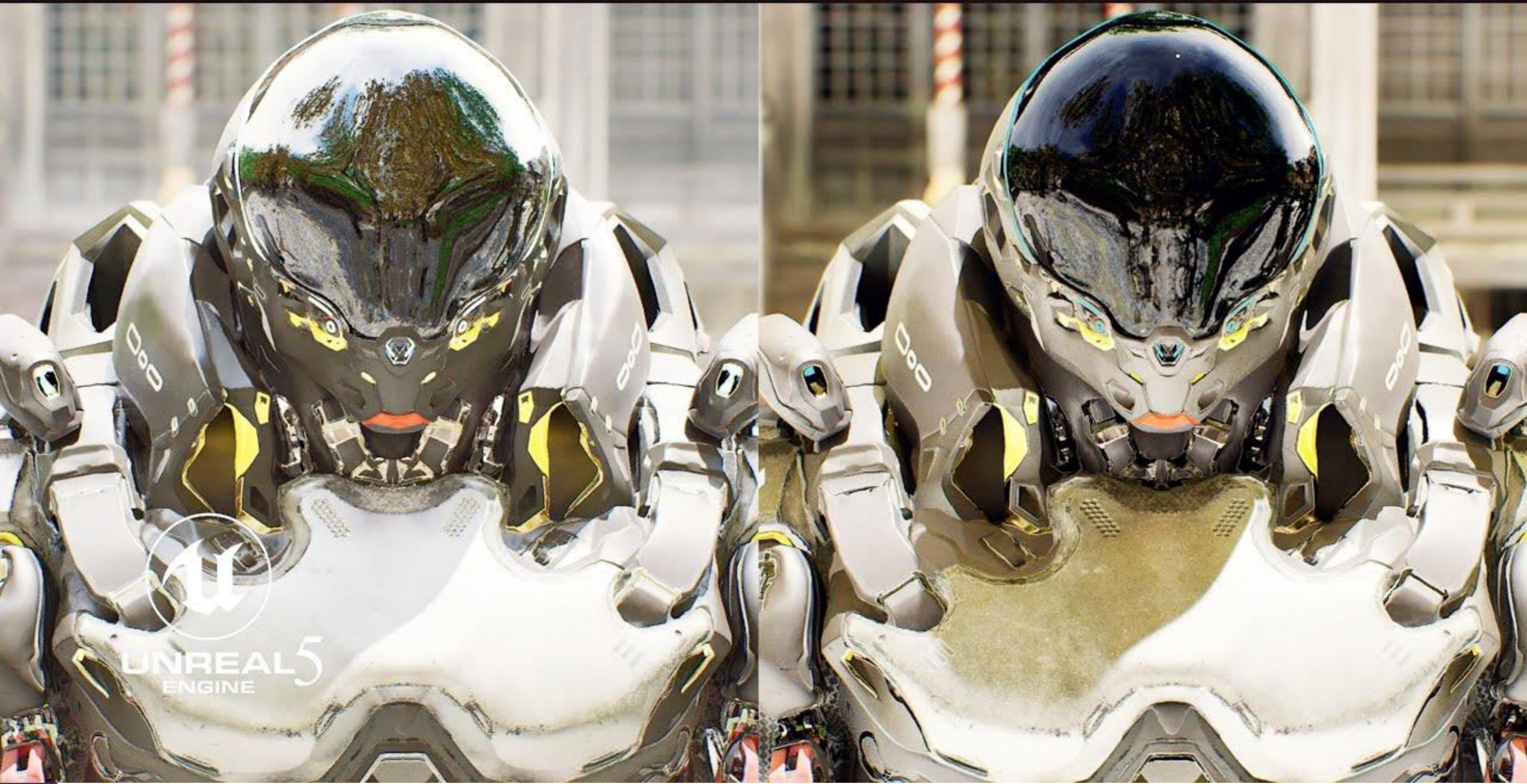



NANITE ON FOLIAGE
UNREAL
ENGINE

Modern UE Technologies - Lumen

- Fully dynamic global illumination and reflection system
- Reacts to changes in real time
- No need for lightmaps

UE5 LUMEN VS OLD RAY TRACING



Useful resources

Style guide:
https://github.com/Allar/ue5-style-guide?tab=readme-ov-file#structure

UE documentation:
https://dev.epicgames.com/documentation/en-us/unreal-engine/blueprints-visual-scripting-in-unreal-engine

UE FAB:
https://www.unrealengine.com/en-US/fabfreecontent

Ali Elzoheiry – games and advanced topics:
https://www.youtube.com/@AliElZoheiry/videos

MENDELU
Faculty
of Business
and Economics

Useful resources

Ryan Laley:
https://www.youtube.com/@RyanLaley/videos

DK 3d – procedural content generation:
https://www.youtube.com/@dk_3d