



모바일 플랫폼에서의 레벨 최적화

🕒 Created	@February 3, 2022 10:26 AM
🏷️ Tags	

- 모바일 환경에서는 PC와 다르게 성능이 제한되므로 여러 부분에서 최적화가 필요합니다.

1. 적은 드로우콜 수 유지

- What is "Draw Call" in Unreal term?
 - A group of polygons that shares the same material.
 - Draw calls are lookups for assets, which happen every frame. The number of draw calls your application uses depends on the number of unique meshes in your scene, as well as the number of unique material IDs each mesh is using. Currently, a high number of draw calls are the biggest contributor to low graphical performance, and you should reduce them as much as possible.
- 쉽게 말해 CPU가 GPU에게 지오메트리 정보를 주면서 Draw하라는 명령을 내리는 것입니다.
- 적은 수의 드로우콜을 유지하기 위해서는 아래와 같은 해결책이 있습니다.
 1. 씬 안에서의 오브젝트의 갯수 줄이기
 2. 오브젝트를 될 수 있는 한 로우폴리로 구성하기
 3. 오브젝트에서 사용하는 Material의 갯수 줄이기

*드로우콜: 오브젝트나 기타요소들을 렌더링하는 횟수를 의미한다.

1) 씬 안에서의 오브젝트의 갯수 줄이기

- 오브젝트의 갯수를 줄이기 위해서는 다음과 같은 방법이 있습니다.

a) Instanced Static Mesh(ISM)

- 같은 오브젝트들의 경우 드로우콜을 따로 하지 않고 1번의 드로우콜로 많은 오브젝트들을 그릴 수 있습니다.

Static Mesh vs. Instance

- **Static Mesh Components are heavy**

- Every single mesh contains:
 - Transform
 - Mesh, materials
 - Physics, collision
 - Lighting
- Rendering: at least 1 draw call per movable SM

- **Single Instance are lightweight**

- An instance contains:
 - Transform
- (Other properties are common for the whole group: mesh, material, etc!)
- Rendering: all instances drawn *at once*

StaticMeshComponent와 InstancedStaticMeshComponent의 차이

- 사용방법은 액터를 만들고 그 액터에 UInstancedStaticMeshComponent를 붙이는 것입니다. 기본이 되는 스태틱 메시를 설정한 후에 액터에서 인스턴스를 늘려가는 방법을 사용하시면 됩니다.
- ISM과 HISM을 어떤걸 사용해야 하는지 판단을 해야할 때가 올 수도 있는데 판단기준은 LOD를 사용하면 HISM을 사용하시면 되고, 그렇지 않으면 ISM을 사용하시면 됩니다.
- 언리얼 4.22 이상에서는 Auto Instancing을 지원합니다. 배치되어 있는 오브젝트 중에서 같은 것은 묶어서 DrawCall 합니다. Directional Light의 모빌리티를 Static으로 바꾸면 Auto Instancing을 시작합니다. 그 이후 모빌리티를 수정하면 됩니다.

b) 액터 병합

- 액터 병합을 함으로써 오브젝트의 숫자를 줄입니다.
- 액터 병합 공식 문서(<https://docs.unrealengine.com/4.27/ko/Basics/Actors/Merging/>)

c) 계층형 레벨 오브 디테일

- 개요(<https://docs.unrealengine.com/4.27/ko/BuildingWorlds/HLOD/Overview/>)
- 사용법은 위 링크에 있습니다.

d) Precomputed Visibility Volume

- 공식 문서(<https://docs.unrealengine.com/4.27/ko/RenderingAndGraphics/VisibilityCulling/PrecomputedVisibilityVolume/>)
- 스마트폰 환경에서 동적 컬링이 제한적이기 때문에 정적 컬링이 요구됩니다.
- Precomputed Visibility Volume은 정적 컬링을 자동으로 계산하게 해줍니다.

스태틱 메시 배치 워크플로 제언

- 레벨을 구성하면서 가장 작은 부분들을 구성하여 그것을 반복적으로 배치할 생각이라면 가장 작은 부분들을 액터병합을 하고 그것을 인스턴스화하여 반복 배치하면 됩니다.
 - 이를테면 벽을 예로 들어보겠습니다. 벽의 층고가 높고(외벽이 수직으로 2개 필요), 벽 위쪽에 몰딩을 가지고 있을 경우, 이런 메시들을 병합하여 하나로 만든 후, 벽들을 인스턴스화하여 배치합니다.
 - 같은 스태틱 메시인 경우 최대한 인스턴스로 만듭니다.
- 배경 오브젝트들을 배치합니다.

- Precomputed Visibility Volume을 배치하여 정적 컬링을 하도록 합니다.
- HLOD를 사용하여 멀리있는 물체는 병합해서 보이도록 설정합니다.
- 언리얼 엔진에서는 LOD를 자동으로 생성할 수 있게 합니다. 이용하여 멀리있는 오브젝트의 삼각형의 갯수를 줄입니다.
- 컬링과 레벨 스트리밍을 고려하여 배경 크기를 정합니다.

참고문서

- 모바일 디바이스용 퍼포먼스 지침서
 - <https://docs.unrealengine.com/4.27/ko/SharingAndReleasing/Mobile/Performance/>
- 모바일 퍼포먼스 팁과 정보
 - <https://docs.unrealengine.com/4.27/ko/SharingAndReleasing/Mobile/Performance/TipsAndTricks/>
- Rendering Optimization for Mobile
 - <https://docs.unrealengine.com/4.27/ko/SharingAndReleasing/Mobile/Rendering/MobileOptimization/OptimizationandDevelop>