**1 . 최적화 수행 전**



- 측정 기준 : 6분 30초 동안 야외 필드에서 몬스터들을 사냥

- 평균 FPS : 19

- Janks : 전체 33 번 , 12번의 Big Janks 발생

- 평균 CPU 사용량 : 14.8%

- 평균 GPU 사용량 : 43%

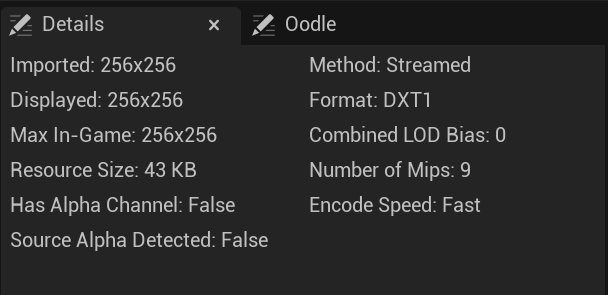
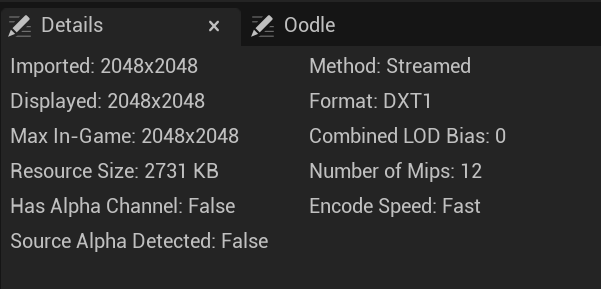
- 평균 메모리 사용량 : 2.1GB

**2 . 1단계 최적화 - 배경 최적화**

**2 - 1 . 텍스처 리사이징**

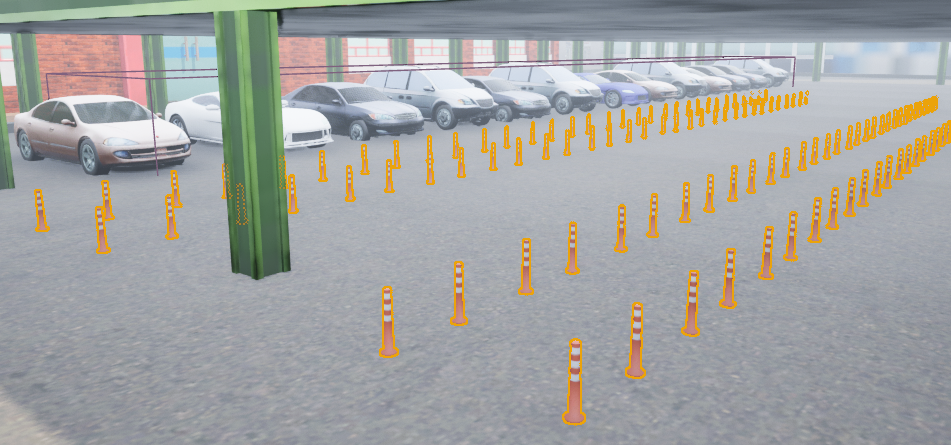
플레이어와 몬스터 같은 주요 요소들의 텍스처들은 2048 -> 1024 or 512 로 텍스처 크기 줄임

배경 요소같은 텍스처들은 2048 -> 512 or 256 으로 다운 스케일링



**2 - 2. UHierarchicalInstancedStaticMeshComponent 사용**

월드에 개별적으로 하나하나씩 배치된 스태틱 메시들이 너무 많았음 -> 이러면 같은 메시일지라도 개별적으로 하나씩 Draw Call 을 해야해서 CPU , GPU 낭비 -> 규칙적으로 나열된 같은 스태틱 메시 액터에 대해서는 UHierarchicalInstancedStaticMeshComponent 사용!

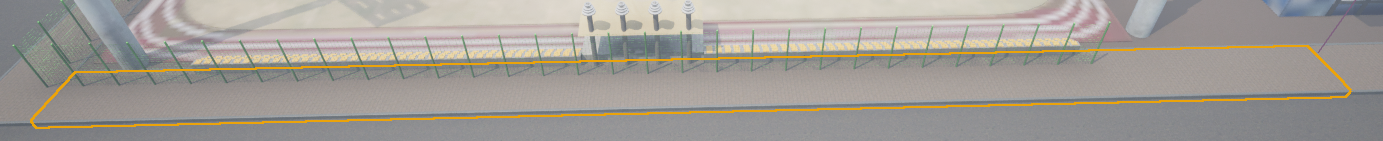


<Hierarchical Instanced Static Mesh 사용>

HISMComponent->SetStaticMesh(MeshAsset.Object); // 한번 GPU 에 DrawCall 요청

HISMComponent->AddInstance(NewTransform); // 이후에는 GPU Instancing 으로 처리



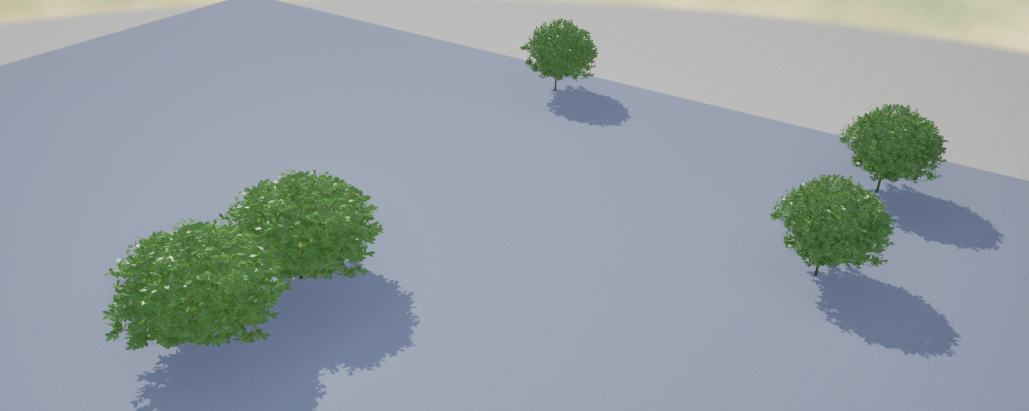


**2 - 3. 월드에 배치된 액터 병합**

불규칙적으로 배치된 같은 스태틱 메시나 머터리얼이 같은 메시들이 낱개로 나누어져 있어서 개별적으로 Draw Call 을 해야 되기 때문에 GPU 부하 발생 -> 액터들을 병합하여 Draw Call 수를 줄임!

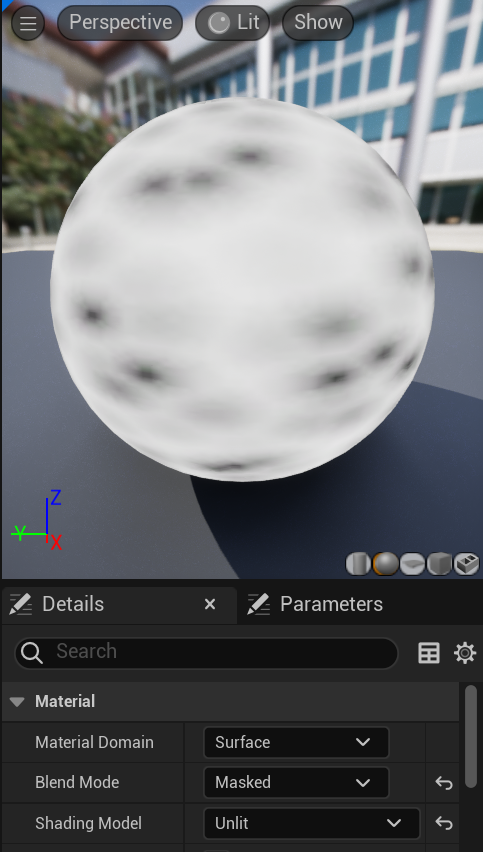
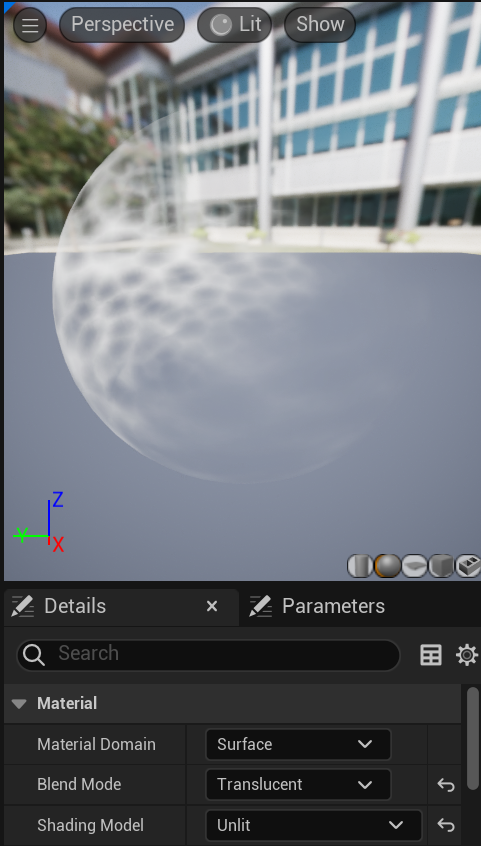
****

가까이 배치된 같은 머터리얼을 가진 액터들에 대해서는 병합을 해서 월드에 배치



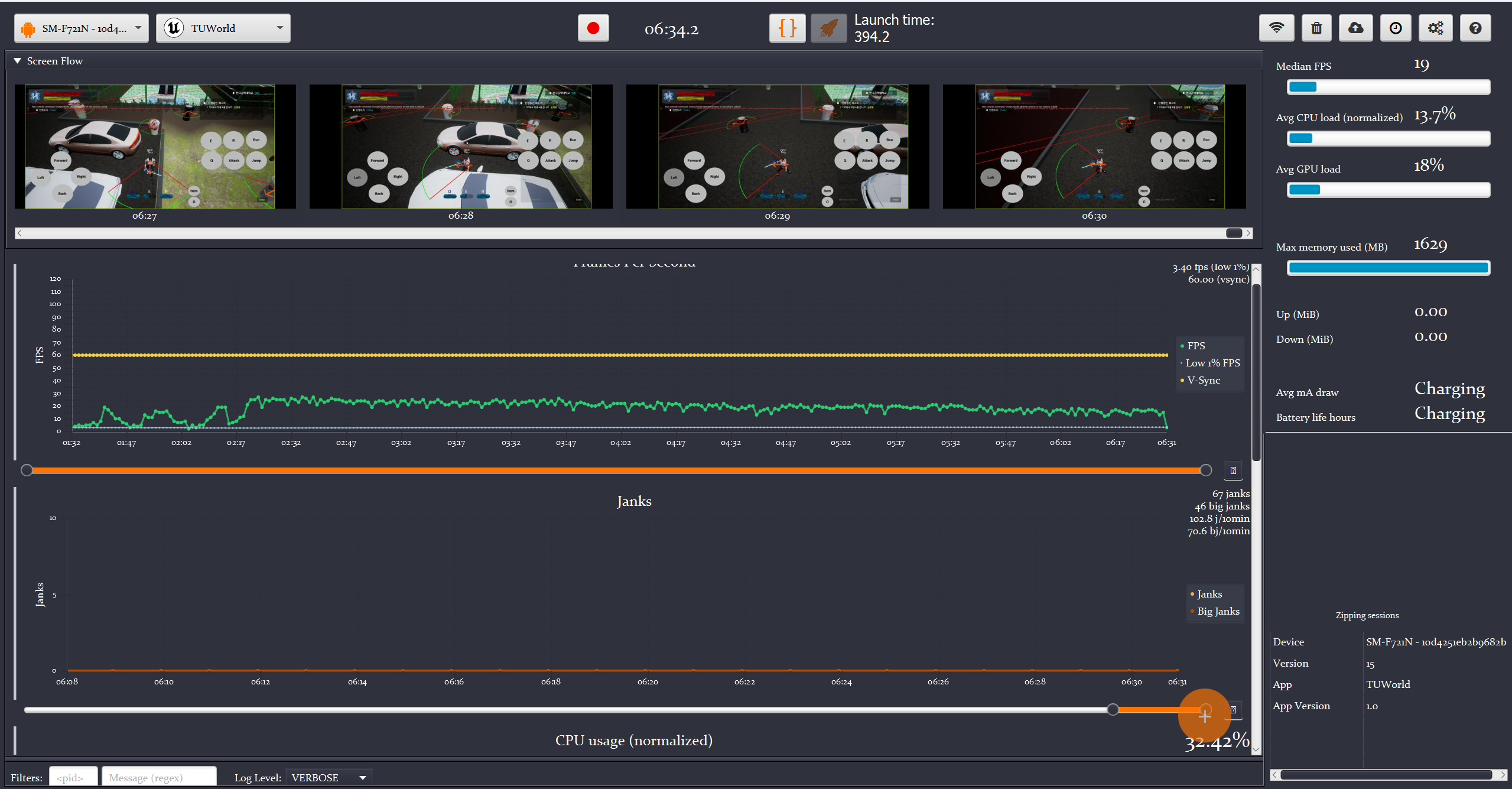
**2 - 4 . Material Blend Mode 를 반투명 머터리얼 대신 Masked 사용하기**

셰이더 복잡도 측면에서 반투명 머터리얼은 여러겹을 렌더링 해야하기 때문에 무거움



**2 - 5 . 텍스처 해상도 최적화**

**2 - 6 . 1단계 최적화 결과**



- 측정 기준 : 6분 30초 동안 야외 필드에서 몬스터들을 사냥

- 평균 FPS : 19

- Janks : 전체 33 번 , 12번의 Big Janks 발생

- 평균 CPU 사용량 : 13.7%

- 평균 GPU 사용량 : 18%

- 평균 메모리 사용량 : 1.67GB