|  |
| --- |
| **MetaX\_최적화 이슈** |

**■ 배경 모델링 최적화**

**모델링**

- 바닥의 경우 4x4 사이즈의 바닥을 연결해서 12x16을 만드는 방법과 12x16 모델을 외부 제작 툴에서부터 제작해서 들여와 배치하는 방법이 존재한다.

- 스태틱 옵션에서 Batching static을 체크하면 동일한 오브젝트라 판단되는 것을 묶어주어 Batches 값을 줄여줄 수 있는 최적화 기능이다.

- 어짜피 동일한 값을 묶어준다면 4x4 하나의 모델만을 활용해도 문제가 없을 것이나 이 묶어주는 것도 연산이 필요한 부분인 것을 인지해햐 한다.

- 그렇다고 규격대로 모델을 제작하여 불러온다면 리소스 용량이 커질 수밖에 없는 것도 사실이다. 물론 묶는 연산은 줄지만 말이다.

- 모바일 프로젝트의 경우 용량과 메모리관리 두 요소를 모두 중요하게 생각해야 한다.

- 모바일 기준에서 확장이 많은 프로젝트의 경우에는 용량관리에 우선을 둬야 할 듯하다. 대신 씬 단위 요소는 최소화되는 것으로 연출을 해야 할 것 같다.

- 모바일이어도 확장보단 한 씬에 처리되는 것이 많은 형태의 프로젝트라면 다양한 형태의 모델을 활용할 수 있는 얘기이므로 조화롭게 구성하여 활용하는 편이 좋을 듯하다.

- 하지만 테스트 해보고싶다. 두 개의 기능을 모두 활용해보고 싶고 어느정도의 퀄리티까지 수용할 수 있는지 여부를 테스트해보고 싶다. 어느정도 욕심은 버리는 선에서 표현도 활용하고, 다양한 모델타입도 활용하고, 배칭하는 형태까지 작업을 해보자.

- 모델링을 익스포트할 때 메터리얼이 적용되고 안되고의 차이로 용량이 차이가 3~4배가량 차이가 난다.

- 가급적이면 맥스에서는 메터리얼을 적용해서 넘기지 말자.

- 탄젠트 노멀의 체크유무에 따라 약 2배가량 용량차이가 난다. 탄젠트 스페이스로 활용을 할 것인지 미켈슨 탄젠트로 활용할 것인지에 따라

체크유무를 결정지어야 하겠지만 모바일의 경우에는 미켈슨을 활용하도록 하자.

**텍스처 및 메터리얼**

- 셰이더의 경우 모바일 셰이더 활용을 권하는 부분이 있으나 범프 / 텍스처를 사용할 경우 메탈릭 텍스쳐를 사용할 수 없게 된다. 현재 우리

프로젝트에서 금속 효과를 위해 스탠다드 셰이더 활용을 해야 할 듯하며 일단 해보고 문제가 발생한다면 교체하는 것을 고려해야 한다.

- 이 또한 맵의 규모가 작다면 활용측면에서 효율적이겠으나 규모가 크다면 지양해야 할 부분인 듯하다.

- 재질은 같은 텍스쳐라면 하나의 재질로 활용하도록 한다. 틸링 값이 다르다는 이유로 재질 개수를 늘리면 setpass 수치가 늘어난다.

- 모델링부터 하나의 재질로 활용할 수 있도록 UV 편집을 진행해서 들여오도록 한다.

- PBS를 위해 스탠다드 셰이더를 사용할 경우 불필요한 슬롯은 비워두면 연산을 안하므로 모바일에서도 PBS를 활용하기 위함이라면 적절하게 슬롯을 활용하는 것이 좋다.

- 또한 텍스처의 사이즈도 화면상에서 확인해보면서 각각의 속성에 따라 사이즈를 조절하는 것도 용량면에서 효율적이다.

- 노멀맵의 경우는 탄젠트 스페이스 또는 Mikkttspace가 있다.

- 탄젠트의 경우 90도 이상 꺾이는 부분을 스무딩 그룹으로 나누어 표면처리를 부드럽게 해주면서 노멀맵을 제작한다.

- Mikkttspace는 스무딩 그룹을 나누지 않고 모양이 안 이쁘게 나와도 스무딩 그룹을 하나로 사용하거나 스무딩 그룹정보를 넘기지 않는다. 이유는 엔진에서 연산 되는 버텍스 개수가 크게 차이 나기 때문이다.

- 탄젠트는 스무디 그룹에 맞춰 UV 나눠줘야 노멀 퀄리티가 좋다.

- Mikkttspace는 하나로 UV 를 편집해줘야 그리고 삼각면의 방향도 적절해야 퀄리티가 좋다.

- 보통 언리얼이나 유니티 둘다 Mikkttspace가 기본으로 되어있는데 언리얼과 유니티HDRP는 탄젠트로, 유니티는 모바일 포함 Mikkttspace로 쓰는 편인 듯하다.

- 서브스턴스 페인터에서 텍스처를 제작할 때에도 언리얼이나 유니티에 따라 Compute Tangent Space Per Fragment를 체크하거나 해제한다.

- Mikkttspace 는 미켈슨 탄젠트 스페이스라고 읽는다.

- 이래저래 그때그때 텍스처의 규격을 정하고 슬롯을 효율적으로 활용하여 메터리얼도 최적화하도록 한다.

- 요즘은 디바이스가 좋아졌다고 하나 일단 가볍게 만들고 보완하는 편이 개발 시 최적화 위험도를 줄일 수 있다고 본다.