Raycast를 이용해 복잡한 구조(Mesh)를 가진 적이 피격되는 정확한 위치에 텍스쳐를 덮입히는 방법

1. Collider를 사용하지 않고, Mesh 위에 RayCast와 충돌하는 지점을 알아내는 방법 필요
   1. Mesh Collider의 경우 복잡한 구조인 적이, 그 수도 많기 떄문에 많은 렉을 유발할 것
   2. 캡슐 콜라이더와 같이 작은 부분들을 적에게 붙여서 사용하는 방법도 가능은 할 것.
   3. Mesh 위에서 충돌된 부분을 정확하게 잡아낸 후, 그 부분에 Decal을 씌우거나 새로운 텍스쳐를 만들어 그려주는 것이 애니메이션 효과도 따라갈 것으로 판단됨
2. 실 적용 방안
   1. Raycast(충돌 확인)

└→ SkinnedMeshRenderer에서 BakeMesh

└→ 모든 triangle과 교차 검사

└→ 일치하는 triangle을 찾으면

└→ UV 보간 (barycentric)

└→ 텍스처 위에 칠하기

* 1. 렌더 텍스쳐를 사용해서 텍스쳐를 복사해서 발생하는 성능이슈 방지 방법 추가. 이 부분은 아직 사용안함. 추후 코드와 원리를 더 이해하여 사용할 가능성 있음.

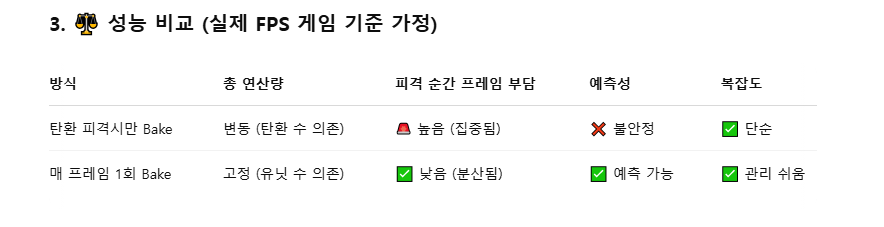
[1] 적 캐릭터는 본인의 RenderTexture를 가지고 있음

[2] 탄환 피격 → 적의 UV 좌표 계산

[3] Graphics.Blit()을 사용해 RenderTexture에 자국 그림

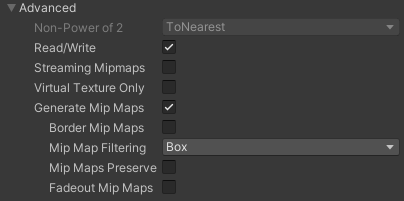
[4] Shader에서 이 RenderTexture를 받아 적 텍스처에 덧입힘

* 1. 테스트 코드를 통해 작동함을 확인. 성능을 위한 코드 개선 방안.
     1. 메쉬를 Bake하는 과정을 피격될 때 -> 적 유닛이 생성될 때? -> 그럼 현재 상태와 다르게 될 수 있다. -> 대안은 Late Update로 주기적으로, 그리고 조금 천천히 미리 업데이트 해둘 것 -> 탄이 발사되는 간격에 따라 달라질 수 있지만, 탄을 쏘는 속도가 몹시 빠른게 아니라면 탄환 피격시에만 처리할 것



* + - * + X GPT 가 발사속도를 너무 높게 잡은 것으로.
    1. 텍스쳐를 복사해서 생성하는 과정 -> 적 유닛이 생성될 때로 옮기기

1. 실 적용 시 주의해야하는 점.
   1. 텍스쳐를 read, write가 가능한 옵션으로 두기



* 1. 텍스쳐의 Format을 Auto가 아닌 다름 값으로 넣어주기

