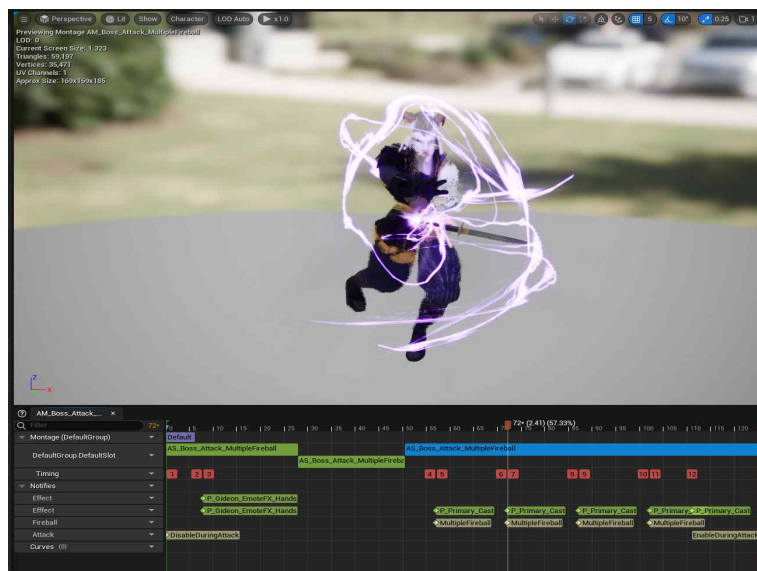


Day 34. 보스 패턴 추가

34-1. 보스 패턴 추가



보스몬스터의 공격에 대한 여러 가지 패턴을 추가하였습니다. 보스몬스터도 일반 몬스터와 마찬가지로 공격패턴에 대하여 Montage를 배열로 묶고 RandRange로 랜덤하게 나타나는 방식으로 설계 했으며 유저를 배회하고 텔레포트 하면서 중간 중간에 패턴을 실행하는 식으로 만들 생각이었습니다. 근접 공격에 대한 구현은 마쳤으므로 원거리 공격이나 특별한 공격에 대한 구현을 만들었습니다. 먼저 기본적인 원거리 공격인 파이어볼 공격을 만들었습니다. 파이어볼의 경우에는 매직 캐릭터를 만들때와 유사하게 충돌판정이 있는 구체를 발사하고 Projectile Movement를 이용하여 위치를 지정했습니다.

그리고 애니메이션을 나누어 구간마다 Rate scale 다르게 하여 연속적으로 파이어볼을 발사하는 패턴을 만들었고 근접한 상황에서는 해당 패턴을 사용하지 못하게 하기 위해 float 값을 Tree에서 입력받아 타겟과 AI Owner간에 거리를 측정해 bool을 반환하도록 UBTDecorator를 만들어 조건으로 사용했습니다. 그리고 캐릭터를 관통하며 공격하는 DashAttack()을 만들었습니다. DashAttack()은 보스몬스터의 Movementmode를 MOVE_Flying으로 나누어 이펙트를 보여주면서 주인공을 관통하며 공격하게 연출하였고 이때 충돌채널이 overlap으로 바뀌면서 onOverlapBegin()함수와 섞어 데미지를 주도록 구현했습니다.

마지막으로 보스몬스터의 등 뒤로 문을 여러개 열고 거기서 망치가 쏟아져 나와 공격하는 패턴을 추가했는데 Arrow Component를 미리 여러개 설치한 뒤 배열로 묶어두고 스킬이 사용되면 Arrow Component에서 Transform 값을 가져와 각 문에 적용시켜 해당 방향으로 스킬이 전개되도록 하였습니다. 그 후 각 문에 대하여 for문으로 TimerHandle을 만들고 전부 Random한 값의 종료값을 주어 시차를 두고 나오도록 연출했습니다.

34-2. 보스 힐 패턴 추가

보스가 특정 체력 밑으로 내려갈 경우 체력을 회복하려 도망가는 패턴을 만들어 보았습니다. 먼저 도망갈 위치를 찾기 위해 EQS를 사용하였습니다. 플레이어의 입장에서 보이지 않는 곳으로 도망가 체력을 회복하는 연출을 하고 싶었고 이것을 위해서 먼저 PathExist로 갈 수 없는 곳을 제한 뒤 Trace를 통하여 타겟의 시야에서 보이는 곳에 대한 값을 다시 bool Match로 뒤바꾸어 보이지 않는 곳으로 바꾸었습니다. 그리고 Test Purpose를 Score only로 설정하여 점수 측정을 Inverse Linear를 주면서 도망 갈 수 있는 위치 중 가장 가까운 곳을 선호하도록 하였습니다.

힐 기능에 대한 구현에 경우에는 간단한게 기존에 사용하던 Damage를 입을 때 적용시키는 함수를 조금 수정하여 거꾸로 체력이 올라가게 하였고 이에 따른 컴포넌트, 인스턴스, UI 값을 바인드 시켜주어 기능을 완성하였습니다.