1. 서버 - 클라이언트 애니메이션 조작

텍스트이(가) 표시된 사진

자동 생성된 설명

플레이어와 몬스터의 애니메이션 종류를 열거형 상수로 정리했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

현재 PlayerMovementComponent는 클라이언트의 키 입력을 바탕으로 애니메이션을 결정한다.

다른 플레이어와 애니메이션을 다르게 하기 위해선 서버에서 애니메이션 정보를 보내주고 클라이언트에 저장해 그 애니메이션을 수행하는 방식으로 이루어지는 것이 좋을 것이라 판단했다.

텍스트, 장치, 계량기, 게이지이(가) 표시된 사진

자동 생성된 설명텍스트, 장치이(가) 표시된 사진

자동 생성된 설명

플레이어와 몬스터의 애니메이션 타입을 담아 보낼 패킷이다.





서버에서 플레이어와 몬스터는 OldAniType과 PresentAniType을 갖는데, 애니메이션 타입이 바뀔 때 (예를 들어 IDLE 상태에서 WALK를 할 때), 그 순간만 패킷을 보내기 위해 만들었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

서버에서 플레이어의 애니메이션 타입이 바뀐 순간에 패킷을 보내고 OldAniType을 수정하는 모습이다.

1. 몬스터 동기화

움직임 동기화

텍스트이(가) 표시된 사진

자동 생성된 설명

몬스터의 FSM이 IDLE과 WANDER가 반복되기만 하고 TRACK 상태로 변환되지 않는다.

하나하나 찾아보니 플레이어와의 위치는 검색하는 함수가 항상 False를 반환해서 일어나는 일이라고 유추했다.

텍스트이(가) 표시된 사진

자동 생성된 설명플레이어 위치, 몬스터 본인 위치를 출력하여 확인해본 결과 플레이어를 움직임에도 불구하고 플레이어 위치가 항상 0으로 고정되어 있었다.



서버의 프레임워크에서 플레이어의 위치를 표현할 때 PlayerMovementComponent의 XMFLOAT3를 사용하고 있지만, 위의 함수에선 Object의 월드 행렬을 받아온다. 이 두 표현 방식의 불일치가 발생하여 일어난 일이었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

따라서 플레이어 위치 변경 때 월드 행렬에도 값을 넣어 주기로 했다.

잔디, 운동 경기, 스포츠이(가) 표시된 사진

자동 생성된 설명 이제 몬스터가 제대로 따라온다.

단, 현재는 처음 들어온 플레이어 하나 만을 따라온다. ▶ 고쳐야 할 점

애니메이션 동기화

플레이어 애니메이션 동기화와 매우 유사하다.

텍스트이(가) 표시된 사진

자동 생성된 설명

서버에서 클라이언트로 애니메이션 타입 패킷을 보내는 모습이다.

텍스트, 스크린샷, 은이(가) 표시된 사진

자동 생성된 설명

클라이언트에서 받아서 저장하는 모습이다.

OldAniType과 PresentAniType이 바뀔 때마다 애니메이션 타입 패킷을 보내주고, 클라이언트는 받은 애니메이션 타입을 저장하여 몬스터의 애니메이션 종류를 결정한다.

몬스터 타겟 동기화

몬스터가 WANDER 상태일 때 클라이언트마다 랜덤 방향으로 몬스터의 Look을 세팅하기 때문에 서버에서 WANDER 상태에서 움직일 방향을 알려줘야 한다.



몬스터의 타겟을 설정해주는 Set함수를 추가해줬다.

텍스트이(가) 표시된 사진

자동 생성된 설명

서버에서 클라이언트로 가는 패킷으로, monster\_id를 갖는 몬스터가 player\_id를 갖는 플레이어를 타겟으로 하라는 정보를 담은 패킷을 정의한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

현재는 테스트를 위해 player\_id가 10인(제일 처음 접속하는 클라이언트)를 항상 타겟이 되도록 설정하여 패킷을 보내고 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

monster\_id와 player\_id를 갖고 Monster와 Player를 찾고, Monster의 타겟을 Player로 설정하는 모습이다.