Archery//

발사체 방향 방법: GEngine으로 뷰포트 사이즈를 구해서 뷰포트 사이즈의 x, y 값을 절반씩해서 크로스헤어 위치를 구했습니다. 크로스헤어 방향에서 정면으로 LineTrace를 해 부딪히는 곳의 벡터 값을 구했습니다. 캐릭터에 손에 달아 뒀던 소켓으로부터 구했던 벡터까지의 방향 벡터를 구해 그곳으로 향하게 했습니다.

이동 블렌드 스페이스 인자: Aim기준 로테이션과 캐릭터 움직임의 x좌표기준 로테이션을 구해 aim로테이션과 x좌표기준 로테이션을 뺀 벡터를 정규화해서 값을 넣어줬습니다.

// 다음 스킬로 넘기는 것 같은건 인자로 bool값을 받아서 true면 ++ false면 --로

UI관련 : USERWIDGET을 상속받아 만든 c++ 클래스를 만들어 에디터에서 만든 위젯 블루프린트의 부모클래스로 설정합니다. PlayerController에 tsubclassof로 받은 클래스 변수하나와 C++로 만든 위젯 클래스 변수를 만들고 CreateWidget 함수를 이용해 위젯에 대한 것을 만들어냈다.

물약이나 스킬 휠 : 아이템 개수가 적어서 enum class로 변수 받아서 switch문으로 사용했다.

Tsubclassof : 그 클래스나 그 클래스의 하위 클래스들을 상속받은 값만 넣어 줄 수 있는 변수,

콜리전 프리셋 : 오브젝트 채널이 Pawn처럼 어떤 객체의 콜리전에 대한 이름. TraceChannel은 오브젝트 채널들의 대한 콜리전 반응을 저장하기 위한 채널. 예를들면 TraceChannel에 폰 오브젝트만 블락하게 만들 수 있다.

몬스터 피격 체크 : 몬스터의 오브젝트 채널만 반응 하는 TraceChannel을 만들어 오버랩멀티바이 채널 함수를 사용해 피격체크를 할 수 있게 했습니다. 다량의 적이 피해를 입는 공격의 경우는 TArray에 피격 당한 몬스터들을 넣어 피격 체크를 했습니다.

블랙홀 : 블랙홀을 놓는 방향으로 향하는 벡터를 구해 틱마다 조금씩 이동하게 만듦.

아이템 획득 : 오버랩되면 델레게이트로 브로드캐스트 시킴.

몬스터 스폰 : getworld()에서 spawn액터로 스폰했음.

타겟팅 고정:

블랙보드 : 데이터 집합.

비해비어트리 : 블랙보드를 기반으로 AI의 행동을 시각화한 트리

컴포짓노드 : 셀렉터, 시퀀스, 셀렉터는 실패시 다음 노드실행, 시퀀스는 실패하면 다음노드 실행하지않음

서비스노드 : 독립적으로는 동작하지 않고 컴포짓 노드에 부착하는 노드. 컴보짓 노드가 실행되는 동안에 반복 작업을 하기 좋음 예를 들면 주인공 캐릭터가 감지되는지 체크할 때 사용

데코레이터 : 어떤 노드로 들어갈지 안 갈지 판단할 때 쓰는 것. 예를들면 공격범위 안에 들어왔는지 true나 false나에 따라 공격노드로 들어갈지 추격노드로 들어갈지 선택 가능.