**AI 개요**

~~--------------------------------------------------------------------------------------------------------------~~

정리 전 상태임.

구현 전 설계

BehaviorTree 간의 포함관계 통해 구현

BT\_Patrol

BT\_Attackable

BT\_Skill

BT\_Trait ( 몹 특성마다 행동 양식을 다르게 하기 위함 -> ex) 체력이 떨어지면 도망쳐서 체력 회복을 하는 특성 , 교전 대상과 일정 거리를 유지하려하는 특성 )

팀구분 : 액터 태그로 함 Player / Enemy / ~~

공부?

AIPerception – AI 캐릭터의 시야 구현 비슷 한 것 인듯.

EnviornmentQuery – 환경 쿼리? 지형 지물 관련 된 것 인듯

Flashback

BT 구조

노드들은 부모 노드에게 성공, 실패를 리턴

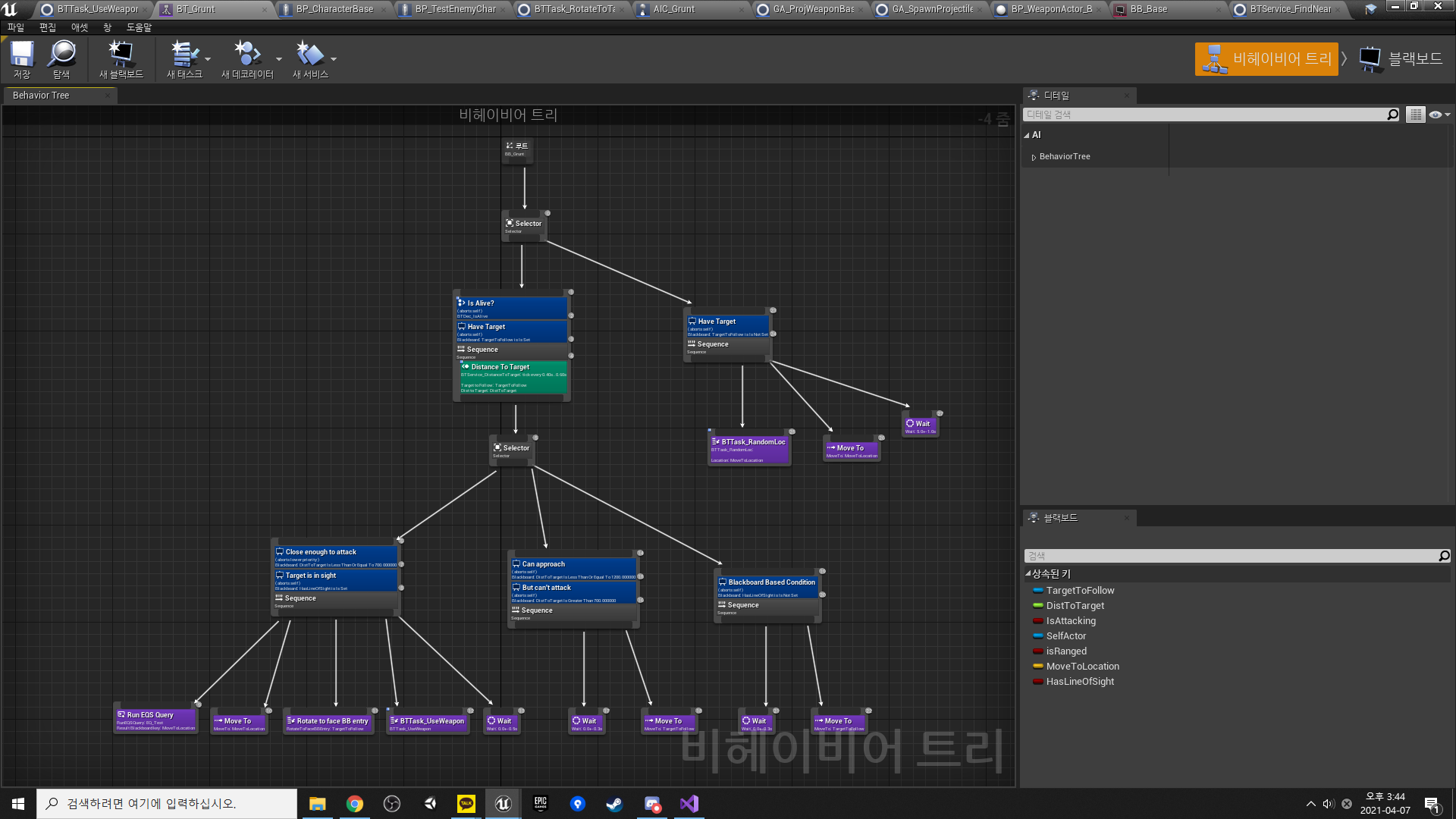
Root -> 엔트리 , 블랙보드 설정 가능

Decorator ->  
조건절이라고도 부르는 것으로, 컴포짓이나 태스크에 붙여서 분기나 노드가 실행될 것인지를 정의, 보통 커스텀하여 사용.

Service ->  
컴포짓 노드에 분기가 실행되는 동안 정해진 빈도에 맞춰서 실행된다. 보통 검사를 하고 그 검사를 바탕으로 블랙보드의 내용을 업데이트하는데 사용.

Composite ->   
컴포짓 노드는 해당 분기가 실행되는 기본 규칙을 정의, 데코레이터, 서비스 부착 가능   
세종류 존재  
1) 시퀸스 -> 왼쪽부터 순차 실행 / 실패한 시점에서 멈춤, 실패를 리턴. 모두 성공해야 성공 리턴  
2) 셀렉터 -> 왼쪽부터 순차 실행 / 성공한 시점에서 멈춤, 성공을 리턴. 모두 실패하면 실패 리턴  
3) 심플 패러렐 -> 전체 노드 트리와 동시에 하나의 태스크를 실행할 수 있다. Finish Mode 설정을 통해서 메인 테스크가 완료되면, 서브 트리를 중단시키고 즉시 노드를 완료시킬지, 아니면 서브 트리를 완료할 때까지 지연시킬지 설정 가능 -> 이것을 이용해 이동하면서 공격 구현 가능.

Task ->  
AI의 이동이나 블랙보드의 값 조정과 같은 작업을 하는 노드로 이 노드에도 데코레이터가 붙을 수 있음.

임시 구조 :

시야 안까지 다가가서 사정 거리 내로 들어오면 eqs 쿼리로 찾아서 정확한 위치 구해서 공격.

멀어지면 다시 쫓아감, 시야에 가로막히는게 있었다면 최종적으로 확인한 위치까지 이동.

4초 뒤에 어그로 풀림. ( 시야 관련은 AIC안에 Perception 참조 , 데미지를 별도로 쓰고 있으므로 퍼셉션에 데미지 말고 별도의 하나를 만들어야 할 듯 / 문서 참고해야 함. )

타겟을 못찾았으면 순찰.

//

피격 반응도 Perception을 이용하여 구현.

별도의 커스텀 Damage Perception 를 만들어서 Tag를 같이 보낼수 있도록 변경함

-> 큰 데미지나 뭔가 데미지 타입을 받은 것에 따라 반응을 다르게 만들 수 있음.

//

Dynamic BT는 게임플레이 태그를 통해 인젝션 위치를 지정할 수 있음.

게임플레이 태그인 트레잇에 따라 HighPriority 트레잇과 LowPriority 트레잇 태그를 나눠서 세분화함.

( 행동 양식에 변화를 주는 트레잇에만 High/Low 의 하위로 생성 )