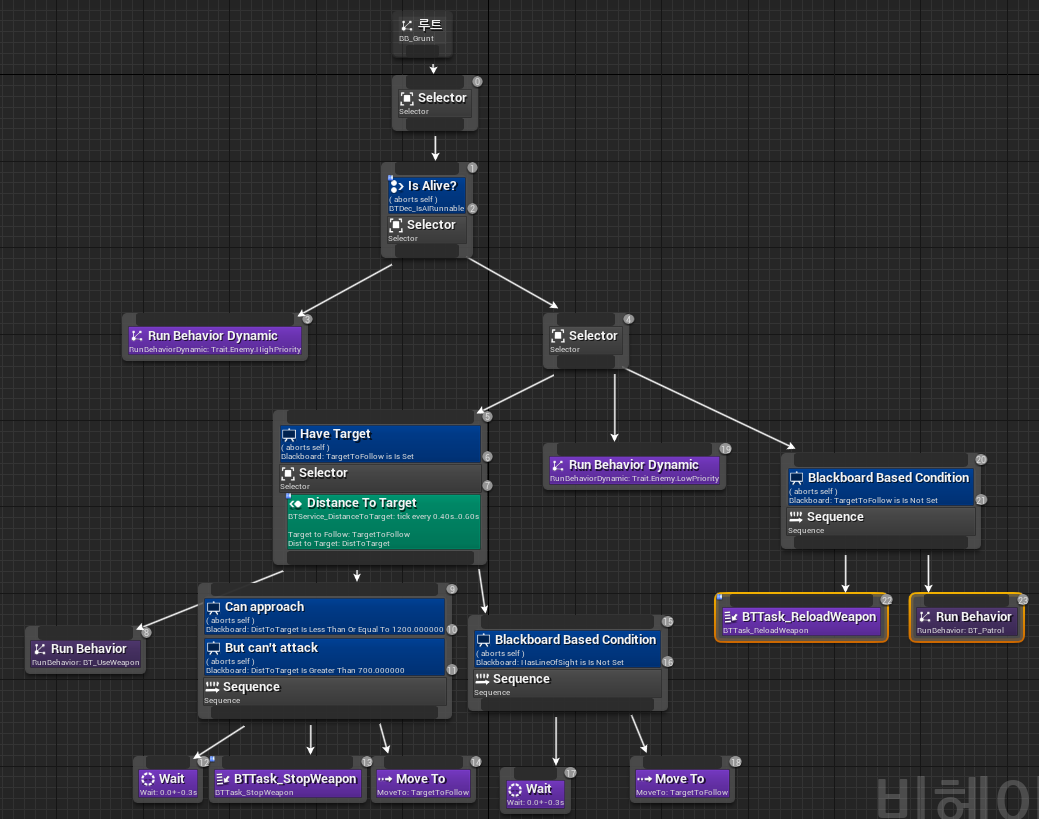
**AI 구현**

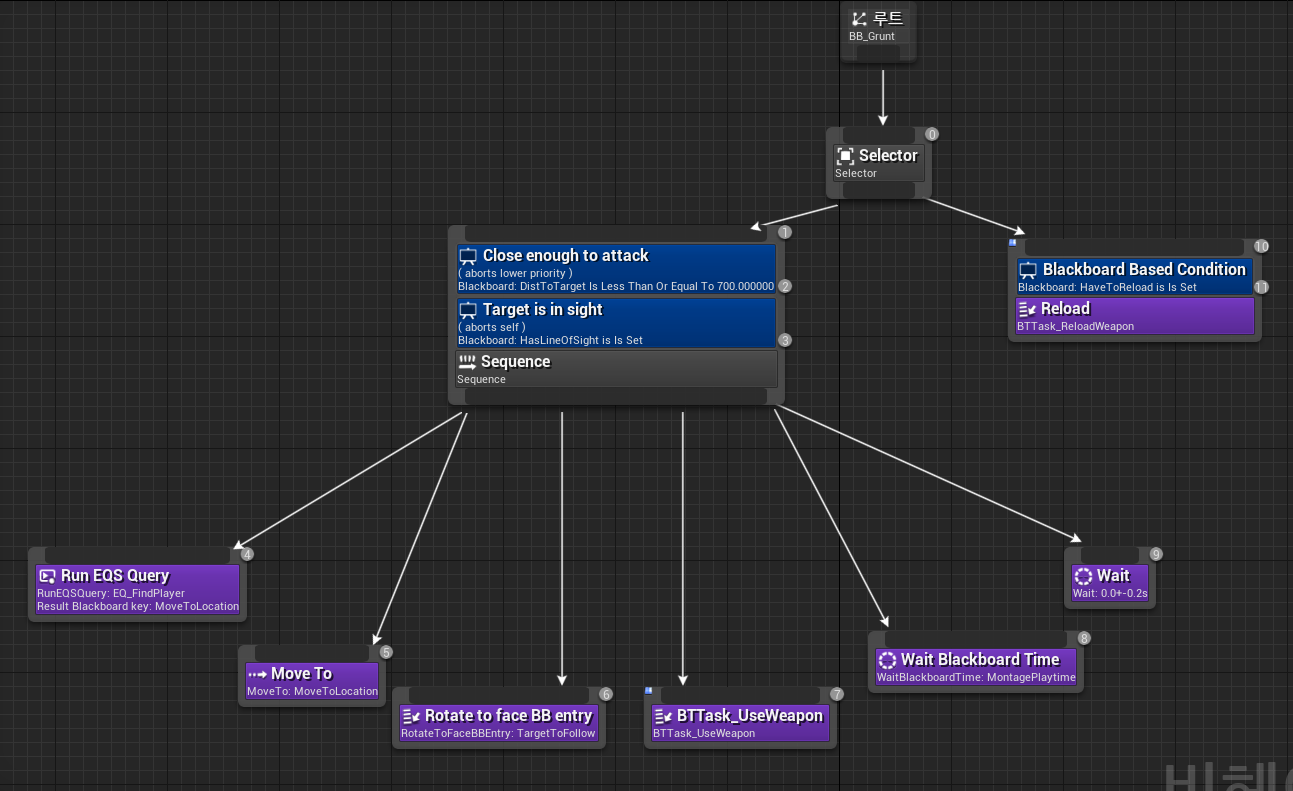
**:: Behavior Tree 구성 ::**

언리얼이 제공하는 AI 구성요소중 BT라는 일종의 스테이트 머신이 있습니다. AI Controller는 이 BT와 BB (BlackBoard : BT가 사용할 값들을 저장하는 요소) 를 사용하여 캐릭터를 조종합니다. 어떤 BT를 사용할지, BT에 어떤 서브 BT를 넣어서 사용할 지 동적으로 구성이 가능합니다

특정 행동 양식마다 BT를 만들어 밀리 캐릭터 / 레인지드 캐릭터 같은 좀 더 고수준의 BT를 행동 양식 (저수준BT) 여러 개를 합쳐서 만드는 형식을 취합니다.



BT\_Grunt ( 캐릭터가 사용하는 BT 예시 )



BT\_UseWeapon ( ‘공격’ 행동 만을 위한 BT )

트레잇에 따른 BT 동적주입 :

위에서 BT\_Grunt에 있는 RunBehaviorDynamic 이 Sub BT를 동적으로 인젝션 할 수 있도록 만들어 놓은 일종의 슬롯입니다. AI Con이 초기화될 때 캐릭터의 Trait을 확인하여 그에 맞는 Sub BT를 맞는 위치에 인젝션 시켜줄 수 있도록 구현 되어있습니다.

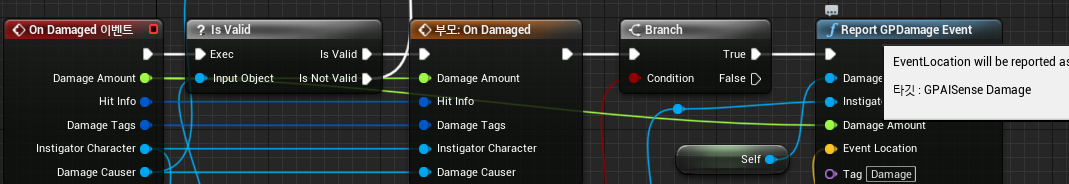
~~--------------------------------------------------------------------------------------------------------------~~

**:: AI의 정보 습득 ::**

언리얼이 제공하는 AI가 정보를 습득하는 방법에는 여러가지가 있겠지만 현재 사용 중인 요소들에 대해서만 설명합니다

**Perception** : 일종의 감각 같은 개념으로 AI Character 의 시각 / 청각 / 데미지 반응 ( 커스텀 데미지 ) 에 대한 처리를 미리 구현된 것을 사용하거나, 별도로 구현하여 사용 할 수 있습니다.

현재 시각에 대한 부분은 기본적으로 구현된 것을 사용하며 데미지 같은 경우 저희의 커스텀 정보이기 떄문에 별도로 구현하여 사용합니다. -> 캐릭터 단에서 커스텀 데미지를 입었을 때 AI 퍼셉션 시스템에 리포트를 하는 함수를 호출하는 형태로 되어있습니다.



**Environment Query** :

EQS는 게임속 환경에 대해 질문을 생성하고, 질문을 던지는 하나의 방법입니다. 현재 게임에서 사용중인 예로는 범위 내에 플레이어 캐릭터가 어디에 있는지를 찾아내는 용도로 사용 하고 있습니다. -> 이를 적용시켜보면 EQS는 플레이어 캐릭터를 찾을 수 있는 가장 최적의 위치를 찾아서 ( 좌표단위가 아닌 어느정도 그리드 단위로 맵을 나누어서 계산합니다 -> 그리드 크기도 조절 가능 ) 그 위치를 저장 / 전달 하여 BT가 그 값을 기반으로 해당 장소로 이동 하는 방식을 취하고 있습니다. 그리고 위의 퍼셉션과 연계하여, 실제로 플레이어 캐릭터를 눈으로 보게된다면 타겟이 정해지고, 그 타겟을 향해 쫓아가거나 공격을 하는 형태가 됩니다.

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

Flashback ( NOTE )

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.

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

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 )