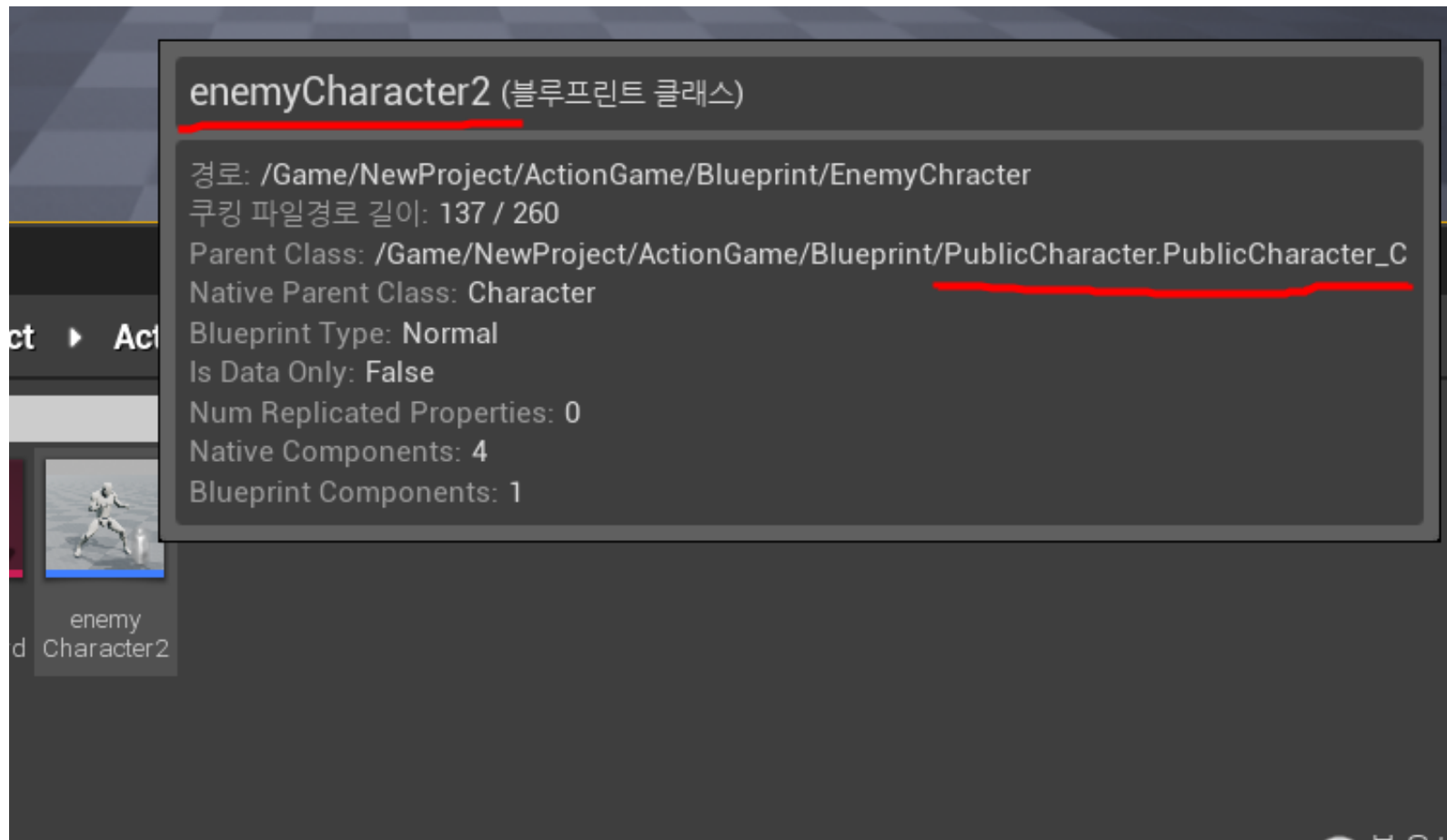


목차

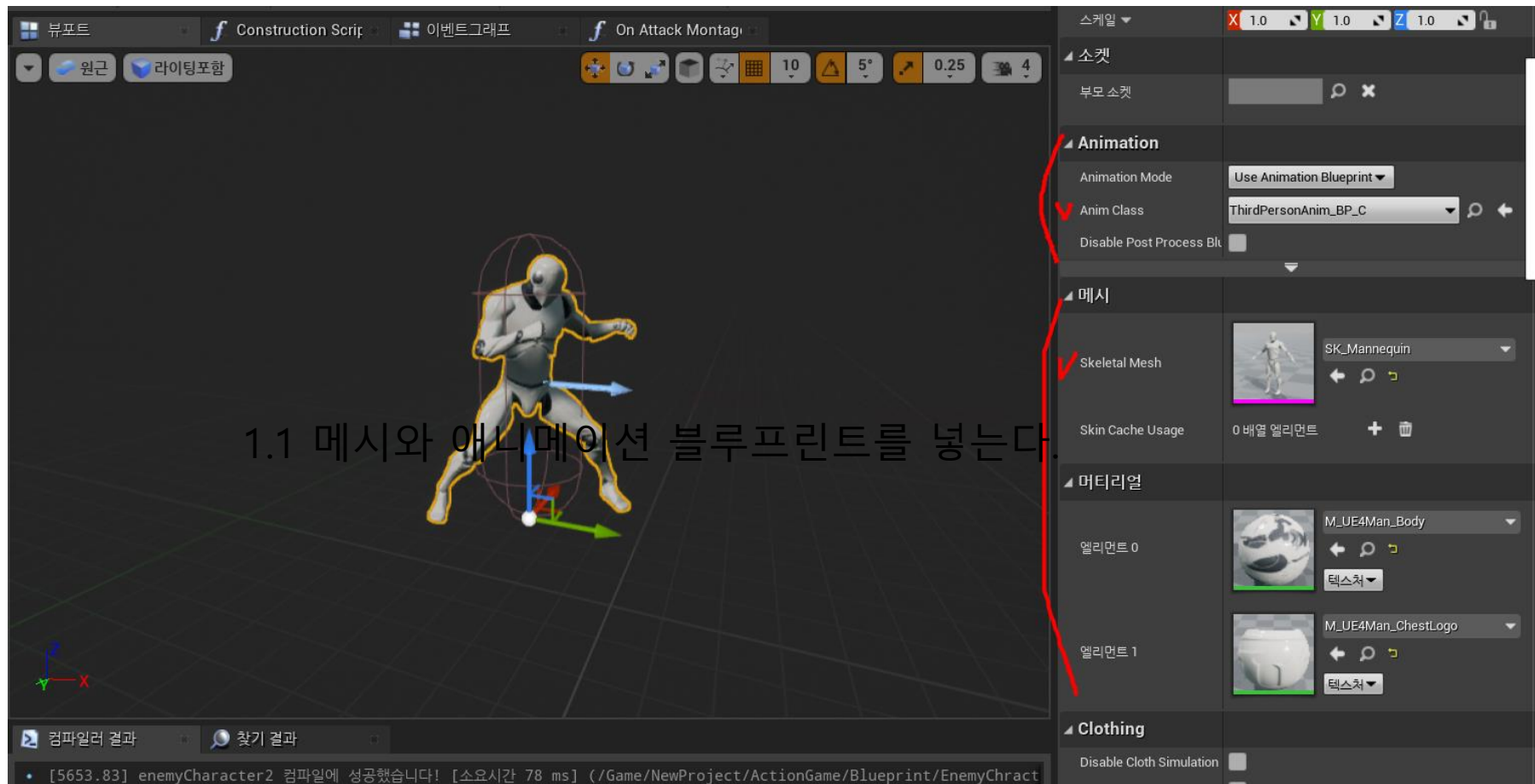
1. **PublicCharacter**를 상속받은 **enemyCharacter2**를 생성한다.
2. **AIController**를 상속받은 **ABAIController**를 생성한다.
3. **enemyCharacter2** 처리1
4. **Blackboard EnemyBlackboard**를 생성한다.
5. **Behaviortree EnemyBehaviortree**를 생성한다.
6. **BTService_BlueprintBase**를 상속받은 **BTService_Detect**를 생성한다.

목차

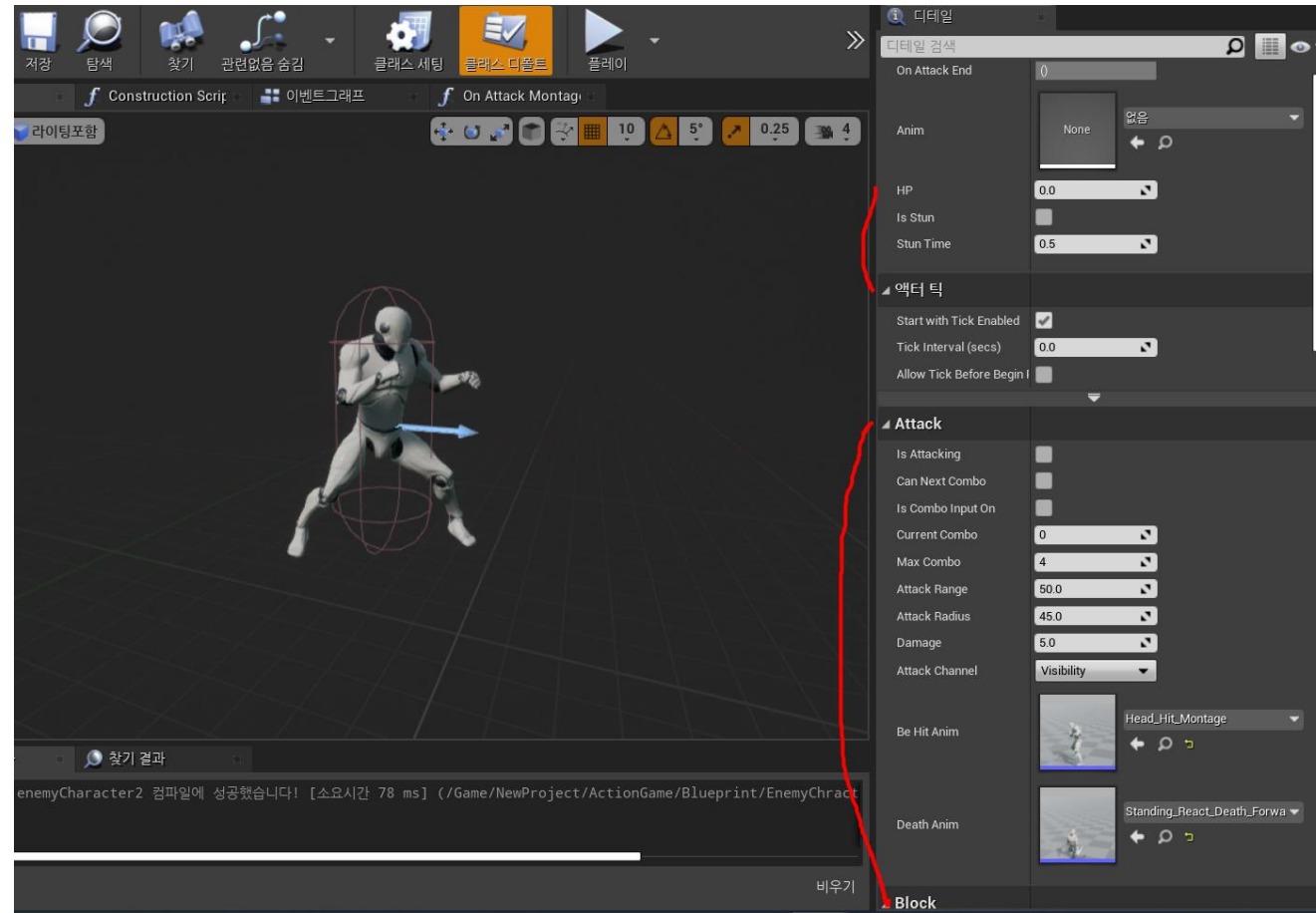
7. BTDecorator_BlueprintBase를 상속받은 BTDecorator_IsInAttackRange를 생성한다.
8. BTTask_BlueprintBase를 상속받은 BTTask_Attack을 생성한다.
9. BTTask_BlueprintBase를 상속받은 BTTask_TurnToTarget을 생성한다.
10. BTDecorator_BlueprintBase를 상속받은 BTDecorator_Follow를 생성한다.
11. enemyCharacter2 처리2



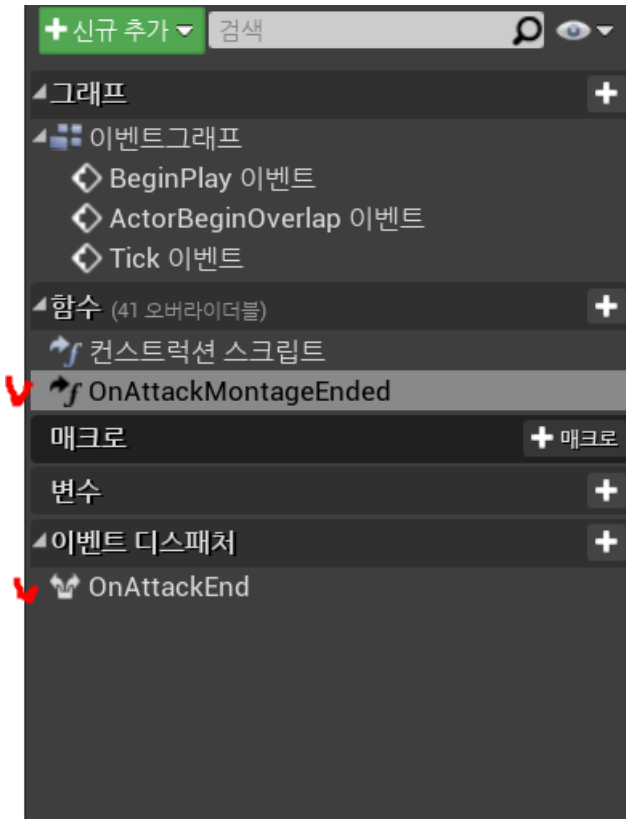
1. PublicCharacter를 상속받은 enemyCharacter2를 생성한다.



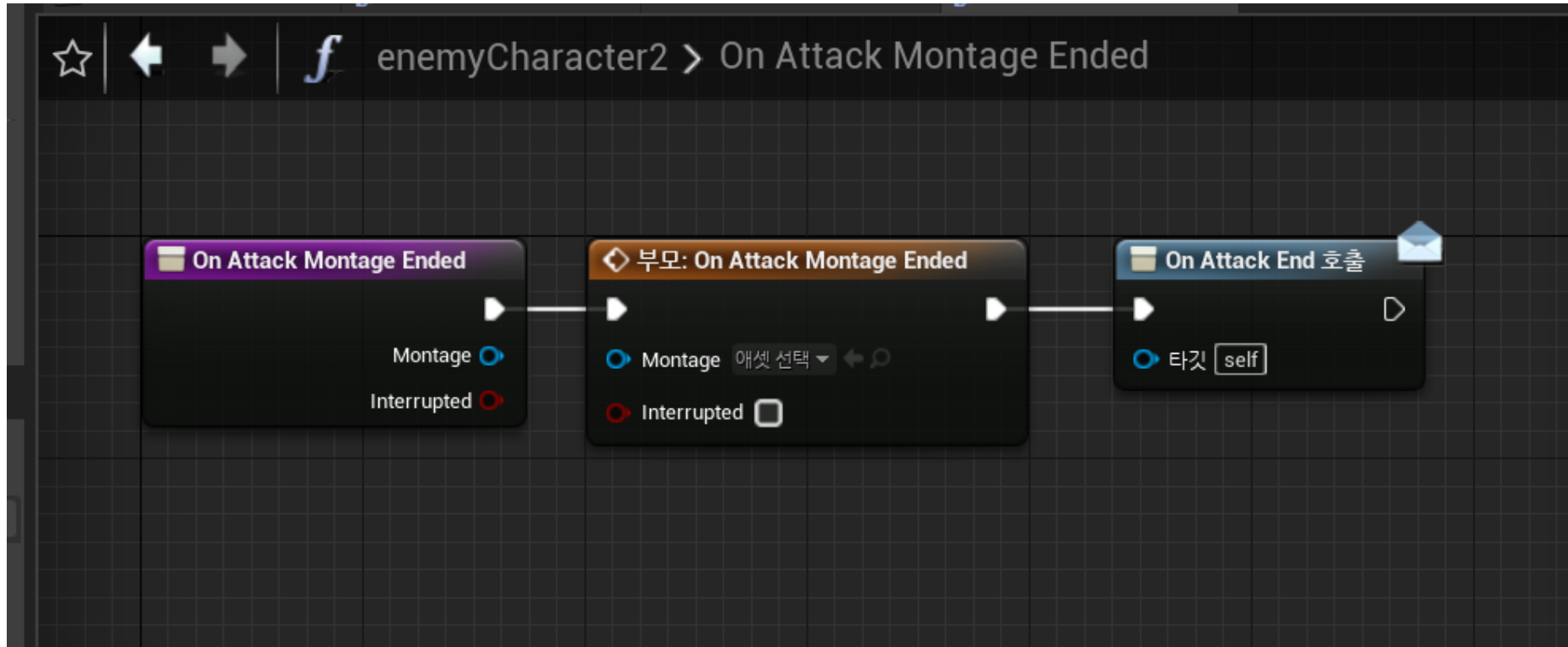
1.1 메시와 애니메이션 블루프린트를 넣는다.



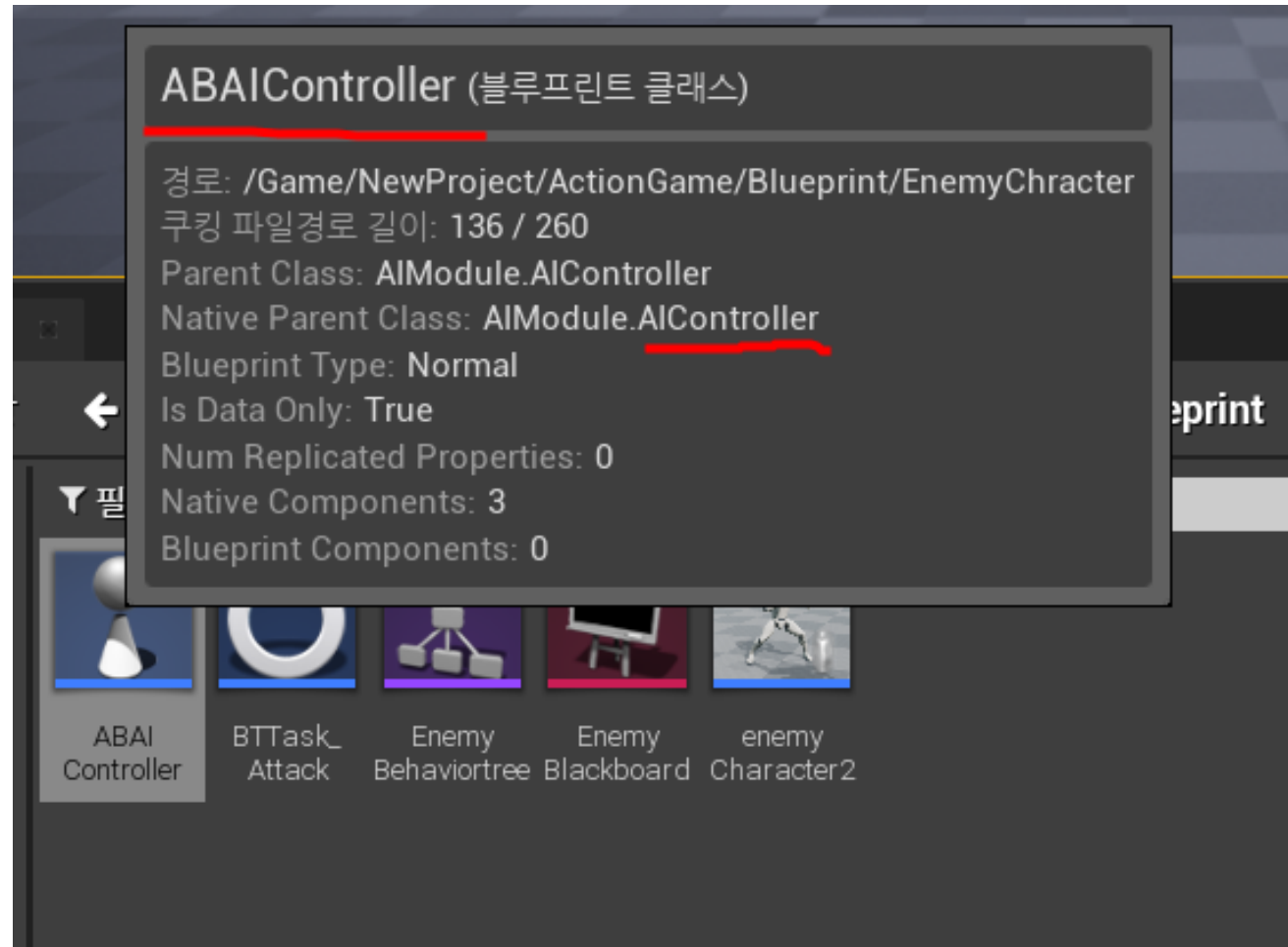
1.2 상속받은 변수들의 값을 조절한다.



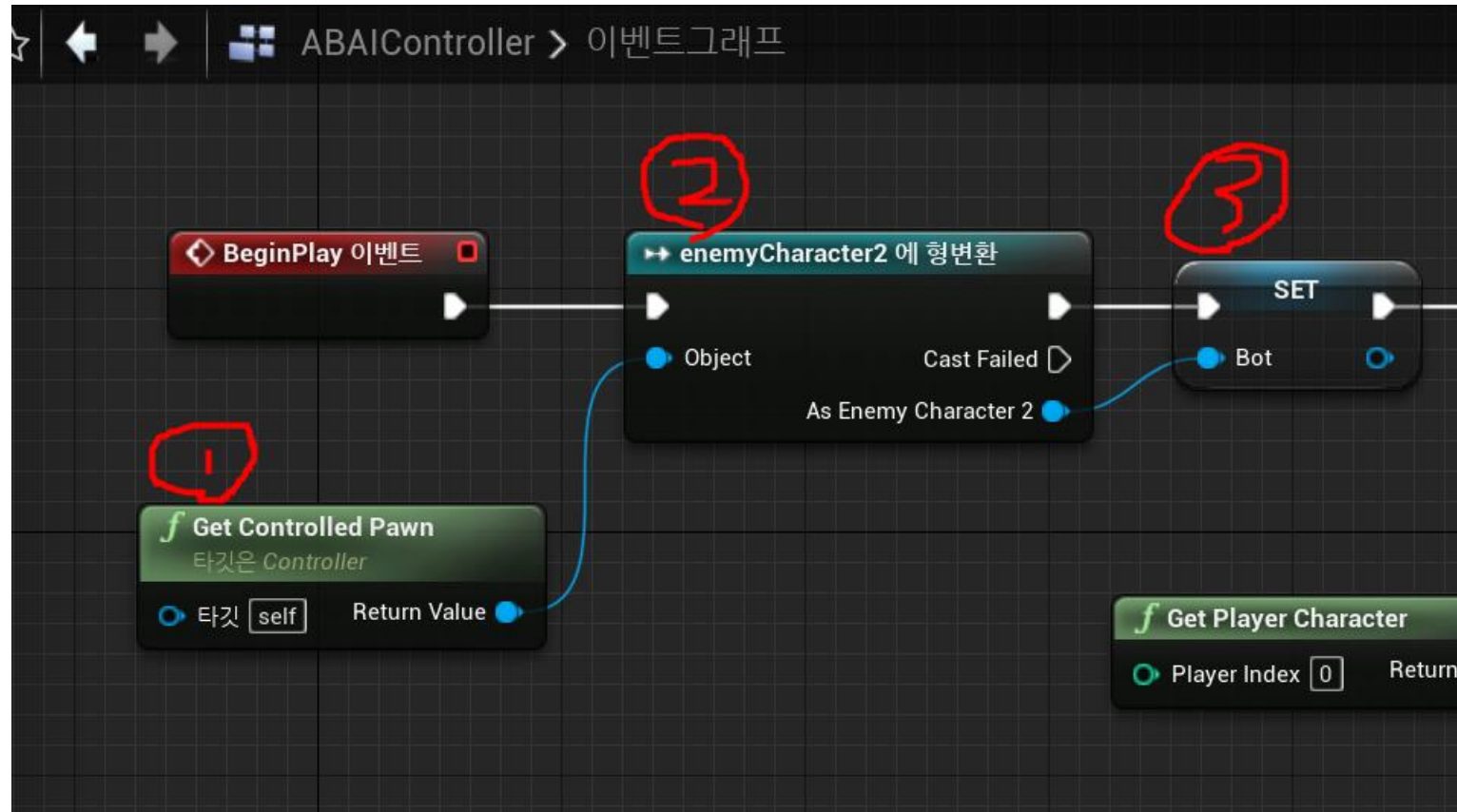
1.3 이벤트 디스패처 OnAttackEnd를 추가, OnAttackMontageEnded 함수를 오버라이드 한다.



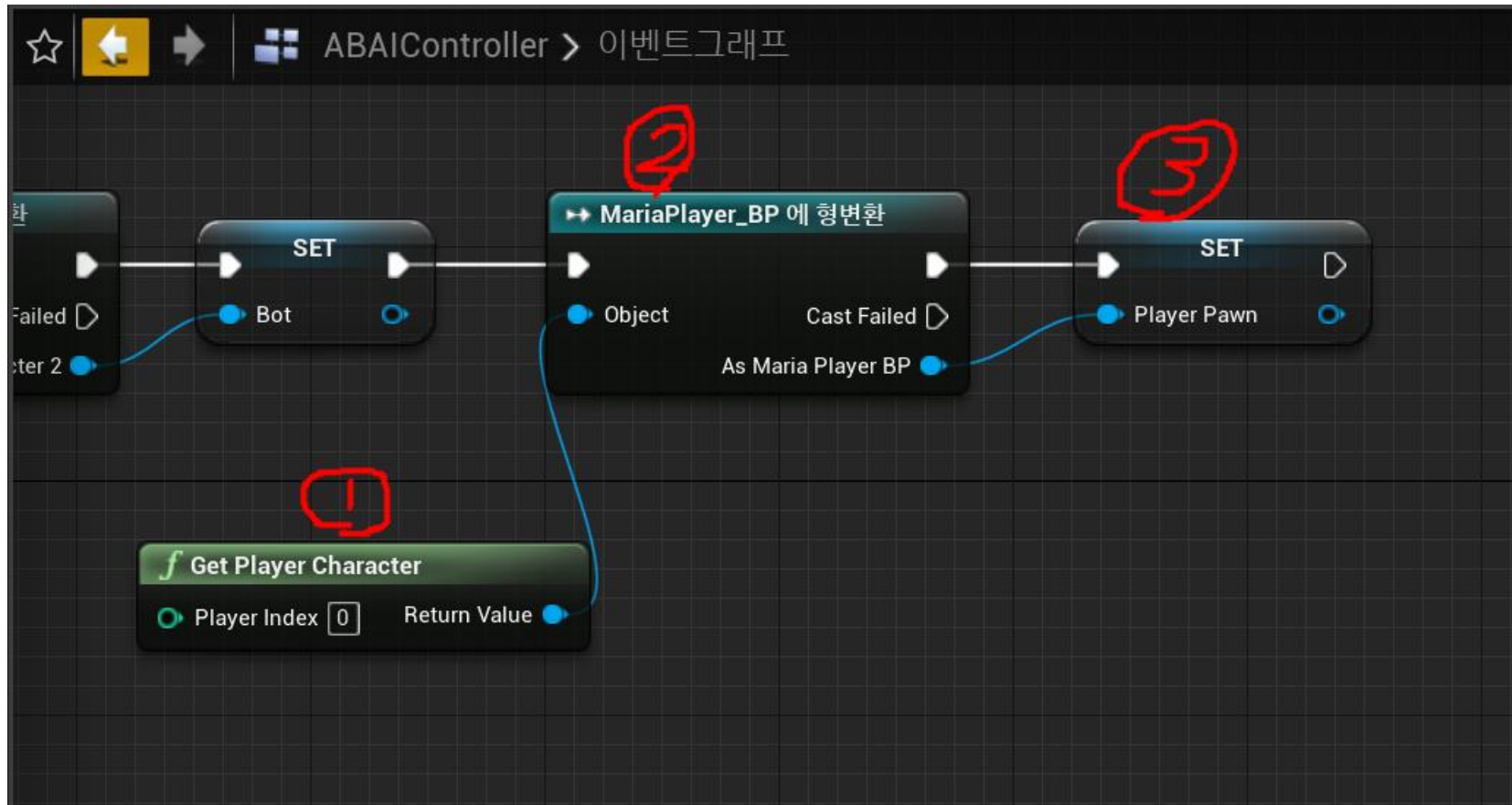
1.4 OnAttackMontageEnded 함수에 OnAttackEnd를 호출한다.



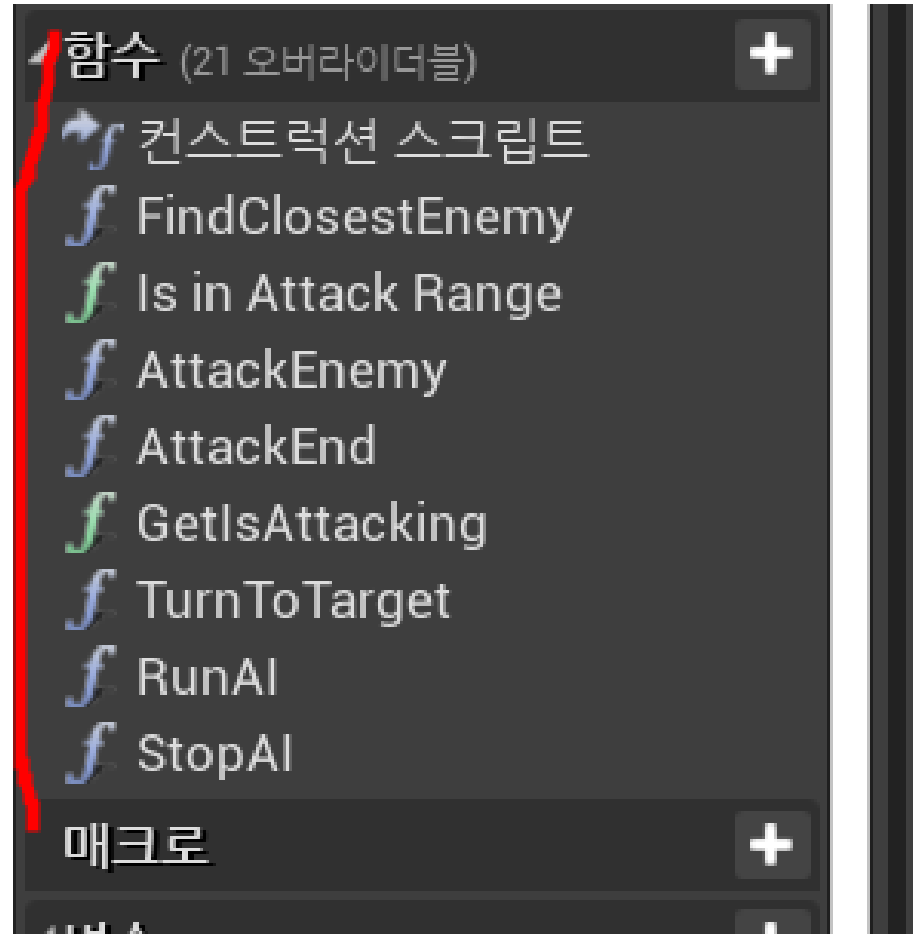
1.4 OnAttackMontageEnded 함수에 OnAttackEnd를 호출한다.



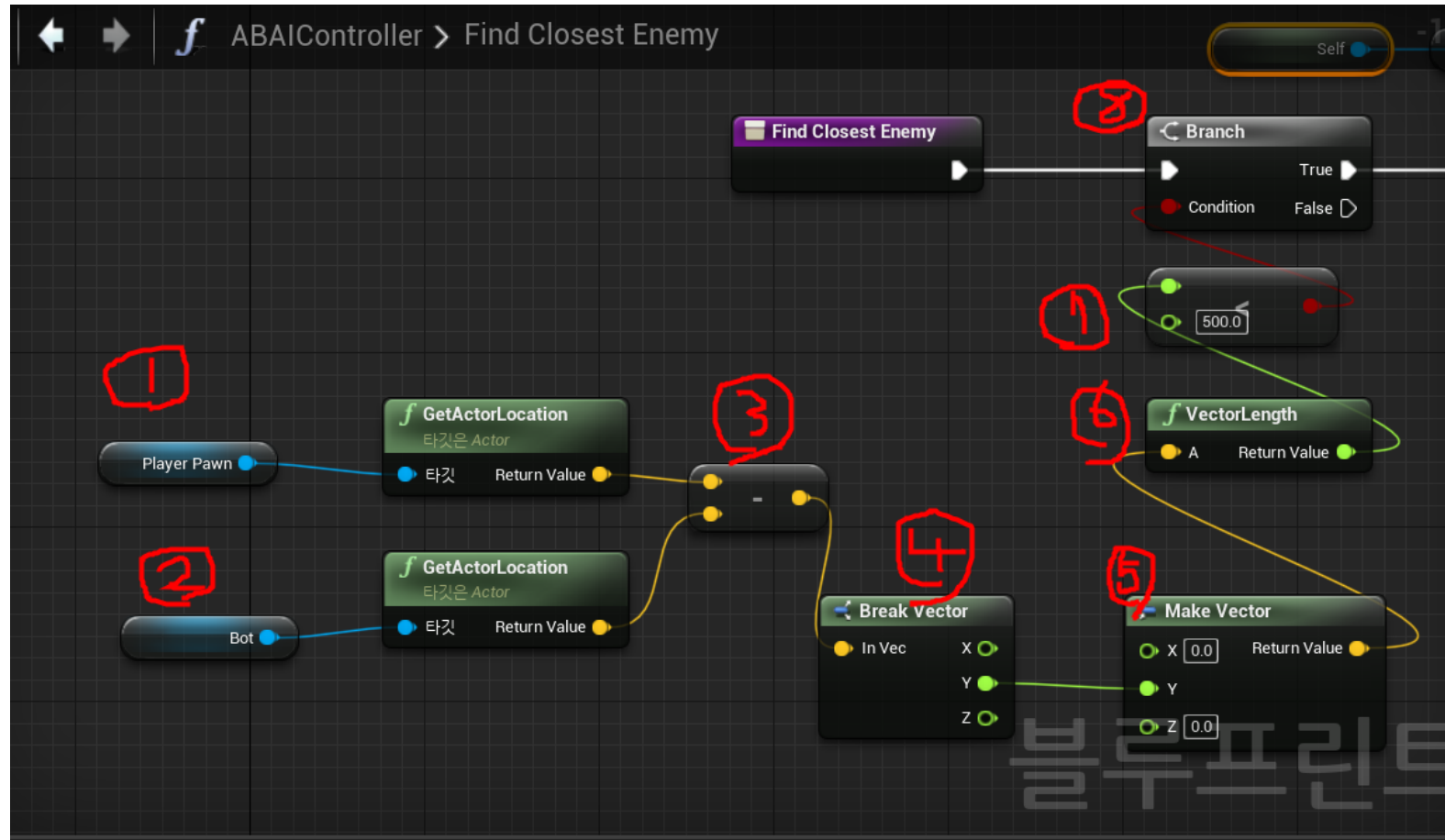
2.1 이 플레이어 컨트롤러가 컨트롤 중인 폰을 enemyCharacter2로 형변환하고 변수로 승격해서 Bot을 생성한다.



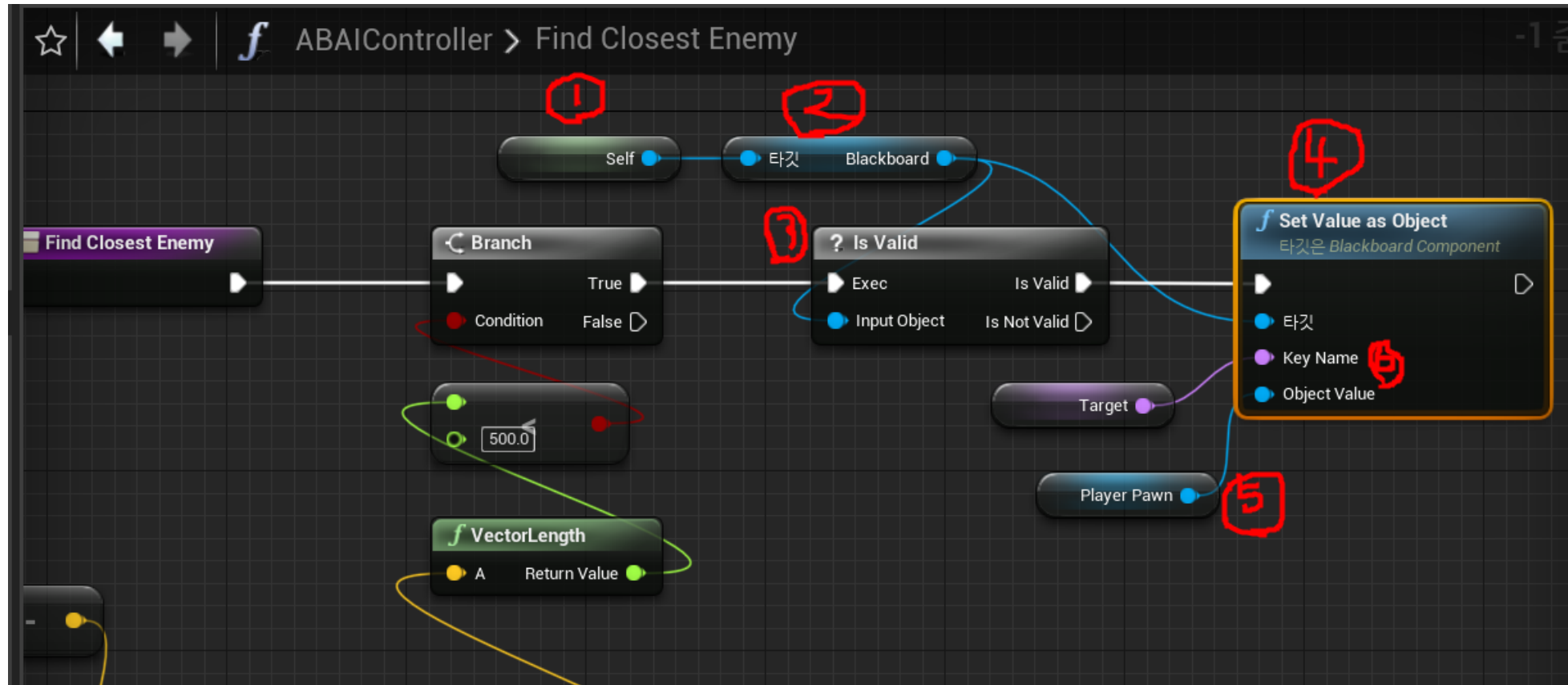
2.2 플레이어 캐릭터를 MariaPlayer_BP로 형변환하고 변수로 승격해서 PlayerPawn을 생성한다.



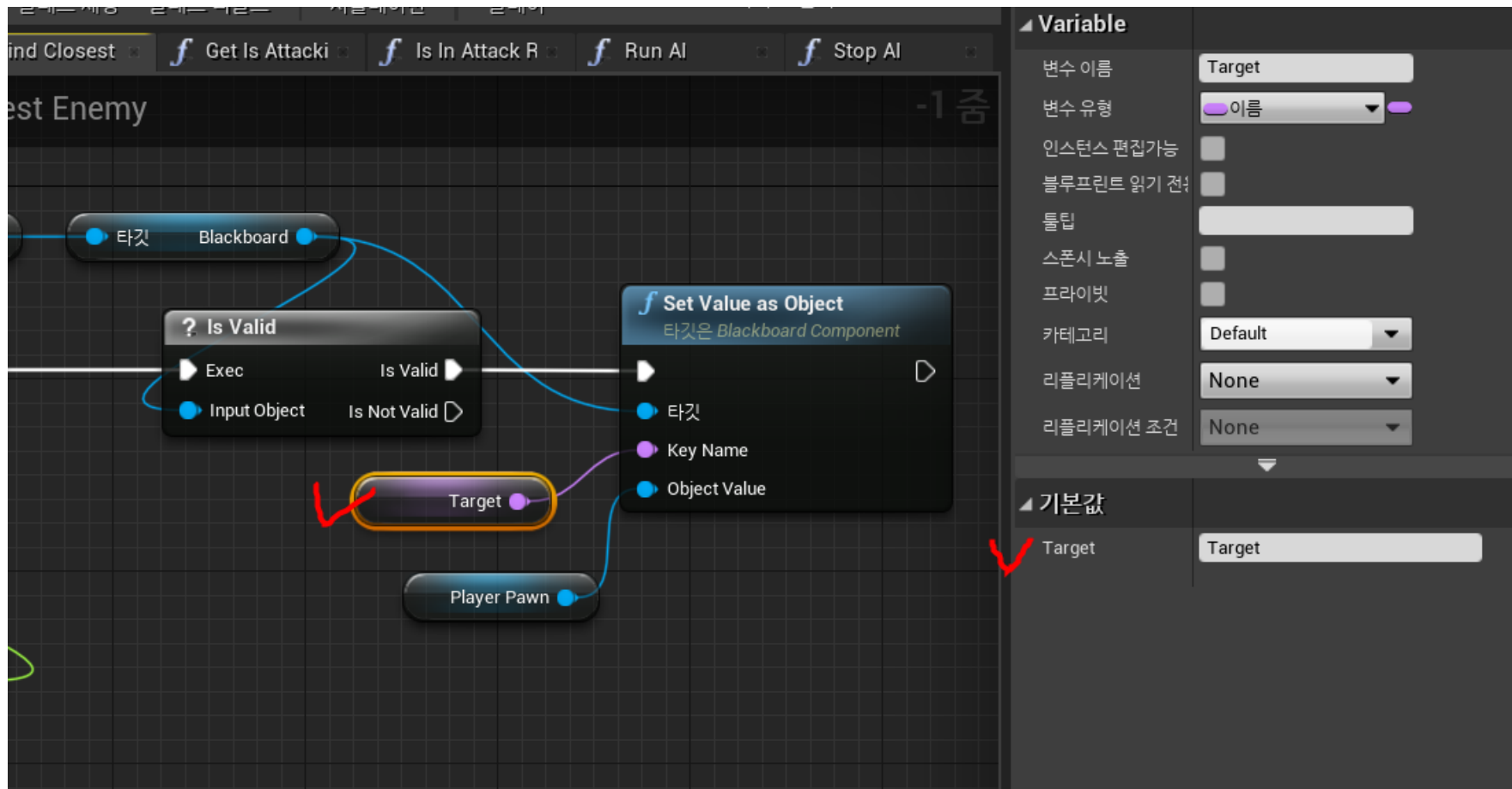
2.3 ABAIController에 추가한 함수들



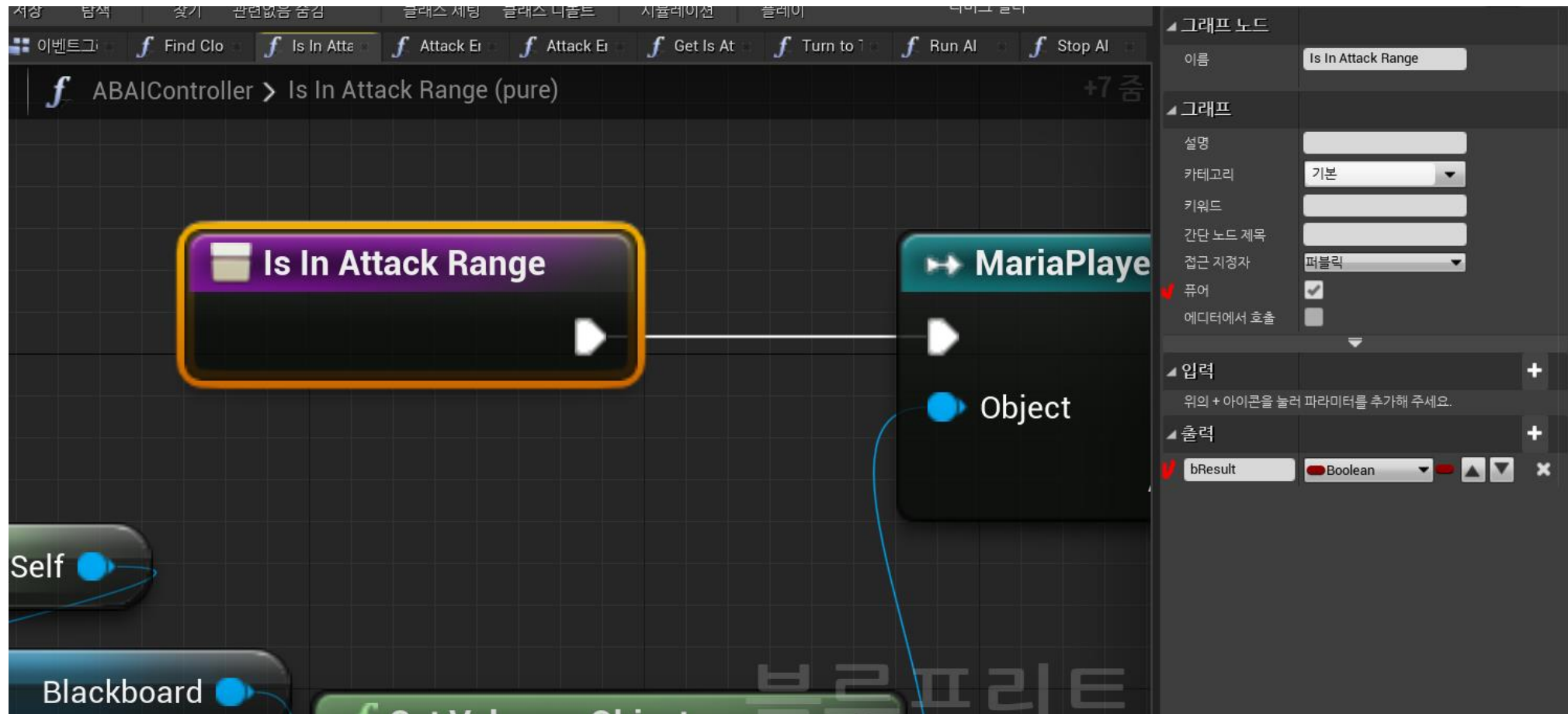
2.4 FindClosestEnemy 함수로 들어가서 PlayerPawn과 Bot의 위치를 뺀 거리가 500보다 작은지 판단한다.



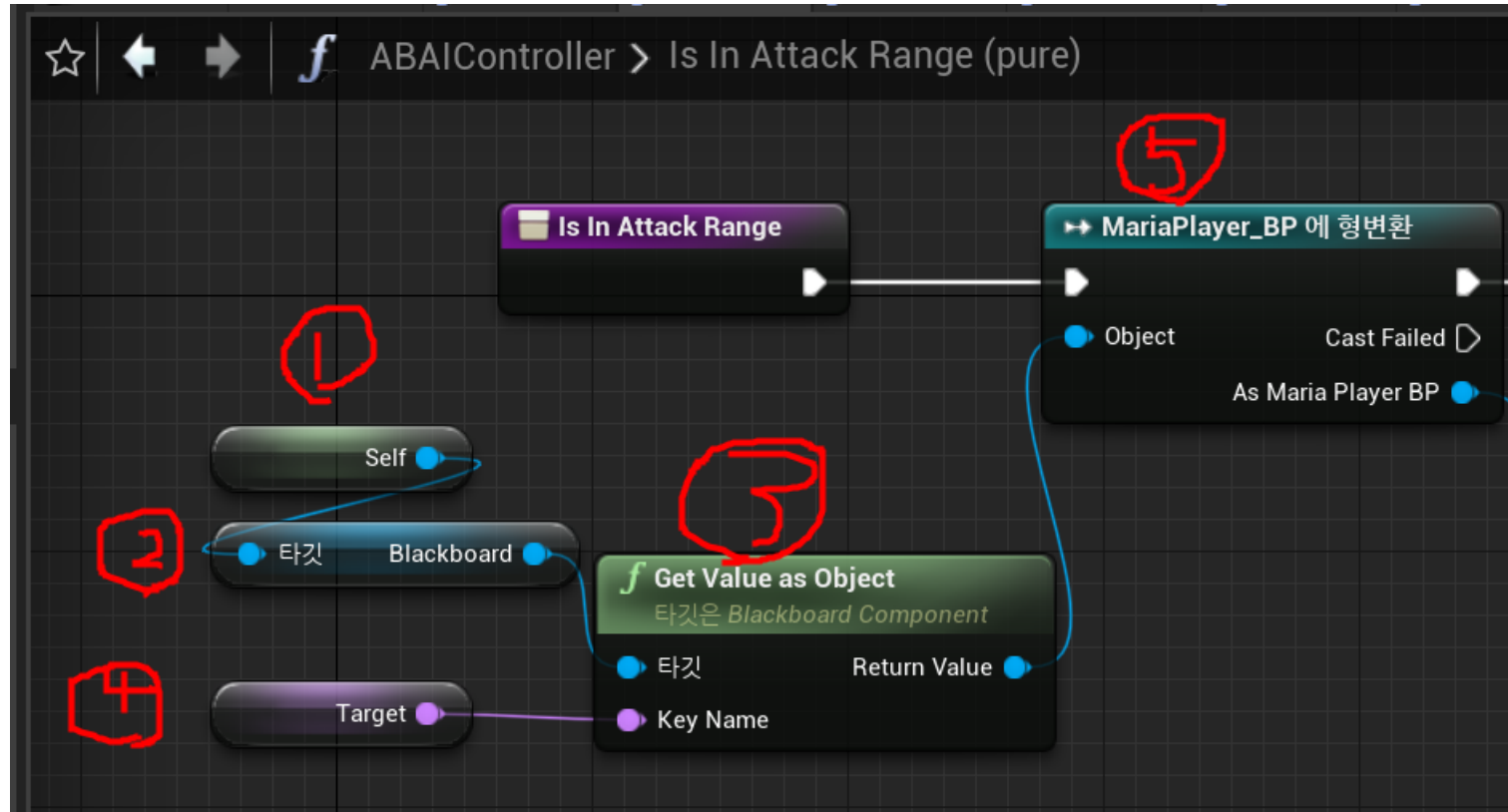
2.5 AIController에 있는 Blackboard가 유효한지 판단하고 유효하면 PlayerPawn을 Object에 넣고 KeyName을 변수로 승격해서 Target을 생성한다.



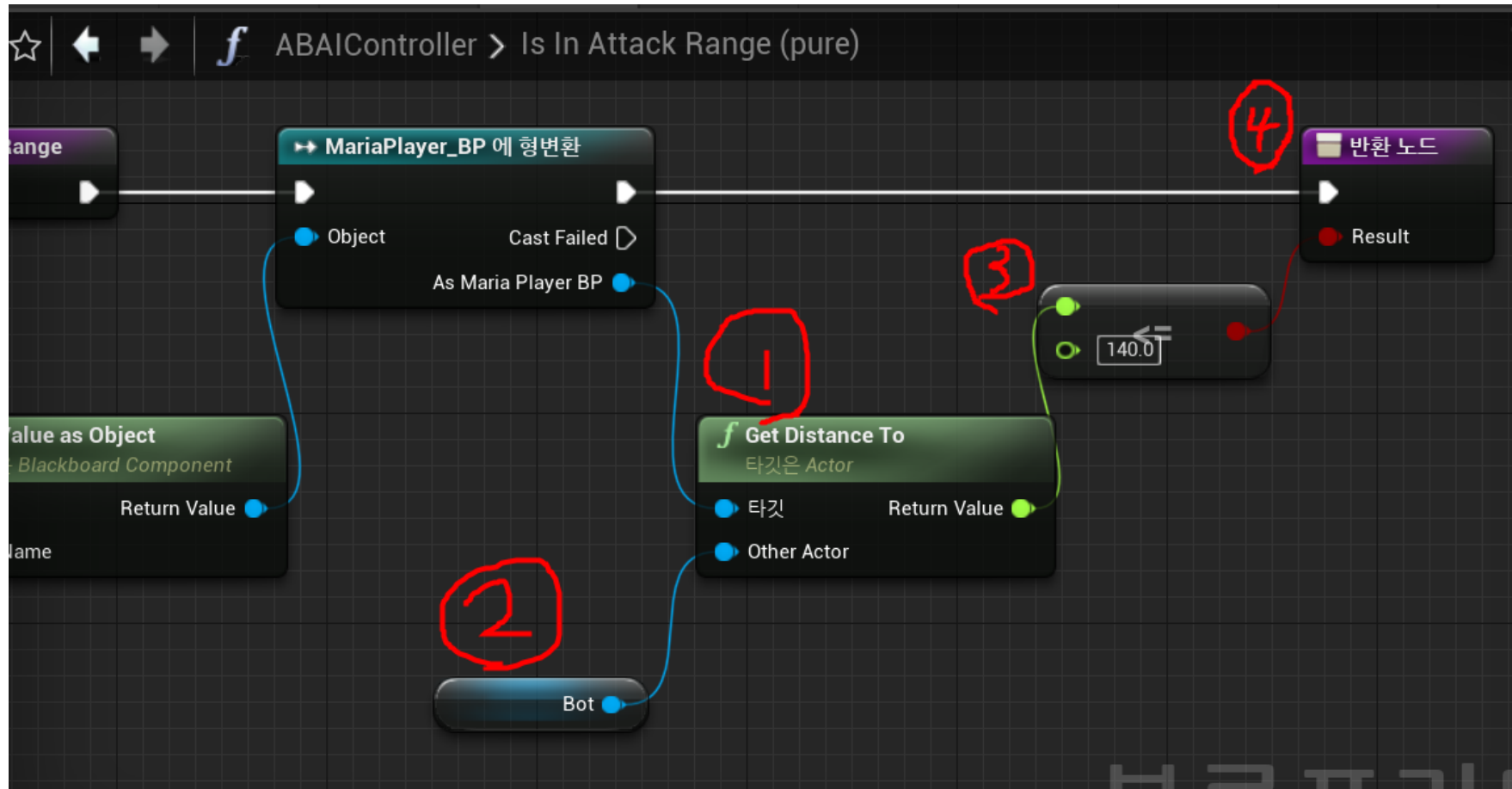
2.6 변수로 승격해서 생성된 Target의 값에 Target을 넣는다.



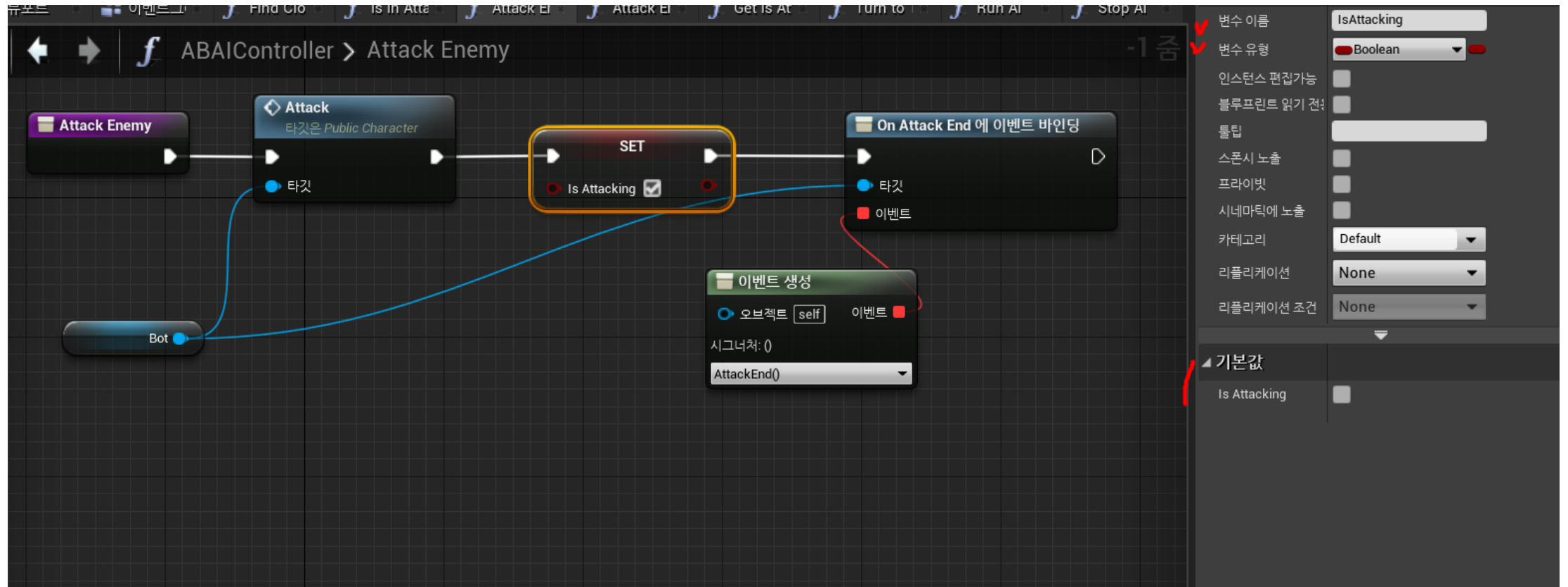
2.7 IsInAttackRange 함수에 들어가서 퓨어 항목에 체크하고, 출력에 boolean 형 bResult를 추가한다.



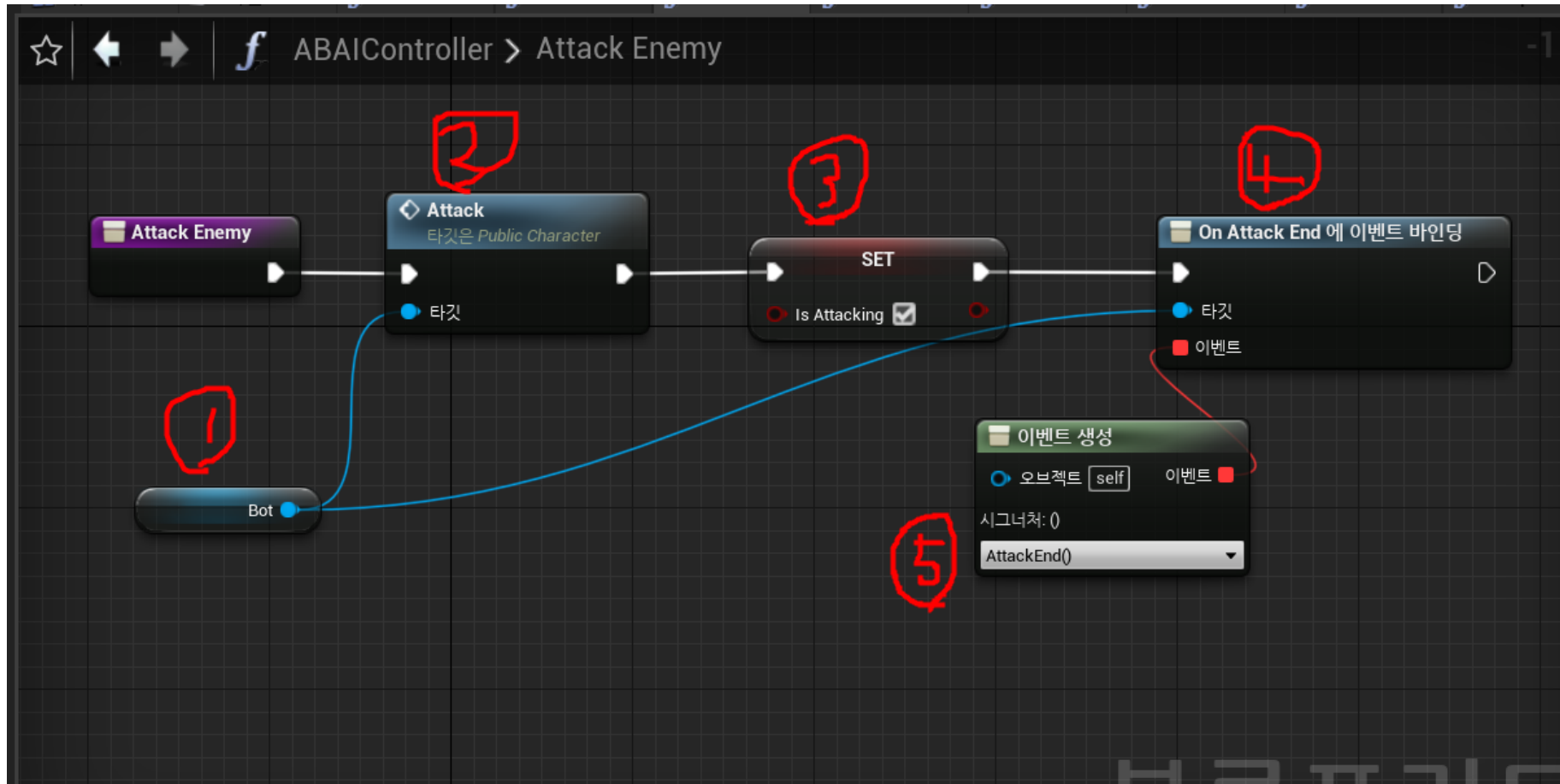
2.8 Blackboard의 Target의 Object를 MariaPlayer_BP로 형변환한다.



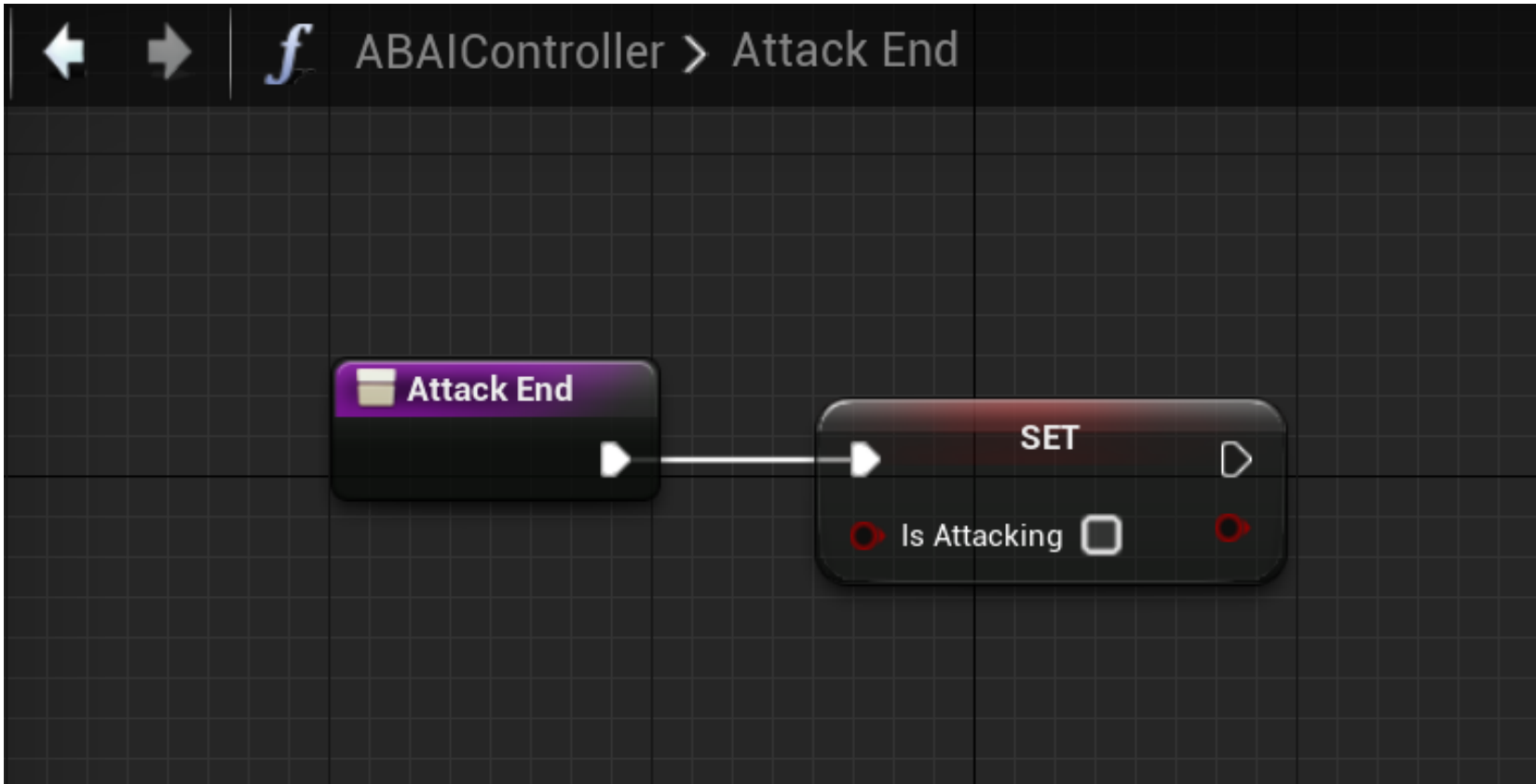
2.9 MariaPlayer_BP는 GetDistanceTo를 호출해서 Bot을 넣고 140보다 작은지 비교해서 결과값을 반환한다.



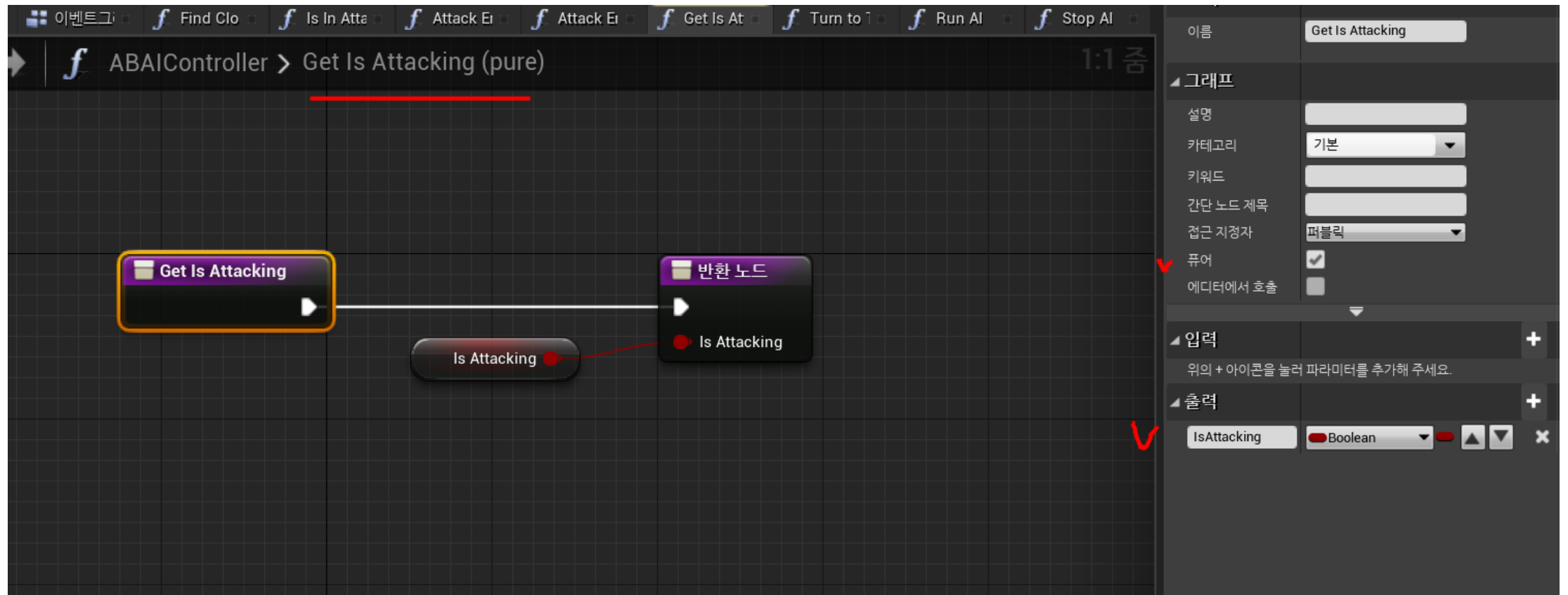
2.10 AttackEnemy 함수에 들어가기 전에 boolean IsAttacking을 선언한다.



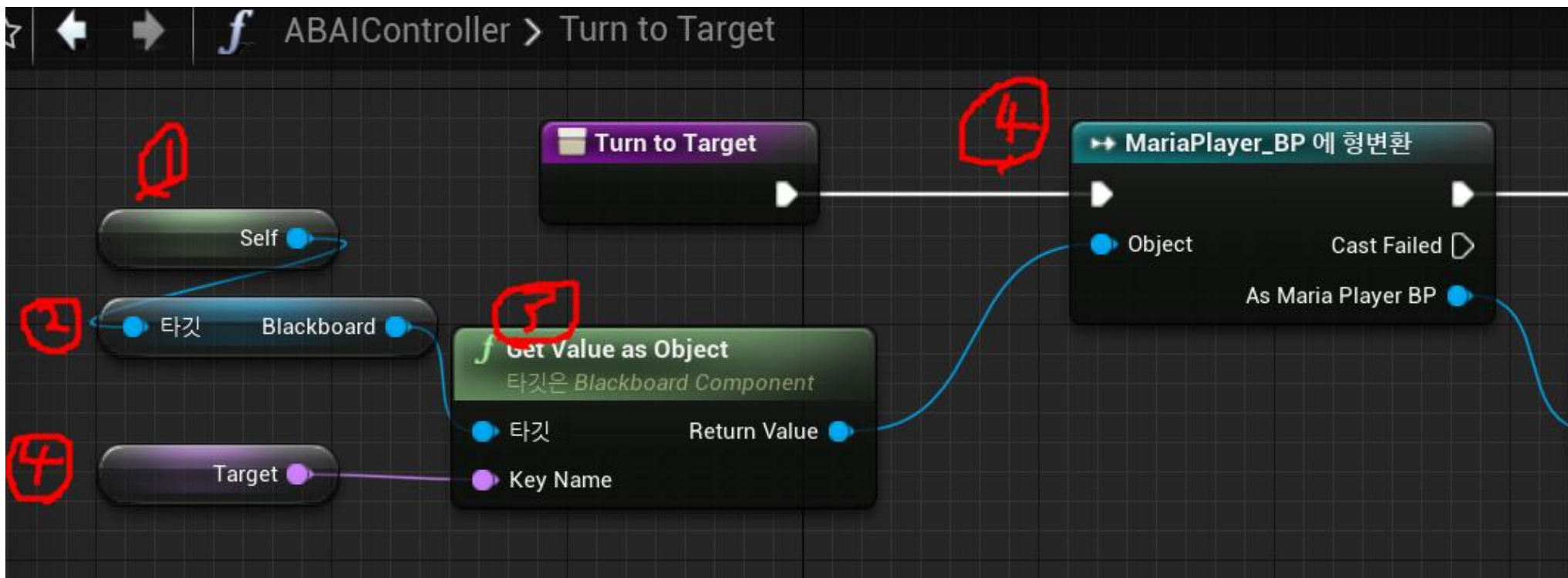
2.11 Bot의 Attack 함수를 호출, IsAttacking을 True로 SET, Bot의 OnAttackEnd를 바인딩, AttackEnd를 생성한다.



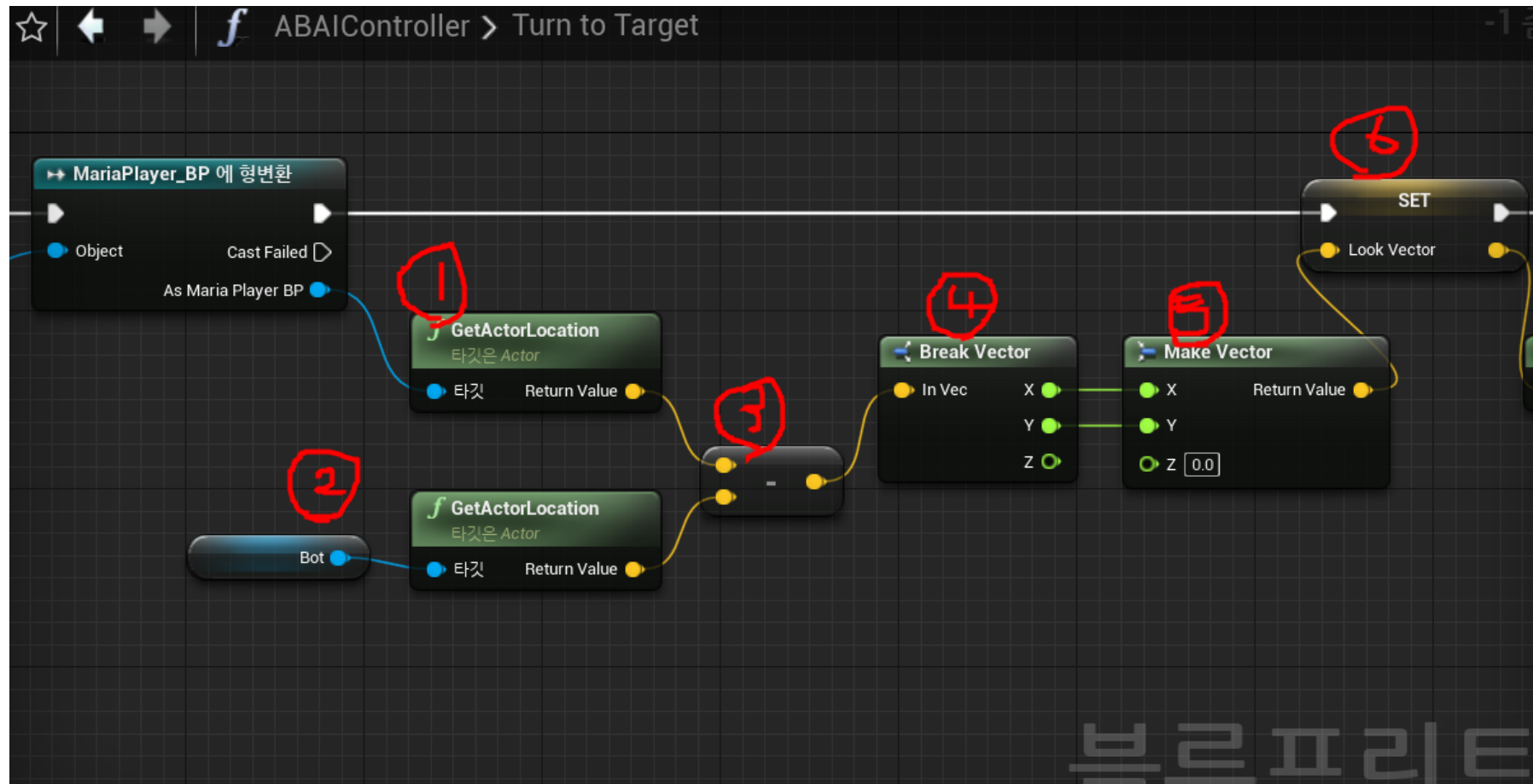
2.12 AttackEnd에 들어가서 IsAttacking을 False로 SET한다.



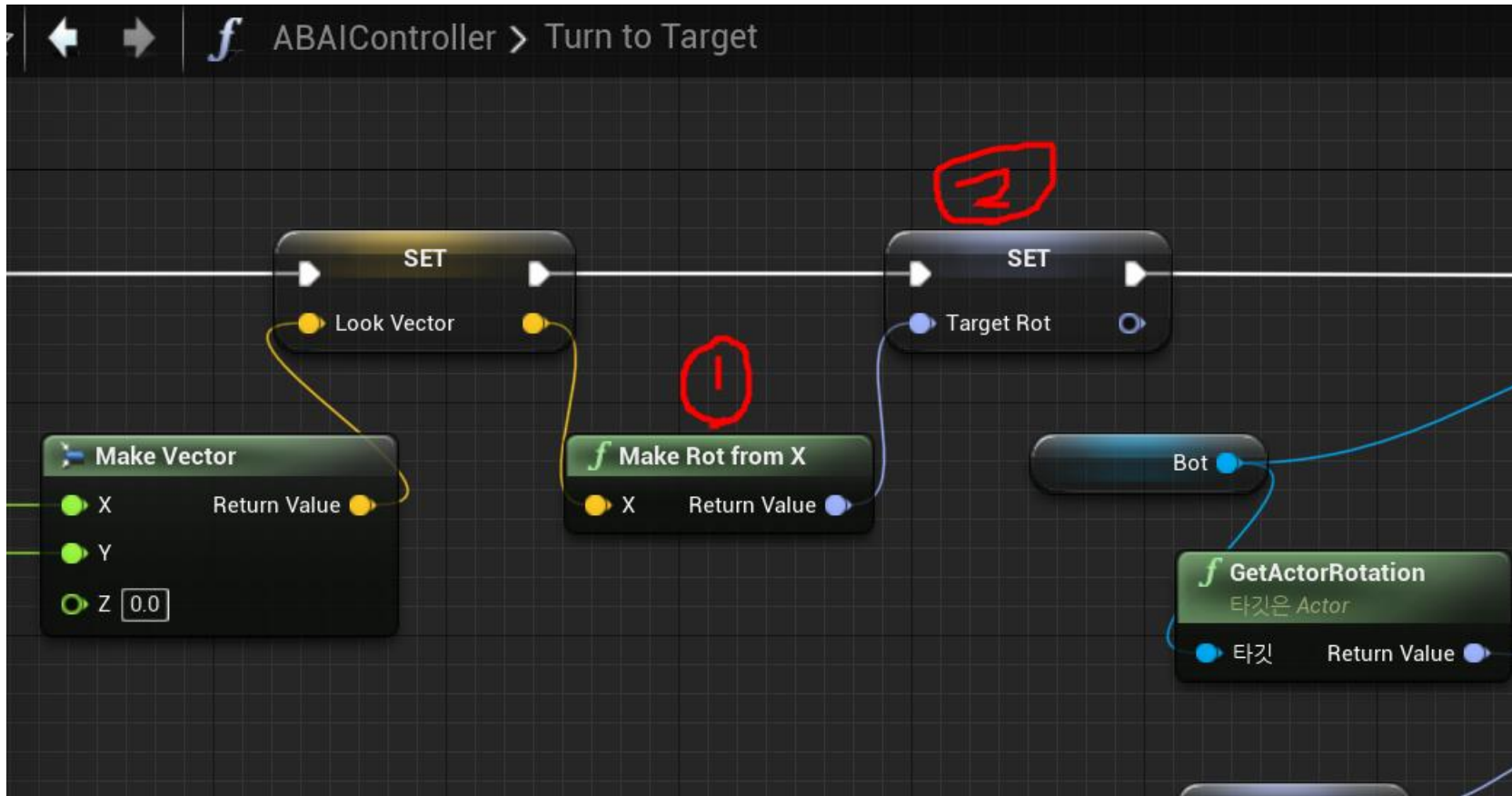
2.13 GetIsAttacking 함수에 들어가서 퓨어에 체크하고 IsAttacking을 반환한다.



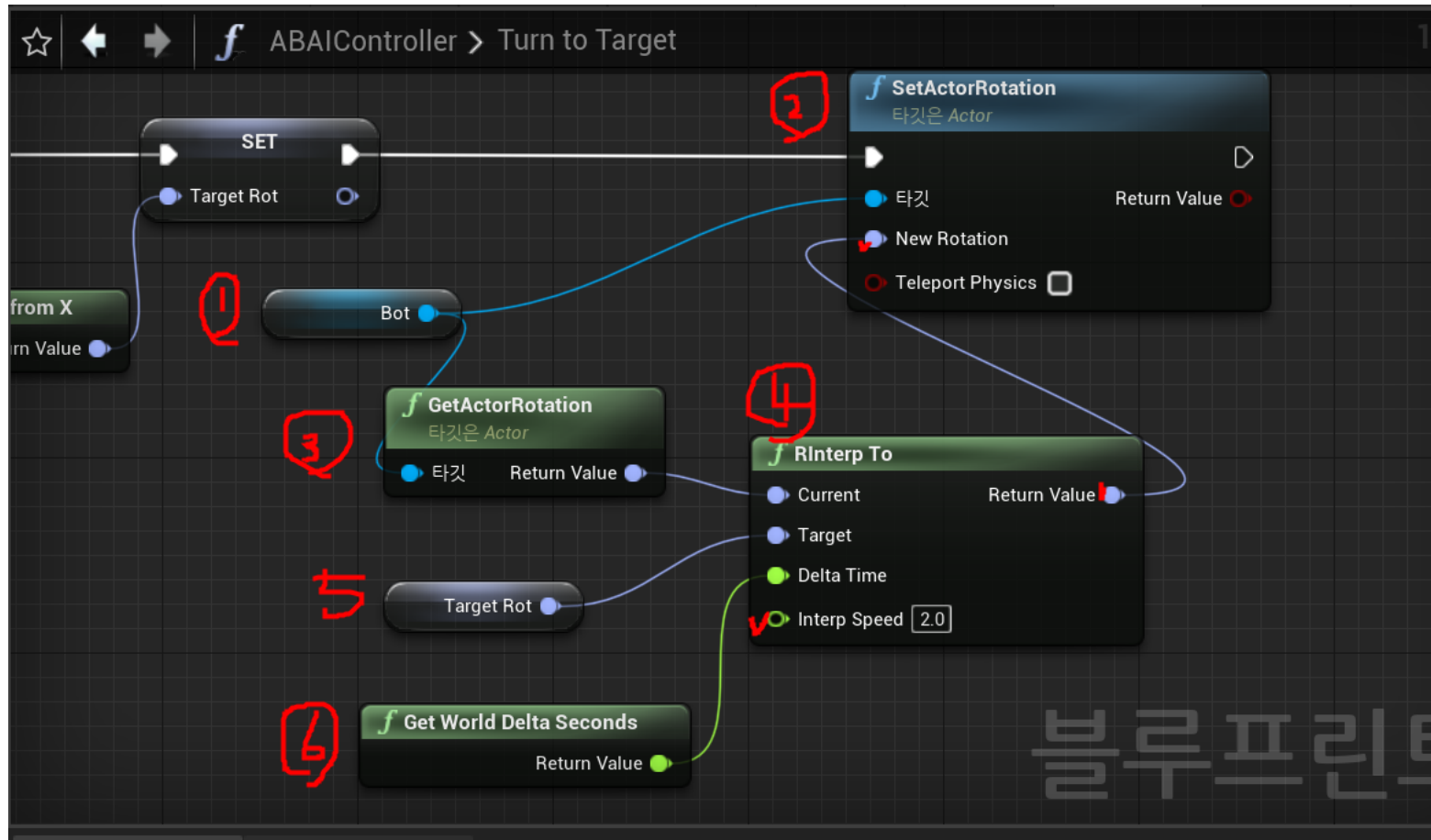
2.14 TurntoTarget 함수에 들어가서 Blackboard의 Target 키 Object를 MariaPlayer_BP로 형변환한다.



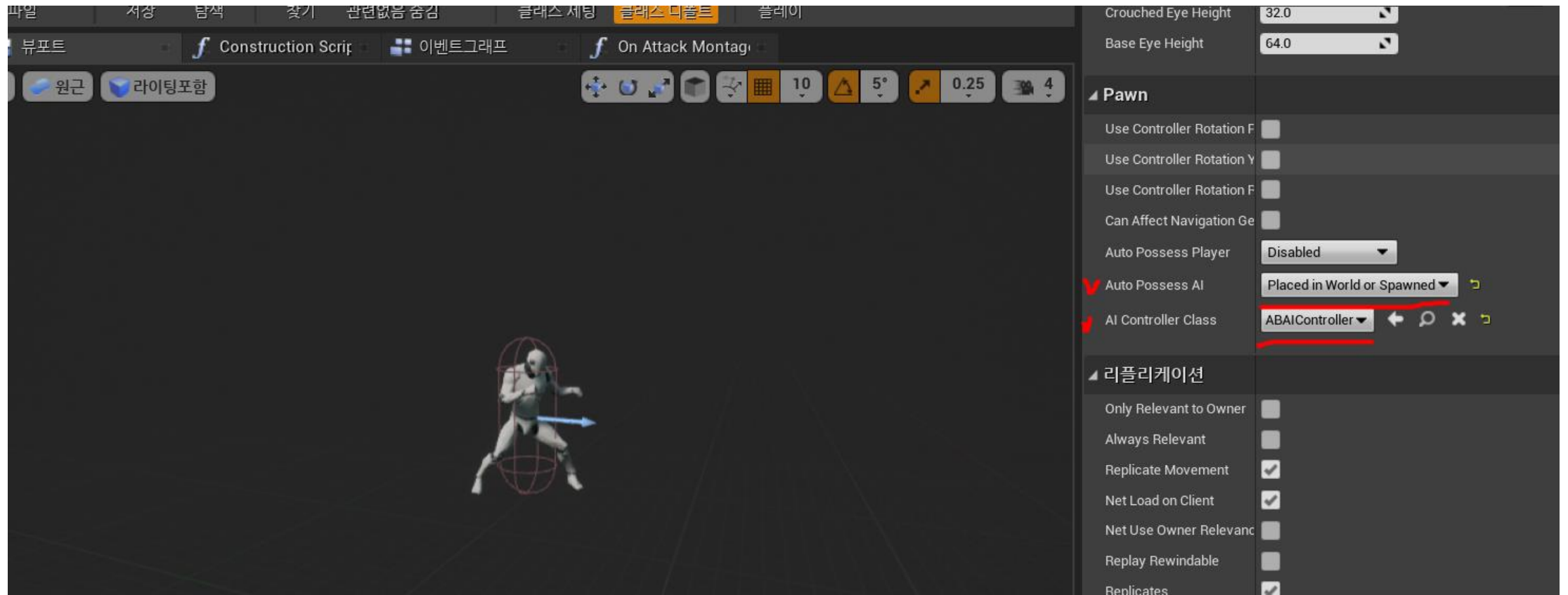
2.15 MariaPlayer_BP와 Bot의 위치를 빼고 XY 값을 변수로 승격해서 LookVector를 생성한다.



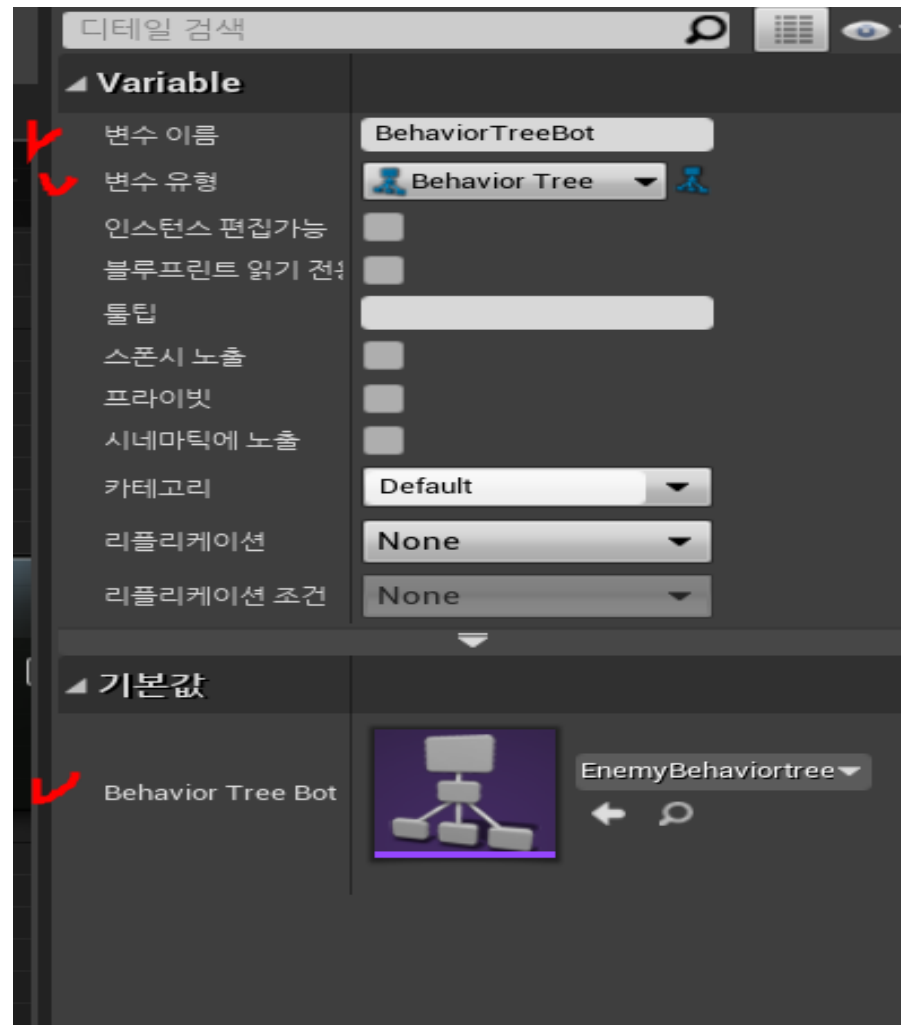
2.16 LookVector를 RotX를 만들고 변수로 승격해서 TargetRot를 생성한다.



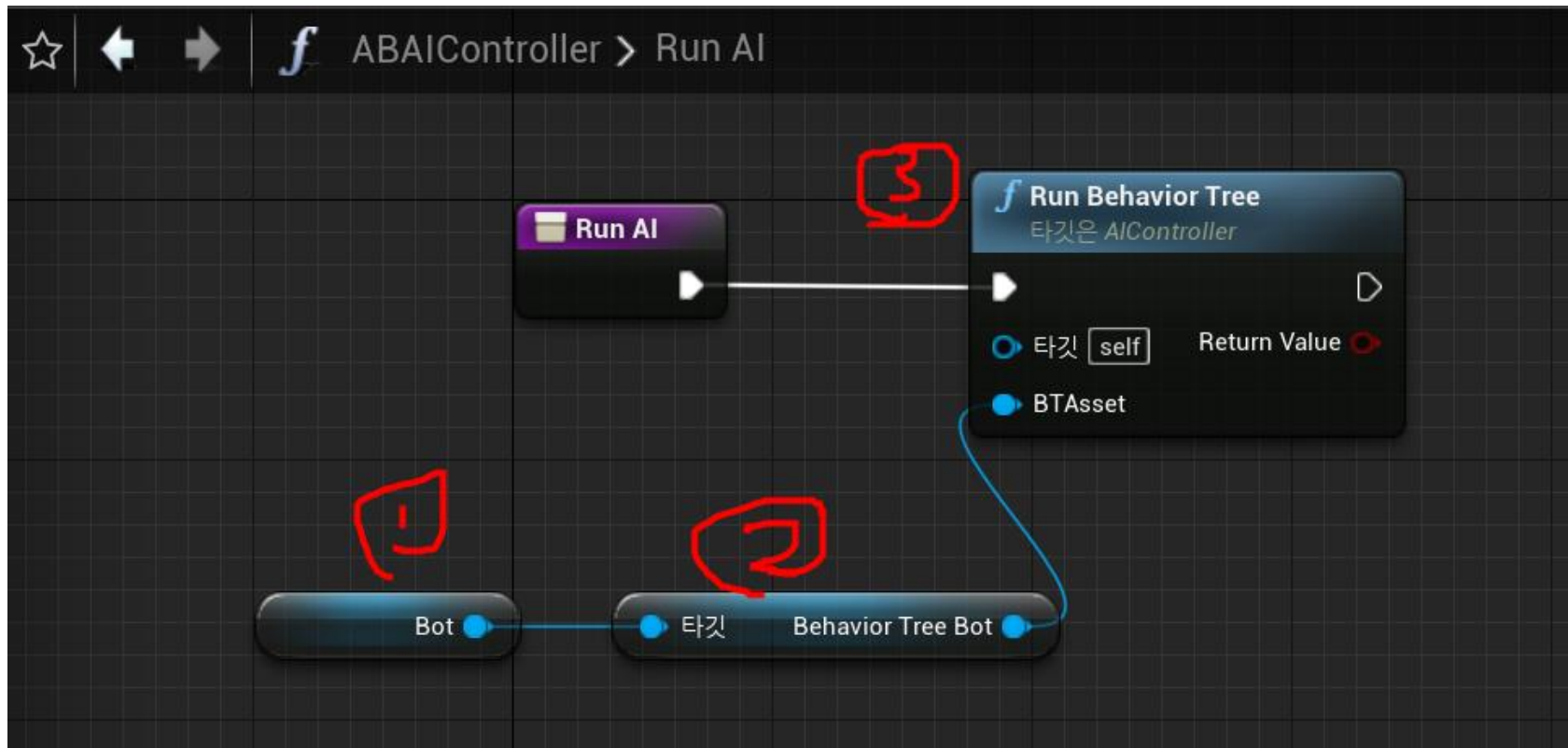
2.17 Bot의 SetActorRoatation을 호출하고 Bot 현재의 회전값을 Target으로 방향으로 넘어가세 한다.



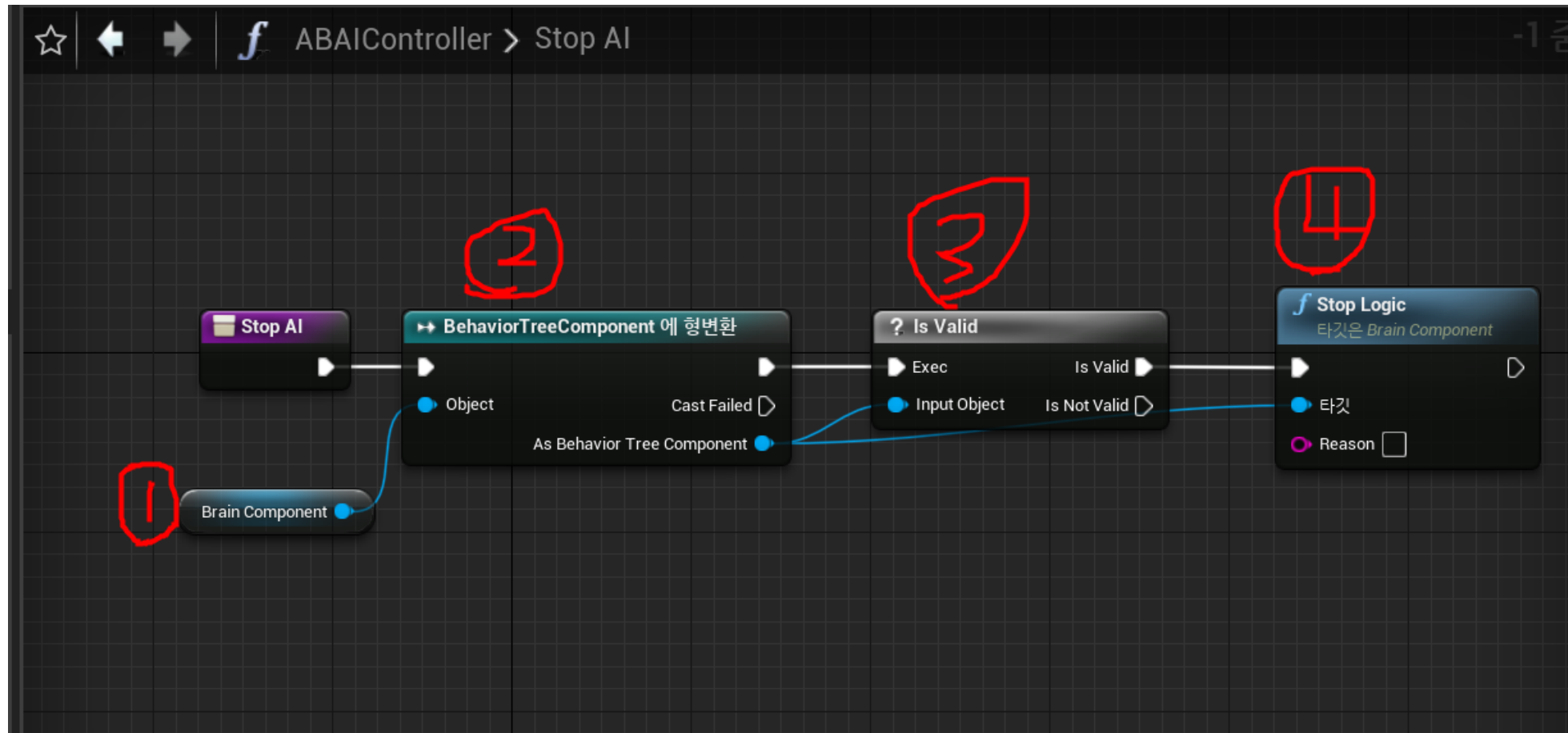
3. enemyCharacter2에 들어가서 AutoPossessAi와 AIControllerClass를 설정한다.



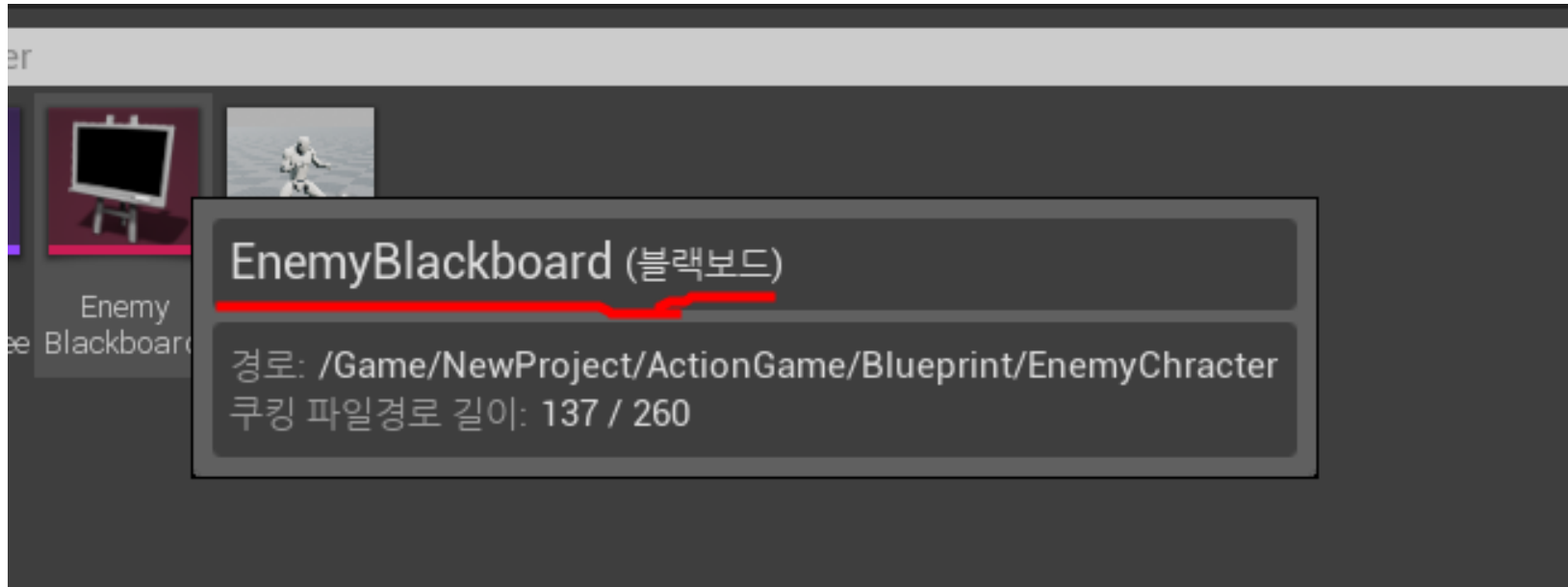
3.1 BehaviorTree 타입 BehaviorTreeBot을 선언한다.



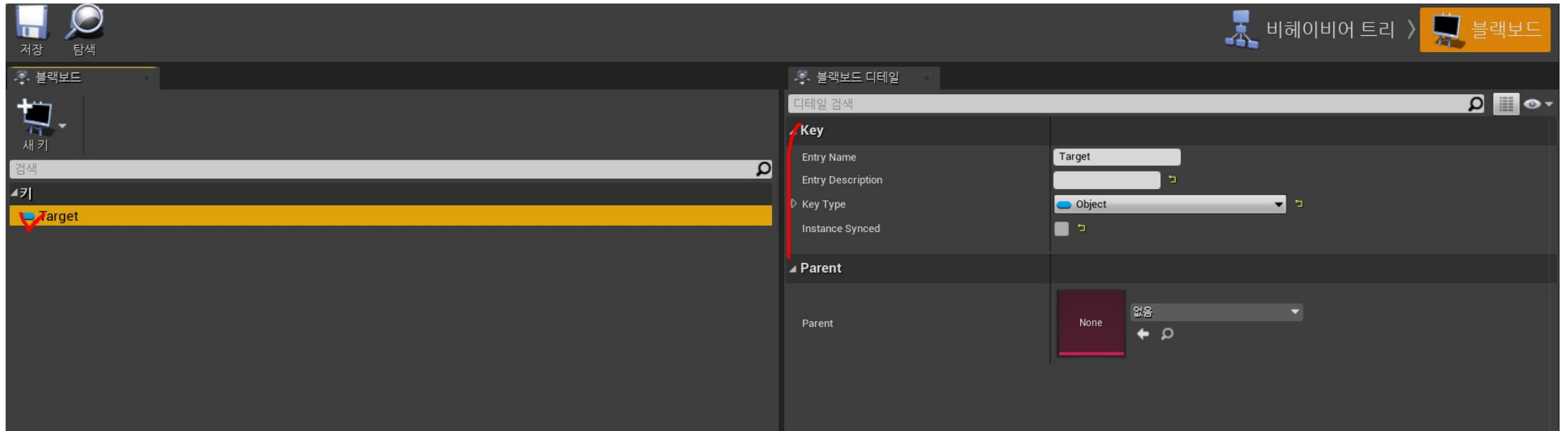
3.2 ABAIController의 RunAI 함수에 들어가서 Bot의 BehaviorTreeBot을 RunBehaviorTree 함수에 넣는다.



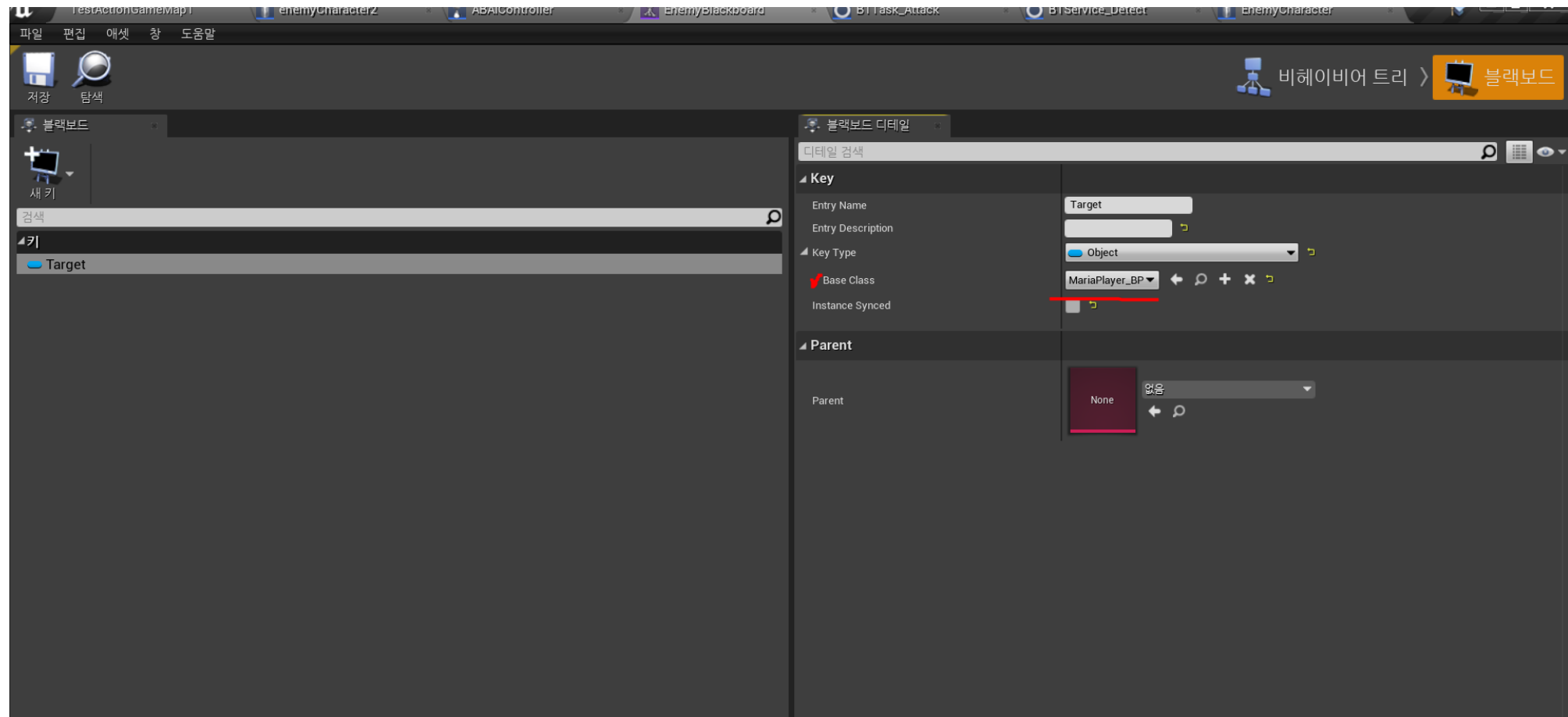
3.3 BrainComponent를 BehaviorTreeComponent로 형변환하고 StopLogic을 호출한다.



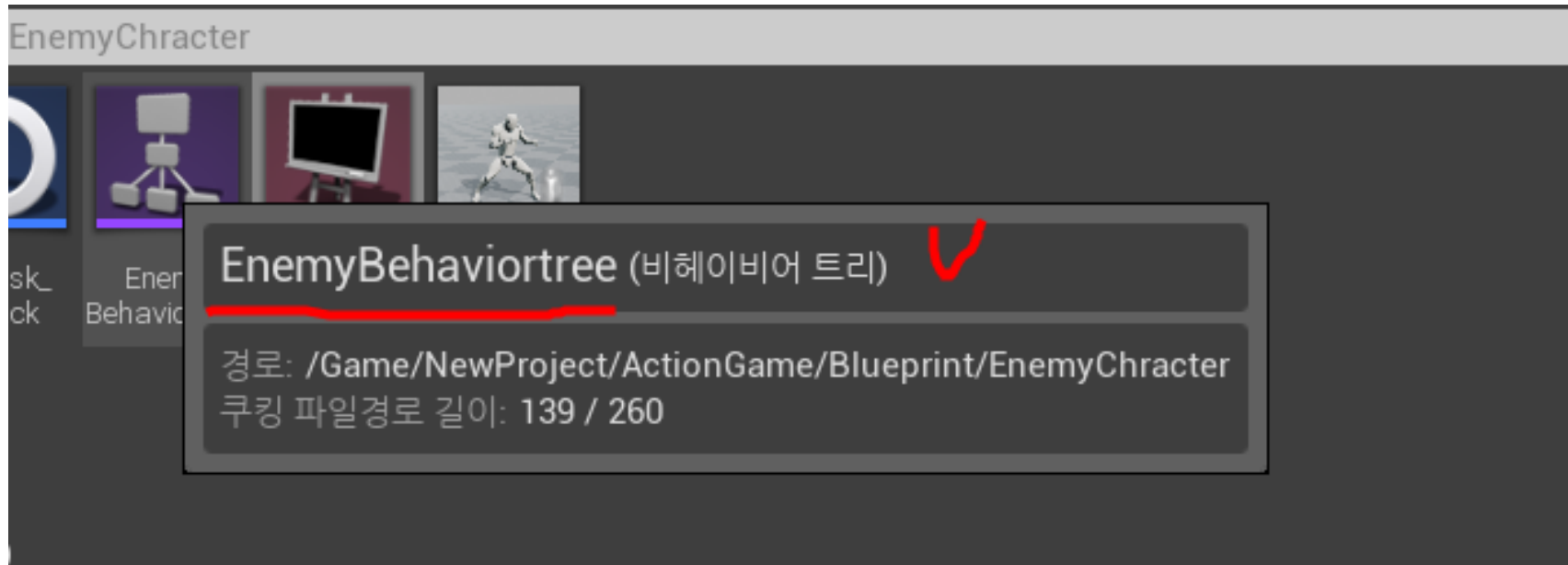
4. Blackboard EnemyBlackboard를 생성한다.



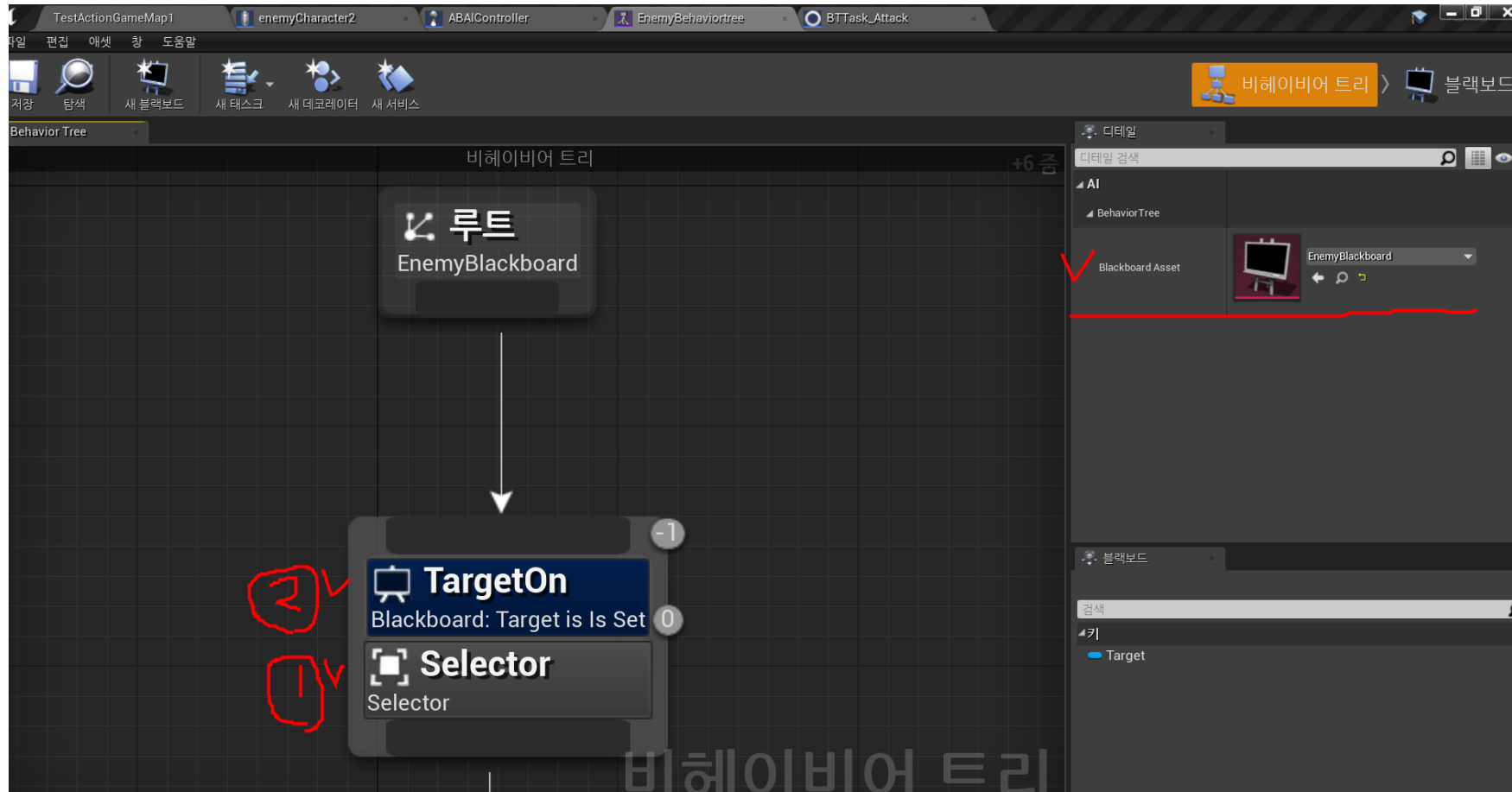
4.1 Object 타입으로 Target 변수를 생성한다.



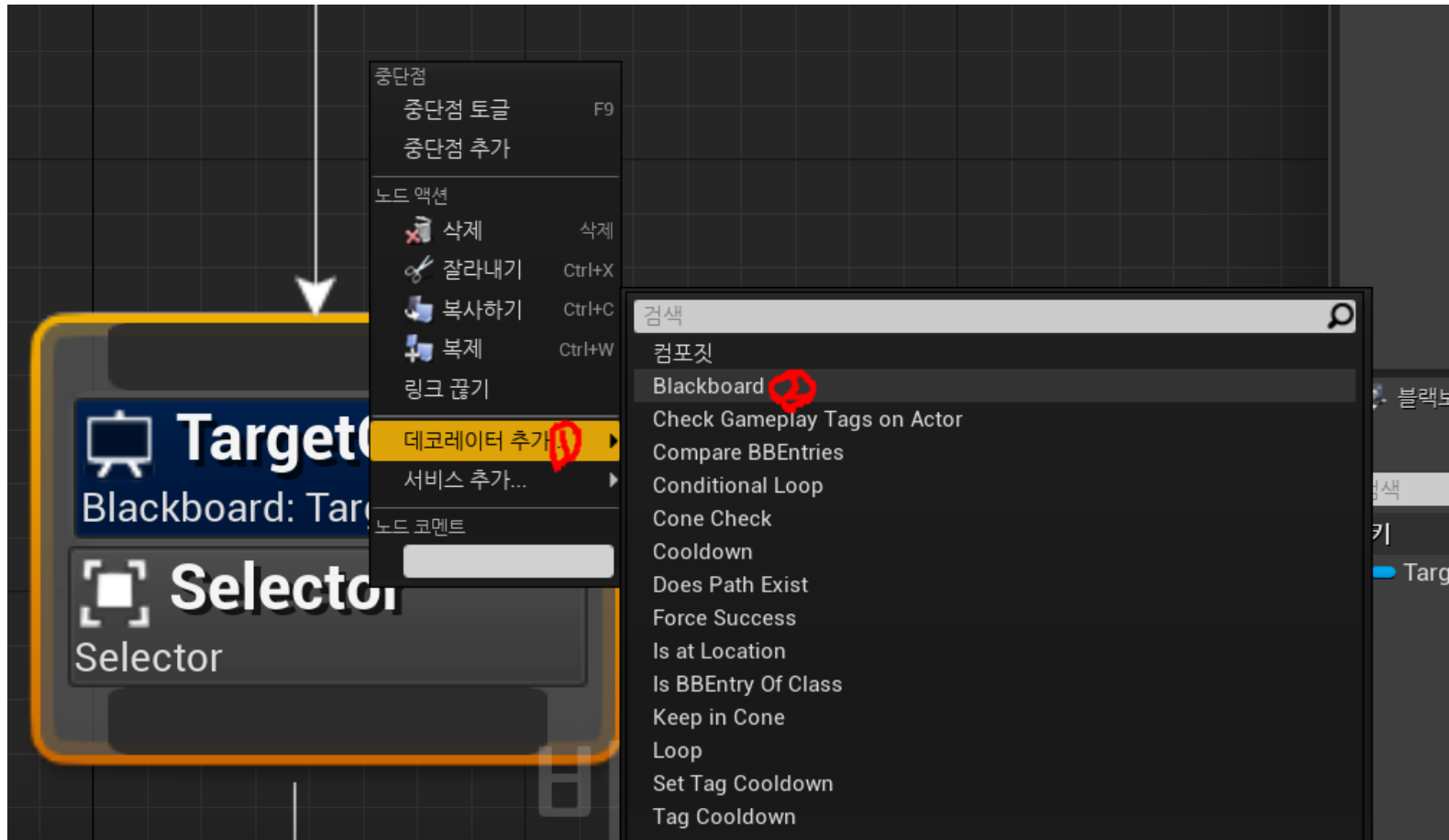
4.2 BaseClass를 MariaPlayer_BP로 한다.



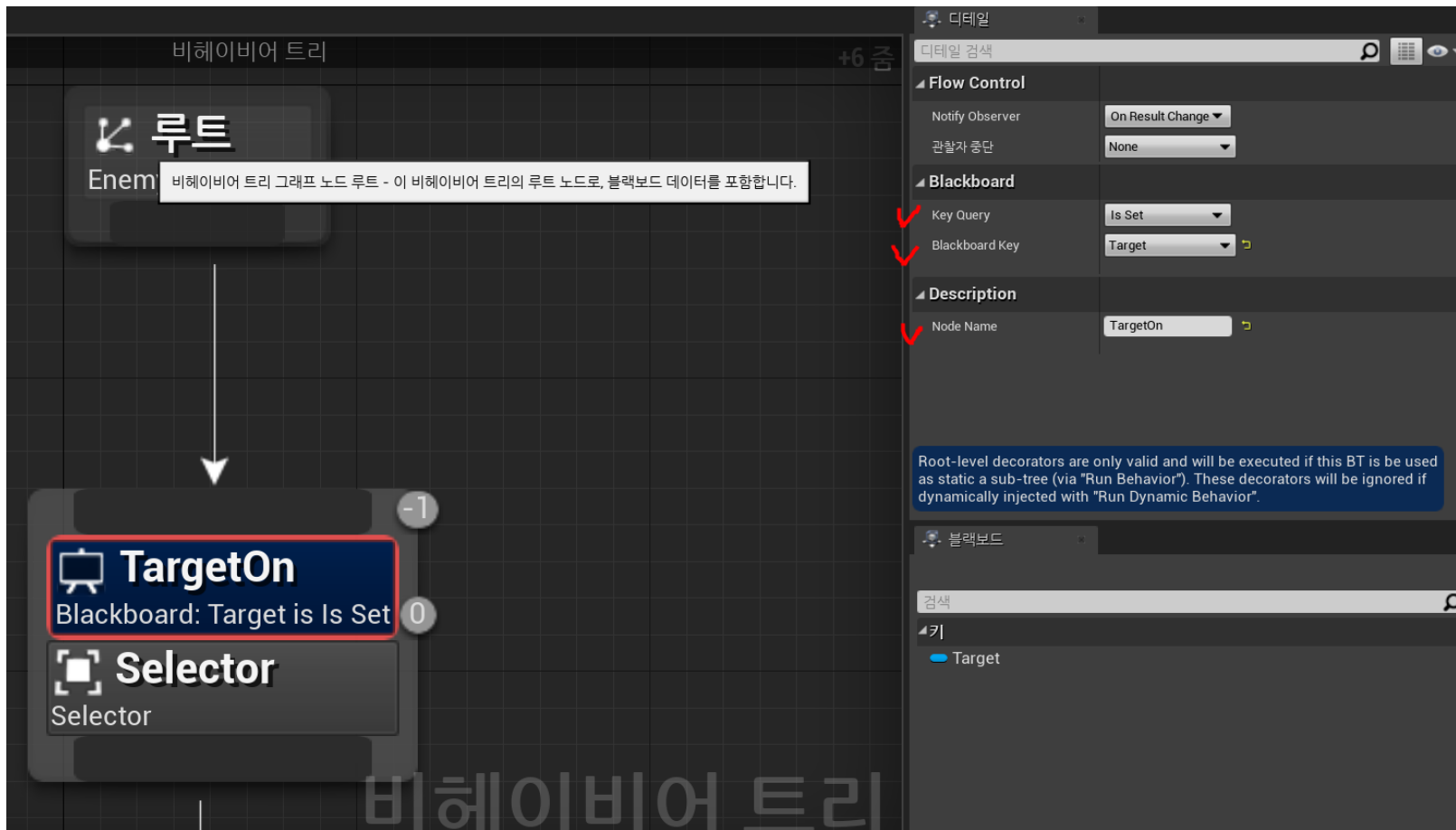
5. Behaviortree EnemyBehaviortree를 생성한다.



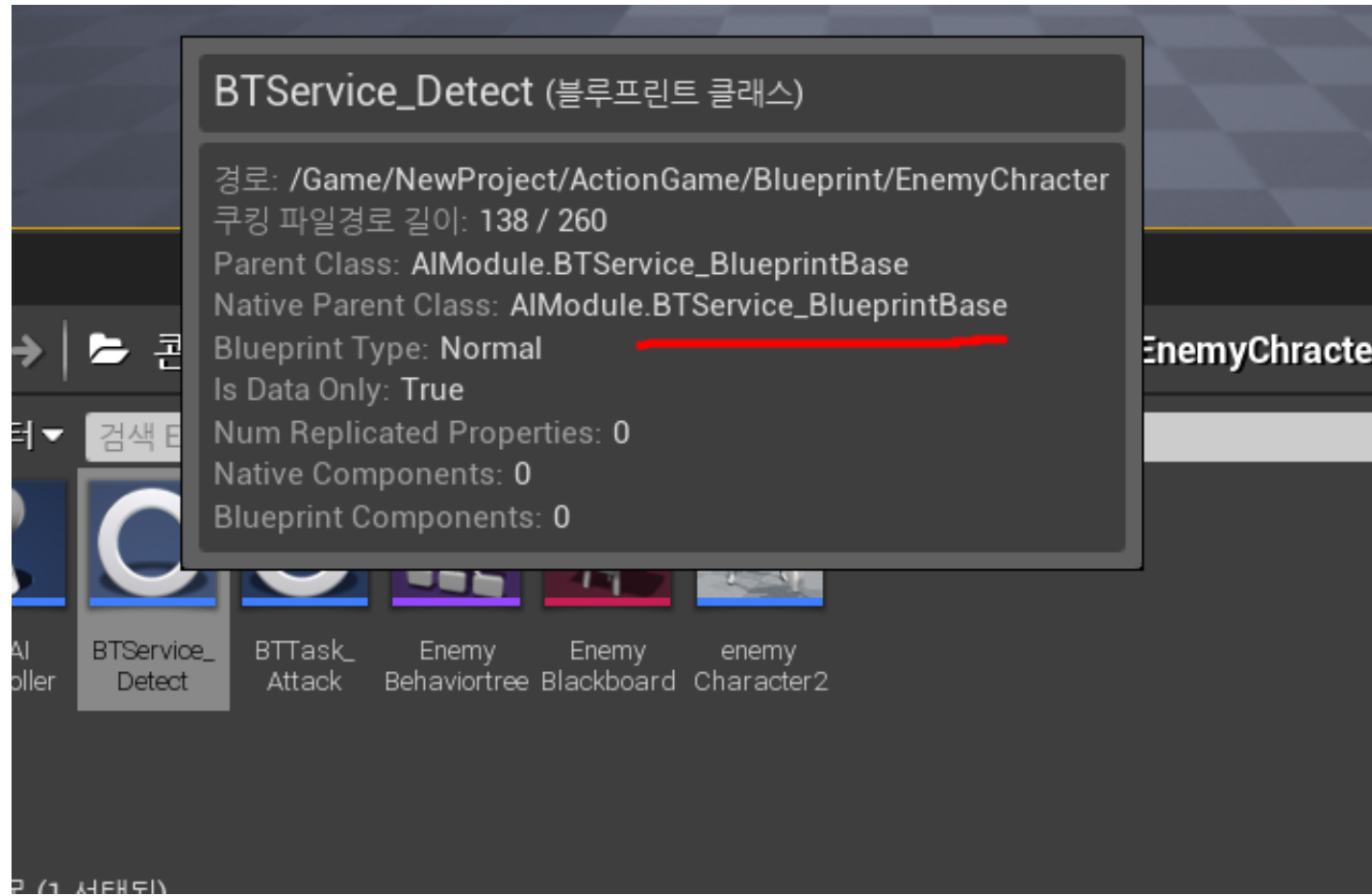
5.1 BlackboardAsset에 EnemyBlackboard가 있는지 확인하고, Composit Selector 노드를 추가한다.



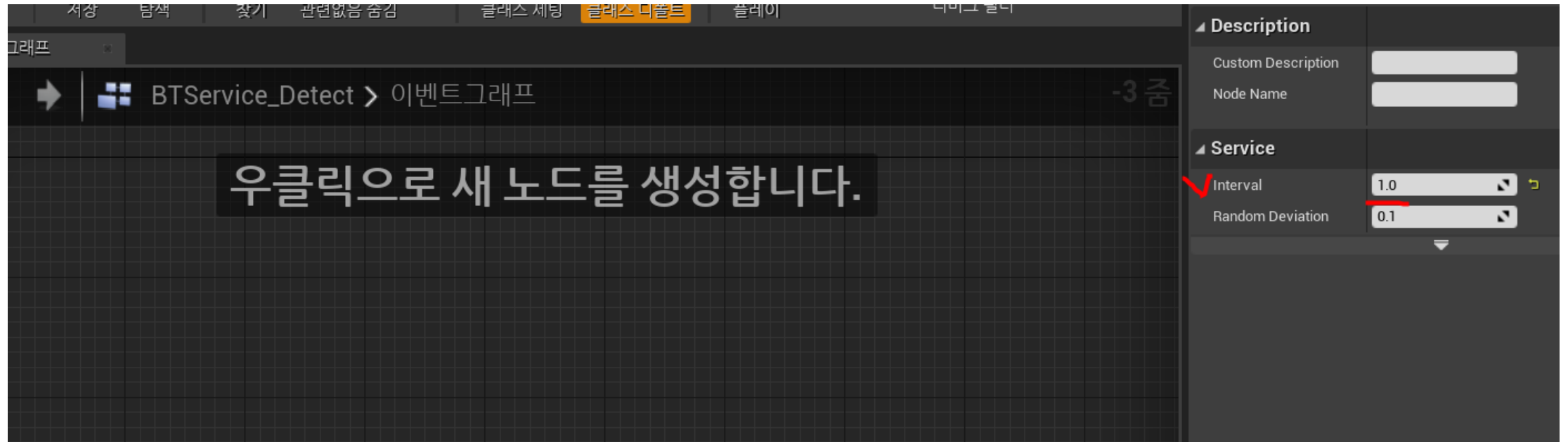
5.2 Selector를 클릭해서 데코레이터 Blackboard를 클릭한다.



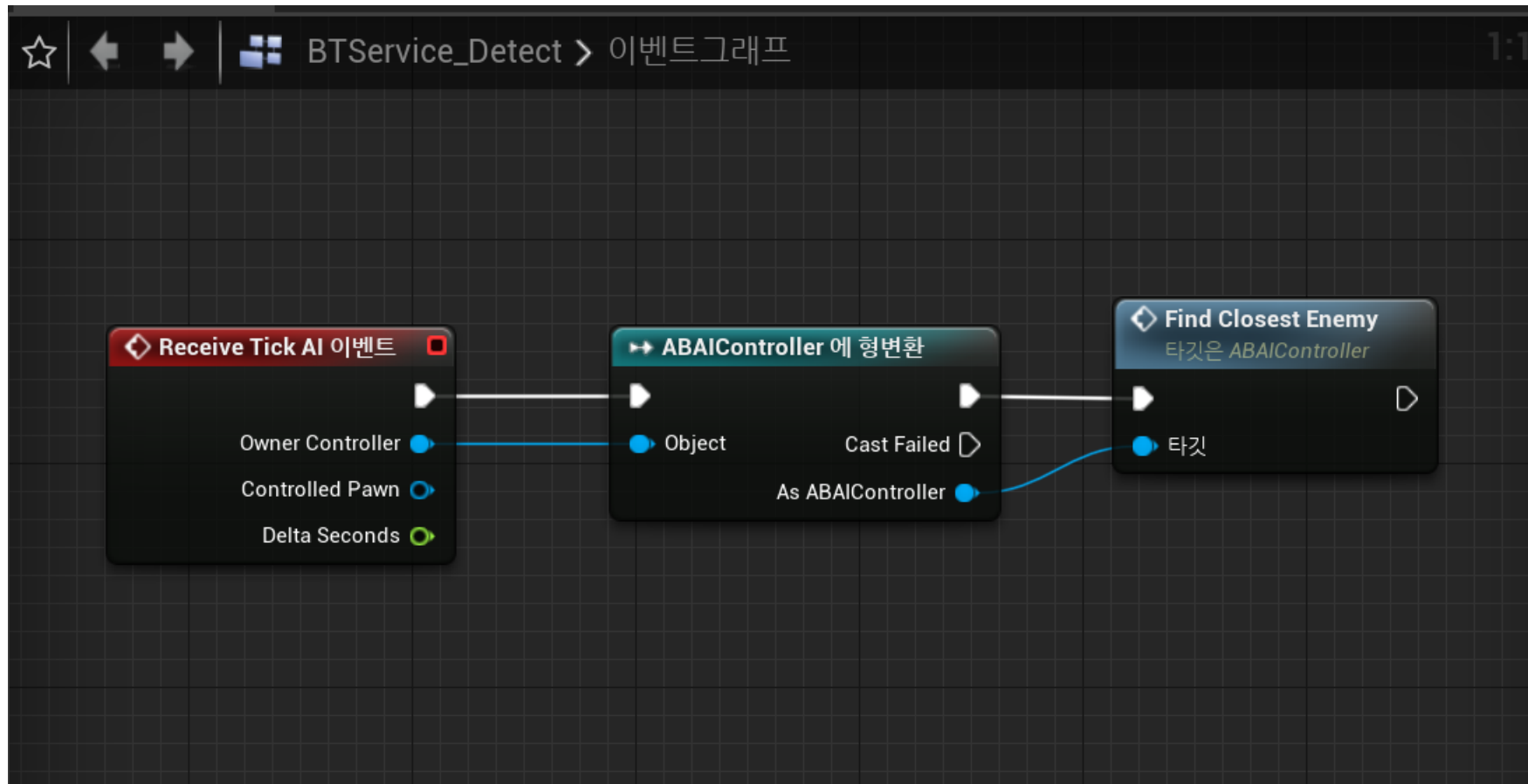
5.3 Node Name을 TargetOn으로 변경하고, KeyQuery와 BlackboardKey를 변경한다.



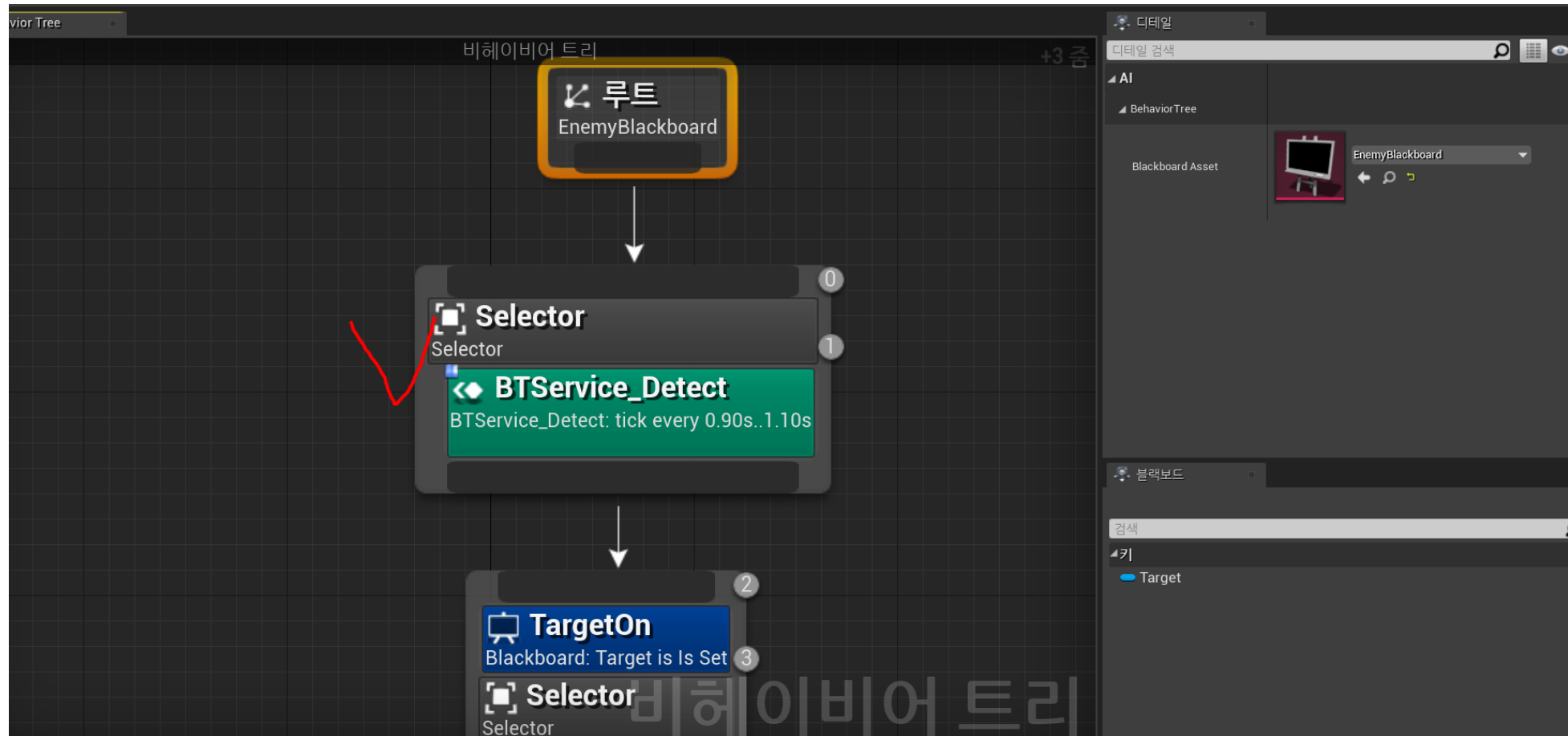
6. BIService_BlueprintBase를 상속받은 BIService_Detect를 생성한다.



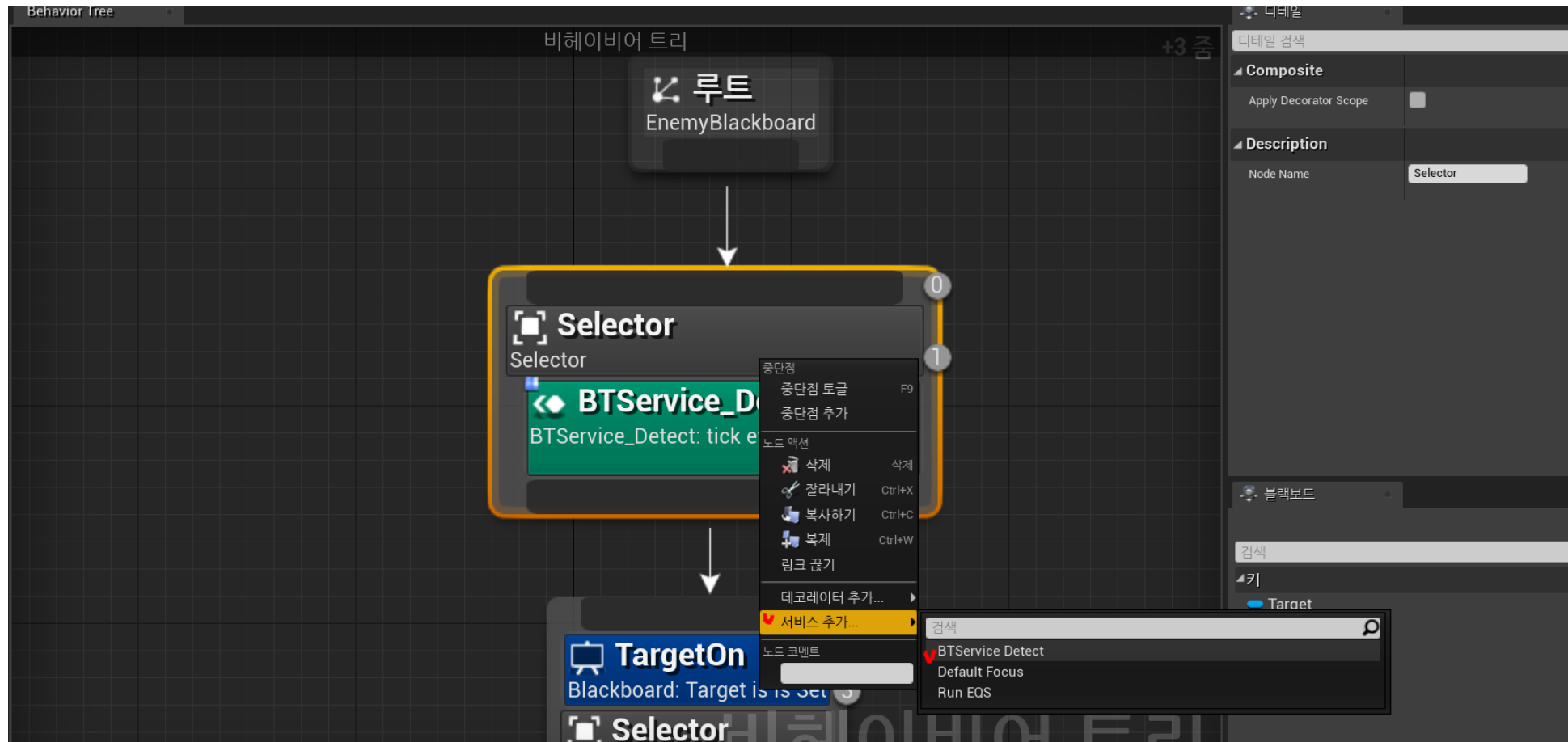
6.1 Interval을 1.0으로 변경한다.



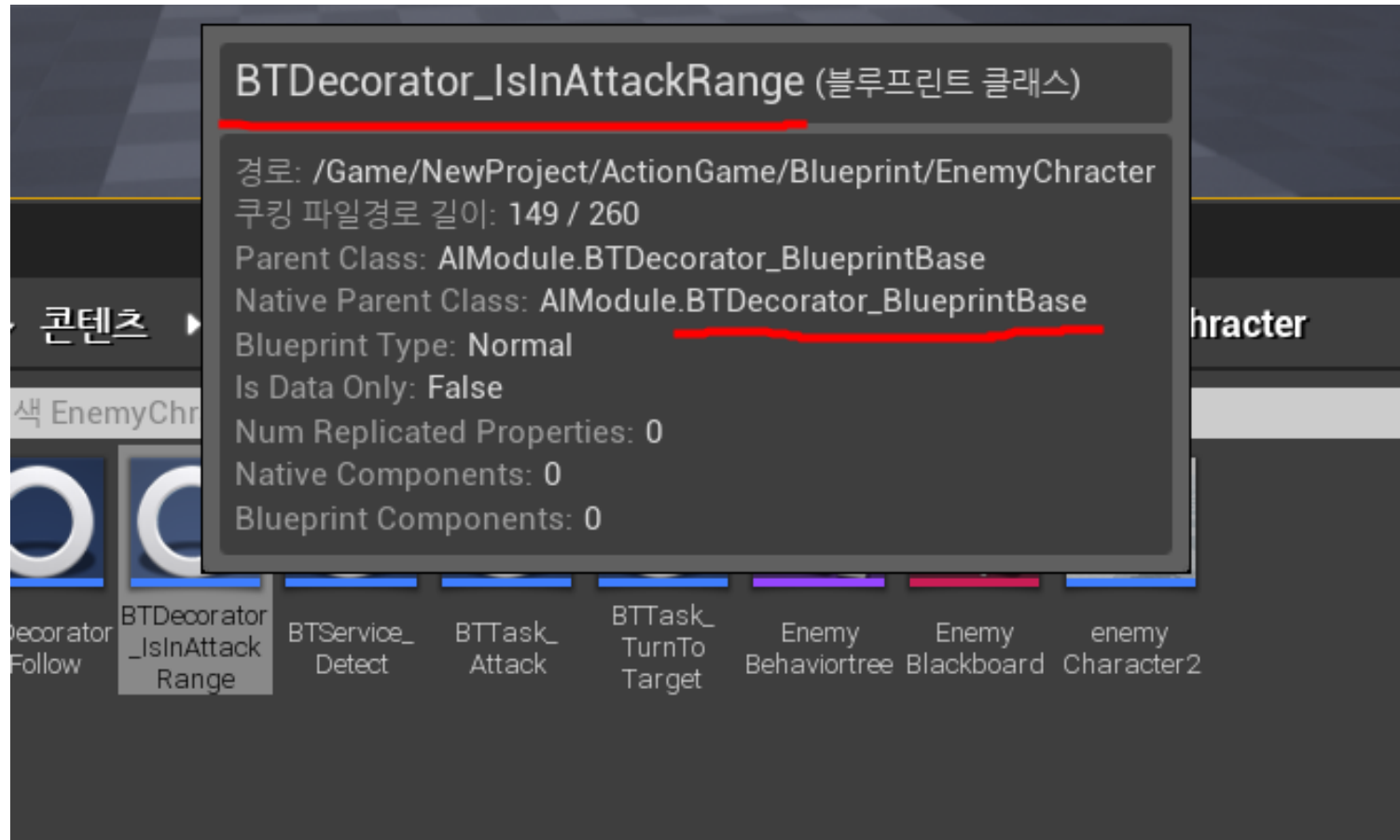
6.2 Tick 이벤트에 ABAIController의 FindClosestEnemy를 호출한다.



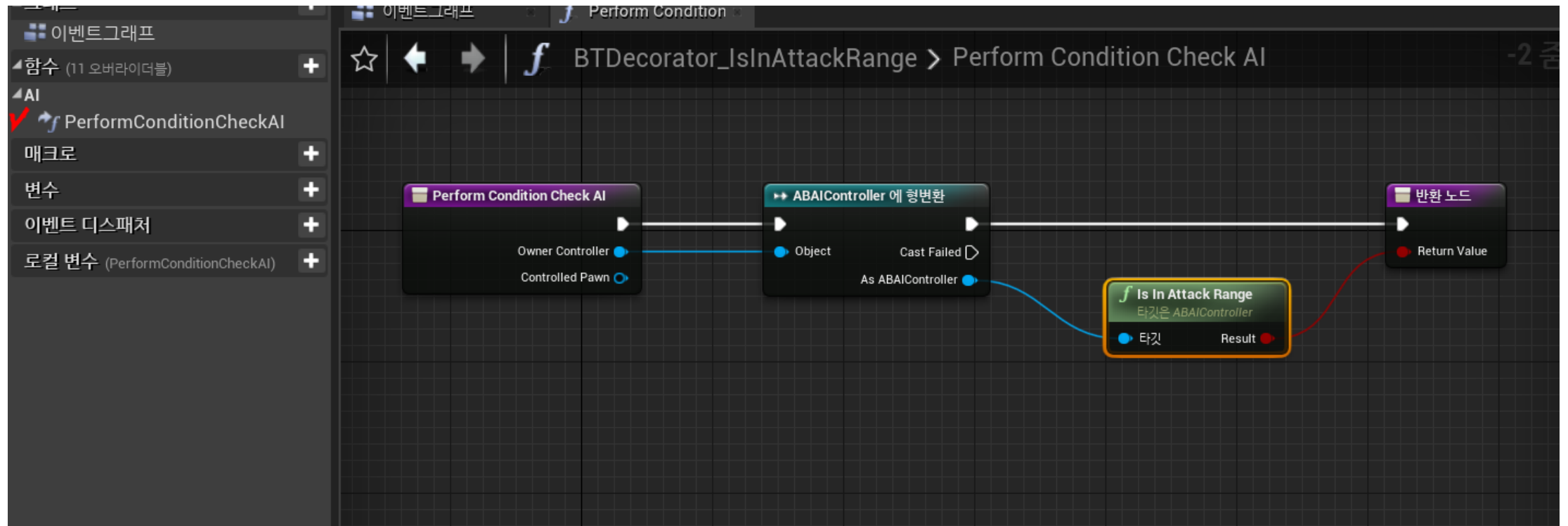
6.3 EnemyBlackboard에 들어가서 Selector 노드를 추가하고, TargetOn이 부착된 Selector의 부모 노드로 한다.



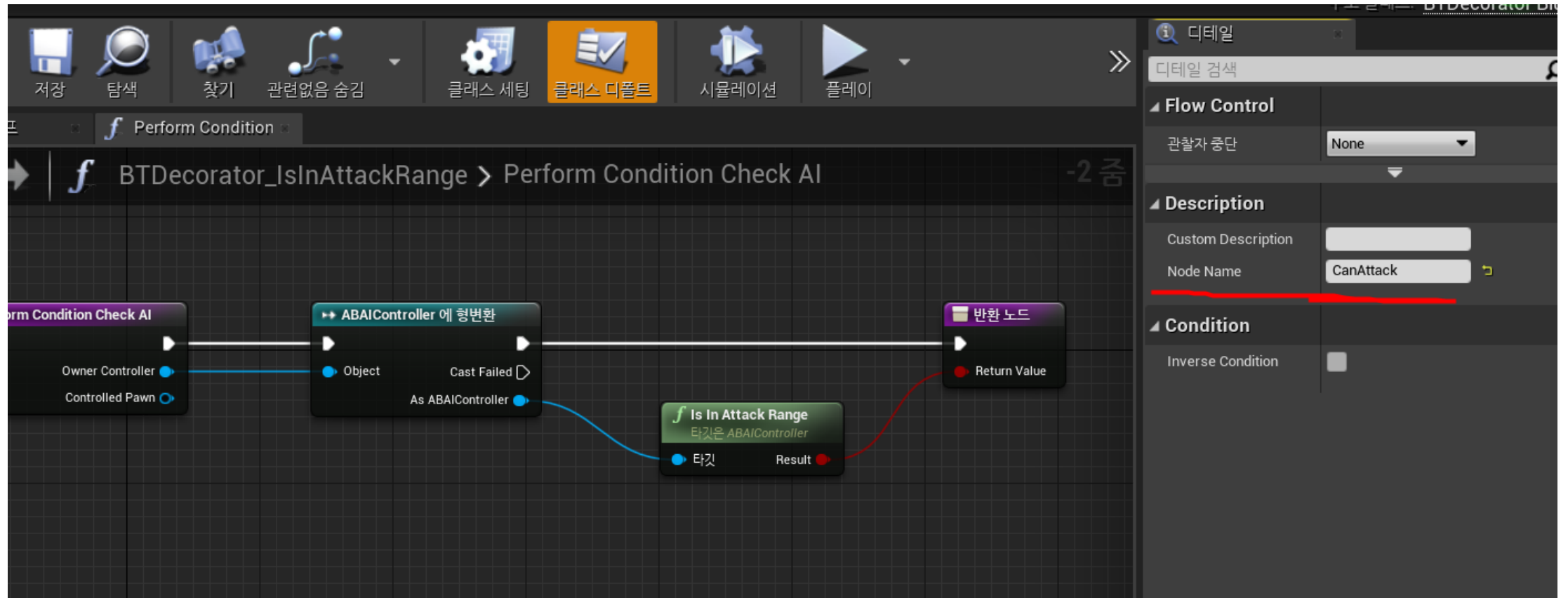
6.4 Selector를 클릭해서 서비스 BTService_Detect를 클릭한다.



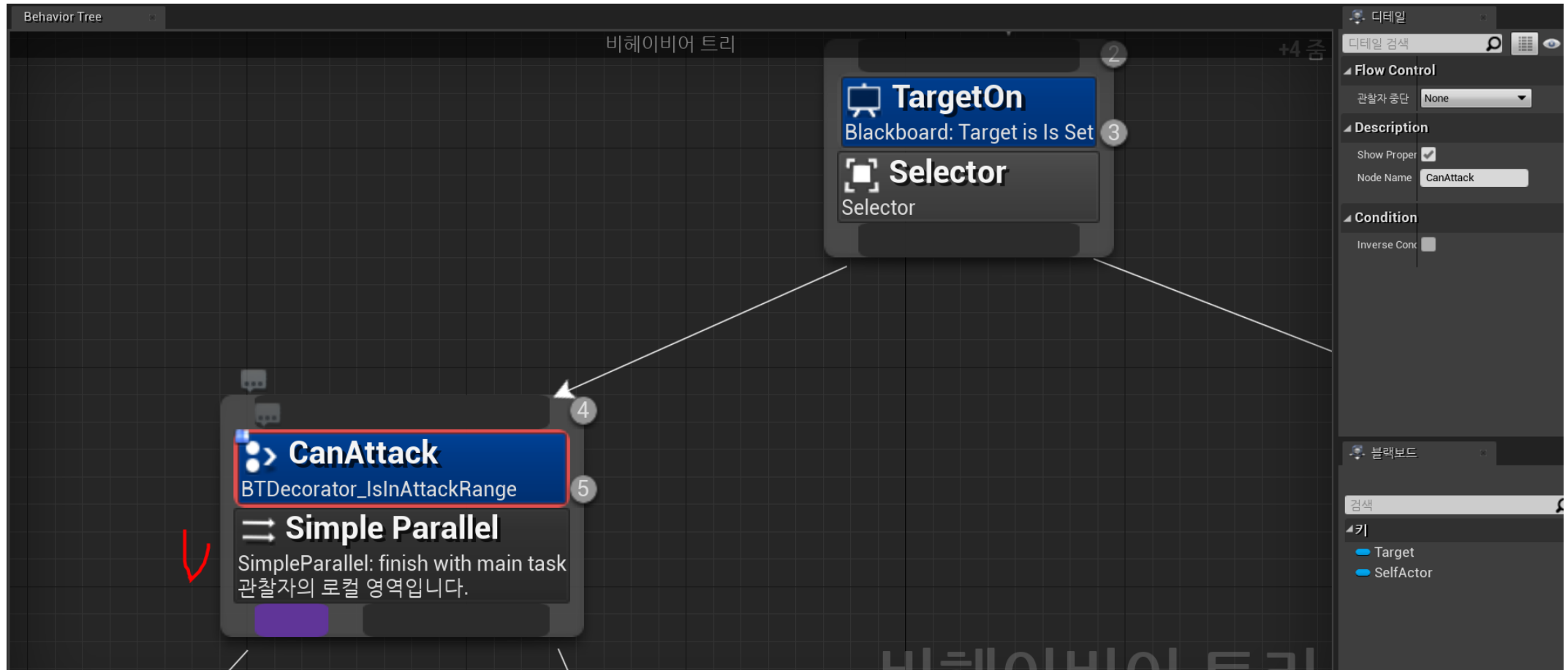
7. BTDecorator_BlueprintBase를 상속받은 BTDecorator_IsInAttackRange를 생성한다.



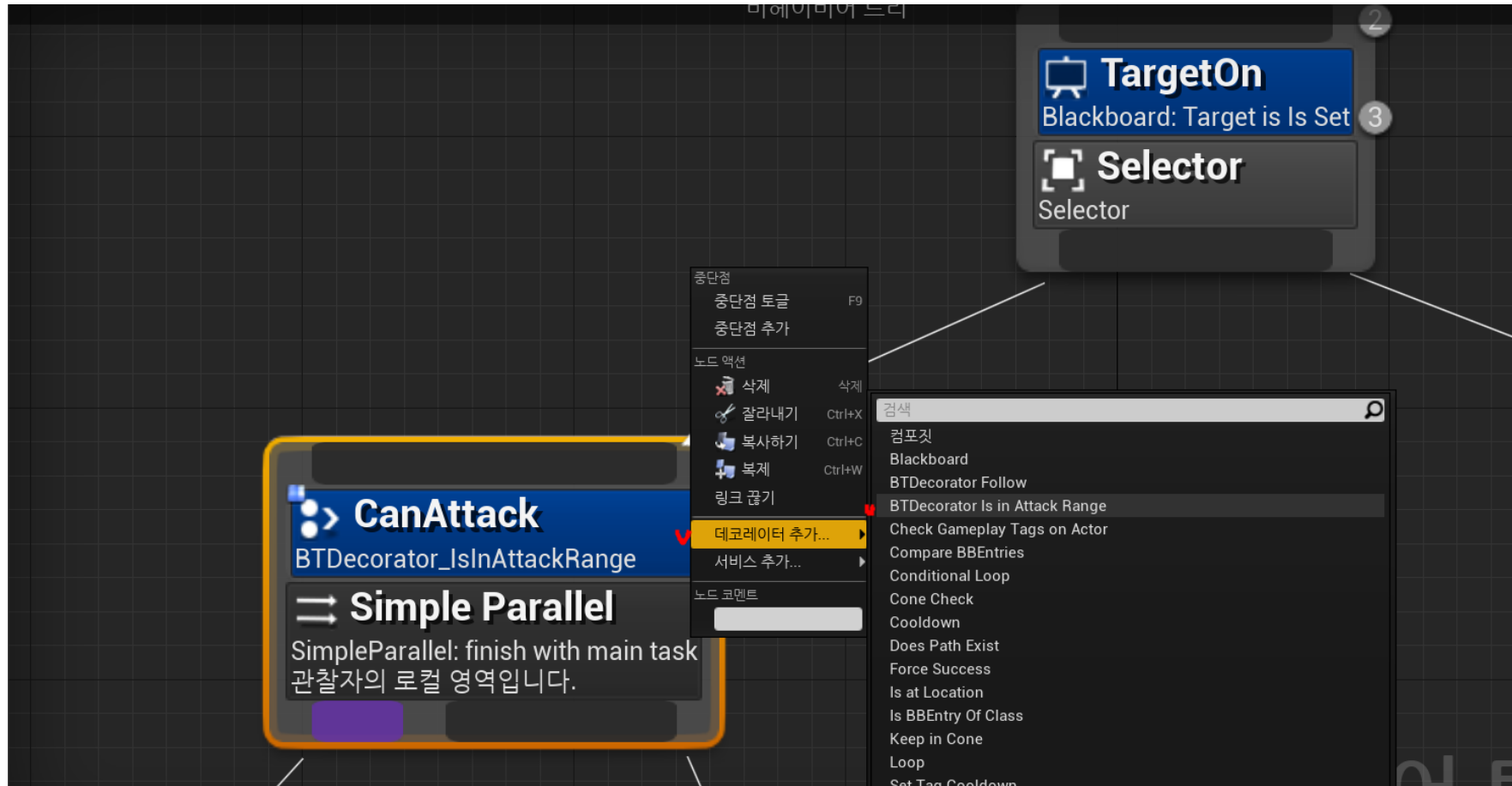
7.1 PerformConditionCheckAI를 오버라이드해서 ABAIController의 IsInAttackRange를 호출하고 결과값을 반환한다.



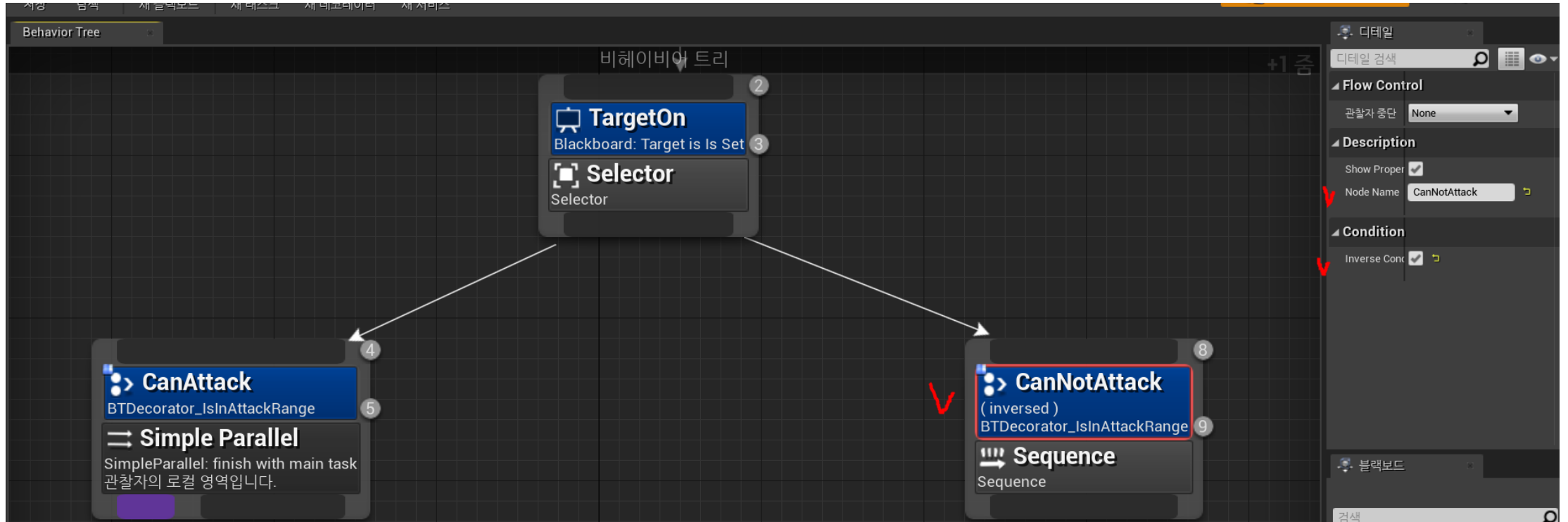
7.2 Node Name을 CanAttack으로 한다.



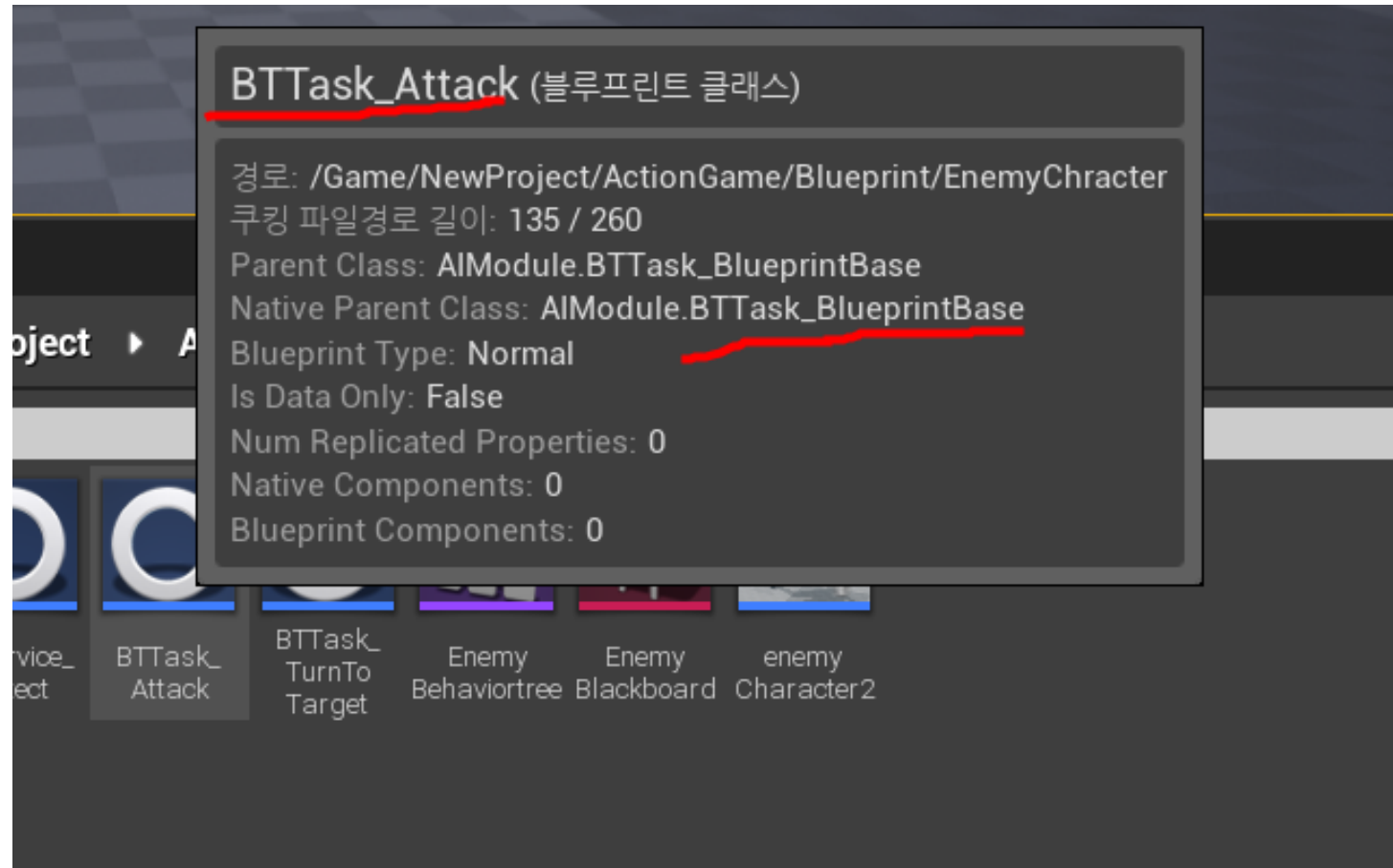
7.3 Comsite SimpleParallel 노드를 생성한다.



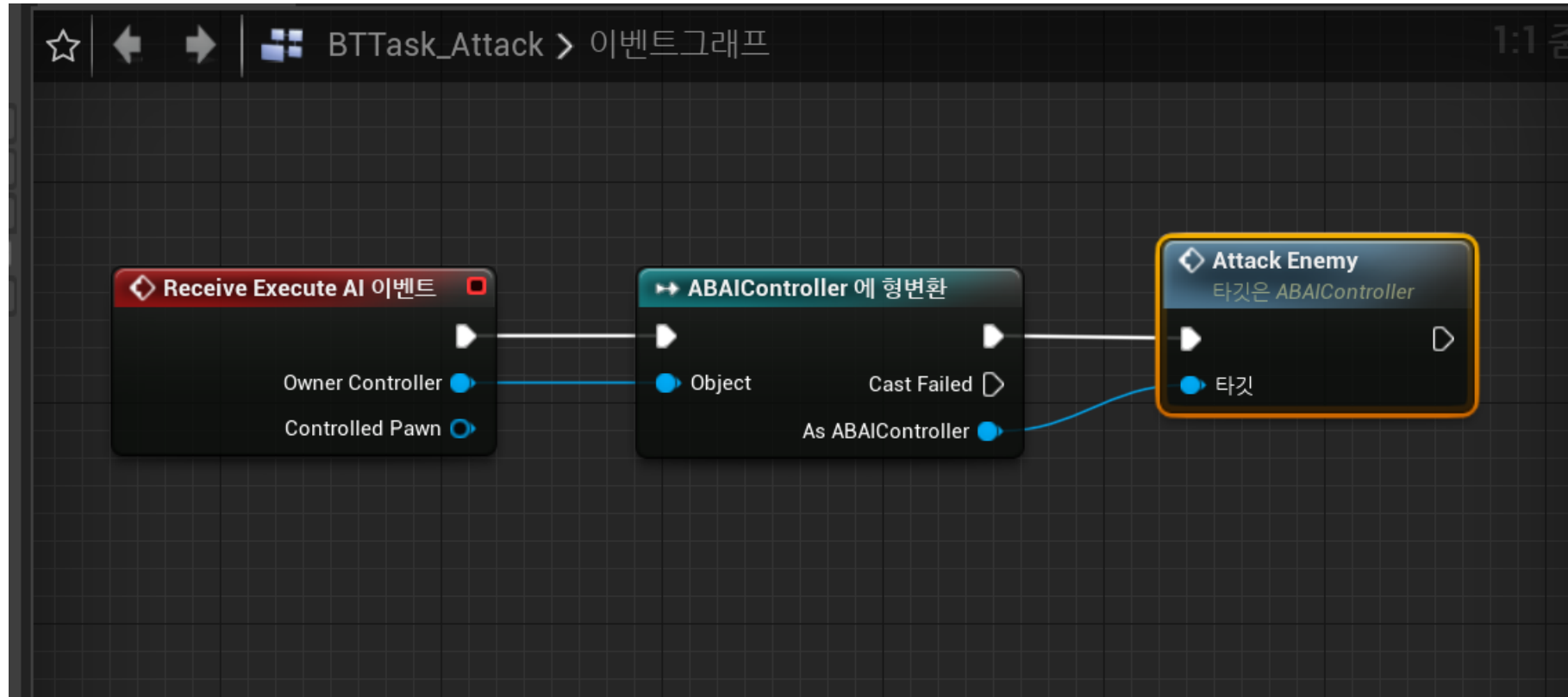
7.4 SimpleParallel에 BTDecorator_IsInAttackRange 데코레이터를 추가한다.



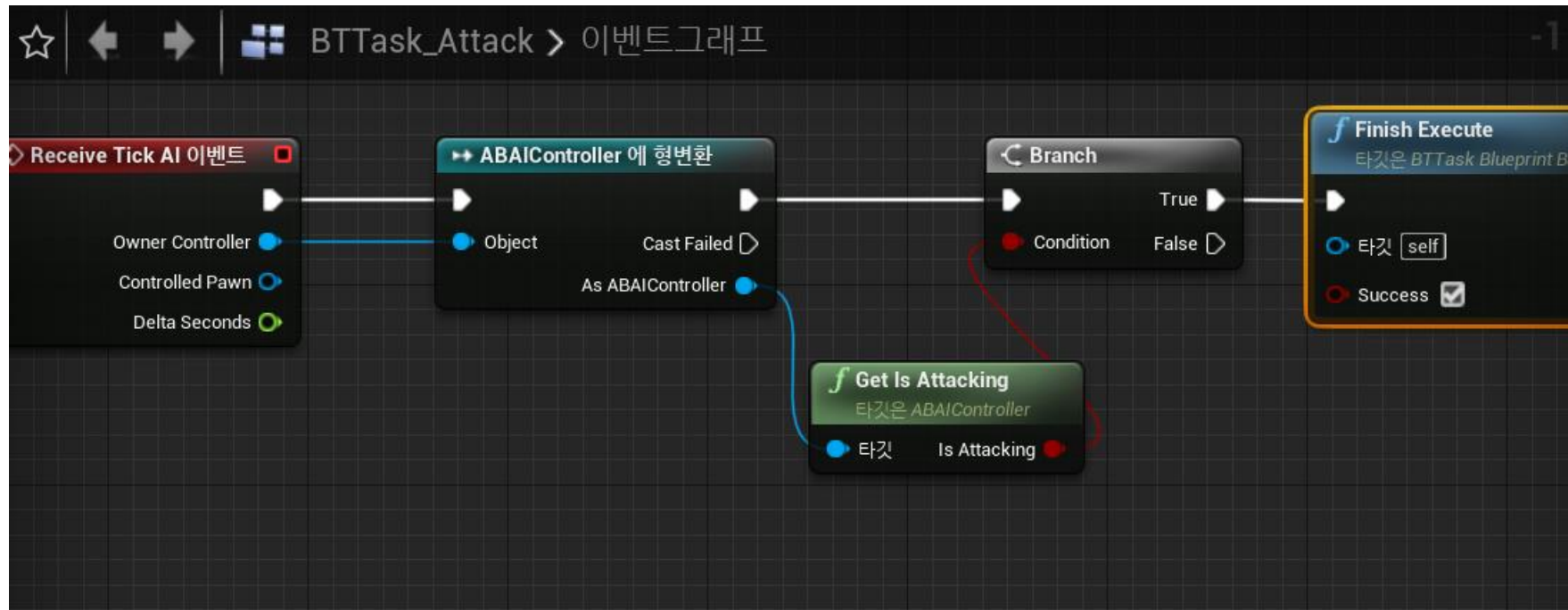
7.5 Sequence 노드를 생성하고 BTDecorator_IsInAttackRange 데코레이터를 추가하고 Node Name과 InverseCondition의 설정을 바꾼다.



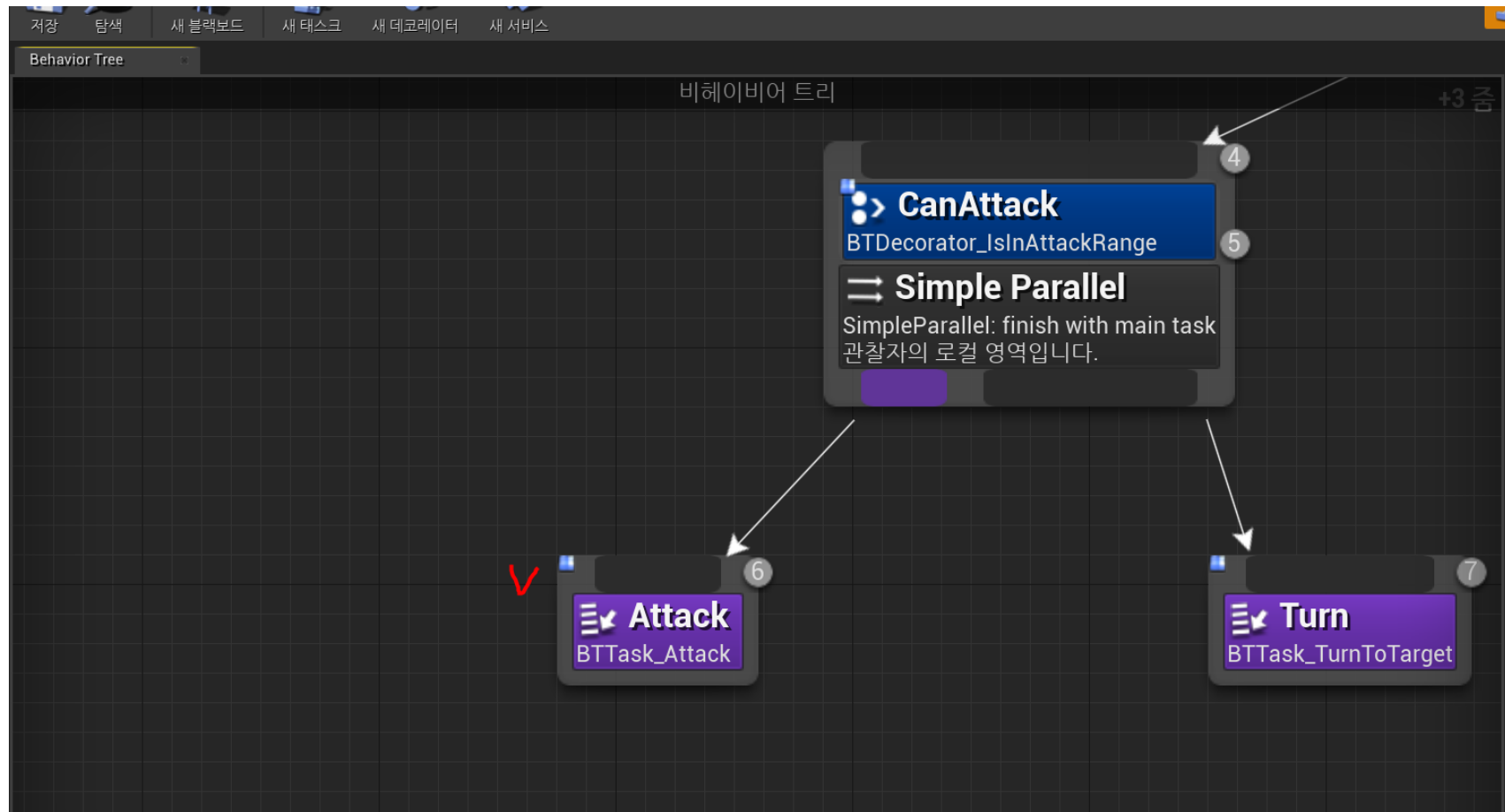
8. BTTask_BlueprintBase를 상속받은 BTTask_Attack을 생성한다.



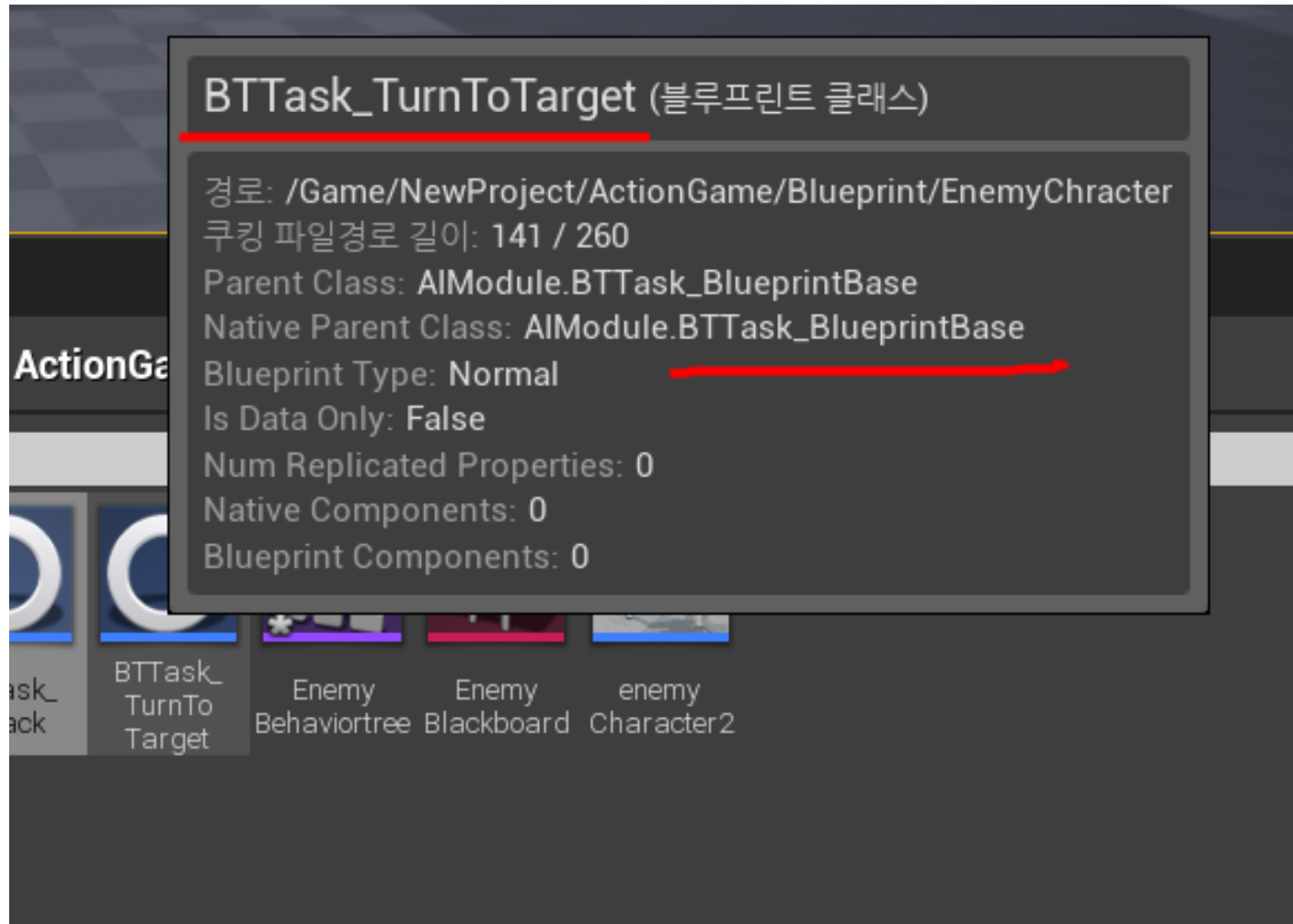
8.1 Execute 이벤트에 ABAIController의 AttackEnemy를 호출



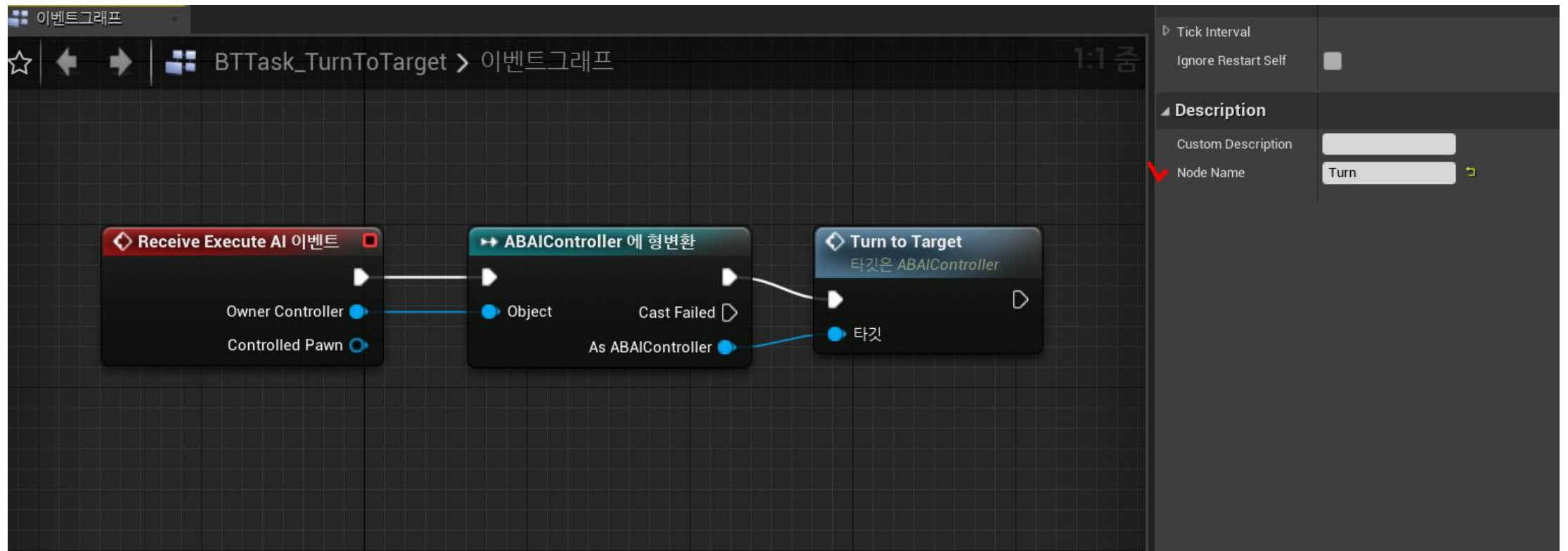
8.2 Tick 이벤트에 ABAIController의 GetIsAttacking 함수를 호출하고 True이면 FinishExecute를 호출한다.



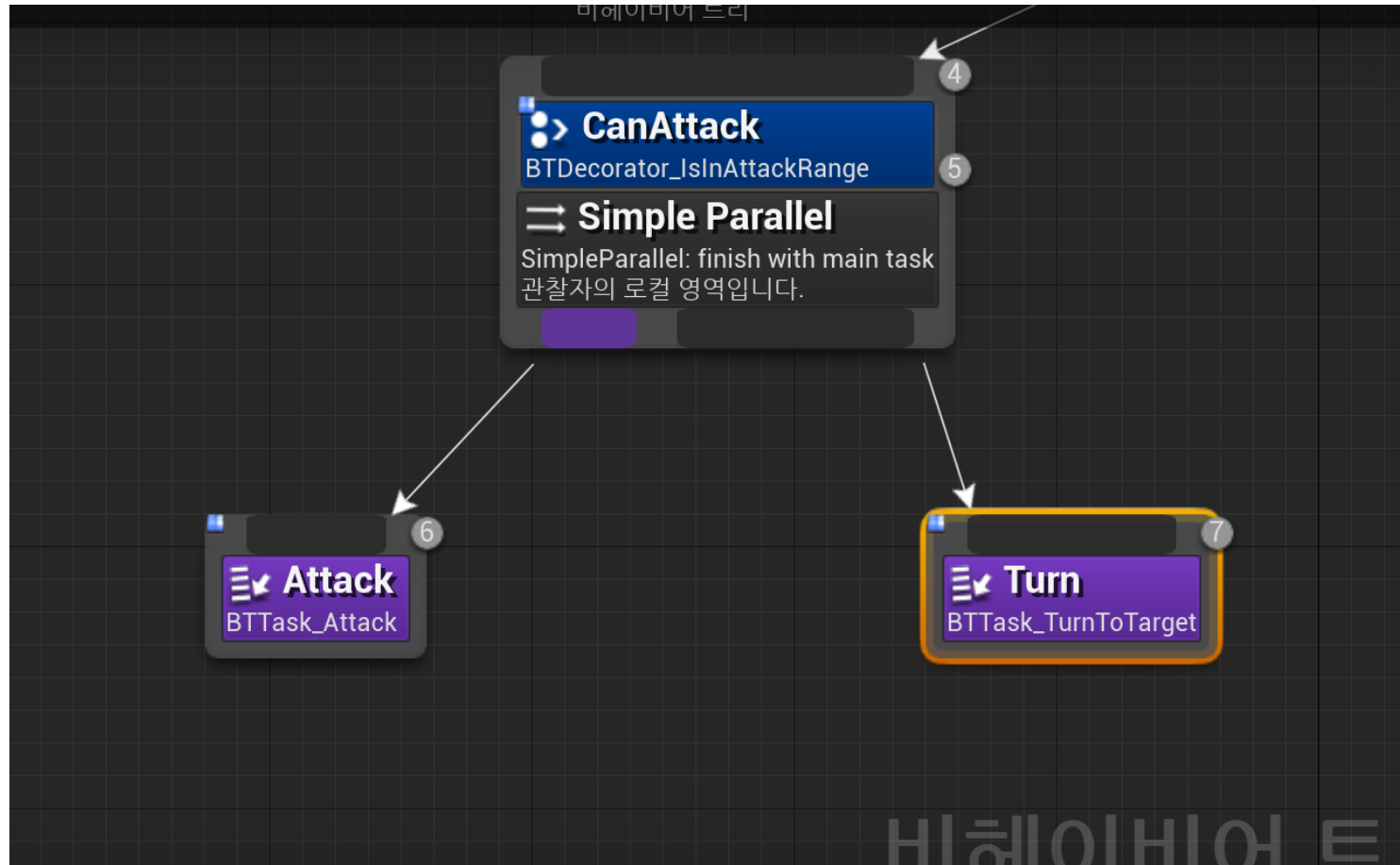
8.3 EnemyBehaviortree에 들어가서 BTTask_Attack을 추가한다.



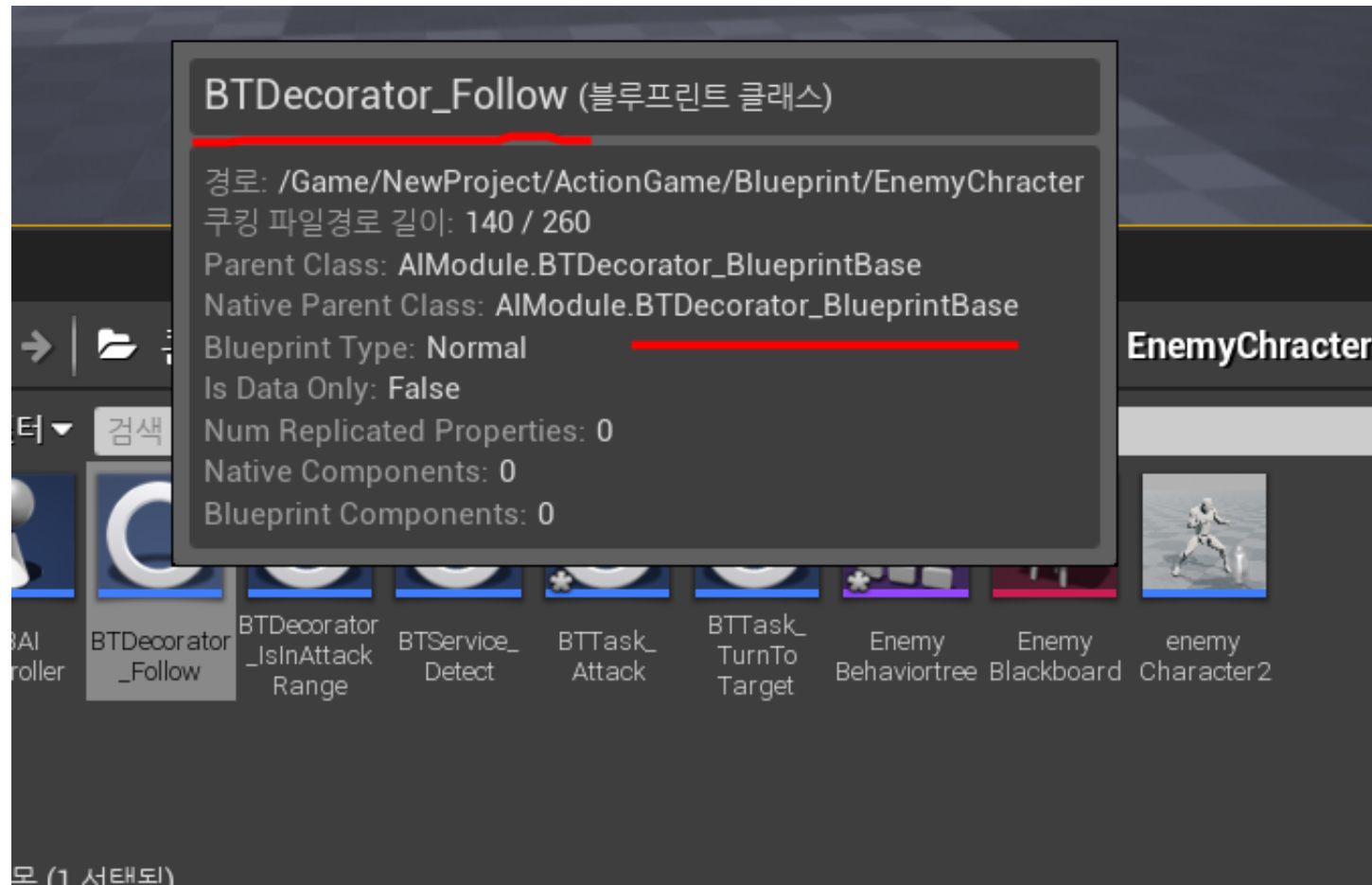
9. BTTask_BlueprintBase를 상속받은 BTTask_TurnToTarget을 생성한다.



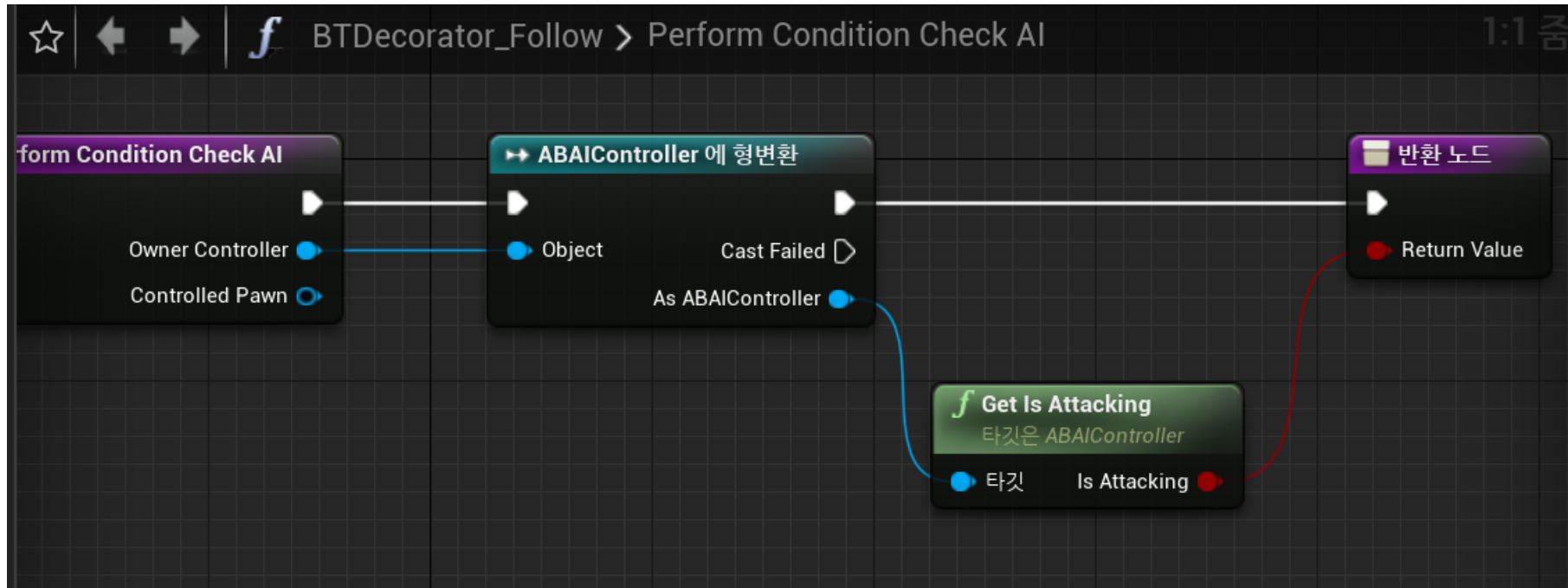
9.1 Node Name을 Turn으로 하고, ExecuteAI 이벤트에 ABAIController의 TurntoTarget 함수를 호출한다.



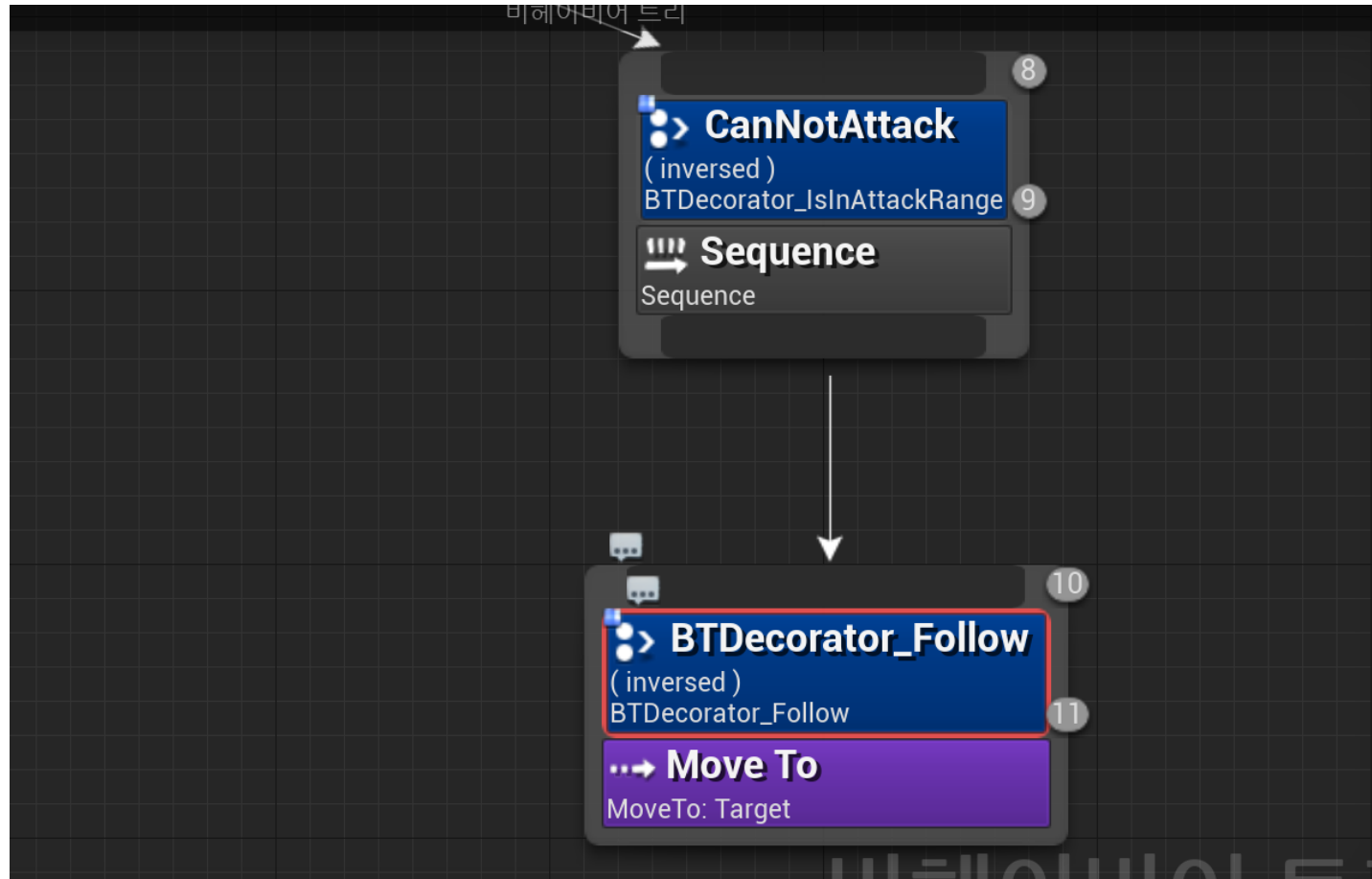
9.2 EnemyBehaviortree에 들어가서 BTTask_TurnToTarget을 추가한다.



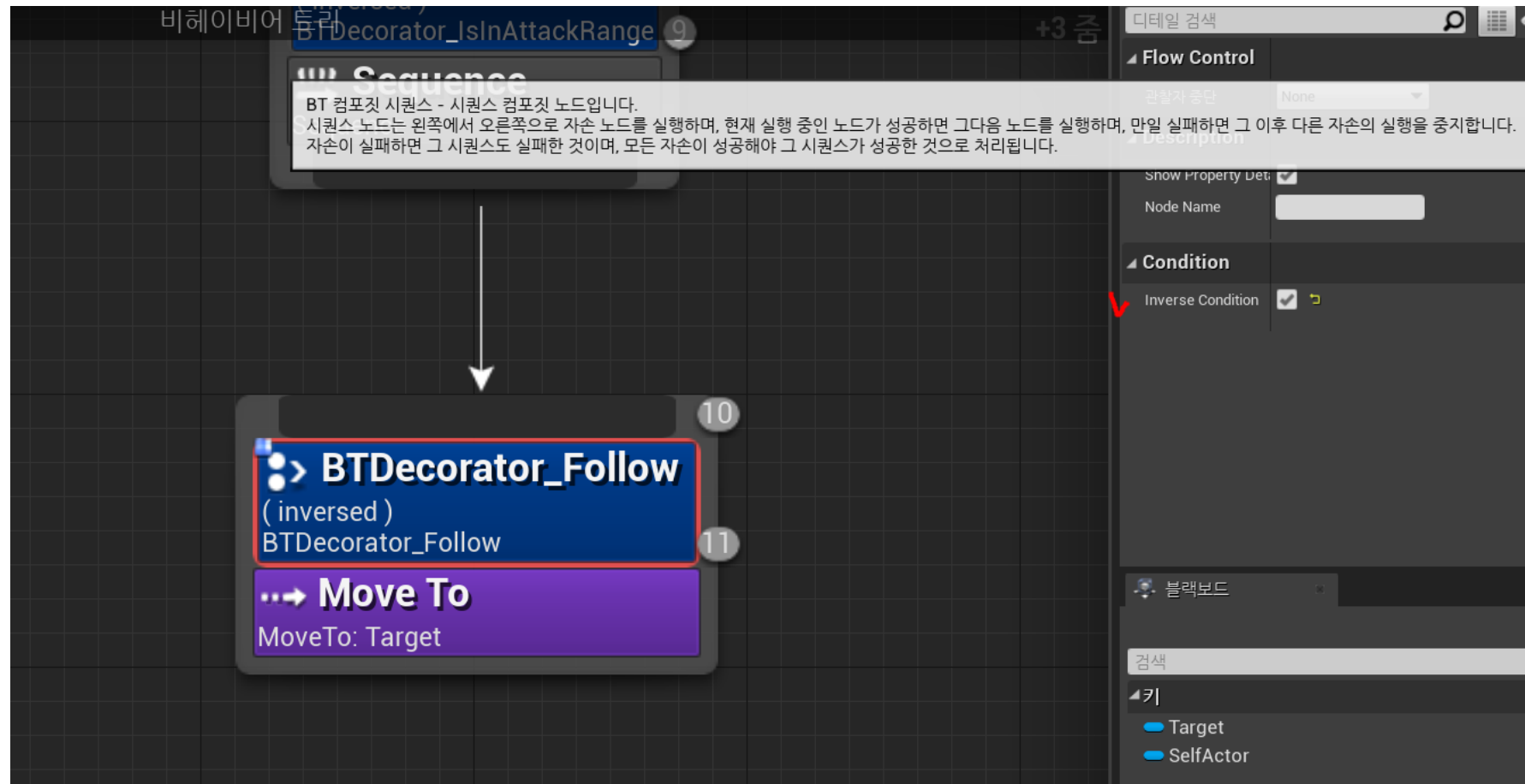
10. BTDecorator_BlueprintBase를 상속받은 BTDecorator_Follow를 생성한다.



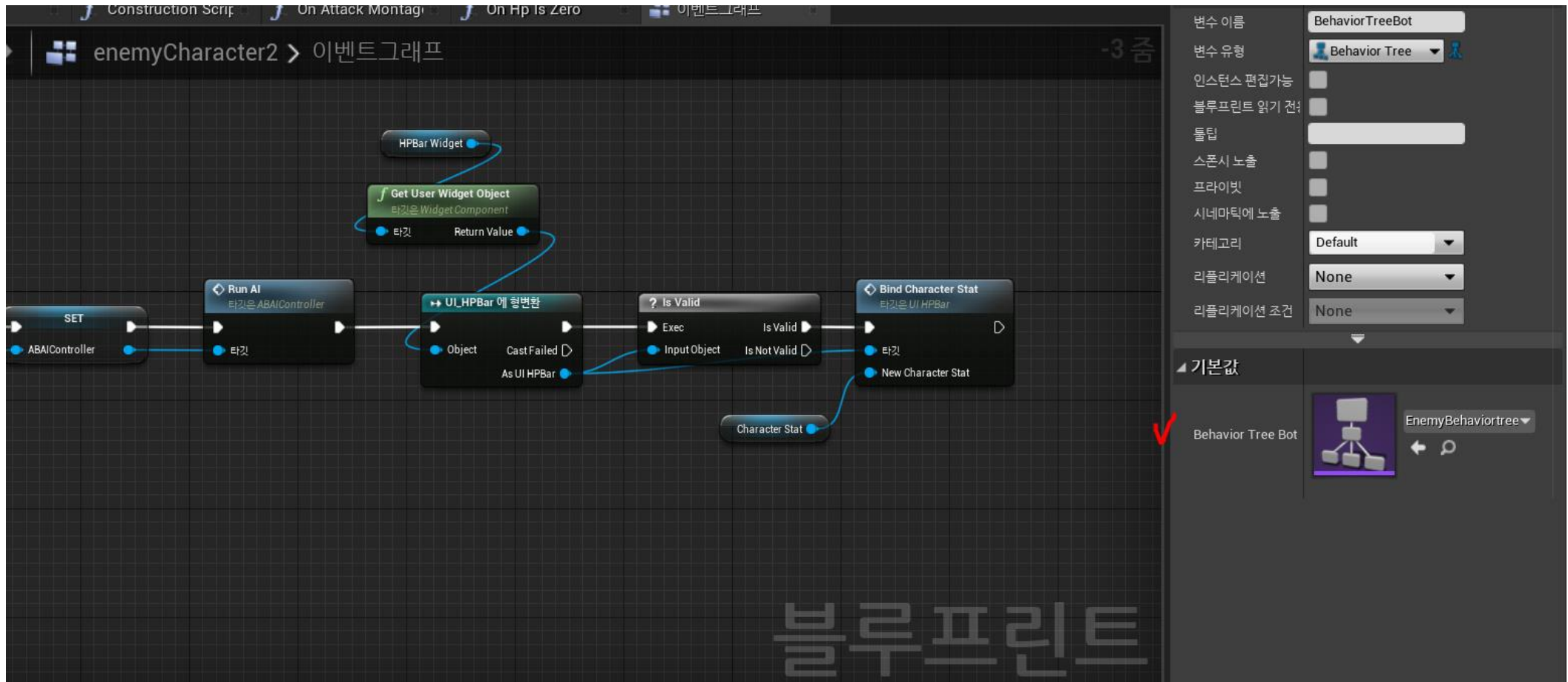
10.1 PerformConditionCheckAI을 오버라이드해서 ABAIController의 GetIsAttacking을 반환한다.



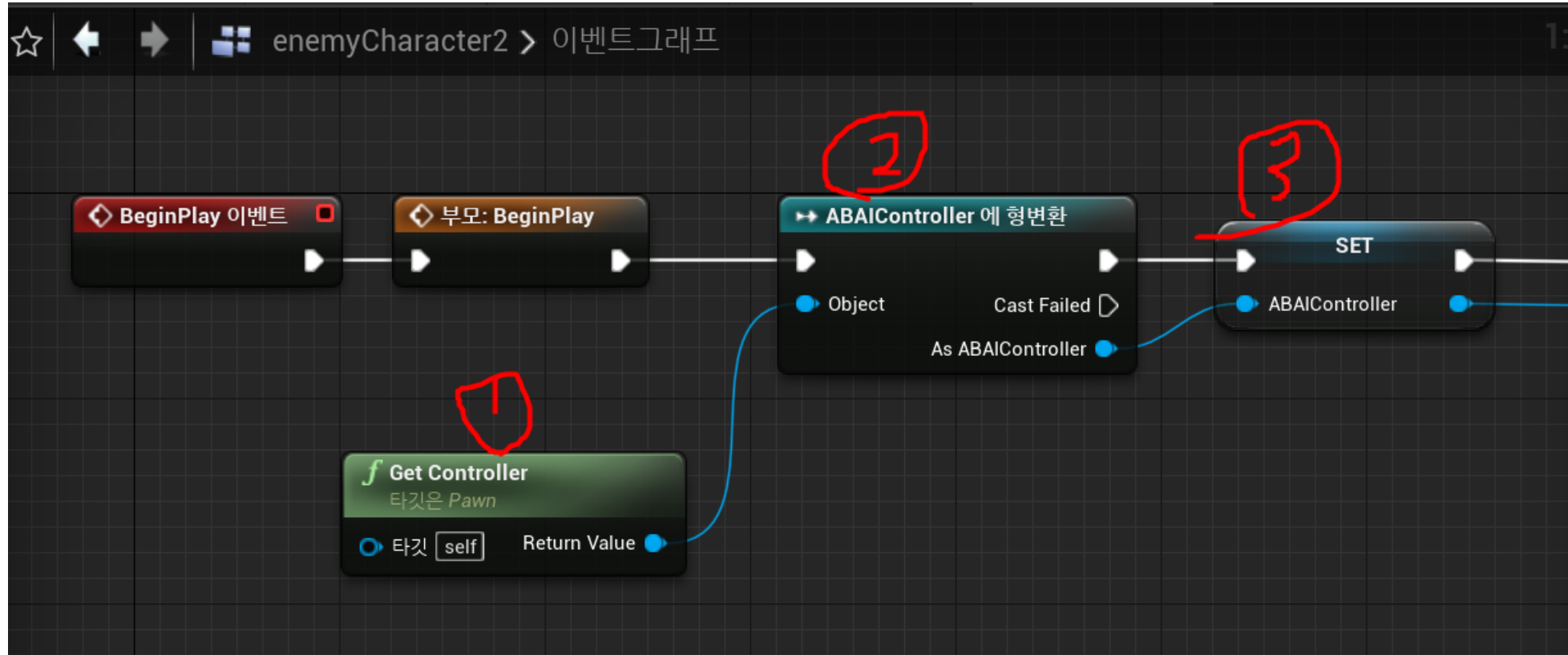
10.2 CanNotAttack의 자식으로 MoveTo Task를 추가하고 BTDecorator_Follow를 추가한다.



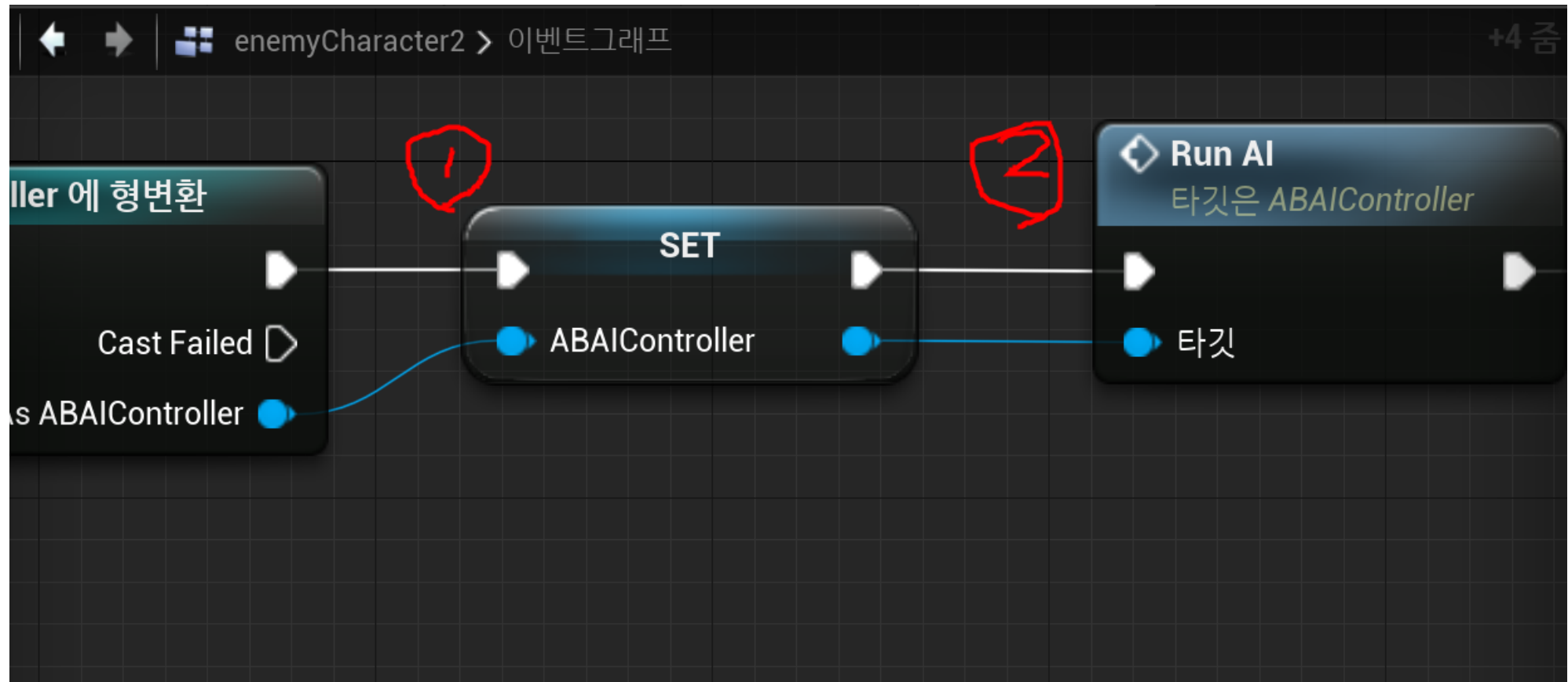
10.3 BTDecorator_Follow의 InverseCondition을 True로 한다.



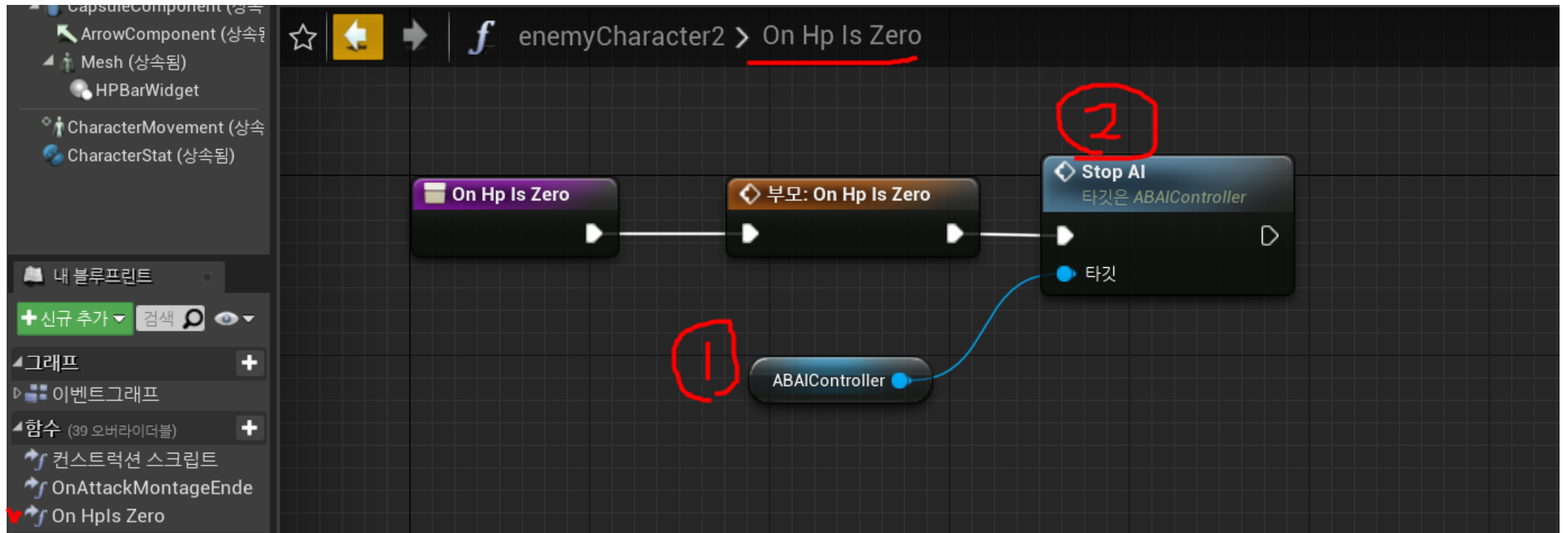
11. enemyCharacter2에 들어가서 BehaviorTreeBot 변수에 EnemyBehaviortree를 넣는다.



11.1 BeginPlay 이벤트에 Controller를 ABAIController로 형변화하고 변수로 승격해서 ABAIController를 생성한다.



11.2 ABAIController의 RunAI 함수를 호출한다.



11.3 OnHpIsZero 함수를 오버라이드 하고, ABAIController의 StopAI 함수를 호출한다.