

이만택 - 5

적 생성 및 AI

은혜로운 교수님의 행동트리

 [언리얼 엔진 비헤이비어 트리 노드 레퍼런스: 데코레이터 | 언리얼 엔진 5.4 문서 | Epic Developer Community](#)

Selector

- 여러 행동 중 하나의 행동을 실행, 왼쪽부터 실행하며 실패하면 다음 행동 실행

Sequence

- 여러 행동을 순차적으로 실행

Parallel

- 동시에 시행

데코레이터(Decorator)

- 조건 기반으로 노드의 실행을 제어
- 자식 노드의 실행 여부를 결정
- 예: 특정 조건을 만족할 때만 행동 실행

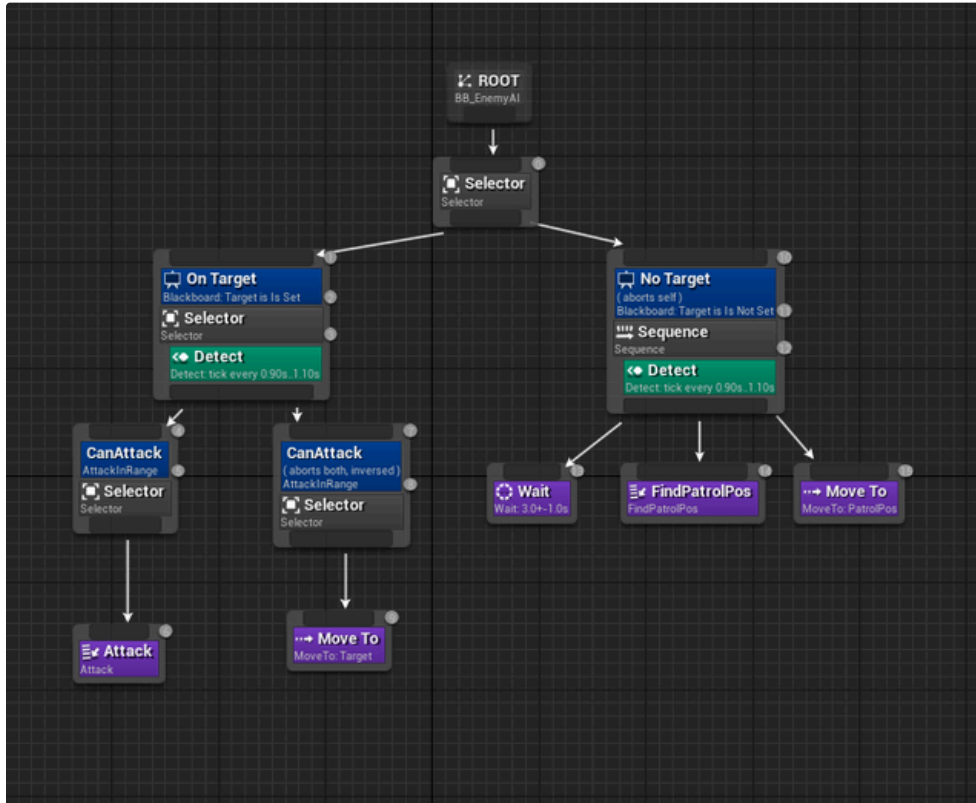
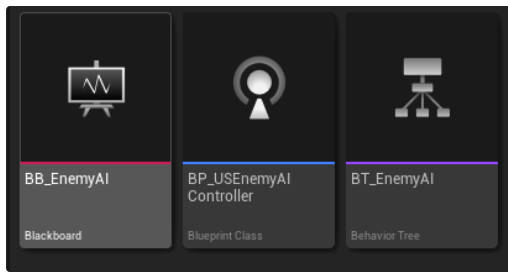
서비스(Service)

- 주기적으로 실행되어 상태를 업데이트
- 주기적인 작업을 통해 행동 트리를 보조
- 예: 주기적으로 적 탐지, AI 상태 체크

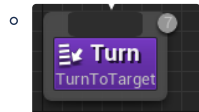
관찰자 중단(Observer Aborts)

- 조건 변화 감지 시 실행 흐름 제어
- 현재 노드 및 자식 노드의 실행 중단
- 예: 조건 변화 시 현재 행동을 중단하고 재평가

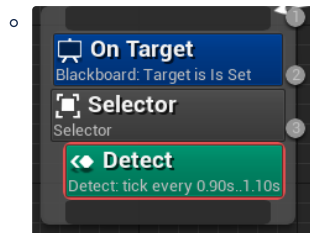
BlackBoard, Behavior Tree 생성



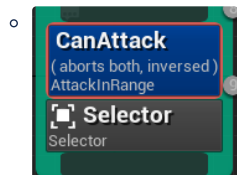
- UBTTask_TurnToTarget : UBTTaskNode (실제 행동을 하는 노드)



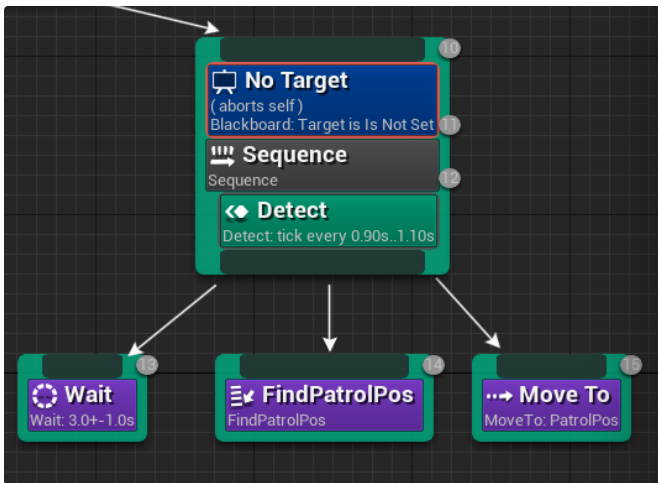
- UBTService_Detect : UBTService



- UBTDecorator_AttackInRange : UBTDecorator



캐릭터 찾는부분



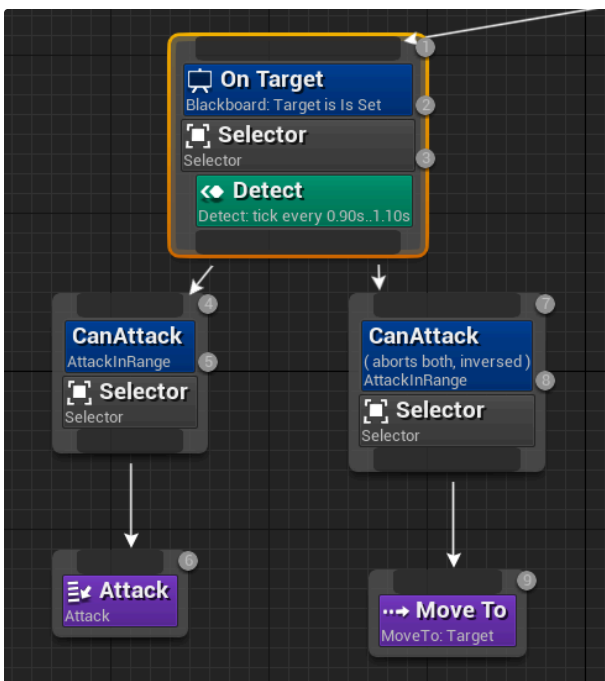
캐릭터를 찾는 행동에서 만약 찾는다면 즉시 변경

Flow Control	
Notify Observer	On Value Change
Observer aborts	Self
Blackboard	
Key Query	Is Not Set
Blackboard Key	Target
Description	
Node Name	No Target

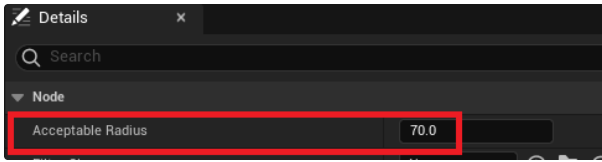
On Value Change - 관찰된 블랙보드 키가 변경될 때만 재평가

Self - 자신 및 이 노드 아래에서 실행 중인 모든 서브트리들을 중단

캐릭터를 찾아서 다가가서 공격



Move To로 캐릭터에게 다가간 이후 공격을 하였을 때 너무 붙어있어서 공격이 맞지 않아서 길이 조절을 함



교수님꺼만 하기엔 허전해서 콤보 추가



근데 공격중에 뒤로 가면 공격이 돌아야 하는데 안됨..

동영상 마지막 썸을 보니 그것도 추가되어 있어서 추가함👍

캐릭터를 찾는 탐색하는 부분이 원형으로 되어있어서 LOS(Line of Sight)를 적용 해보려함

일정 거리에 따라서 라인 트레이스를 하여서 확인

```
1  for (int32 i = 0; i < NumberOfSections; i++)
2  {
3      int32 L_CurrentAngle = DAngle* i - HalfAngle;
4
5      // 회전 축
6      FVector RotationAxis(0.0f, 0.0f, 1.0f);
7
8      // 회전 각도 (도 단위)
9      float RotationAngle = L_CurrentAngle;
10
11     // 벡터 회전
12     FVector RotatedVector = GetActorForwardVector().RotateAngleAxis(RotationAngle, RotationAxis) * ExternalRadius;
13     FHitResult HitResult;
14     FCollisionQueryParams QueryParams;
15     QueryParams.bTraceComplex = true;
16     QueryParams.AddIgnoredActor(this); // 이 액터는 트레이스에서 제외
17
18     bool bHit = GetWorld()->LineTraceSingleByChannel(
19         HitResult,
20         StartPoint,
21         EndPoint,
22         ECC_Pawn,
23         QueryParams
24     );
25 }
```

unreal5Study DebugGame Unreal Editor 2024 05 28 22 24 29



실제 시야를 보여주기 위해서 UProceduralMeshComponent를 통해서 시야를 그려줌

```
1 TObjectPtr<class UProceduralMeshComponent> ProceduralMesh;
2
3 FVector LLeftPoint;
4 FVector LRightPoint;
5 TArray<FVector> Vertices;
6 TArray<int32> Triangles;
7 for (int32 Index = 0; Index < HitArray.Num() - 1; ++Index)
8 {
9     if (HitArray[Index].bBlockingHit)
10    {
11        LLeftPoint = HitArray[Index].Location;
12    }
13    else
14    {
15        LLeftPoint = HitArray[Index].TraceEnd;
16    }
17
18    if (HitArray[Index + 1].bBlockingHit)
19    {
20        LRightPoint = HitArray[Index + 1].Location;
21    }
22    else
23    {
24        LRightPoint = HitArray[Index + 1].TraceEnd;
25    }
26
27
28    Vertices.Add(StartPoint);
29    Vertices.Add(LLeftPoint);
30    Vertices.Add(LRightPoint);
31
32    Triangles.Add(0 + Index * 3);
33    Triangles.Add(2 + Index * 3);
34    Triangles.Add(1 + Index * 3);
35 }
36 TArray<FVector> Normals;
37 TArray<FVector2D> UVs;
38 TArray<FProcMeshTangent> Tangents;
39 TArray<FColor> VertexColors;
40 ProceduralMesh->CreateMeshSection(0, Vertices, Triangles, Normals, UVs, VertexColors, Tangents, true);
```



라인 트레이스로 확인하고 그 위치를 삼각형으로 만들어서 그려주는 형식

선이 많아지면 성능이 안좋아지는데 선을 너무 적게 그리면 부드럽게 나오지 않음

실컷 만들고..보니 UAISenseConfig_Sight, UAIPerceptionComponent가 이미 있음..

```
1 // Perception component 생성
2 AIPerceptionComponent = CreateDefaultSubobject<UAIPerceptionComponent>(TEXT("AIPerceptionComponent"));
3 // 시각 감지 기능 추가
4 SightConfig = CreateDefaultSubobject<UAISenseConfig_Sight>(TEXT("SightConfig"));
5
6 SightConfig->SightRadius = 2000.0f; // 탐지 거리
7 SightConfig->LoseSightRadius = 2200.0f; // 탐지 제한 거리
8 SightConfig->PeripheralVisionAngleDegrees = 90.0f; // 시야 각
9 SightConfig->SetMaxAge(1.0f); // OnPerceptionUpdated 업데이트 시간
10 SightConfig->AutoSuccessRangeFromLastSeenLocation = 0.0f; // 감지 된곳에서 다시 체크
11 SightConfig->DetectionByAffiliation.bDetectEnemies = true; // 적 감지
12 SightConfig->DetectionByAffiliation.bDetectNeutrals = true; // 중립 감지
13 SightConfig->DetectionByAffiliation.bDetectFriendlies = true; // 아군 감지
14
15 AIPerceptionComponent->ConfigureSense(*SightConfig);
16 AIPerceptionComponent->SetDominantSense(SightConfig->GetSenseImplementation());
17
18 BeginPlay()
19 AIPerceptionComponent->OnPerceptionUpdated.AddDynamic(this, &AUSEnemyAIController::OnPerceptionUpdated);
```

원래는 벽 뒤에 있는 캐릭터를 감지하지 않기 위해서 위에 작업을 한건데 이미 만들어져 있고 가져다 쓰기만 하면 되었음...

unreal5Study DebugGame Unreal Editor 2024 05 29 23 04 18



급격히 하려던게 사라짐... 큰일남

문제점을 발견함ㅜㅜ



캐릭터가 이동을 하지 않으면 업데이트를 하지 않음....

추가적으로 한번 걸러주는 작업이 결국 필요한 상태가 됨

물론 지금 상태에선 발견 즉시 찾는 상태에서 공격상태로 변경할 경우 내 캐릭터가 움직여야 하니 문제가 없을 것 같긴함...

아무튼 그냥 그대로 쓰기엔 뭔가 마음 같지 않음

AIController에 넣는거라 행동트리에 넣는것도 애매하고 그냥 한번 거르는 필터 정도로 생각해야 할 듯