

Day 11. AI Perception

11-1. AI Perception

```
SetPerceptionComponent(*CreateOptionalDefaultSubobject<UAI PerceptionComponent>(TEXT("AI Perception")));

SightConfig = CreateOptionalDefaultSubobject<UAI SenseConfig_Sight>(TEXT("Sight Config"));
SightConfig->SightRadius = 500.f;
SightConfig->LoseSightRadius = 600.f;
SightConfig->PeripheralVisionAngleDegrees = 90.f;
SightConfig->AutoSuccessRangeFromLastSeenLocation = -1.f;
SightConfig->PointOfViewBackwardOffset = 200.f;
SightConfig->NearClippingRadius = 200.f;
SightConfig->SetMaxAge(0.f);
SightConfig->DetectionByAffiliation.bDetectEnemies = true;
SightConfig->DetectionByAffiliation.bDetectNeutrals = true;
SightConfig->DetectionByAffiliation.bDetectFriendlies = true;

GetPerceptionComponent()->SetDominantSense(*SightConfig->GetSenseImplementation());
GetPerceptionComponent()->OnTargetPerceptionUpdated.AddDynamic(this, &AMonsterAIController::HandleSightSense);
GetPerceptionComponent()->ConfigureSense(*SightConfig);
```

저번에 구현했던 몬스터가 플레이어를 찾는 AI를 고치기로 했습니다. OverlapMultiByChannel로 플레이어를 찾던 것을 AIPercetpion을 사용하는 방향으로 고칠 예정인데 AI Perception을 사용하기 위해서는 SenseConfig에 대한 선언이 필요했습니다. AI Perception에는 촉각, 청각, 시각 등 다양한 감지하는 방식이 있었습니다. 일단은 시각으로 구현할 생각이므로 AISenseConfig를 Sight로 선언하여 AI Controller에 등록했습니다. 초기 세팅을 하는데 대부분의 속성들이 생소한 것 들이 많았습니다. AutoSuccessRangeFromLastSeenLocation 같이 음수로 값을 넣어줘야 시각적으로 타겟을 확인할 때만 타겟이 감지되어 벽 뒤에 숨은 플레이어를 찾을 수 없다던가 SetMaxAge의 값을 0으로 주어야 시각 정보가 감지 정보가 즉시 업데이트 돼서 지속적으로 쫓아오는 상황을 방지할 수 있다던가 하는 처음 듣는 속성들이 많았고 시각 감지 범위, 시야각도, 시야 거리 등에 대한 변수값도 어떤 물리값으로 적용되는지 알 필요가 있었습니다. AI Perception에 관하여 어느정도 공부한 뒤에 세팅을 끝마쳤고 플레이어를 찾는 과정은 HandleSightSense라는 함수를 만들어 actor와 stimulus 값을 받아 어떤 센서가 인식 되었는지, 어떤 캐릭터가 인식 되었는지 확인 후에 BlackBoard의 Target에 값을 넣어 완료했습니다.