## C++: RayCast

장애물 처리

충돌된 액터가 플레이어일 때만 공격 실시

```
Evoid ARangedMonster::HasObstacle()

{
FHitResult outHit;

FVector Start = this->GetActorLocation(); // 레이캐스트의 시작점
FVector End = Start + this->GetActorForwardVector() * 100000; // 레이캐스트의 종료점

FCollisionQueryParams CollisionParams;

CollisionParams.AddIgnoredActor( this ); // 현재 액터는 충돌 검사에서 무시

// 레이캐스트를 수행하고 충돌 정보를 outHit에 저장.
bool blsHit = GetWorld()->LineTraceSingleByChannel( outHit , Start , End , ECC_Visibility , CollisionParams );

if (blsHit)
{
    DrawDebugLine( GetWorld() , Start , End , FColor::Red , false , 0.1f , 0 , 5.f );
    UE_LOG( LogTemp , Warning , TEXT( "Obstacle detected!" ) );
}
else
{
    DrawDebugLine( GetWorld() , Start , End , FColor::Green , false , 0.1f , 0 , 5.f );
    UE_LOG( LogTemp , Warning , TEXT( "No Obstacle detected!" ) );
}
```

```
if (blsHit)
{

// 충돌한 액터를 처리

AActor* HitActor = outHit.GetActor();

if (HitActor)
{

// 충돌한 액터가 플레이어가 아니라면

AProjectDCharacter* player = Cast<AProjectDCharacter>( HitActor );

if (player)
{

RangedAttack();

}

else
{

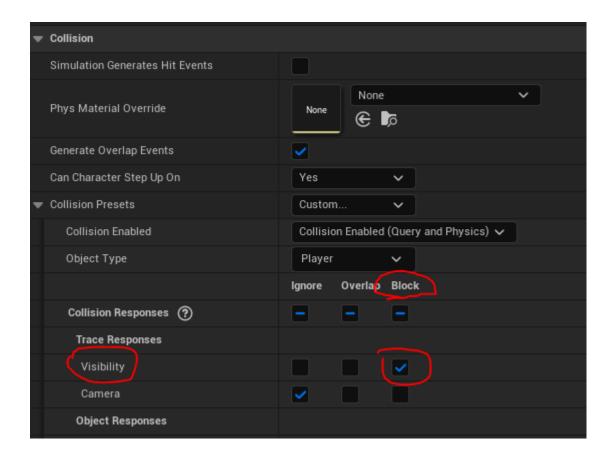
//PatrolState로 전환

MonsterFSM->state = EMonsterState::Move;

anim->animState = MonsterFSM->state;
}
}
```

C++: RayCast

## Visibility→Block 처리 !!



C++: RayCast 2