

Day 25. 일반 발사/Zoom

25-1. 일반 발사 구현

```
void AGunCharacter::Attack()
{
    if (ShouldAttack && !IsRunning) {
        IsAiming = true;
        if (!IsZooming) {
            WeaponGun->ZoomShoot();
            GetWorld()->GetTimerManager().SetTimer(FireHandle, FTimerDelegate::CreateLambda([&]() {
                WeaponGun->ZoomShoot();
            }), 0.2f, true);

            FTimerHandle waitHandle;
            IsAimWaving = true;
            GetWorld()->GetTimerManager().SetTimer(waitHandle, FTimerDelegate::CreateLambda([&]() {
                IsAimWaving = false;
            }), 1.5f, false);
        }
        else {
            WeaponGun->Shoot(GetMesh()->GetSocketTransform("gun_barrelSocket"));
        }

        if (GetWorld()->GetTimerManager().IsTimerActive(AimWaitHandle)) {
            GetWorld()->GetTimerManager().ClearTimer(AimWaitHandle);
        }

        GetWorld()->GetTimerManager().SetTimer(AimWaitHandle, FTimerDelegate::CreateLambda([&]() {
            IsAiming = false;
        }), 10.0f, false);
    }
}
```

Zoom을 하지 않는 상황에서 일반적인 공격에 대하여 고민해 보았습니다. 처음에 생각한 것은 카메라가 향하고 있는 방향으로 총을 쏘는 공격이었지만 앞서 캐릭터들과 너무 달라 이질감이 느껴졌습니다.. 그래서 Zoom 상태에서 총을 발사하는 상황을 강제하게 만드는 플레이를 요구고 실용성이 떨어지더라도 그냥 캐릭터가 향하고 있는 정방향으로 총을 쏘는 공격으로 만들었습니다.

캐릭터가 총을 견착한 애니메이션으로 움직이는 것은 부자연스러기 때문에 총의 견착 유무에 따라 비전투모드와 전투모드로 나누었고 전투모드에 대한 돌입은 Zoom을 하거나 일반 발사를 하는 것으로 조건을 만든 뒤 이 전투모드에 대한 상태는 따로 FTimerHandle을 만들어 관리하게 하였습니다. 그리고 이 FTimerHandle에서 전투 상태와 관련된 bool 변수를 true로 만들고 입력이 있다면 TimerHandle을 ClearTimer하고 다시 TiimerHandle을 갱신하고 없다면 10초 후에 변수를 false로 만드는 형태로 만들었습니다.

25-2. Zoom 구현

일반 공격에 대한 구현을 마쳤으므로 Zoom에 대한 구현을 했습니다. Zoom은 마우스 오른쪽 키에 press와 release로 바인딩하였고 입력이 있을 때 Tick()함수에 있는 SprimgArm의 TargetArmLength 값을 100.0f로 확대되게 하고 관련된 bool 변수들의 값이 바뀌게 하였습니다. 그런데 Tick에서 값을 주다보니 실시간으로 값이 반영되어 부자연스러울 정도로 빠르게 Zoom 상태가 되었고 해당 문제를 고치기 위하여 FInterpto를 주어 값을 조금 보간하였습니다. 전투모드와 비전투모드 각각에 대한 Zoom도 고려해 필요한 작업들도 처리해주었습니다. 그리고 라이플 무기에도 Shoot()에 더해 ZoomShoot() 함수를 만들어 라이플 총구 소켓 대신 컨트롤러의 Transform 값을 GetPlayerViewPoint함수에 OUT 매개변수로 주어 바뀐 Transform 값을 얻고 이것을 LinetraceSingleByChannel에 넣어 line을 쏘는 형태로 총의 발사를 구현하였습니다. 마지막으로 Zoom이 끝날때는 관련된 값들을 원래대로 돌아오는 방식으로 설계하였습니다.