



# 오브젝트가 카메라 시야 가릴 때

## 방법 1: 오브젝트를 투명하게 만들기

: 카메라와 플레이어 사이에 있는 오브젝트를 감지하여 투명하게 만들기

### 1. 라인 트레이스(Line Trace)로 감지하기

먼저, 카메라와 플레이어 사이에 있는 오브젝트를 감지하기 위해 라인 트레이스를 사용

```
void AMyCharacter::CheckCameraObstruction()
{
    // 카메라와 플레이어 위치
    FVector CameraLocation = CameraComponent->GetComponentLocation();
    FVector PlayerLocation = GetActorLocation();

    // 라인 트레이스 결과를 저장할 변수
    FHitResult HitResult;

    // 라인 트레이스 설정
    FCollisionQueryParams Params;
    Params.AddIgnoredActor(this);

    // 라인 트레이스 실행
    bool bHit = GetWorld()->LineTraceSingleByChannel(HitResult, CameraLocation, PlayerLocation, ECC_Visibility);

    if (bHit)
    {
        AActor* HitActor = HitResult.GetActor();
        if (HitActor)
        {
            // 오브젝트를 투명하게 처리
            UMeshComponent* MeshComponent = HitActor->FindComponentByClass(UMeshComponent::StaticClass());
            if (MeshComponent)
            {
                MeshComponent->SetVisibility(false);
            }
        }
    }
}
```

```

        MeshComponent->SetRenderCustomDepth(true);
        MeshComponent->CustomDepthStencilValue = 1;
    }
}
}
}

```

## 2. 후처리(Post Process)로 투명 처리

- 라인 트레이스로 감지한 오브젝트를 후처리 효과를 통해 투명하게 만들기
- 스텐실 버퍼를 사용하여 특정 오브젝트만 투명하게 처리할 수 있음

```

// PostProcessVolume 설정
APostProcessVolume* PostProcessVolume = GetWorld()->SpawnActor<APostProcessVolume>();
PostProcessVolume->bUnbound = true;

FWightedBlendable WightedBlendable;
WightedBlendable.Object = MyPostProcessMaterial; // 투명
처리를 위한 포스트 프로세스 머티리얼
PostProcessVolume->Settings.WightedBlendables.Array.Add(WightedBlendable);

```

## 3. 후처리 머티리얼 설정

- 투명 처리를 위한 후처리 머티리얼을 설정
- 이 머티리얼은 스텐실 버퍼 값을 읽어 해당 값을 가진 오브젝트를 투명하게 처리

```

// Material 설정 (머티리얼 에디터에서 설정)
if (CustomDepthStencilValue == 1)
{
    // 투명 처리 로직
}

```

## 방법 2: 카메라 위치 조정

: 카메라와 플레이어 사이에 오브젝트가 있을 때 카메라 위치를 조정하여 시야를 확보하는 방법

```
void AMyCharacter::AdjustCameraPosition()
{
    FVector CameraLocation = CameraComponent->GetComponentLocation();
    FVector PlayerLocation = GetActorLocation();

    FHitResult HitResult;
    FCollisionQueryParams Params;
    Params.AddIgnoredActor(this);

    bool bHit = GetWorld()->LineTraceSingleByChannel(HitResult, PlayerLocation, CameraLocation, ECC_Visibility, Params);

    if (bHit)
    {
        // 카메라를 히트된 위치로 이동
        FVector NewCameraLocation = HitResult.Location;
        CameraComponent->SetWorldLocation(NewCameraLocation);
    }
}
```