

Network : Owner, Role, NetConnection의 이해

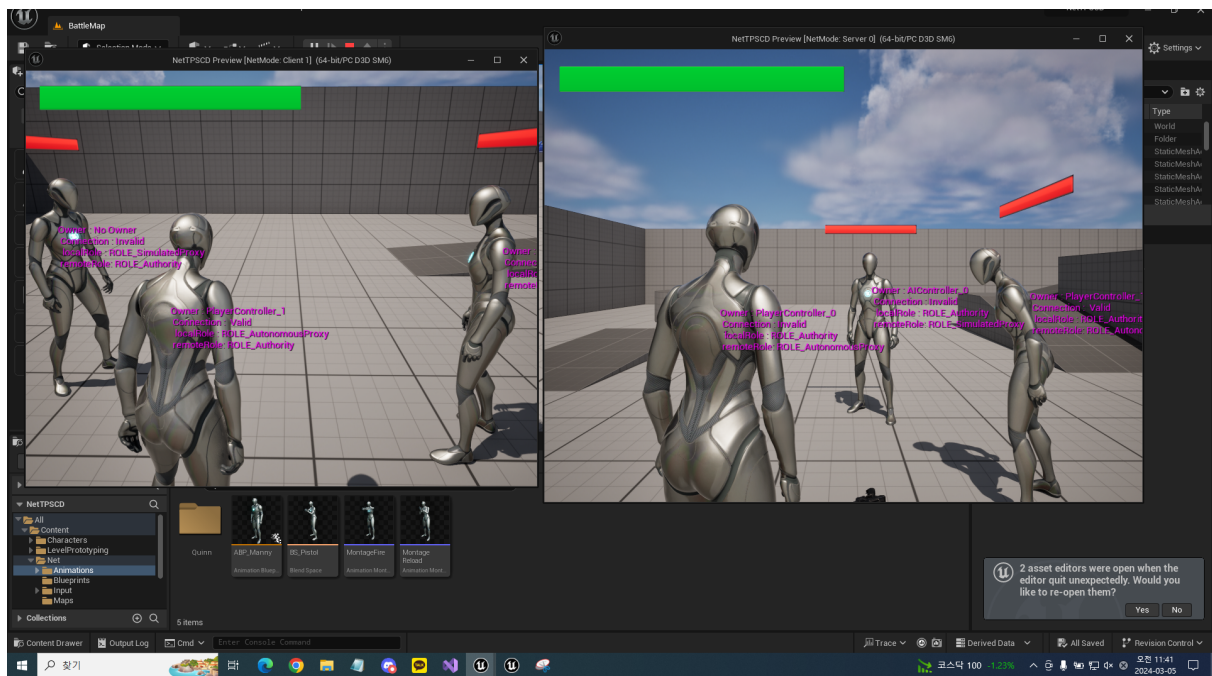
- NetInfo 출력하기
 - NetActor 구현 및 Owner 변경하기
 - 속성 리플리케이션을 이용한 회전
-

Tick 에서 호출해주기

```
void ANetTPSCDCharacter::PrintNetLog()
{
    //UE_LOG( LogTemp , Warning , TEXT( "11" ) );
    //오너가 있는가?
    FString owner = GetOwner() ? GetOwner()->GetName() : TEXT( "No Owner" );
    //NetConnection이 있는가
    FString conn = GetNetConnection() ? TEXT( " Valid " ) : TEXT( "Invalid " );
    //LocalRole
    FString localRole = UEnum::GetValueAsString<ENetRole>( GetLocalRole() );
    //RemoteRole
    FString remoteRole = UEnum::GetValueAsString<ENetRole>( GetRemoteRole() );

    FString str = FString::Printf( TEXT( "Owner : %sWn Connection : %sWn localRole : %sWn remoteRole: %s" )
        , *owner , *conn,*localRole,*remoteRole );
    FVector loc = GetActorLocation() + FVector( 0 , 0 , 50 );
    DrawDebugString( GetWorld() , loc,str,nullptr,FColor::Magenta,0,true,1.2f );
}
```

리슨서버로 플레이



```
//속성 리플리케이트
UPROPERTY(Replicated)
float rotYaw;

void SelfRotation(const float& DeltaTime);

virtual void GetLifetimeReplicatedProps(TArray<FLifetimeProperty>& OutLifetimeProps) const override;
```

```

void ANetTestActor::SelfRotation( const float& DeltaTime )
{
    //회전
    //만약 서버라면
    if(HasAuthority())
    {
        // : 실제로 회전하고 그 결과를 rotYaw변수에 담고 싶다
        AddActorWorldRotation( FRotator( 0 , 360 * DeltaTime , 0 ) );
        rotYaw = GetActorRotation().Yaw;
    }
    //그렇지 않고 클라이언트라면
    else
    {
        // 현재 회전값의 Yaw값을 rotYaw값으로 반영하고 싶다
        FRotator rot = GetActorRotation();
        rot.Yaw = rotYaw;
        SetActorRotation( rot );
    }
}

```