

# Network : 개요

NetTPS 프로젝트 생성 및 프로젝트 설정

언리얼 네트워크의 개론

- 다중접속기반 게임프레임워크 이해
- 서버, 클라이언트 모델
- Replication 시스템 이해
- 데이터 동기화 와 RPC 개념이해

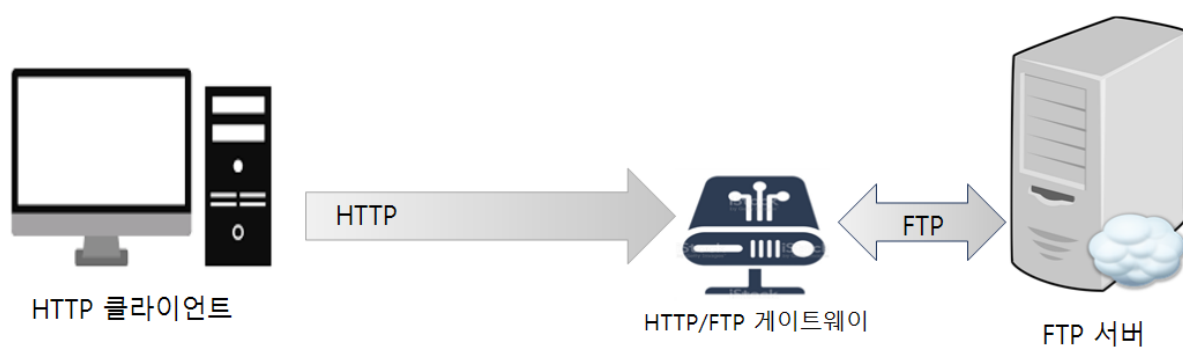
**라우터 (IP 공유기) :** 한 개 이상의 근거리 통신망(LAN) 간에 데이터를 전달하는 게이트웨이

→인터넷 프로토콜(IP)을 사용하여 데이터가 포함된 IP 패킷을 전송

**게이트웨이 :** 컴퓨터 네트워크에서 서로 다른 통신망, 프로토콜을 사용하는 네트워크 간의 통신을 가능하게 하는 컴퓨터나 소프트웨어를 두루 일컫는 용어,

→ 다른 네트워크로 들어가는 입구 역할을 하는 네트워크 포인트

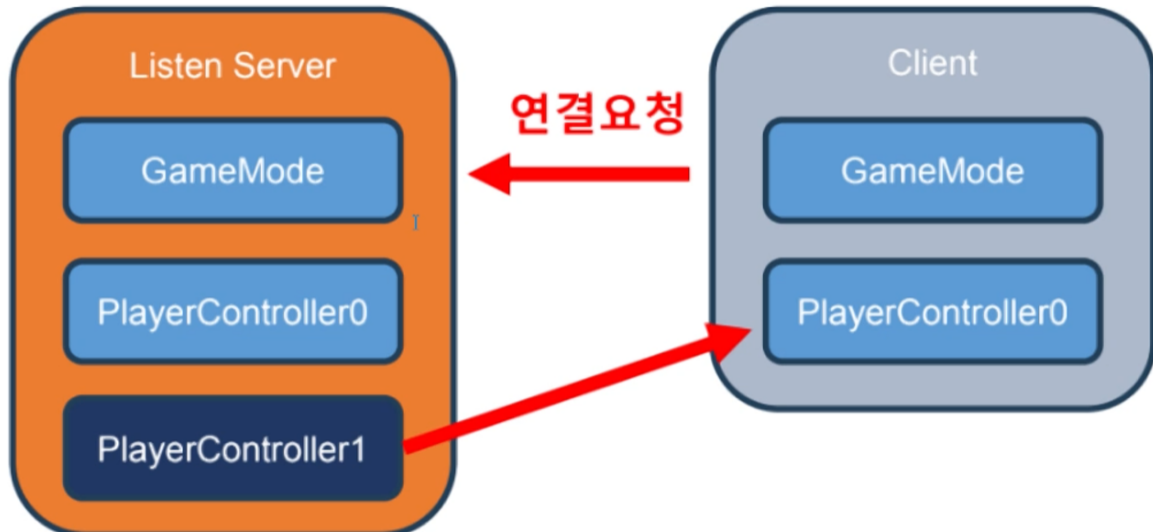
→ 클라이언트와 서버 사이의 중개자 역할



## 언리얼 네트워크

언리얼 엔진 내부에서

클라이언트 접속 → 연결 요청 → 서버에서 PlayerController 생성



클라이언트 쪽에서 플레이어 움직여 → 입력값을 서버에 전달 → 서버에서 실제 플레이어 이동 시킴 → 값을 다시 클라이언트로 전달

⇒ 실제로 모든 일은 서버에서 처리, 클라이언트는 요청만

서버에 원본이 있고 클라이언트에 허상

## 언리얼 네트워크 구조 – 연결에 따른 흐름

### 1. 네트워크 진행 흐름

#### ◆ Client 접속

1. 새로운 Client 가 접속하면 서버의 GameMode 에서 로그인을 허락할지 말지를 결정할 수 있다.
2. 로그인이 허락되면 서버는 새로운 PlayerController 를 하나 만든다.
3. 이렇게 만들어진 PlayerController 는 Client 에도 복제가 되며 이를 Proxy 라고 한다.
4. PlayerController 를 통해 Connection 을 갖게되며 이를 이용해 Client 는 서버와 통신할 수 있게된다.



#### ◆ Client Login 부터 게임 시작까지

1. PreLogin : 처음 Client 가 접속 요청을 하면 게임모드의 PreLogin 함수가 불린다. 이 곳에서 Client 의 연결을 허용할지 여부를 결정한다. **리슨서버 로그인시에는 호출되지 않는다.**
2. Login : Client 의 Player Controller 를 만드는 단계
3. PostLogin : 플레이어에 필요한 기본 설정을 마무리 하는 단계
4. StartPlay : 게임의 시작을 알리는 단계
5. BeginPlay : StartPlay 함수로 부터 모든 액터의 BeginPlay 함수가 호출된다.

I

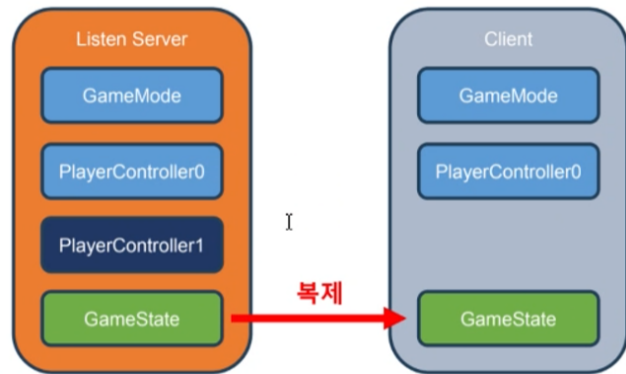


## GameMode

이곳에 게임 룰을 작성 ex) 점수

## ◆ Gamemode 역할

1. 네트워크상에서 서버에만 존재한다.
2. 클라이언트의 연결 권한을 갖는다.
3. 게임 진행에대한 권한을 갖는다.(예. 게임시작)
4. 서버에서 클라이언트의 PlayerController 가 만들어지고 플레이어 설정까지 끝나면 PostLogin 함수가 호출된다.
5. 게임모드는 서버에만 존재하기 때문에 클라이언트에 직접 명령을 내리지 않고 GameState 클래스를 이용해 전달한다.
6. 이후 GameMode 의 StartPlay 함수를 호출하면 GameState 객체를 이용해 클라이언트의 BeginPlay 함수가 호출되도록 명령을 내린다.



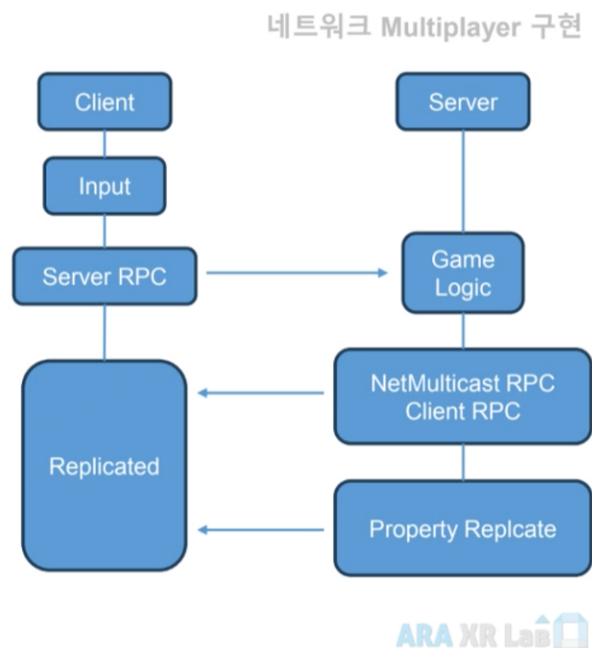
## 1. 변수 리플리케이션

## 2. RPC (Remote Procedure Call) 함수 리플리케이션

## 언리얼 네트워크 구조

### ◆ 통신처리 구조

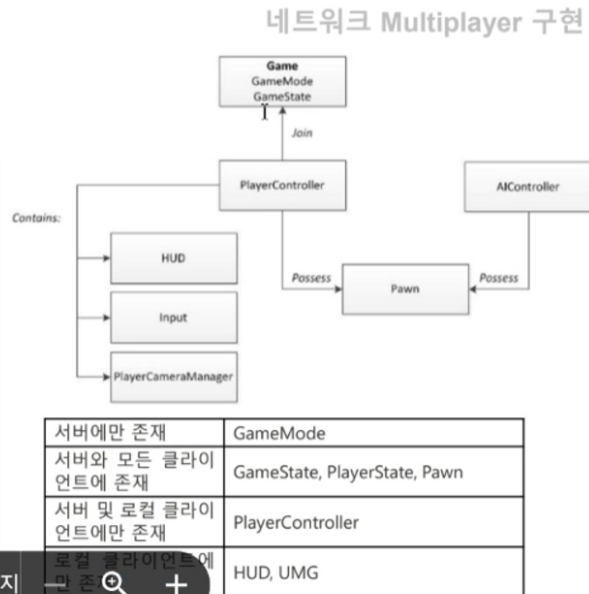
- client 에서는 자신의 단말기에서 Input 을 받는다.
- client 에서 동기화를 위해 단독으로 게임 로직에대한 처리는 절대 하지 않는다.
- Server 에 RPC 를 요청해서 서버로부터 처리를 유도한다.
- Server 에서만 유일하게 게임로직을 처리할 수 있다.
- Server 에서는 자신이 처리한 내용을 2가지 방식으로 클라이언트한테 복제(Replicate) 시킬 수 있으며, 이는 RPC 와 Property Replicate 를 이용한다.
- 모든 객체(액터포함)의 생성과 소멸은 Server 만이 담당을 한다.(예외 UI, effect 같은 cosmetic 요소들)



## 언리얼 네트워크 구조

### ◆ 네트워크 게임 프레임워크

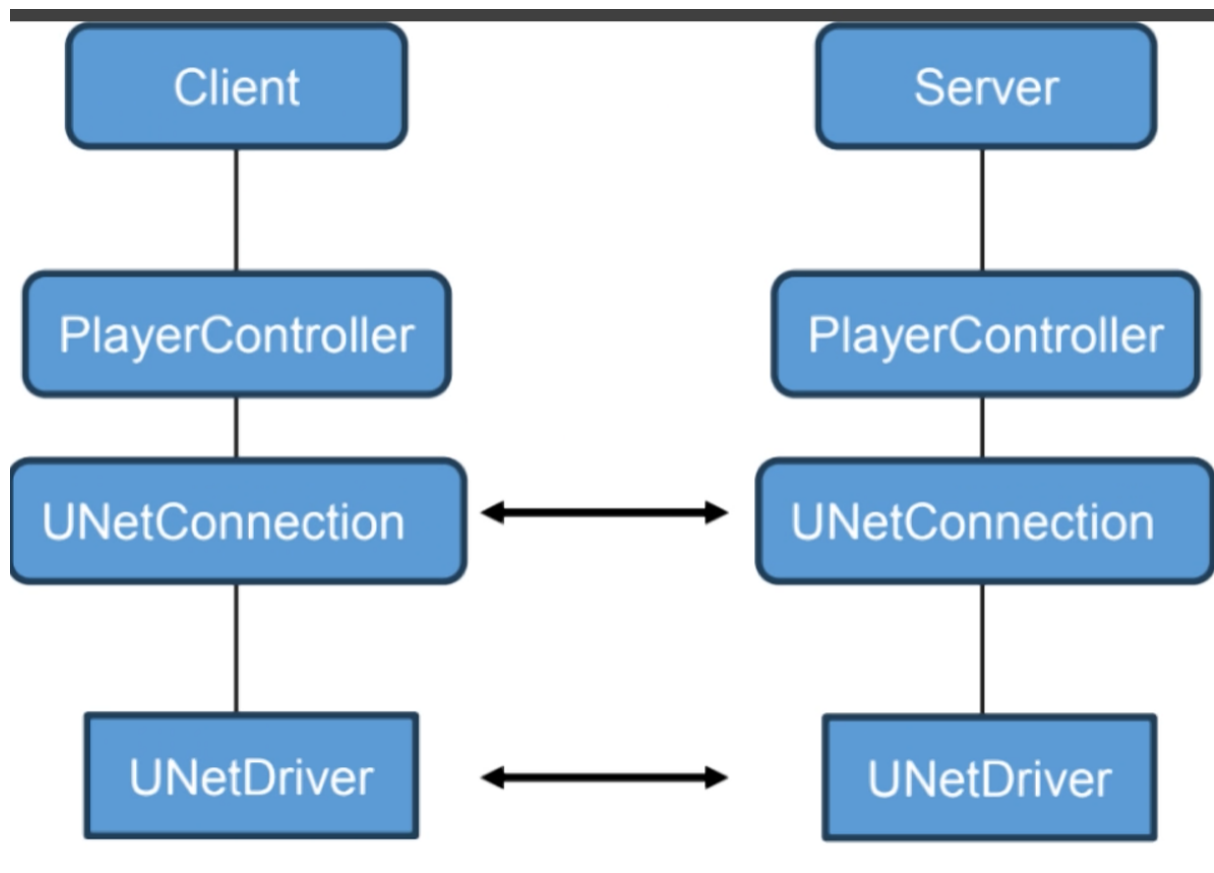
GameMode	게임의 기본 규칙을 정의한다.
GameState	게임의 상태를 모니터링 해주는 역할을 한다. (경과 시간, 각 팀의 점수 등)
PlayerState	플레이어의 이름, 킬/데스/어시스트 비율, 선택 종족 등의 정보를 관리한다.
PlayerController	폰에 빙의해서 유저의 의도를 전달하는 역할을 한다.
Pawn	플레이어를 시각적으로 표현한다.
HUD	디스플레이에 기본적인 모양(shape)과 텍스트를 그리는 역할을 한다.
UMG	복잡한 그래픽 요소를 화면에 그리는 역할을 한다.



## 언리얼 통신 방법

플레이어 컨트롤러만 통신 가능 ( ai controller 는 안됨)

UNetConnection을 통해 통신



서버 : write, response, recieve

클라이언트: read, request, send


editor preference에서

always on top

allow late joining

### 언리얼엔진 네트워크 멀티플레이어 관련(설명 & 사이트모음)

GameMode -서버에서만 존재 -클라이언트가 알 필요가 없는 게임  
관련 정보들을 관리 - 모든 요소들을 지정할수있다? (디테일창에서) -  
ex) 아이템스폰? GamelInstance -서버와 각 클라이언트마다 별도로

 <https://ljhyunstory.tistory.com/45>

언리얼 공부