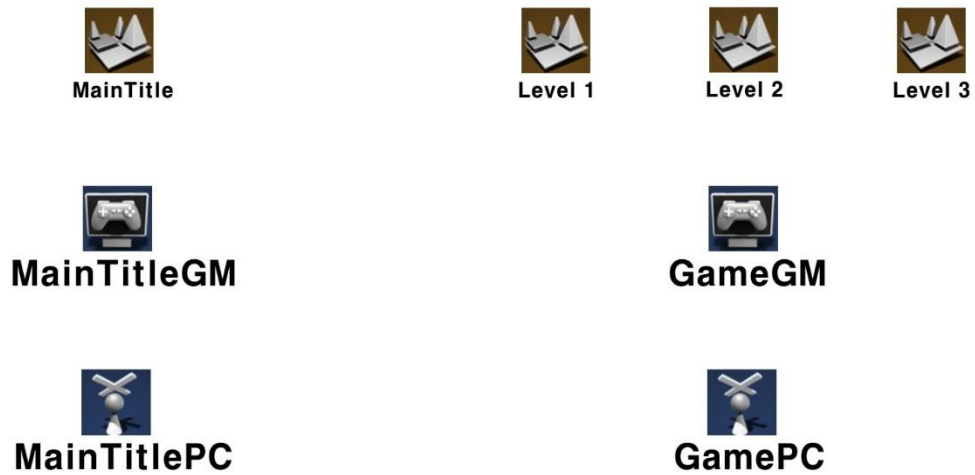


싱글게임



싱글 게임에서는 상호 베타적인 콘텐츠들을 레벨 단위로 끊는 경우가 많음

(ex) (Title – Robby - InGame)

GameMode 나 PlayerController 들도 레벨 별로 1개씩 만듬

GameMode에서 다 처리하기 때문에 따로 State가 필요하지 않음.

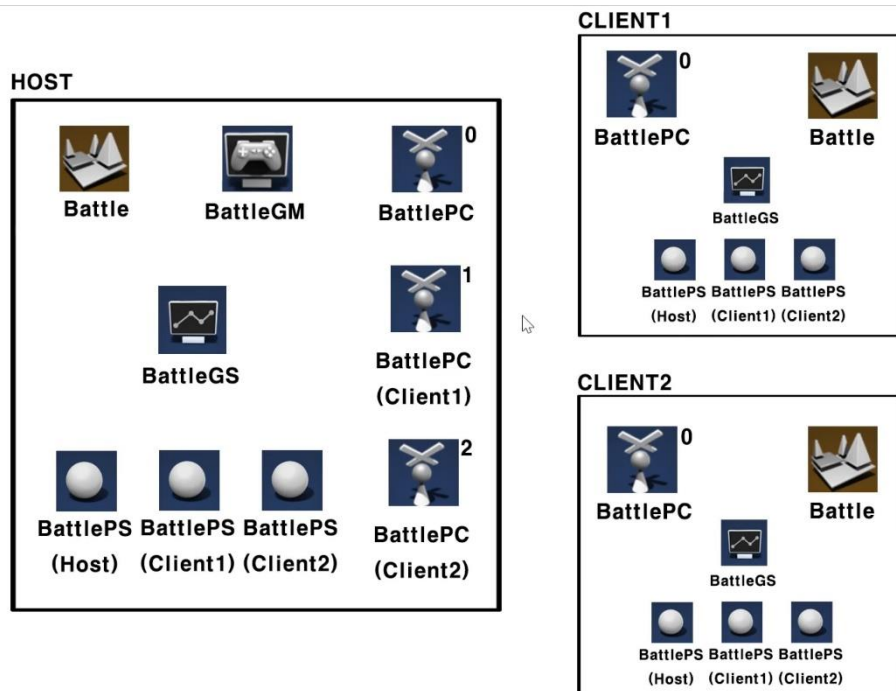
로컬멀티게임



로컬 멀티(화면 분할) 게임은 기본적으로 싱글 게임과 같은 구조이지만, Player Controller가 여러 개 있는 구조.

로직은 한 개이므로 GameMode 역시 1개

멀티 게임 구조



멀티 게임 구조는 호스트가 서버 역할을 하고 클라이언트들이 호스트 컴퓨터에 접속해서 게임을 진행하는 방식. 따라서, GameMode가 호스트에만 존재 (게임 로직에 대한 코드 존재)

언리얼에서 네트워크 연결을 위해 소켓을 여는 것은 레벨 이동을 통해서 하는 것이 정석

(Open Level에서 옵션으로 ?listen 켜줌.)

● 접속 흐름

먼저, 호스트가 Battle 레벨에 들어가고, 레벨을 로드함. 이때 호스트이기 때문에 GameMode가 존재하고, Battle Player Controller 0 를 가짐. GS(Game State)와 PS(PlayerState) 가짐

Game State는 게임 진행 메시지, 게임 상황, 팀별 점수 등 전체 클라이언트 들이 알아야할 정보들
Player State는 남이 알아야하는 내 정보들, 내가 알아야하는 남의 정보들 저장하는 곳

이때 클라이언트가 호스트에게 접속 요청을 보내고 호스트가 battle 레벨을 load하라고 알려줌

레벨 load를 완료하면, 호스트에게 알려주고 *** 이때 호스트는 클라이언트를 위한 PC와 PS를

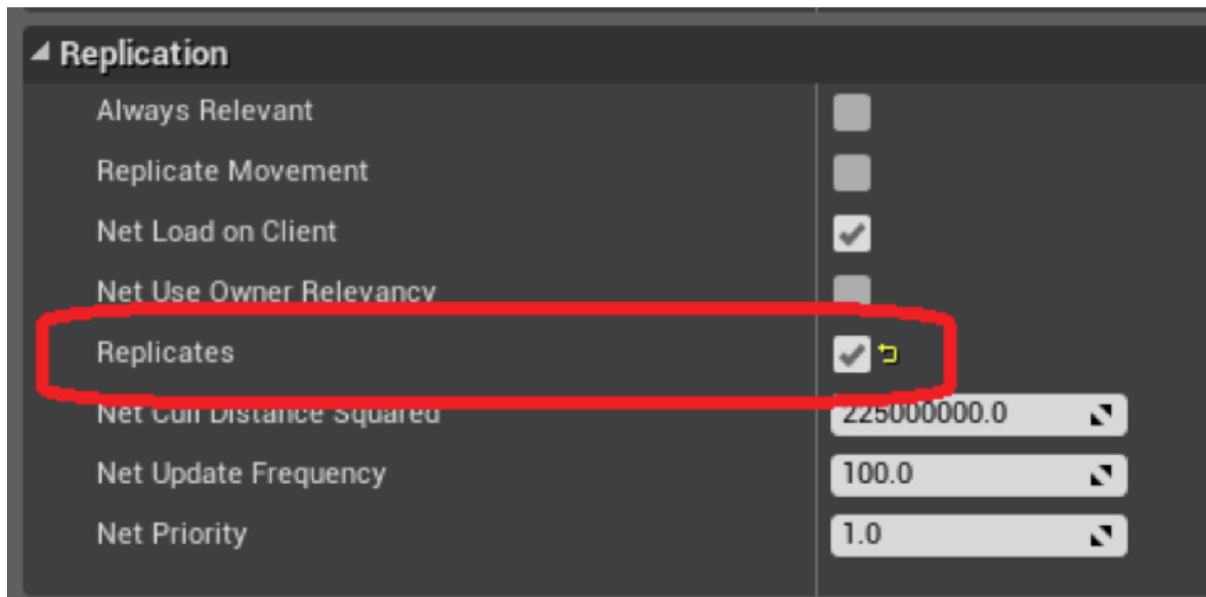
호스트 내에 새로 만듦. (BattlePC-Client1, BattlePS-Client1) 이걸 클라이언트에게 복사하라고 시킴
(PC는 Client1 에게만, PS는 전체 클라이언트 에게 복사) *** GS도 클라에게 복사.

이제, GS, PS, PC를 가지고 서버-클라간 통신 하는데, 이때 사용하는 기능이 replication.

- Replication

리플리케이션이란 복제와 같은 의미인데, 같은 이벤트나 변수에게 한 곳에서 실행하면 다른 곳에서 같은 일을 하도록 하게 하는 기능을 뜻함. 즉, 리플리케이션된 변수나 이벤트는 하나의

클라에서 실행했을 때 호스트 또는 다른 클라에게 같은 일을 하도록 혹은 값이 바뀌도록 해주는 기능. 언리얼에서는 이 기능을 이용해서 네트워크 통신. 일단 블루프린트 클래스 디폴트에서 리플리케이트 옵션을 켜줘야 함.



리플리케이션을 해주면 옵션에 따라 해당 클라 뿐만 아니라 다른 클라의 PC, 호스트의 PC와 연결해서 통신 가능

리플리케이션은 크게 변수 리플리케이트와 이벤트 리플리케이트

(다른 것도 있지만, 간단하게 설명하기 위해서)

*** 참고로 리플리케이션은 서버에서 클라로만 실행됨 ***

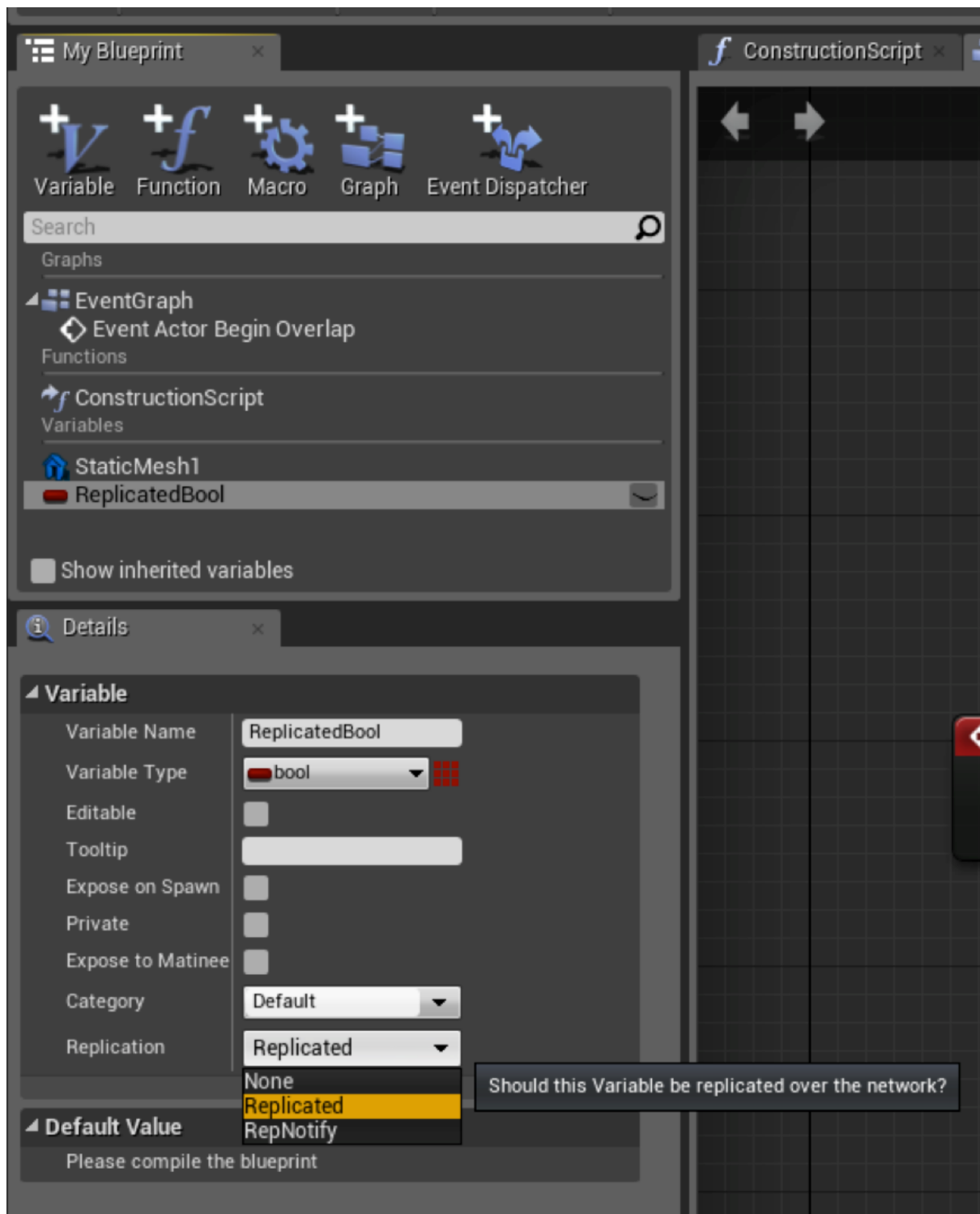
클라에서 서버로 통신을 할 수 없다는 소리가 아님(이거는 뒤에서 이벤트 옵션으로 다시 설명)

- 변수 Replication

변수를 리플리케이트 해주면 서버에서의 이 변수 값은 모든 클라이언트에 전송됨.

디테일 패널에서 Replication 옵션을 바꿔주면됨

None / Replicated / RepNotify 3가지



'RepNotify'는 리플리케이션이긴 한데 노티파이 이벤트가 클라 서버 양쪽에서 호출되도록 함
 이건 변수만 바꾸는게 아니라 그 바뀐 후 추가적인 작업을 해줘야 할 때 사용함
 ex) 체력 변경 후 (메시 교체, 사운드 재생 등등)
 RepNotify 선택하면 함수 자동 추가 (OnRep_변수명) 본문만 추가해주면 자동으로 호출됨

- 이벤트 리플리케이션 (RPC, Remote Procedure Call)

이벤트는 한 기계에서 호출된 뒤 실행되는 다른 곳으로 리플리케이션 가능

커스텀 이벤트 만들면 디테일 패널에 Replicates 옵션이 있음

Not / Run on Server / Run on owning Client / Multicast

- Run on Server (서버에서 실행)

클라이언트에서 서버로 통신할 때 사용, 클라이언트에서 이 이벤트 노드에 도달하면, 서버에게 이 이벤트를 실행하라는 패킷을 보내고 클라이언트에서는 실행하지 않고 그냥 넘어감.

- Run on owning Client (소유중인 클라이언트에서 실행)

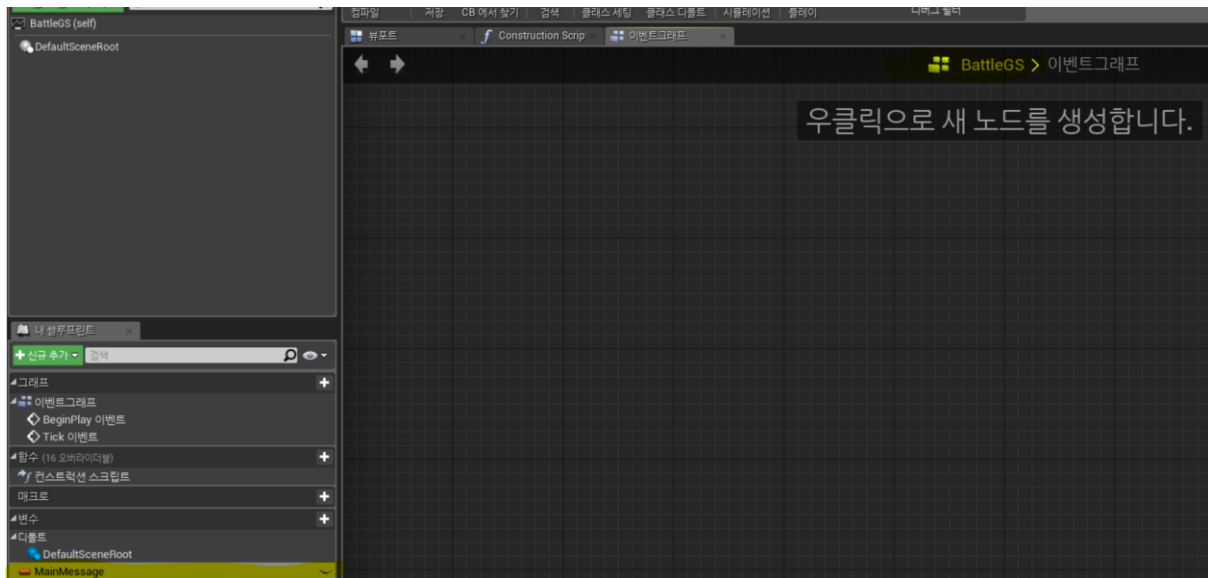
서버에서 클라이언트들에게 보내는 이벤트로 소유 중 클라이언트에게 패킷을 보내 실행하도록 함

- Multicast (모두가 실행)

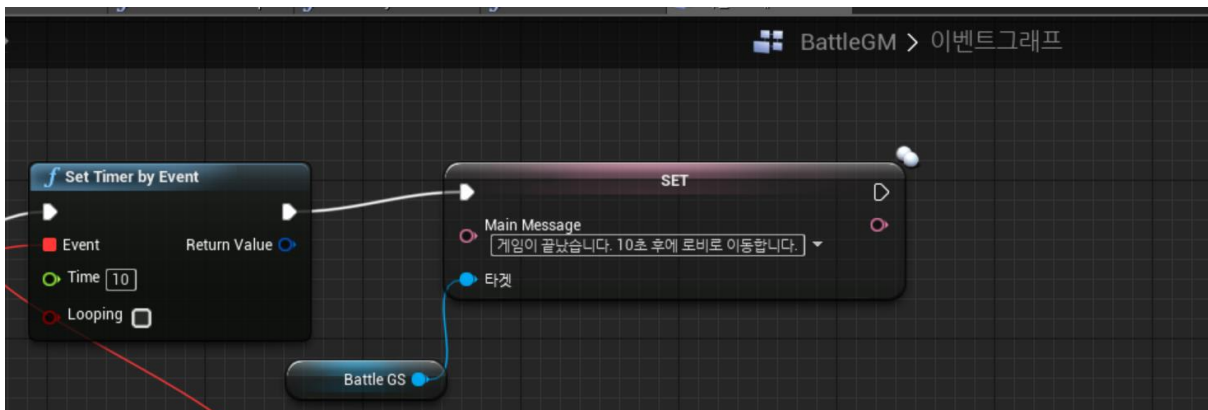
서버에서 호출되어 서버와 연결된 모든 클라이언트에서 실행되며, 리플리케이션 되는 아무 액터에서나 호출 가능. 모두에게 실행 가능해서 편하지만, 패킷이 지나치게 많이 돌아다니기 때문에, 적절한 곳에만 사용해야함.

이 옵션들을 적절히 사용해서 통신을 해야함.

- Game State 예시 (게임 진행 메시지 보내기)



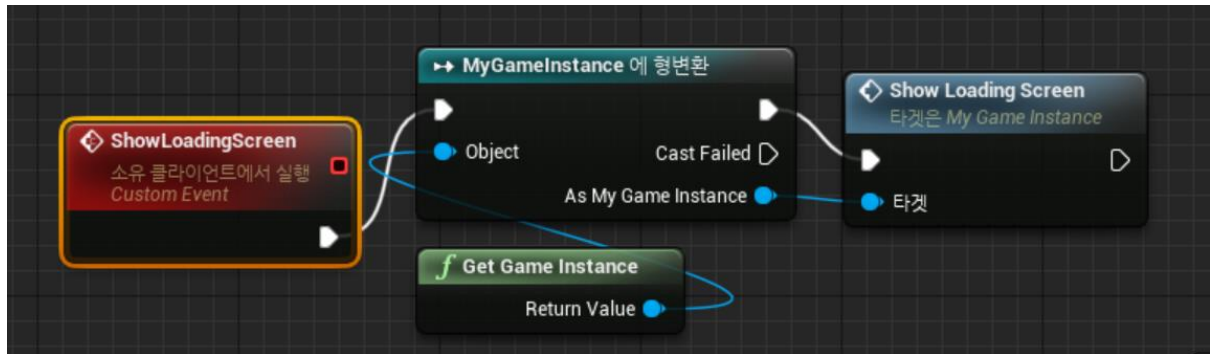
Game State에 text 변수 하나 선언해두고 리플리케이션 옵션 켜줌



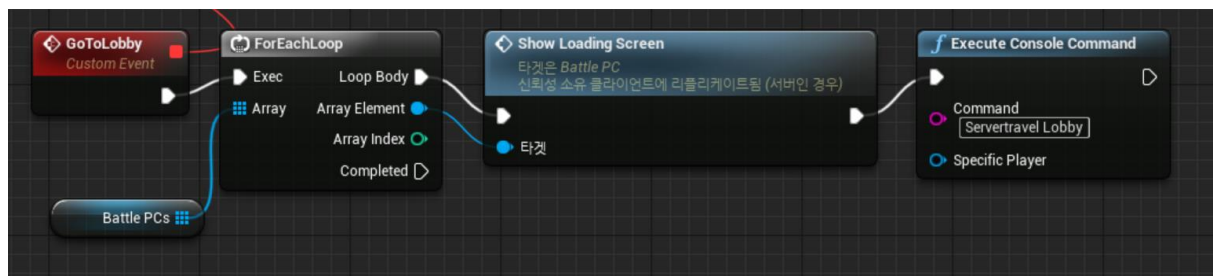
Game Mode(서버)에서 특정 이벤트 마다 Set Message 해줌

그러면 모든 클라에서 이 변수가 바뀌고 바뀐 값으로 사용 가능

- 소유 중인 클라이언트 리플리케이션 예시

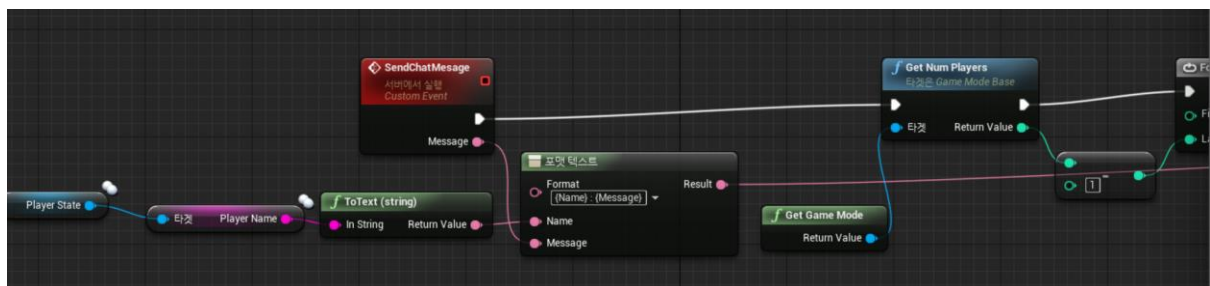


Custom Event 만들 때, 소유 클라이언트에서 실행으로 만들어 놓고.

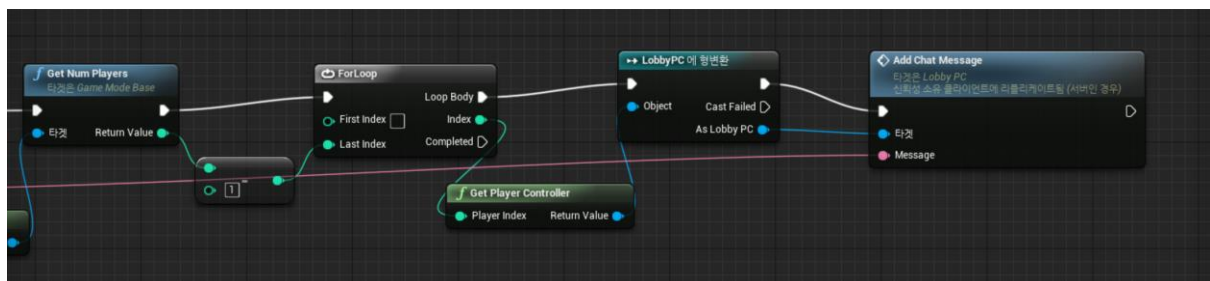


각각의 플레이어 컨트롤러에게 이 이벤트를 실행시키면 됨

- 서버에서 실행 리플리케이션 예시

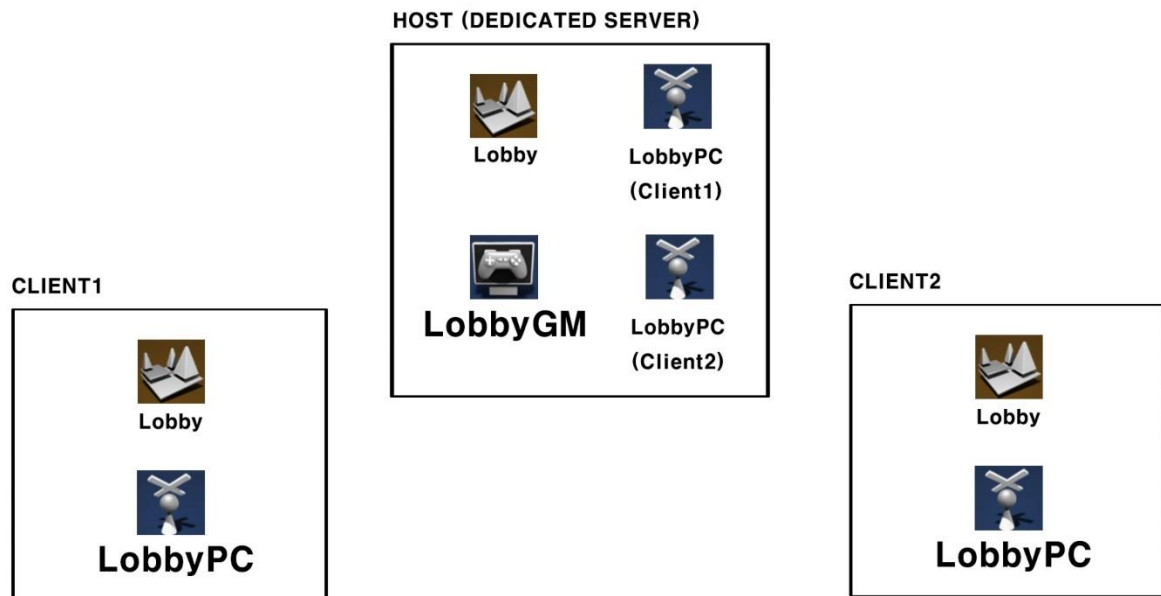


PlayerState에서 리플리케이션한 텍스트를 가지고 서버에서



각 클라에게 다시 보내줌 Add Chat Message(소유 클라 리플리케이션)

Dedicated Server 구조



위에서 설명한 멀티는 호스트 서버임 (호스트도 하나의 클라이언트)

이 경우 호스트가 플레이 중 렉이 걸리거나 접속이 해제되면 게임 자체가 멈춰버림

그래서 나온 것이 데디케이티드 서버

이 서버에서는 게임 로직만 돌리고 (GM), 입력이나 렌더링을 하지 않음 (PC)

즉, 서버는 서버 역할에만 집중하고, 클라들이 접속을 해제해도 게임에 영향을 주지않음.