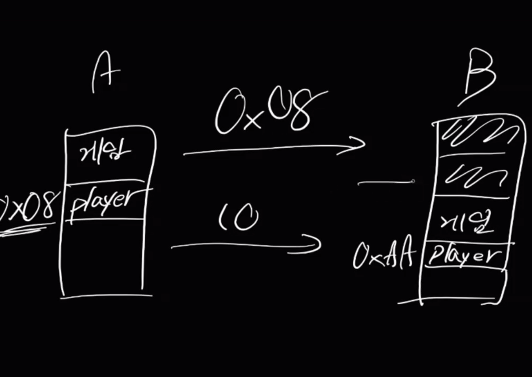
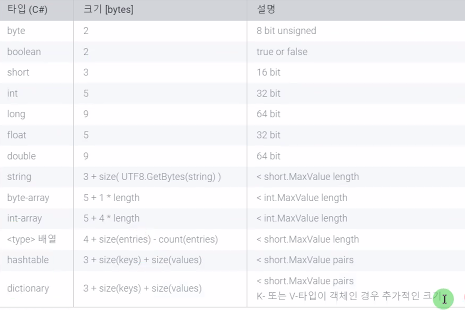
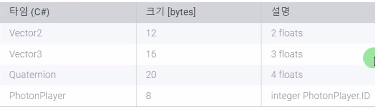
**네트워크 4**

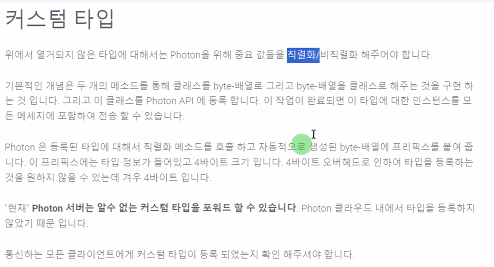


네트워크에서는 참조를 사용할 수 없다. 주소값을 전달하는것이기에 당연히 다른 기기에서도 같은 주소에 같은 값이 있을 가능성이 낮기에 그렇다.



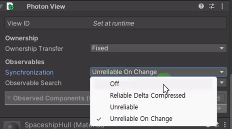


네트워크를 통해 전달 가능한 타입

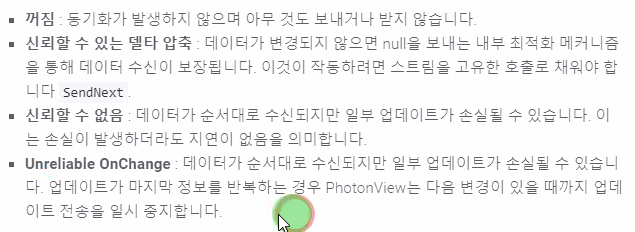


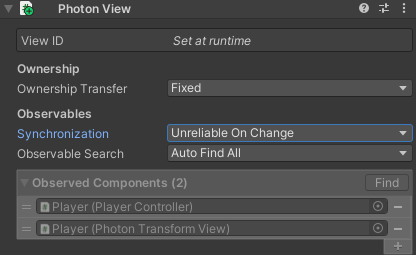
직렬화한 커스텀 타입또한 보낼 수 있다.

Photon View

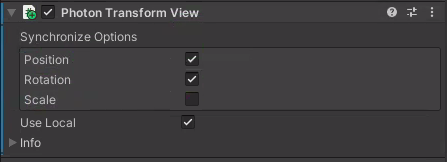


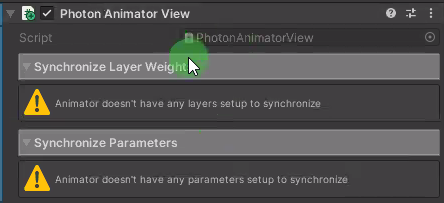
동기화 하는 주기.





어느것을 동기화 할지에 대한 리스트 (Auto find하면 관련된 View Components를 가져와 넣어준다.)





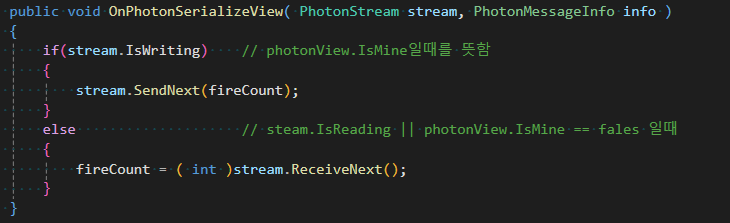
이 외에 Photon Rigid Body View 등 다양한 동기화 components가 있다.

또한 PhotonView를 참조하는 물체는

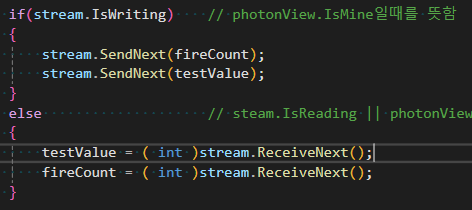


MonoBehaviorPun을 사용함으로서 를 사용해 손쉽게 바로 불러들일 수 있도록 만들어줄 수 있다.

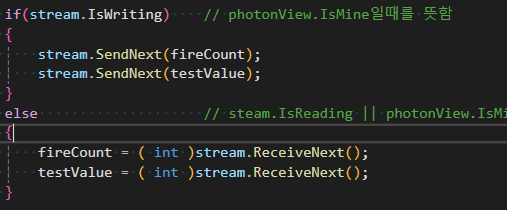
또한 성질을 가지고 있으면 PhotonView를 통해 리스트에 자동추가되어 동기화가 가능하다. 이 성질을 가지고 있다면



OnPhotonSerializeView를 통해 동기화할 정보를 넣어서 보낼 수 있다.



네트워크는 순차적으로 stream으로 받은 순서로 받는데 위와같이 넣으면 fireCount값을 넣은 것을 testValue에 넣게되는 문제가 발생한다. 똑같이 보낸갯수와 맞지않으면 문제가 발생한다.

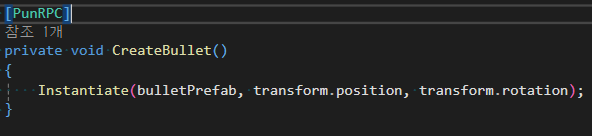


순서는 맞춰서 넣고 보낸 것은 꼭 받도록 하자.

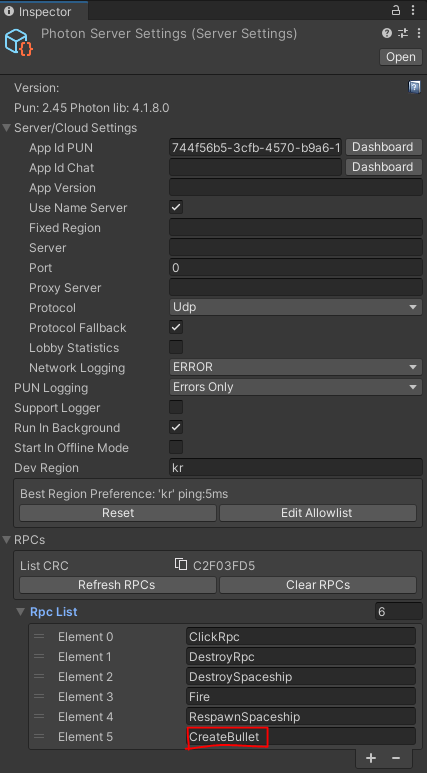


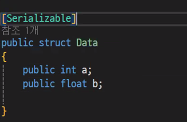
굳이 보내야 한다면 이런식으로 컴포넌트로 보내면 된다.

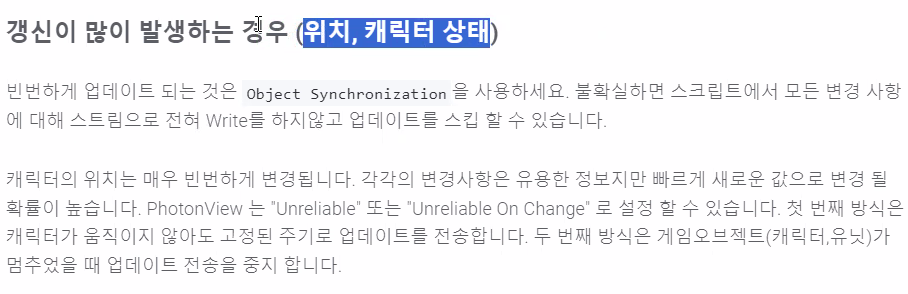
**PunRPC**

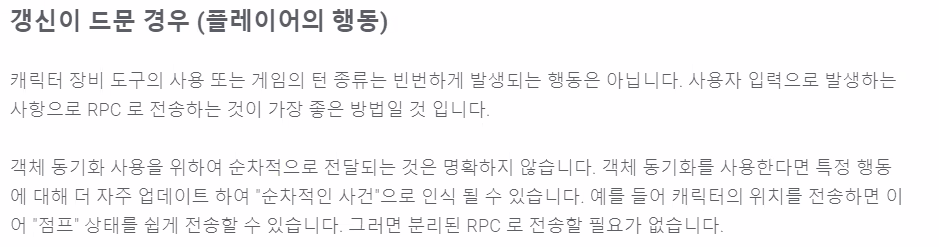


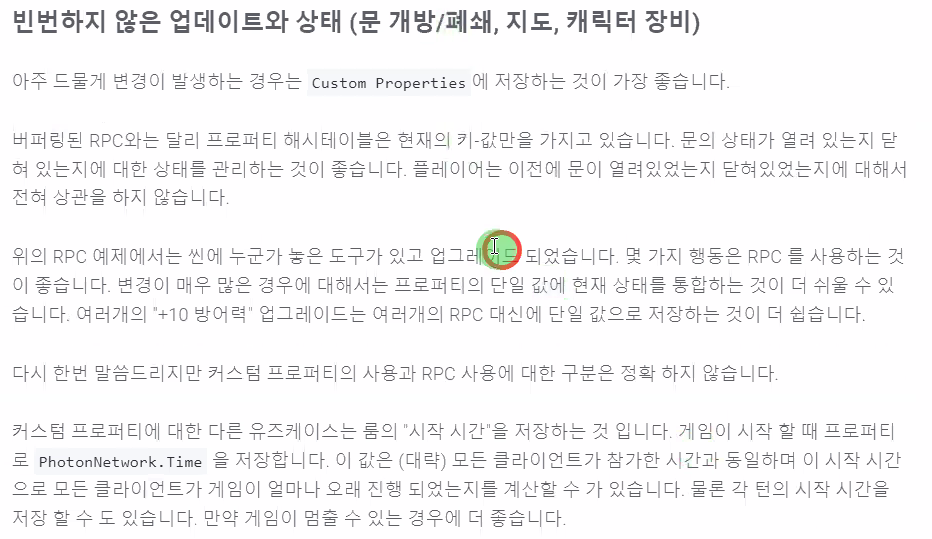
PunRPC 라는 어튜리뷰트를 달아 Photon에서 우선적으로 찾을 수 있도록 만들어준다.

이런식으로 RPC List에 추가해준다. (이름이 중복되면 어떤 곳에서 실행했는지 햇갈려할 수 있다.)

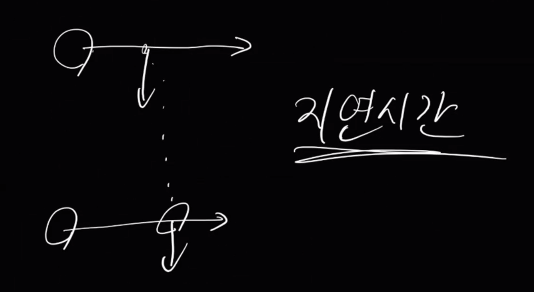
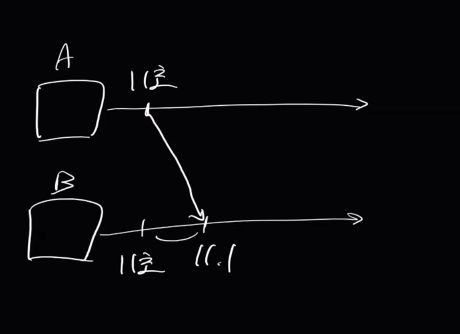
직렬화 가능해야만 보낼수 있다.

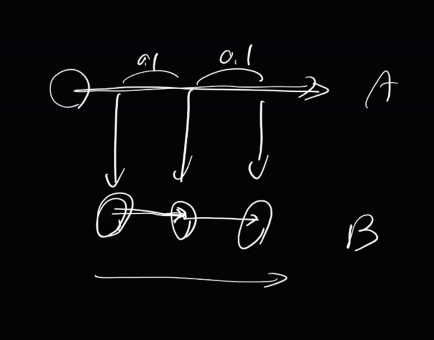
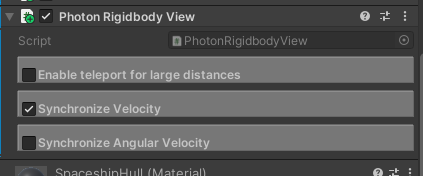




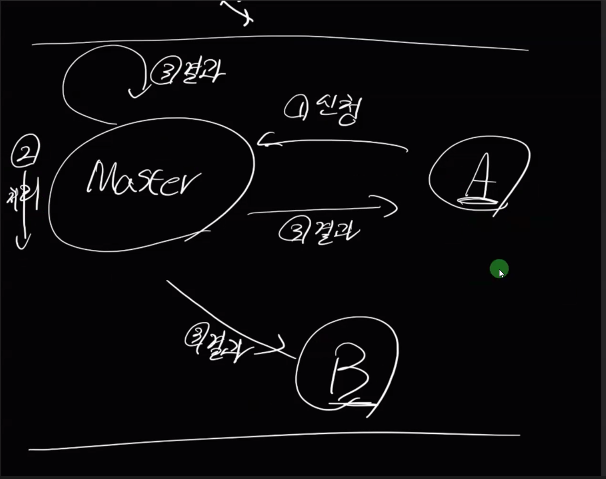


PING





지연보상을 줌으로서 부드럽게 움직이도록 한다.



하나의 결과물이 필요한 경우 이런식으로 처리하고 통보하는 방식으로 처리