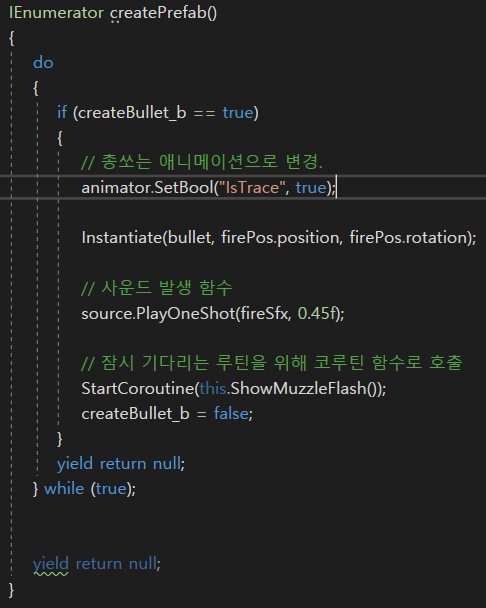
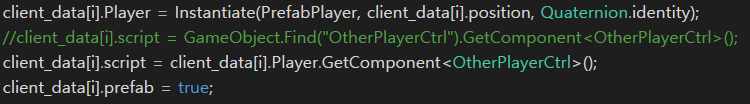
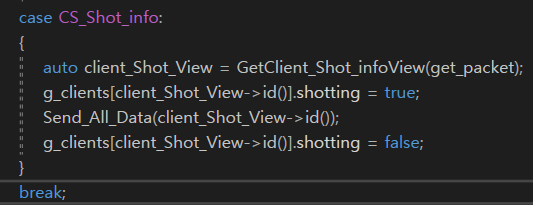
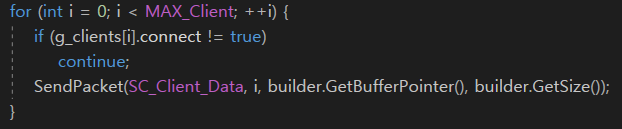
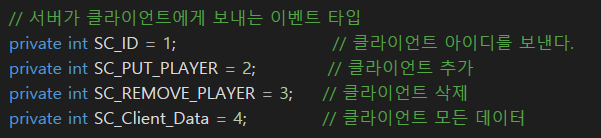
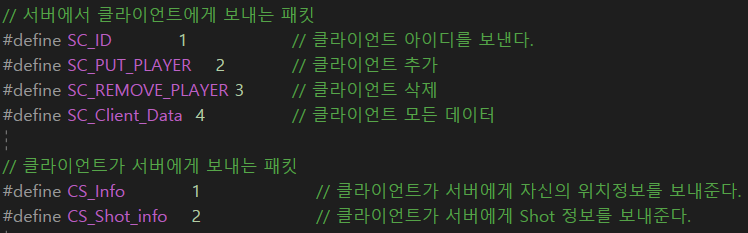
02월 15일 작업일지:

**클라이언트에서 Shot시 다른 클라이언트에서 해당 애니메이션 작동.**

OtherPlayerCtrl.cs 스크립트에서 다른 클라이언트 들이 Shot시 본인 클라이언트에서도 정상적으로 작동할 수 있게 수정.

* OtherPlayerCtrl.cs의 경우 NetworkCtrl.cs 함수에서 불러오게 되므로, 유니티 자체에서는 외부 함수에서는 프리팹 생성 및 객체에 대한 접근 권한이 없음.
  + 
  + Coroutine을 이용하여 외부 함수가 아닌 내부 함수에서 작동을 하게 만들었음.
  + 
  + 외부 함수를 사용할 수 있게 GetComponent를 사용하였다.
* **클라이언트 Update로 인하여 서버에 클라이언트가 Shot 에 대한 정보를 입력 받아도 Update의 속도로 인하여 Shot에 대한 정보가 사라짐**
  + 
  + 서버에서 클라이언트 들에게 전송을 할 때 Shot에 경우 한번 더 따로 전송을 할 수 있게 코드를 수정함.
* **서버가 클라이언트에게 데이터를 전송하였지만 Shot에 대한 정보가 들어오지 않음.**
  + 
  + 서버는 자신에 보낸 클라이언트에게만 전송을 하고 있었음.
    - 다른 클라이언트들이 잘 보이는 이유는 구조체를 이용하여 다른 클라이언트들의 위치를 모두 묶어서 전송을 하였기 때문
  + 클라이언트에게 전송 시 모든 클라이언트(접속된)에게 전송하도록 코드 수정
* 서버, 클라이언트 이벤트 타입이 어떤 건지 알 수 있게 추가
  + 
  + 
  + ↑ C#의 경우 Define이 없으므로, private int를 사용하여 이벤트 타입을 설정.
  + 
  + ↑ C++의 경우 protocol.h에 Define으로 추가함.
* **서버를 연동하여 총을 발사하여 각 클라이언트에서 보이는 화면.**

