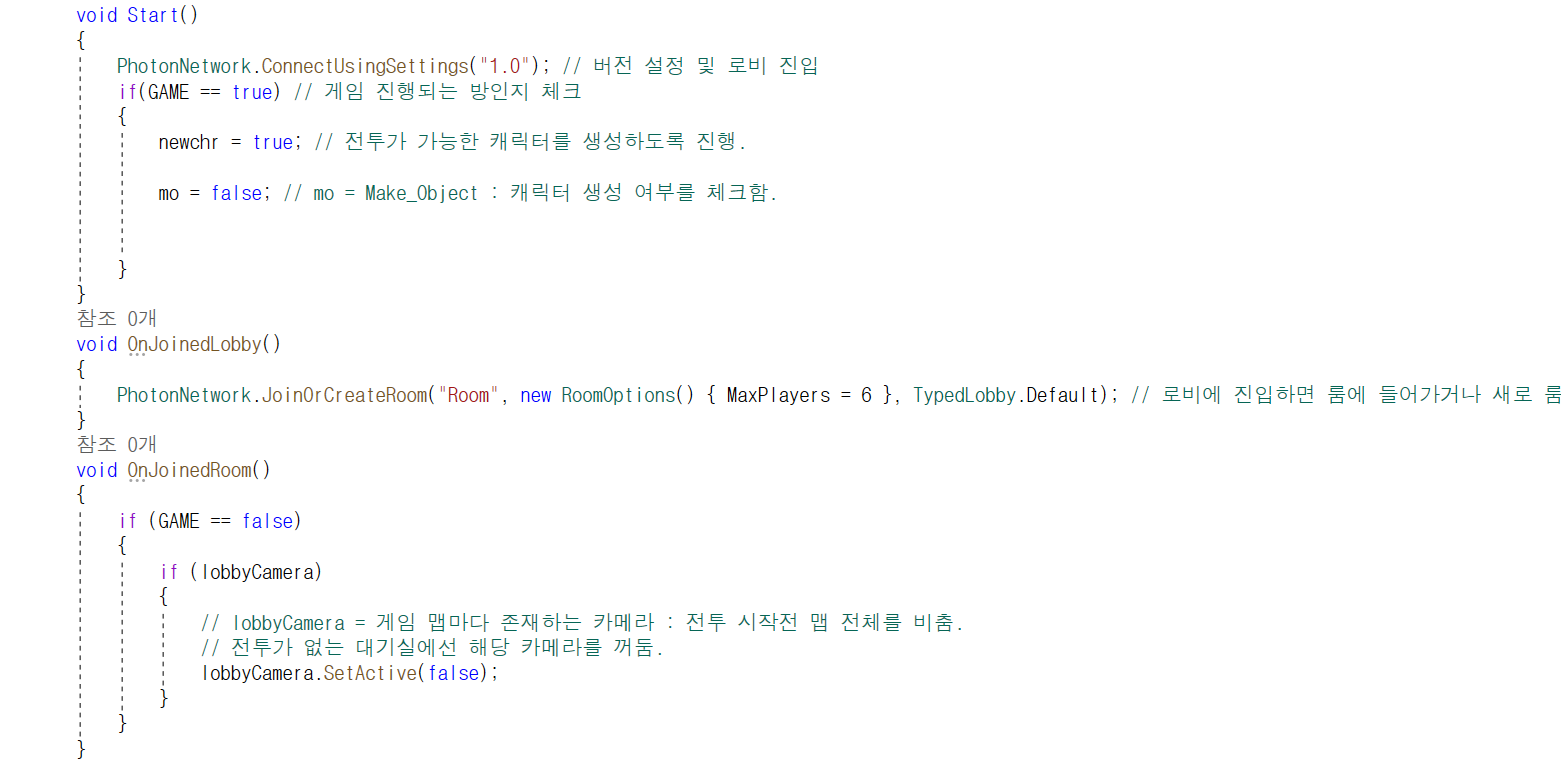
[서버 내 캐릭터 동기화 관련 클래스]

1. **PhotonManager\_클래스 :** 포톤 서버에 접속해 플레이어가 자신의 캐릭터를 생성하도록 돕고, 게임 내 진행을 책임진다. 맵에 한 개만 존재해야 함.

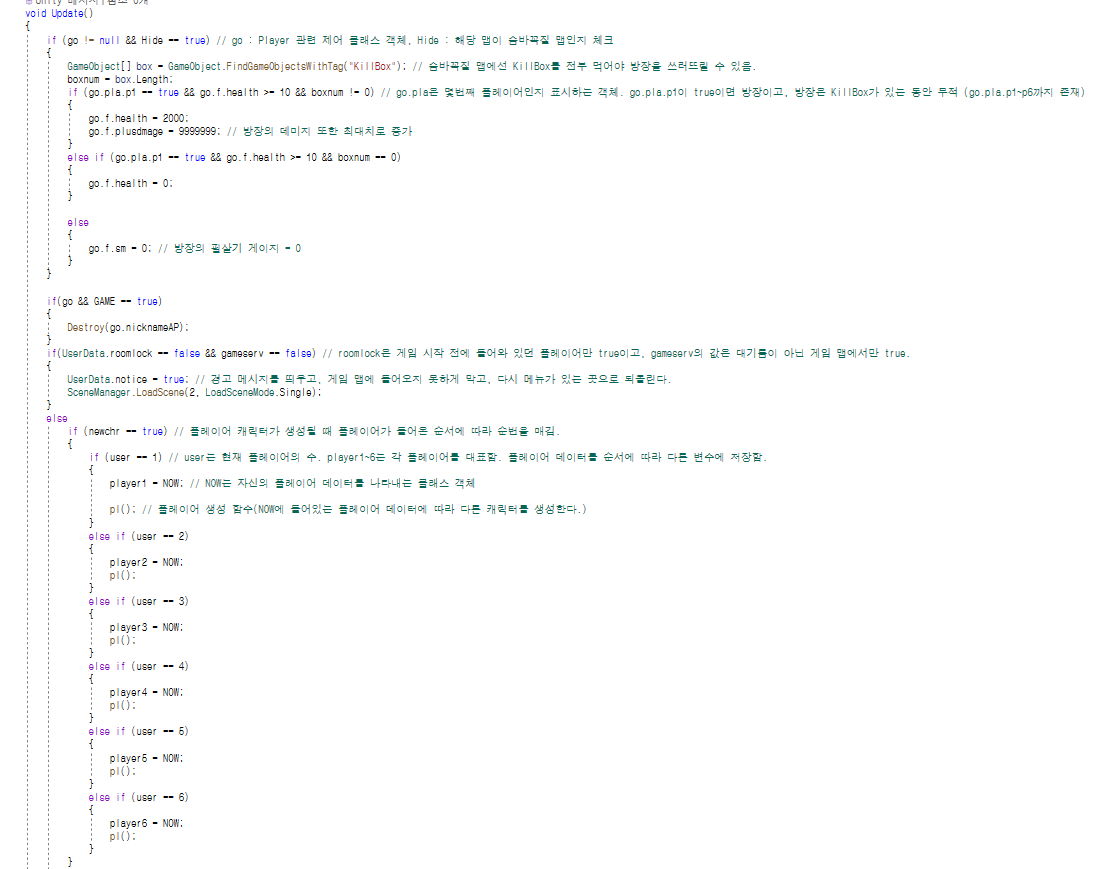
**[코드설명]**



* 해당 클래스는 자동으로 시작하자마자 서버(로비)에 접속해 룸에 들어가도록 작성됨.
* 또한, 캐릭터 생성 여부를 FALSE로 체크하고, 캐릭터 생성을 진행하도록 함.
* mo 변수는 자신의 캐릭터 생성 후 true로 바뀜.
* OnJoinedLobby(로비 접속 시), OnJoinedRoom(룸 접속 시)은 Photon.MonoBehaviour를 상속한 PhotonManager 클래스에서 별도 호출 없이 자동 호출된다.
  + 캐릭터는 **GAME 변수**를 통해 대기 룸인지 게임이 진행되는 곳인지 체크해

캐릭터의 전투 가능 여부를 결정한다.

* **lobbyCamera 변수**는 게임이 진행되는 레벨에서 게임 맵 전체를 보여주는 카메라.



* Update에서 플레이어 생성과 관련된 작업이 실행된다.
* 플레이어가 생성되지 않을 경우 고려해 반복적으로 결과 체크해 실행하도록 하기 위함.
* 게임 내 모드가 **숨바꼭질 모드와 일반 전투 모드**로 나뉘는데 이를 고려해 숨바꼭질 모드일 경우에만 방장을 무적으로 설정
* 현재 유저의 수에 따라 현재 플레이어의 순번을 다르게 설정함.
* 여기서 정한 순번은 다른 플레이어가 나가지 않는 이상 바뀌지 않음.
* NOW는 플레이어의 정보를 저장한 객체로 해당 객체에 있는 데이터로 서버에 캐릭터를 생성한다.
* pl 함수에 해당 내용이 들어간다. (PhotonNetwork.Instantiate 함수 이용)

1. **PhotonCheck 클래스 : 플레이어를 실시간으로 체크하고, 이를 게임에 반영해 문제가 발생하지 않도록 한다.**

**[코드 설명]**



* Update에서 Userspace를 이용해 유저 수를 지속적으로 갱신하도록 함.
* 유저 수를 갱신한 후에 실시간으로 화면 내 유저의 닉네임 표시 부분도 갱신함.
* pm은 PhotonManager 객체를 의미하며 newchr의 값을 true로 바꾸는 이유는

새로운 유저 데이터를 등록하기 위해서이다. 유저 데이터를 등록하는 이유는 캐릭터의 스텟 및 착용 아이템 등을 서버에서 동기화하기 위함.