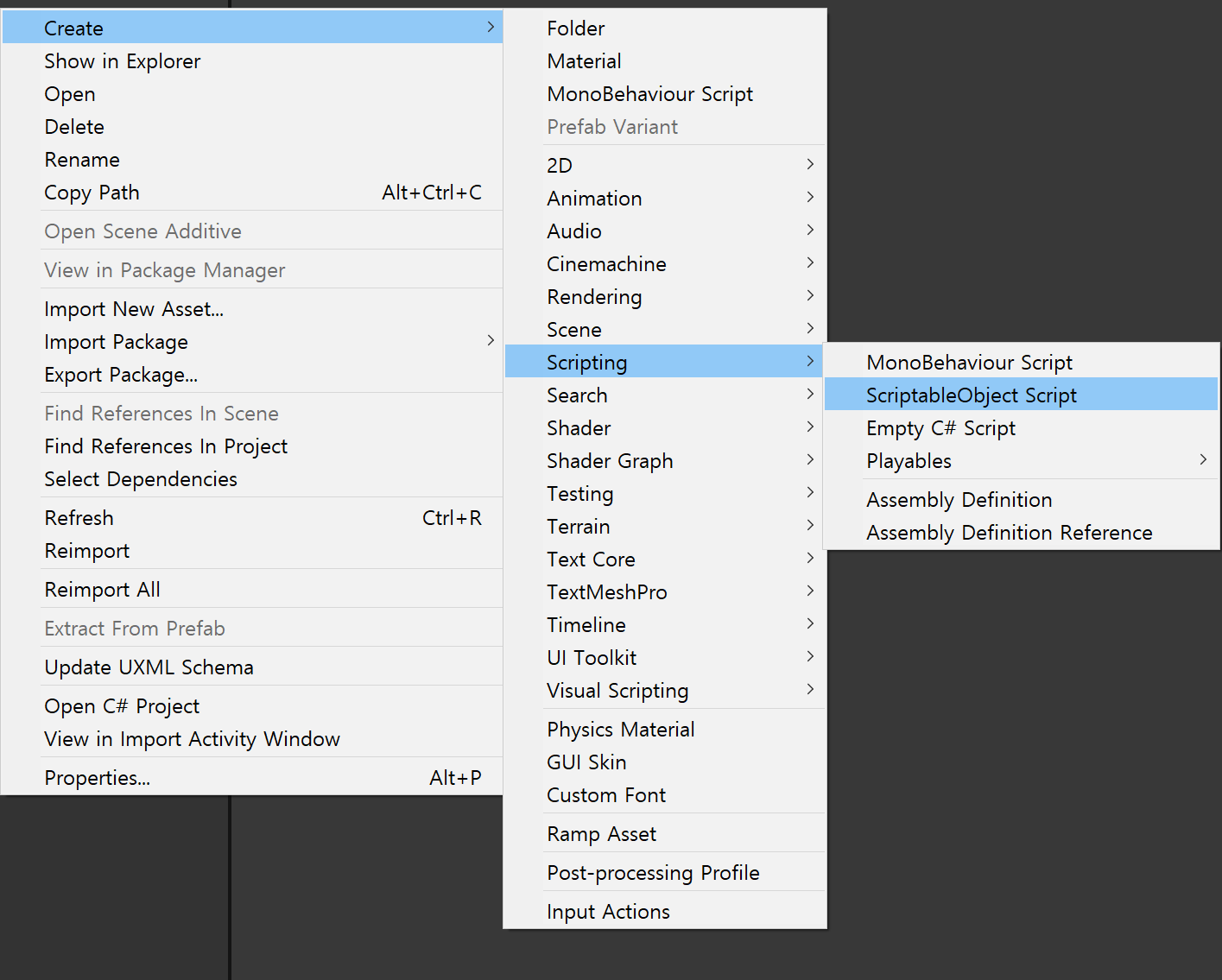
Scriptable Objcet

데이터들을 한 군데 모아서 메모리를 절약할 수 있는 디자인

<https://unity.com/kr/blog/engine-platform/6-ways-scriptableobjects-can-benefit-your-team-and-your-code>



using UnityEngine;

[CreateAssetMenu(fileName = "BarrelDataSO", menuName = "Scriptable Objects/BarrelDataSO")]

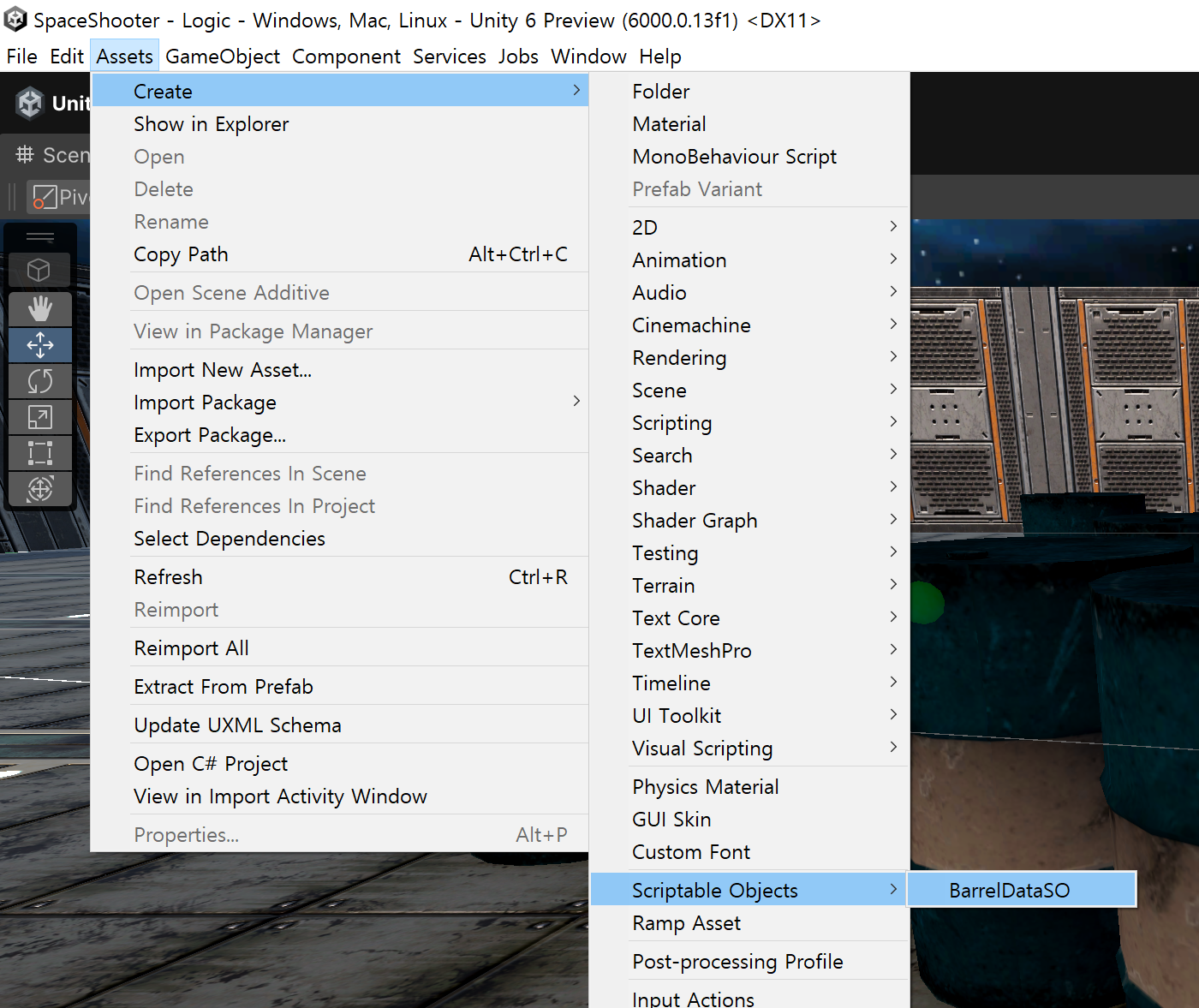
public class BarrelDataSO : ScriptableObject

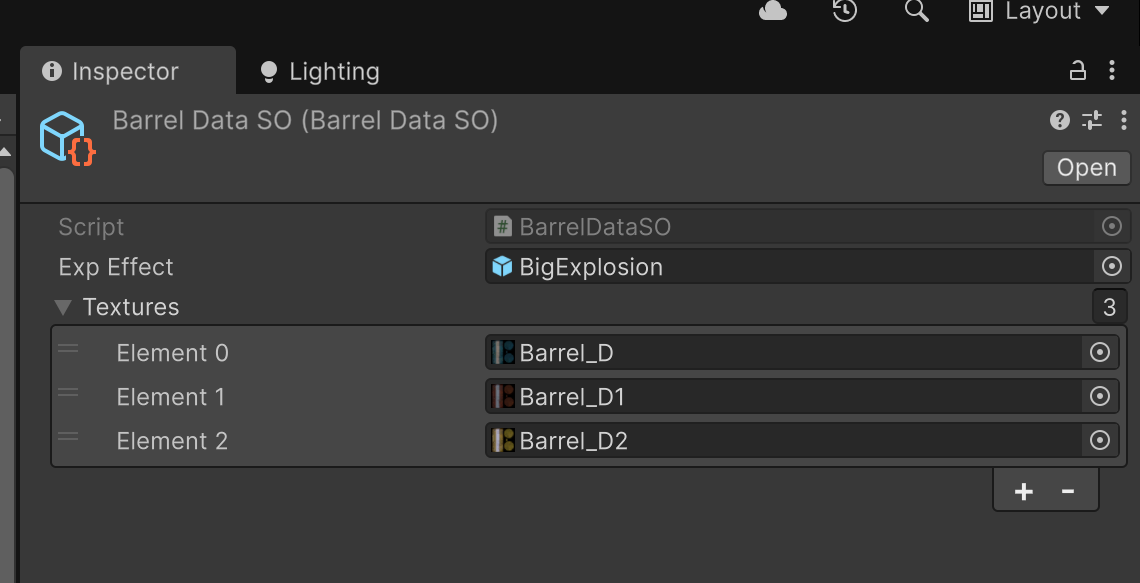
{

    public GameObject expEfeect;

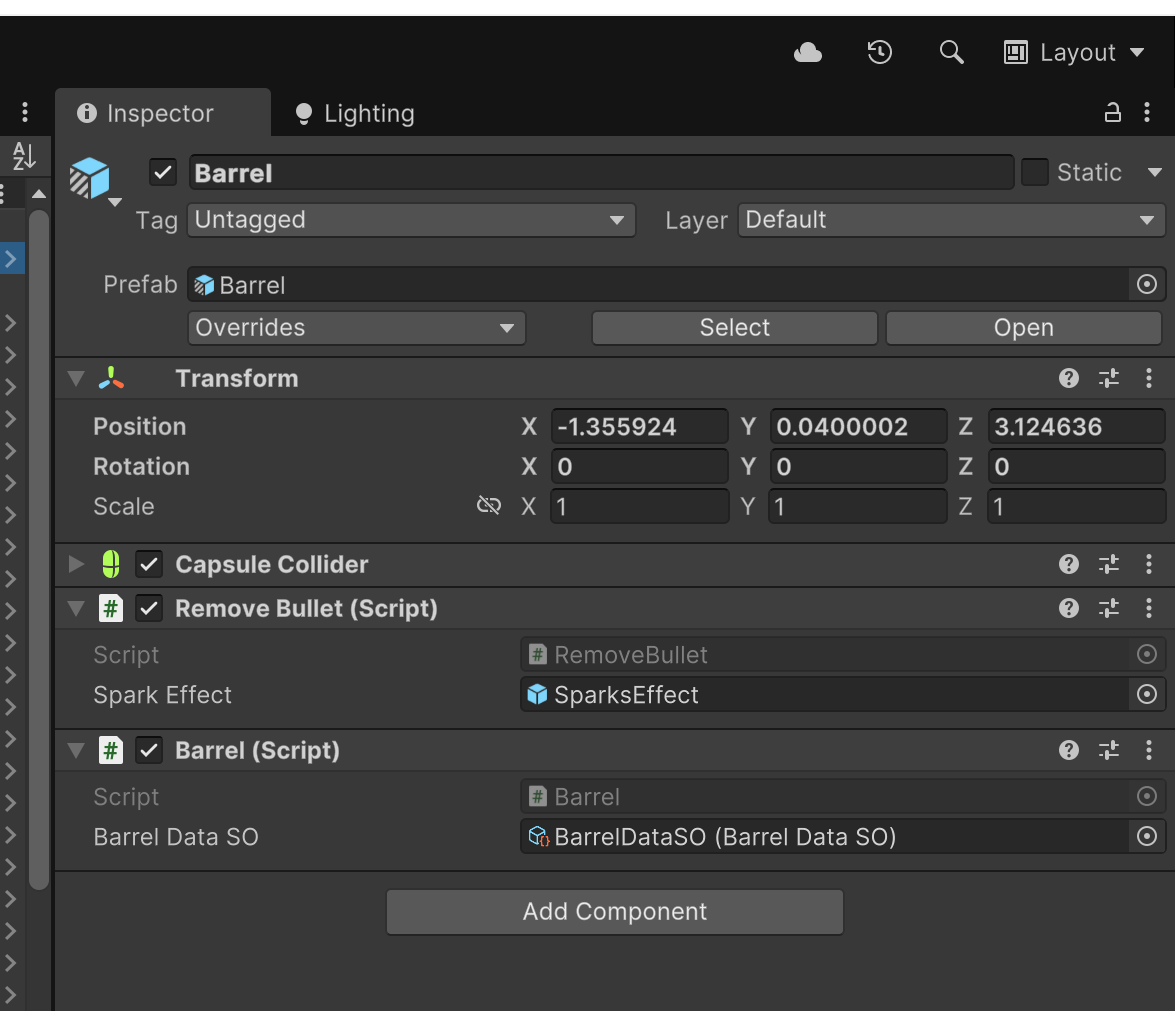
    public Texture[] textures; // 기존에 사용했던 것으로 쓰면 수정할 때 편하다

}

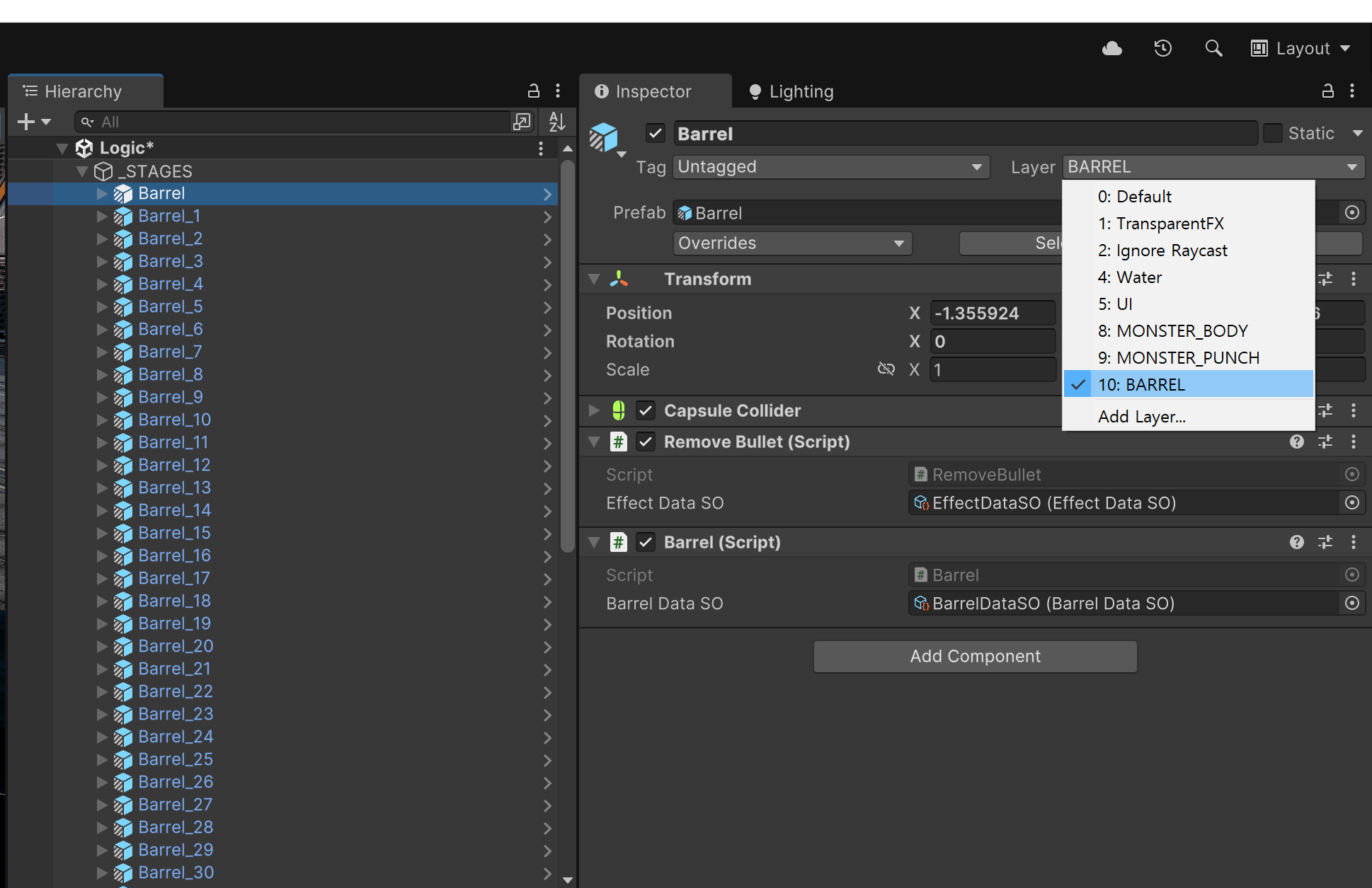


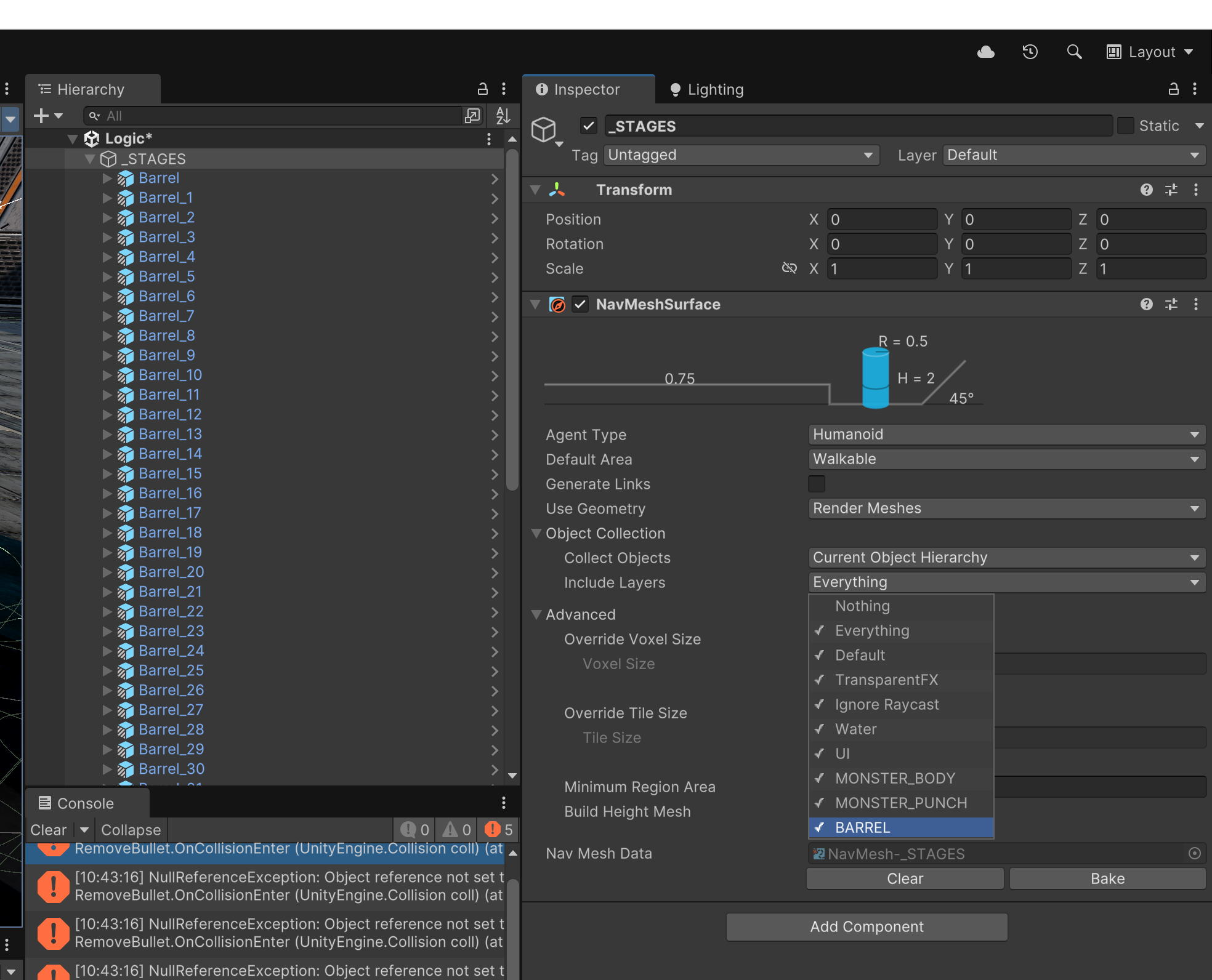


새로 생긴 barrel data so 에 텍스쳐 더해주기



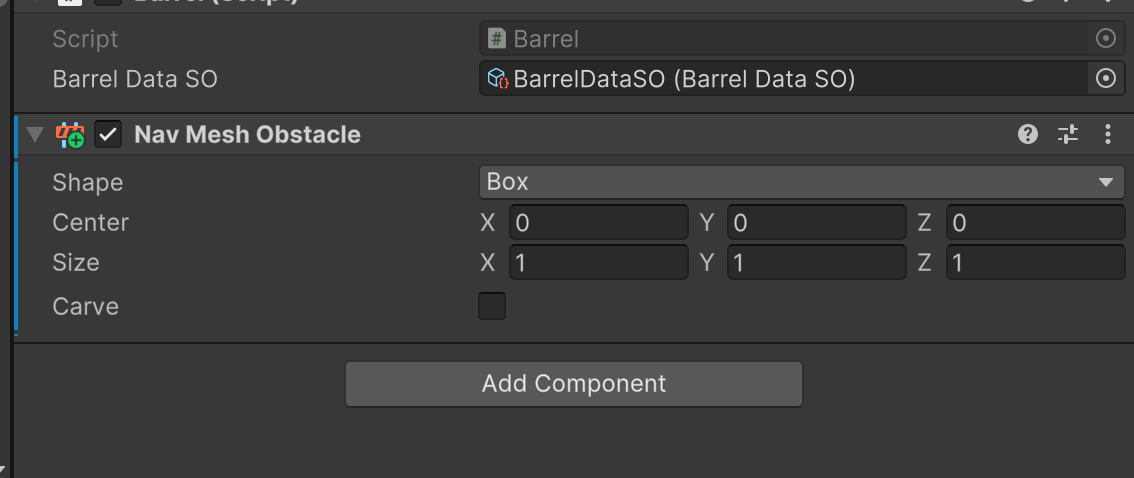
배럴 스크립트에 barrel data so 연결해주기 \*\*\* overrides 에서 꼭 apply 해줘야한다 \*\*\*



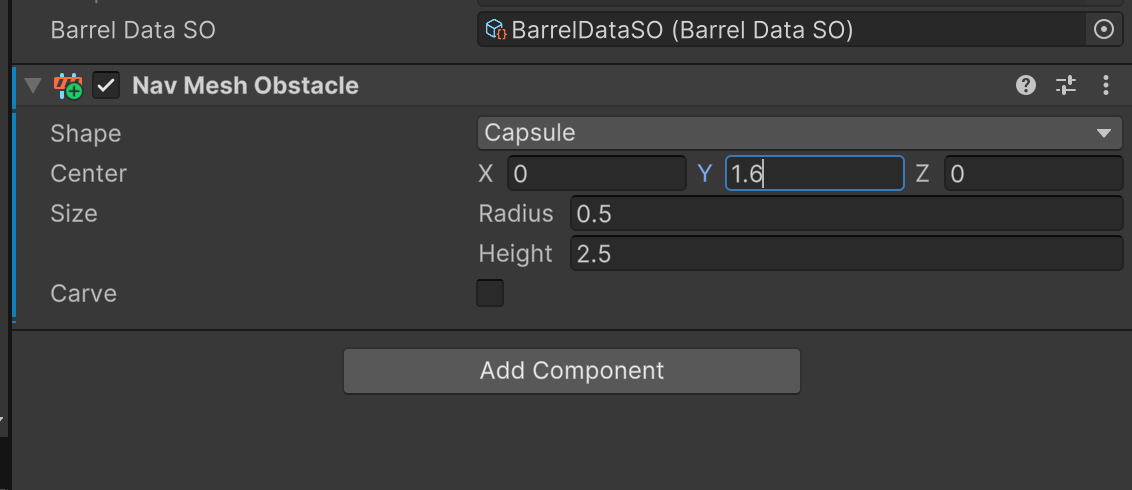


배럴이 터지고 나면 그 위치를 장애물로 인식하지 않기 위해서 레이어를 추가해서 컨트롤함.

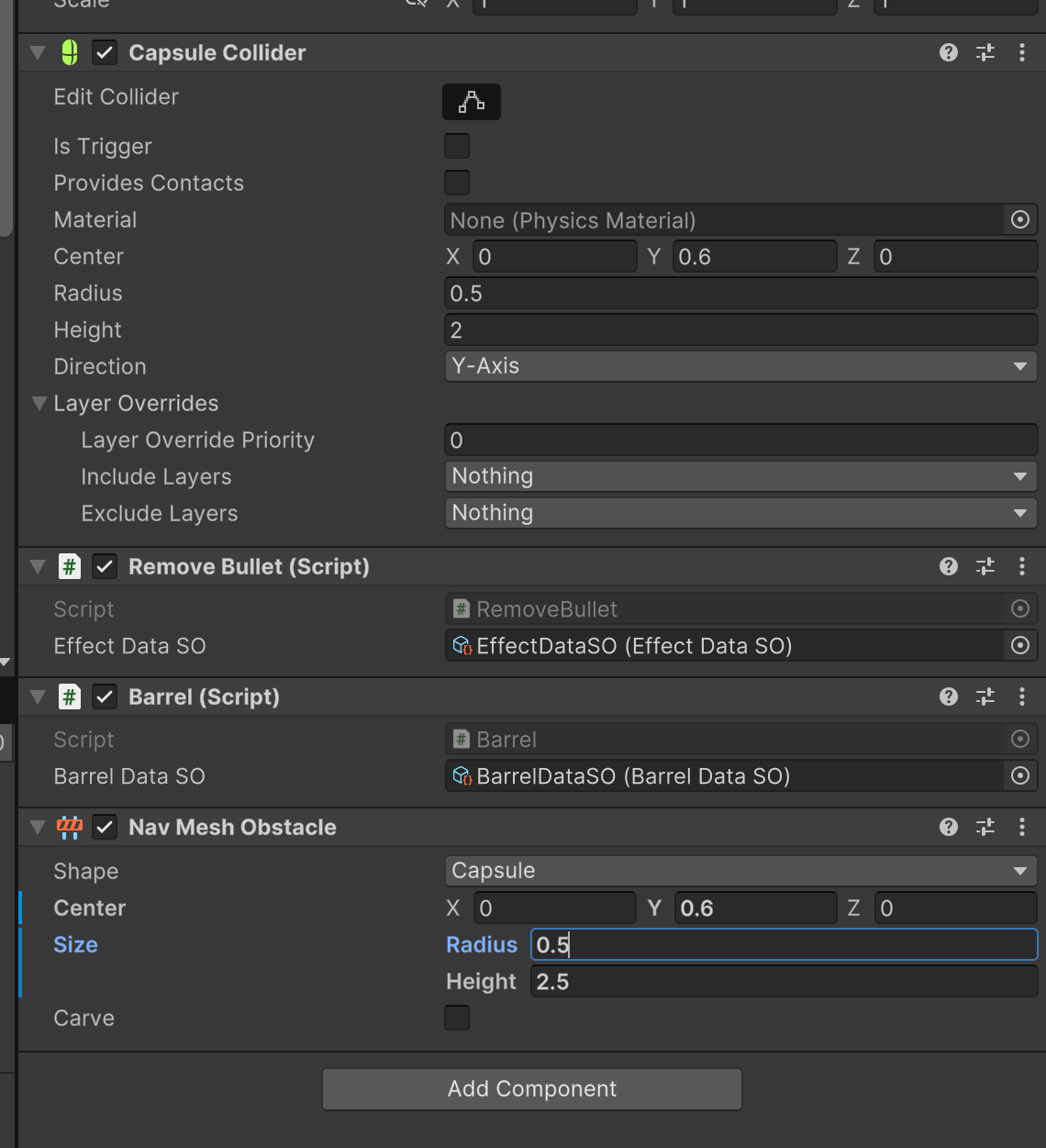
이렇게까지 하면 모든 네브에서 다 인식 안함.



Nav mesh obstacle 추가



캡슐로 바꾸고 크기 조절해줌



위에 캡슐콜라이더랑 사이즈 맞춰주면 됨

인스턴시에이트, 디스트로이가 유니티 개발 할 때 제일 부하가 많이 걸리는 함수.

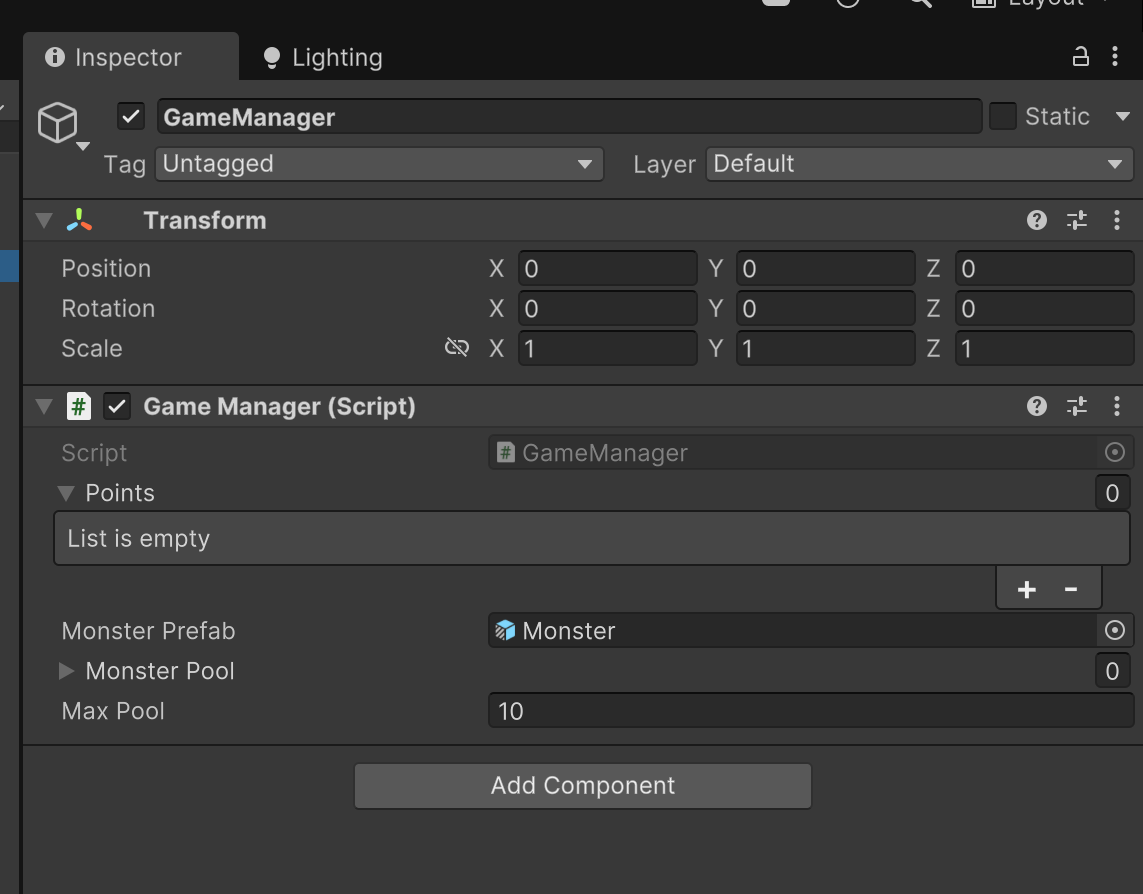
그래서 object pooling 을 사용해서 부하를 줄일 수 있다.

예) 지금까지는 총알을 총 앞에 동적으로 대기 시켜놓은 상태에서 인스턴시에이트, 후에 벽에 부딪히고나면 디스트로이 되게 만들어 놨지만 이것들을 Pool 에 대기시켜놓고 사용(active true)한 후에 active false 시켜서 사라지게끔 만드는 것.

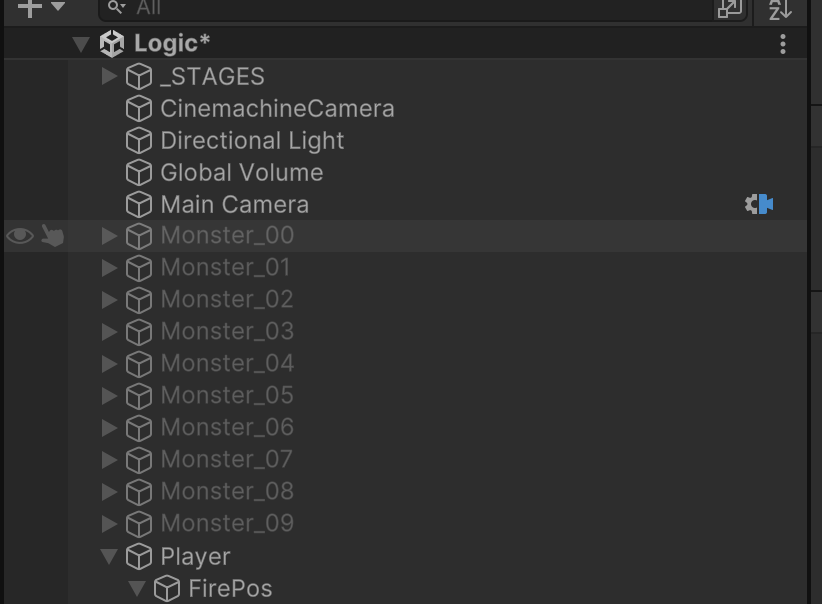
1. object pooing 직접 구현, 2. unity에서 IObject pool 을 사용하는 방법.



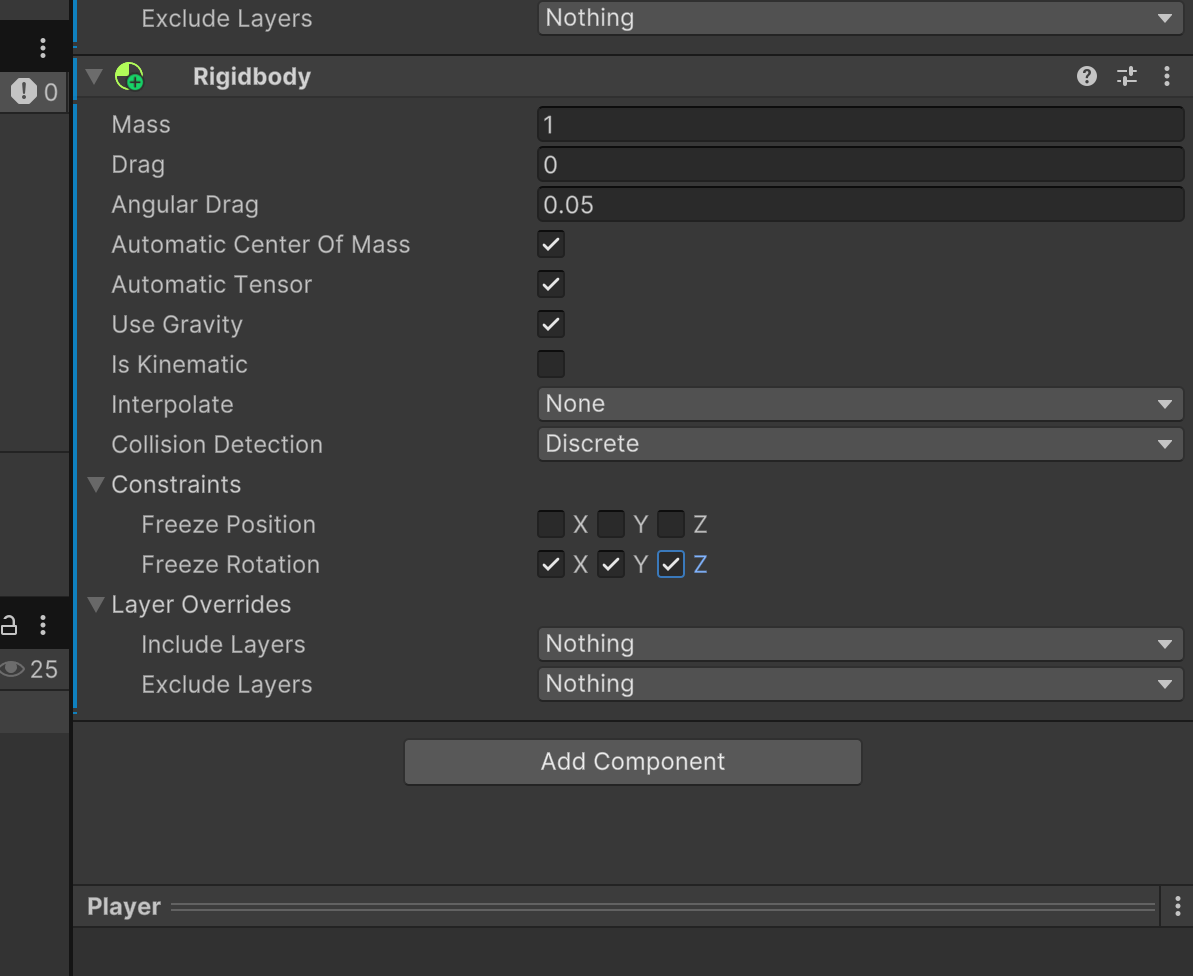
코딩 후에 보면



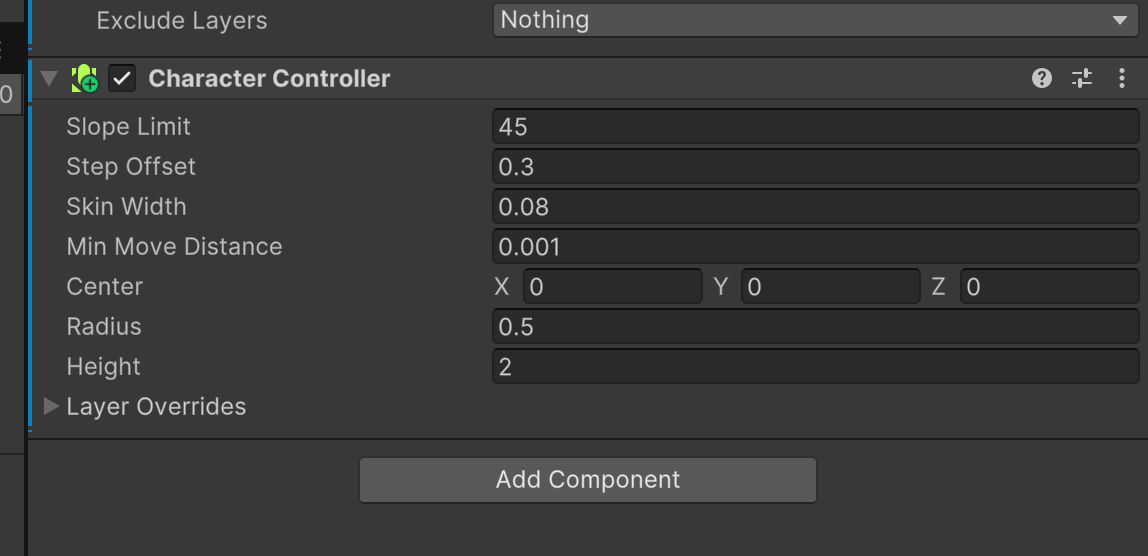
Max pool 이 생성 되어있고 10 쓰여있다



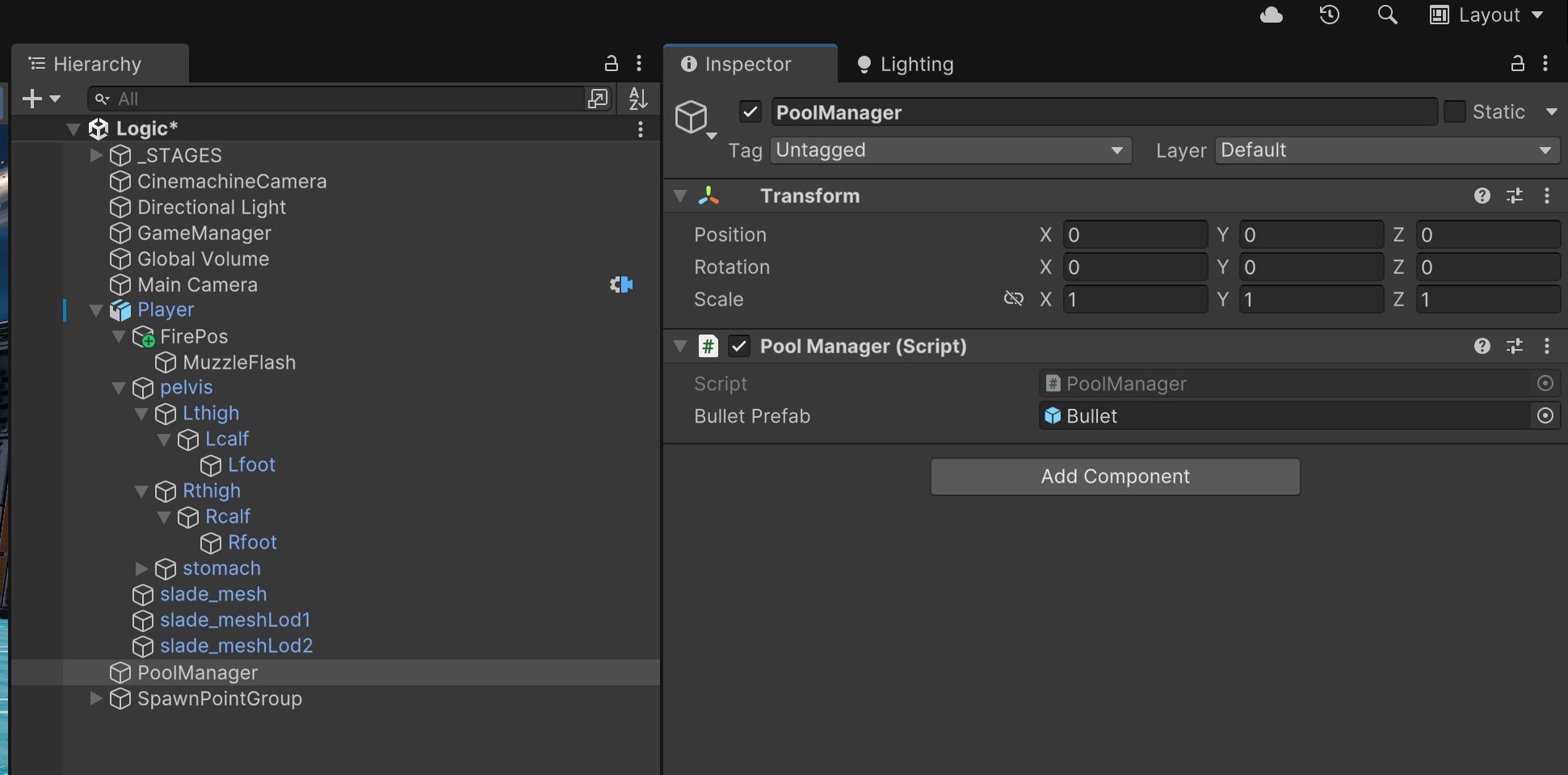
시작 해보면 몬스터들이 비활성화 된 상태로 생성



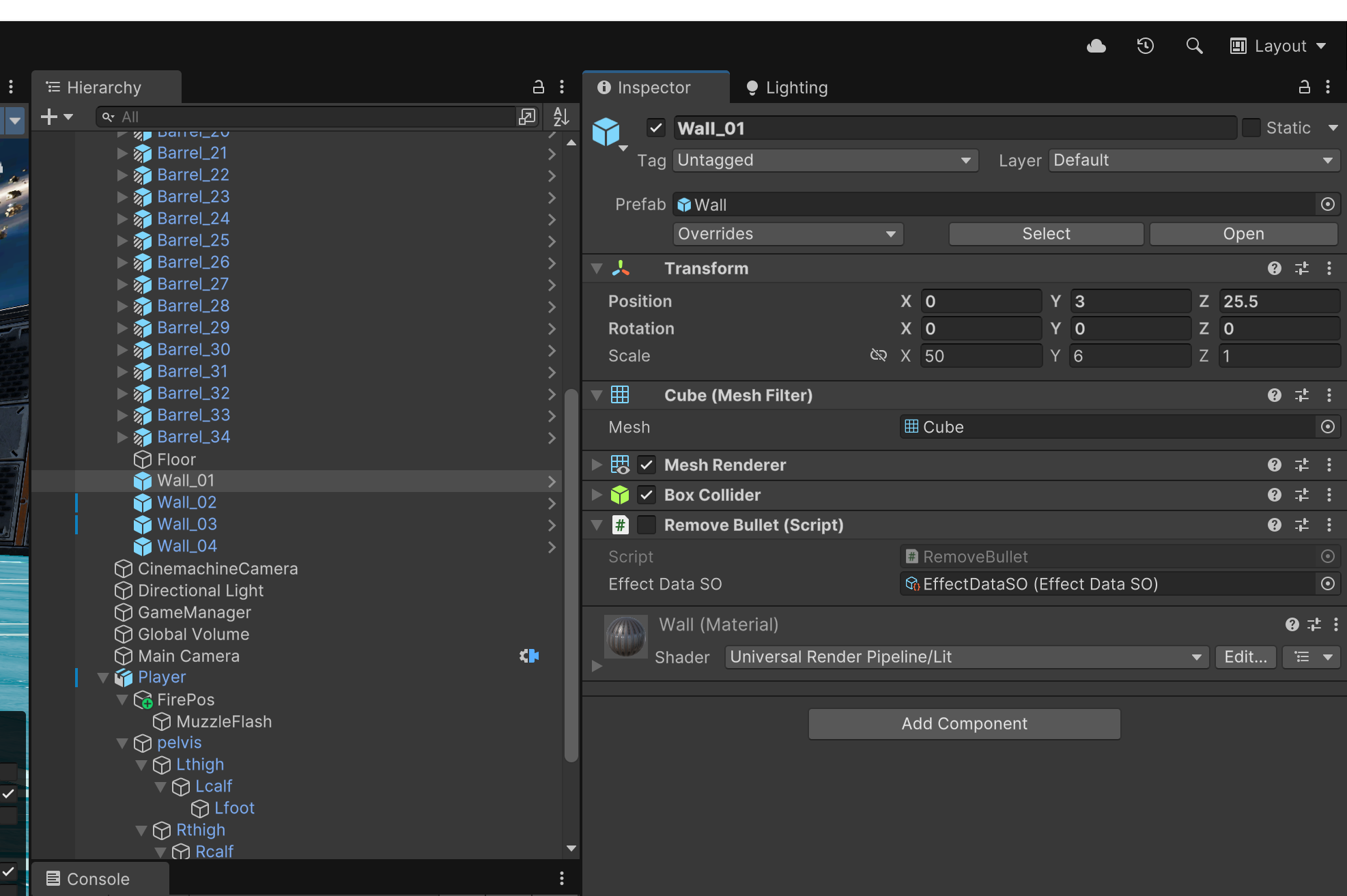
플레이어한테도 rigidbody 추가해주는데, 그냥 두면 부딪혔을 때 막 굴러다님 그래서 xyz 좌표를 freeze 해준다. 그런데 이 상태면 , 매 프레임마다 계산하기 때문에 벽에 부딪히거나 했을 때 떨림 현상이 발생한다.



그래서 rigidbody지우고, 캐릭터컨트롤러를 추가해준다.

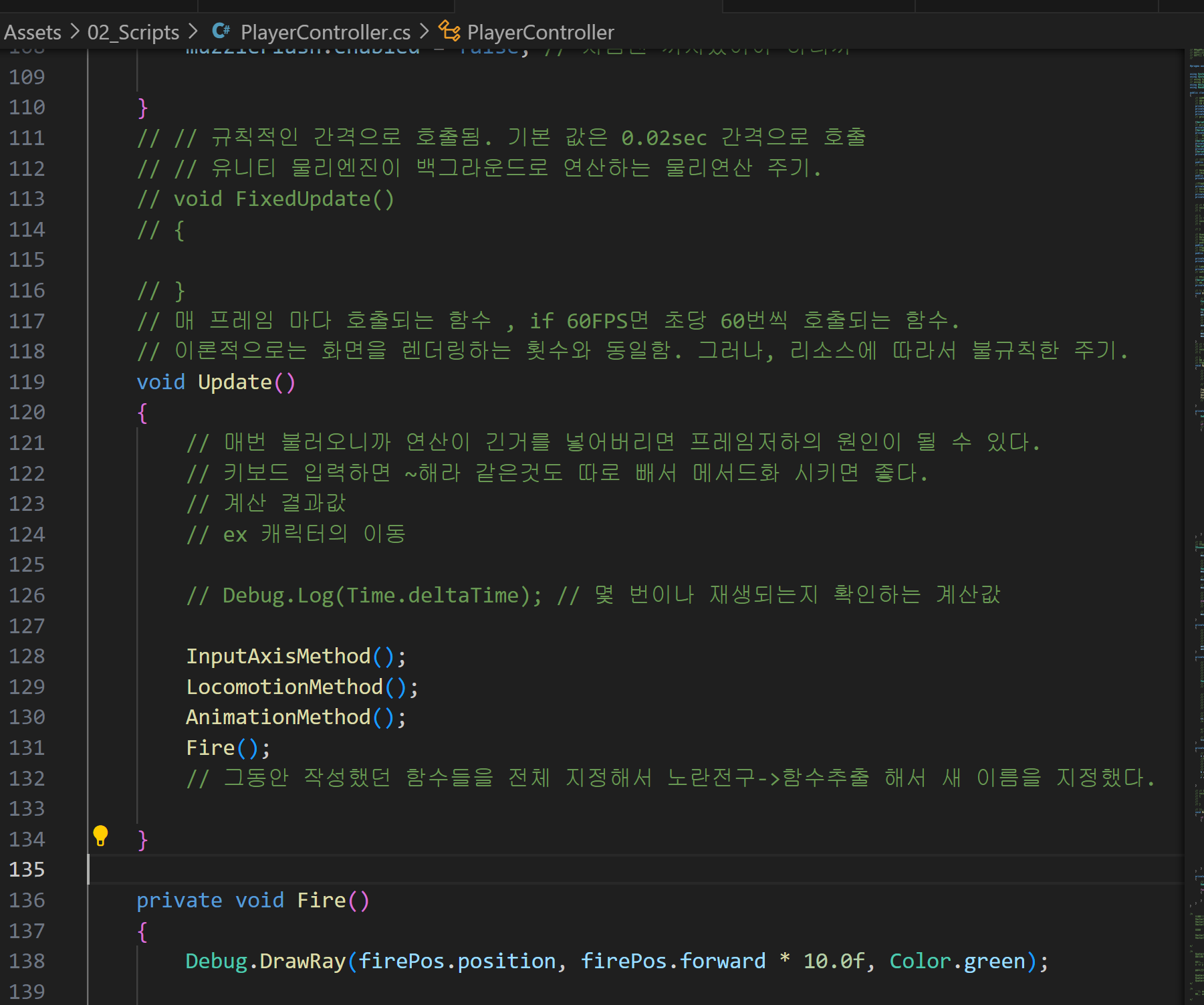


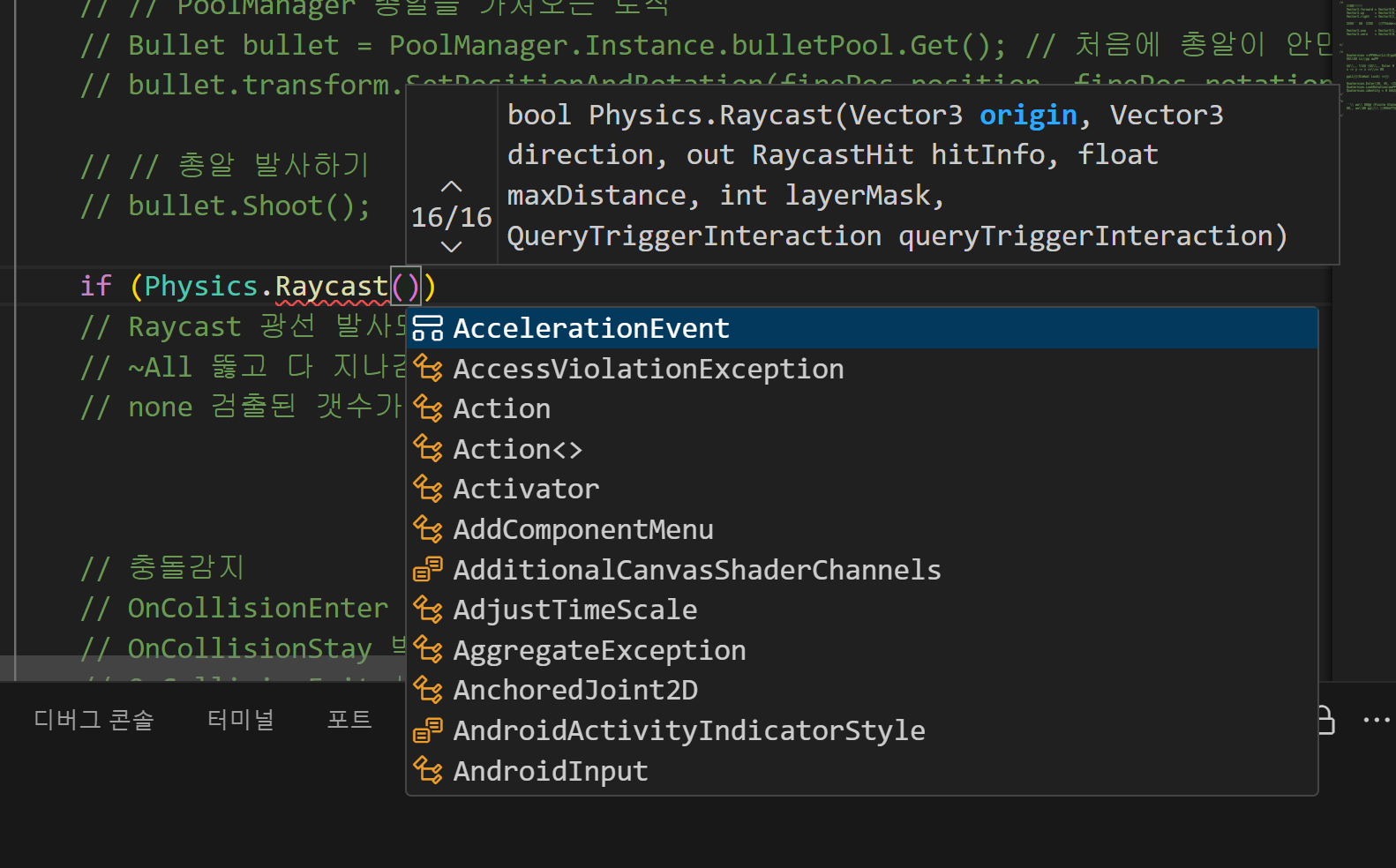
PoolManager 코드 작성후에 프리펩 추가



Oncollision Enter 를 총알에 추가하기 전에 배럴과 벽에 있는 리무브불렛 스크립트 비활성화

저격모드!!!! 만들기





발사 원점과 총구가 발사되는 방향에서 10m 간 raycast 를 발사, 뭔가에 닿으면 hit 라는 true 값을 리턴해줌.