

그래스핑 심화, 총잡기 및 총알 발사!

1. 이준

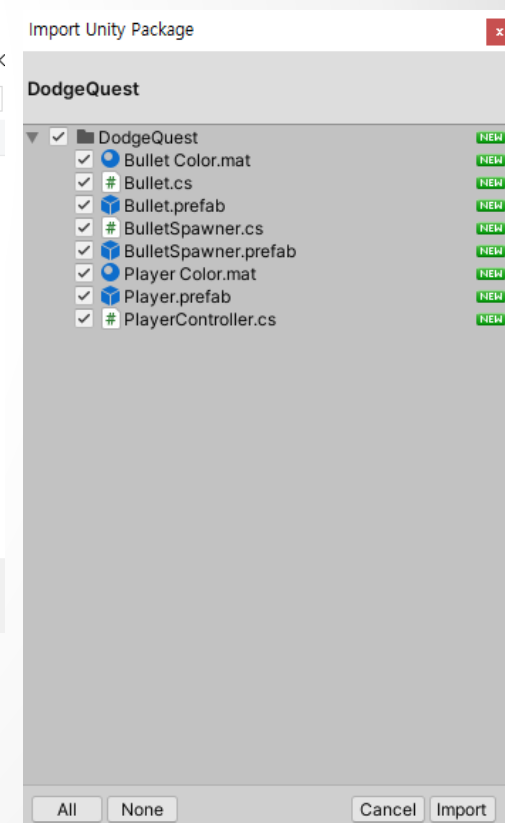
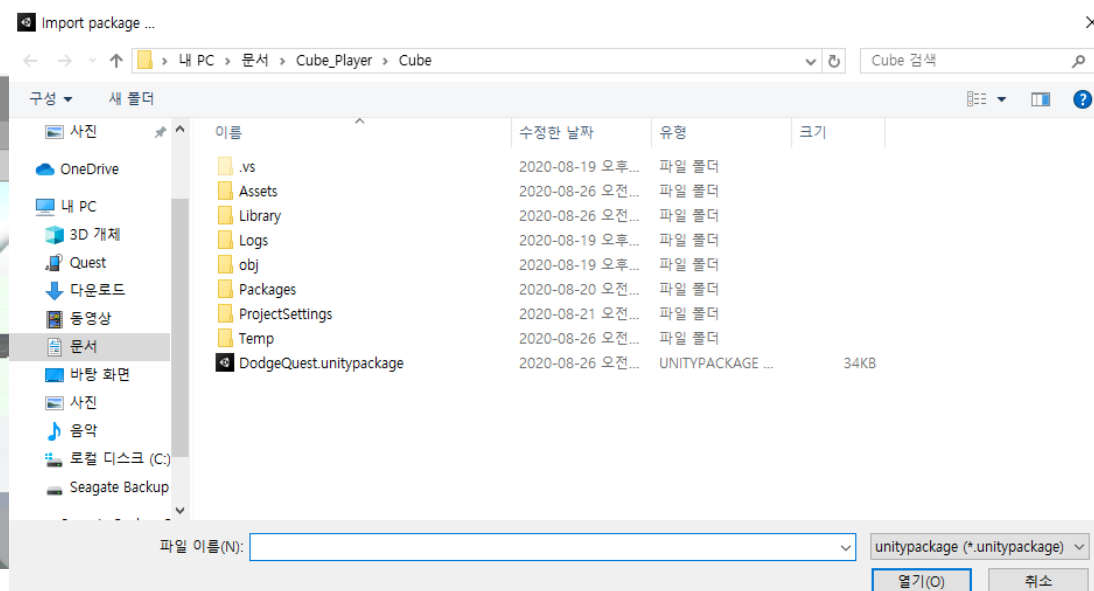
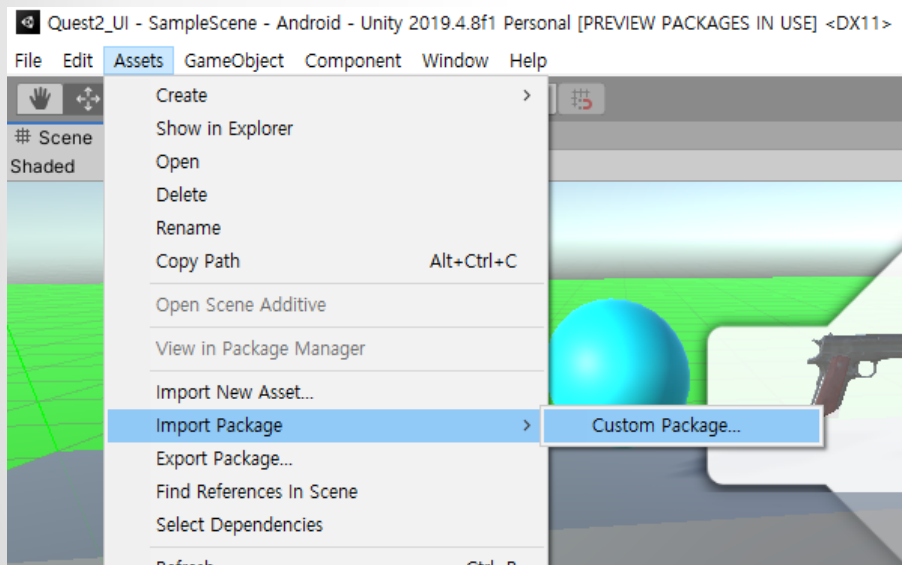
전투 시스템 만들기!

- 닳지 플레이어에서 사용한 총알 피하기 및 전투를 만든다면?
- 적이 총알을 발사하는 경우!
 - 고개를 돌려서 피해야함!
- 내가 적을 총알을 쏘서 공격하는 경우!
 - 적도 나의 총을 맞으면 죽어야 함! (피하는거는 일단 없음!)

전투 시스템 만들기!

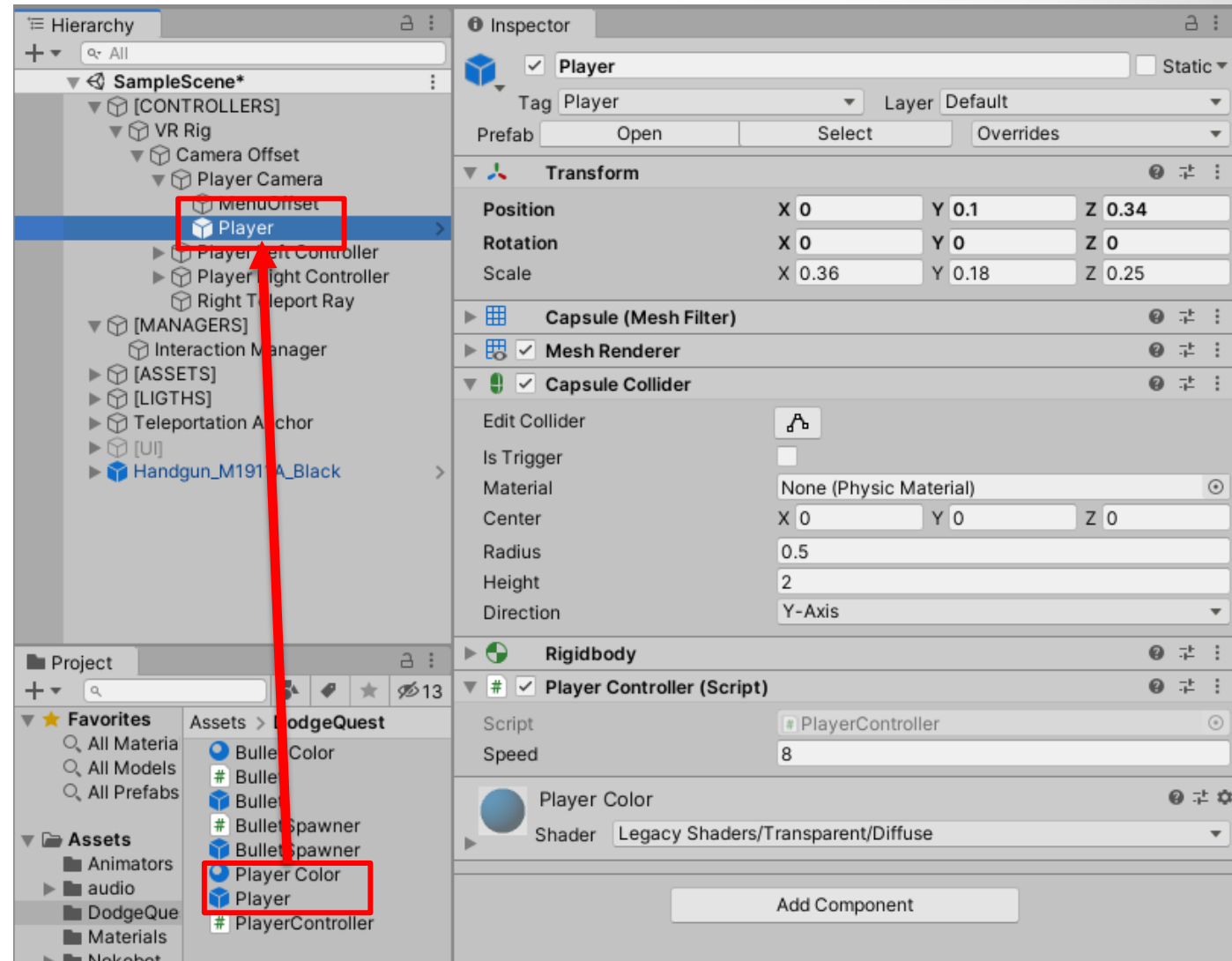
- 닷지 퀘스트의 유니티 패키지 다운로드!

- <https://drive.google.com/file/d/1f4I2EsH5OYBLUBKM2nlsX-6R8dJyLM1c/view?usp=sharing>
- 지난 시간 만들었던 핵심 로직을 패키지로 만들어 둬!
- 재활용을 하여 속도를 높혀 봅시다!



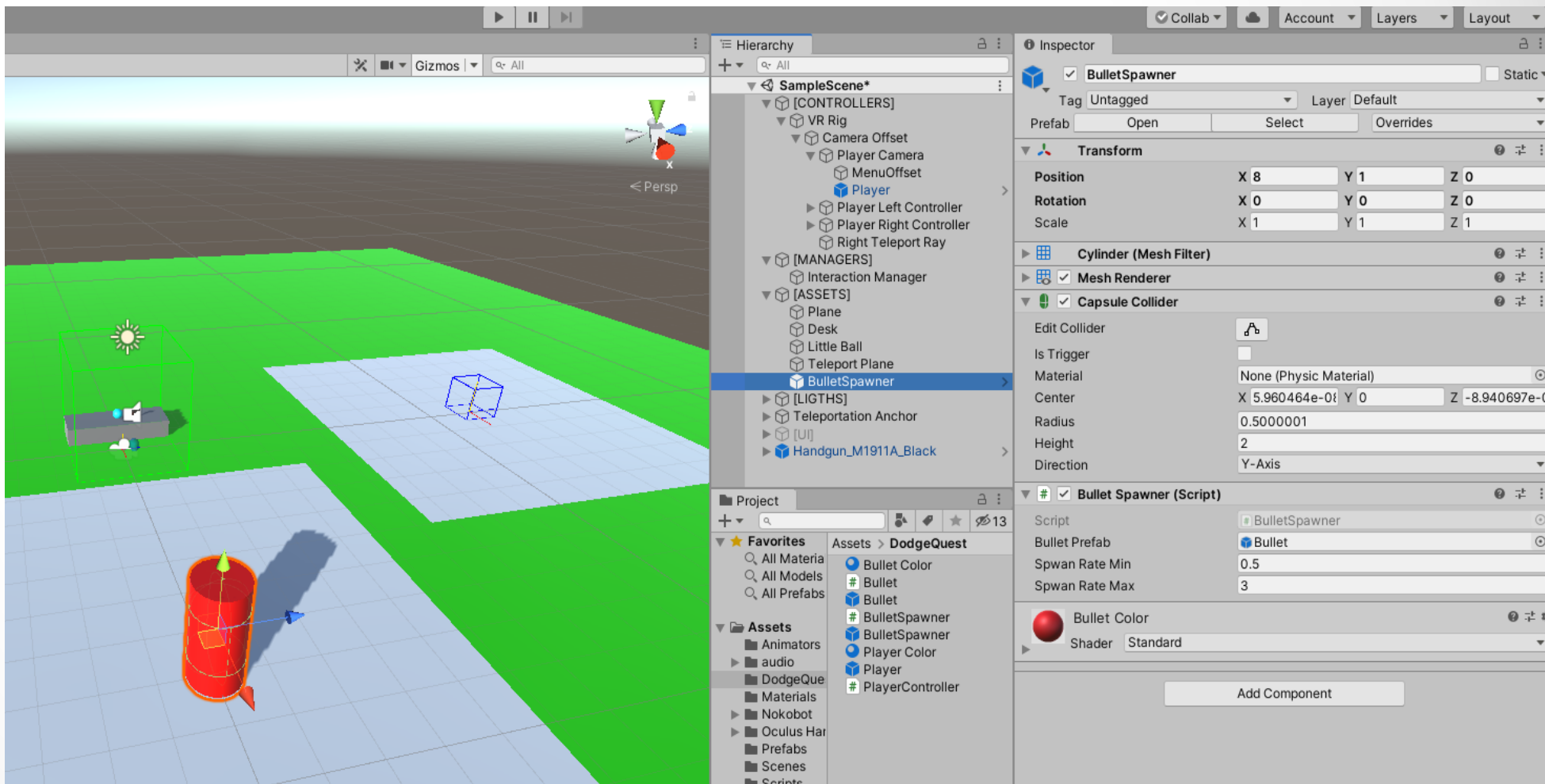
전투 시스템 만들기!

- 먼저 Player Prefab을 Player Camera 밑에 추가!



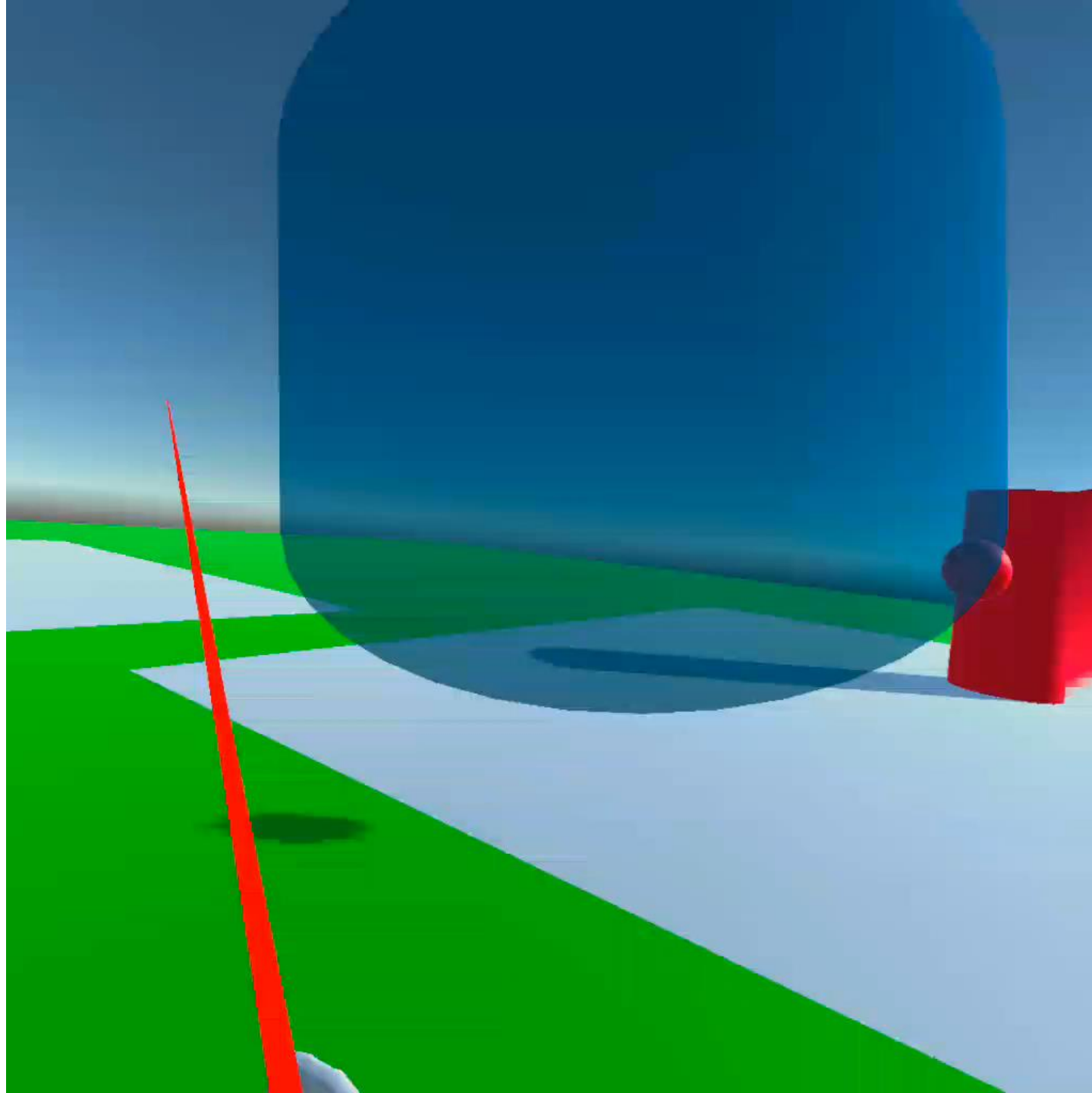
전투 시스템 만들기!

- BulletSpawner도 배치해보자!



전투 시

- 패키지를 사용하는 경우 쉽게 재활용 가능!
- 아직 게임로직을 구성이 필요

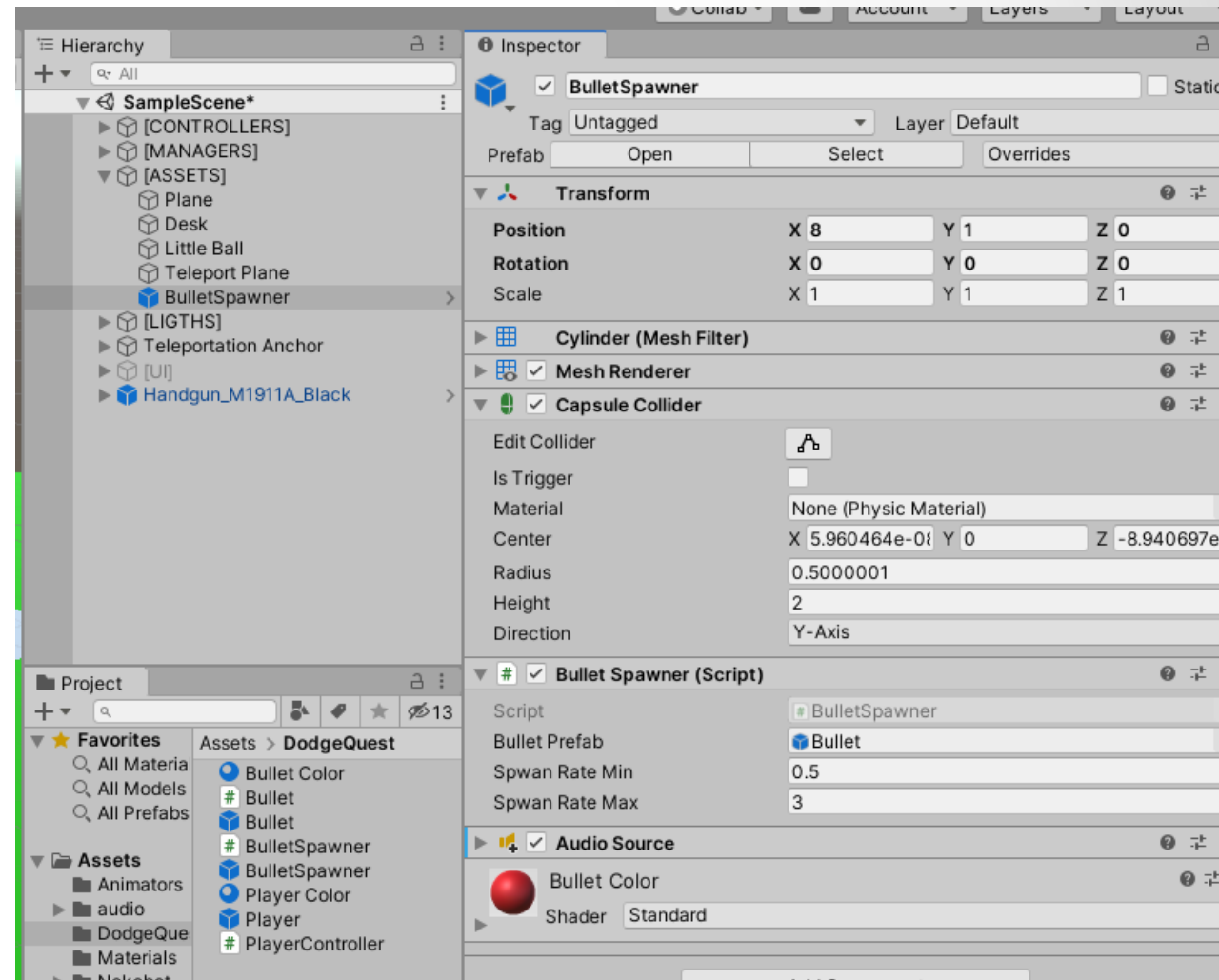


전투 시스템 만들기!

- BulletSpawner의 체력 설정 `HP = 100;`
- 내 총알의 공격력 설정 `BulletAttack = 35;`
- 즉 세방을 맞아야 BulletSpawner를 죽일수 있음!

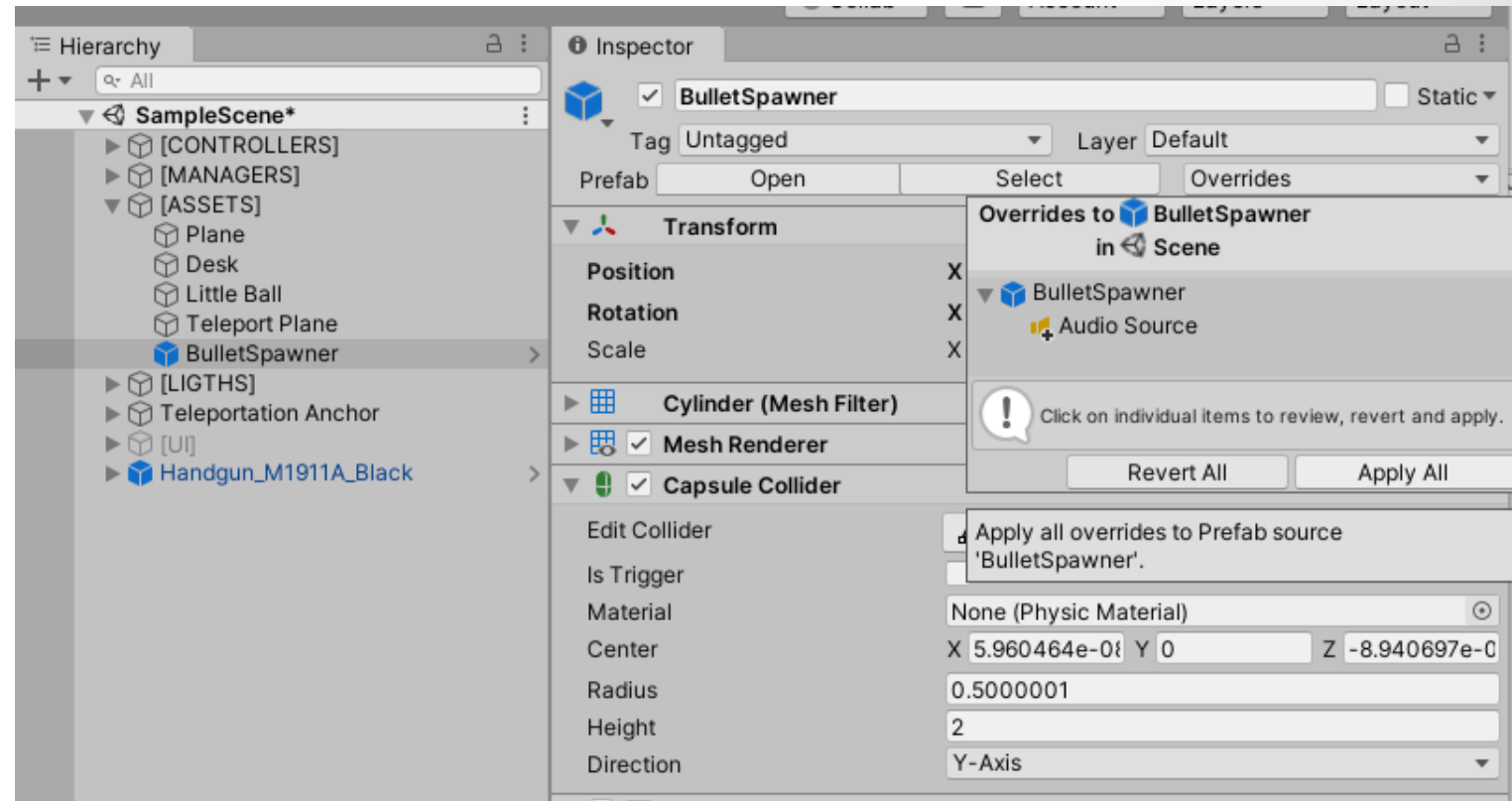
전투 시스템 만들기!

- BulletSpawner 에 총알 발사 사운드 넣기
- AudioSource를 추가하기!



전투 시스템 만들기!

- 프리팸에 적용 하려면 항상 Overrides -> Apply All을 실행 해야함!



전투 시스템 만들기!

- BulletSpawner.cs 파일 수정
- HP=100.0f 추가
- 사운드 재생을 위한 fireClip 및 fireAudio 추가

```
Assembly-CSharp
1  using System.Collections;
2      using System.Collections.Generic;
3      using UnityEngine;
4
5  public class BulletSpawner : MonoBehaviour
6  {
7      public GameObject bulletPrefab;
8      public float spwanRateMin = 0.5f;
9      public float spwanRateMax = 3f;
10
11     private Transform target;
12     private float spwanRate;
13     private float timeAffterSpawn;
14
15     public float hp = 100.0f;
16     public AudioClip fireClip;
17     AudioSource fireAudio;
18
19
20     // Start is called before the first frame update
21     void Start()
22     {
23         timeAffterSpawn = 0f;
24         spwanRate = Random.Range(spwanRateMin, spwanRateMax);
25         target = FindObjectOfType<PlayerController>().transform;
26
27         fireAudio = GetComponent<AudioSource>();
28
29     }
```

전투 시스템 만들기!

- BulletSpawner.cs 파일 수정
- Update 함수에서, 총알발사 사운드 플레이 추가!

```
32 void Update()  
33 {  
34     timeAffterSpawn += Time.deltaTime;  
35  
36     if(timeAffterSpawn >= spwanRate)  
37     {  
38         timeAffterSpawn = 0f;  
39  
40         GameObject bullet = Instantiate(bulletPrefab, transform.position, transform.rotation);  
41         bullet.transform.LookAt(target);  
42         fireAudio.PlayOneShot(fireClip);  
43  
44  
45         spwanRate = Random.Range(spwanRateMin, spwanRateMax);  
46     }  
47  
48 }
```

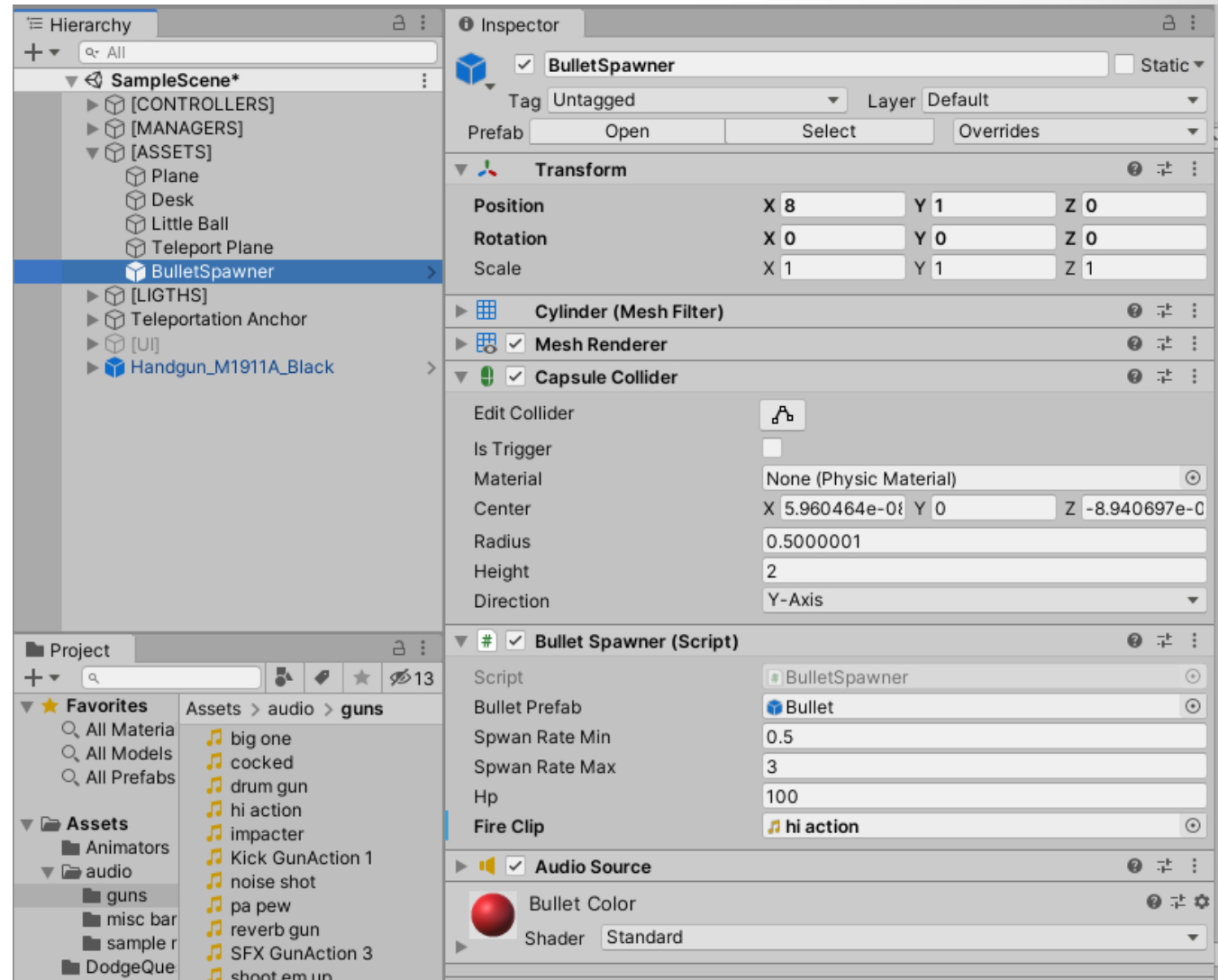
전투 시스템 만들기!

- BulletSpawner.cs 파일 수정
- GetDamage 파일, 총알의 공격력 만큼 체력을 깎다가 0이 되면 객체 자체를 비활성화

```
53 public void GetDamage(float amount)
54 {
55     hp -= amount;
56
57     if(hp < 0)
58     {
59         //Bullet Spawner가 죽음 SetActive를 false로 설정
60         gameObject.SetActive(false);
61
62     }
63 }
64 }
```

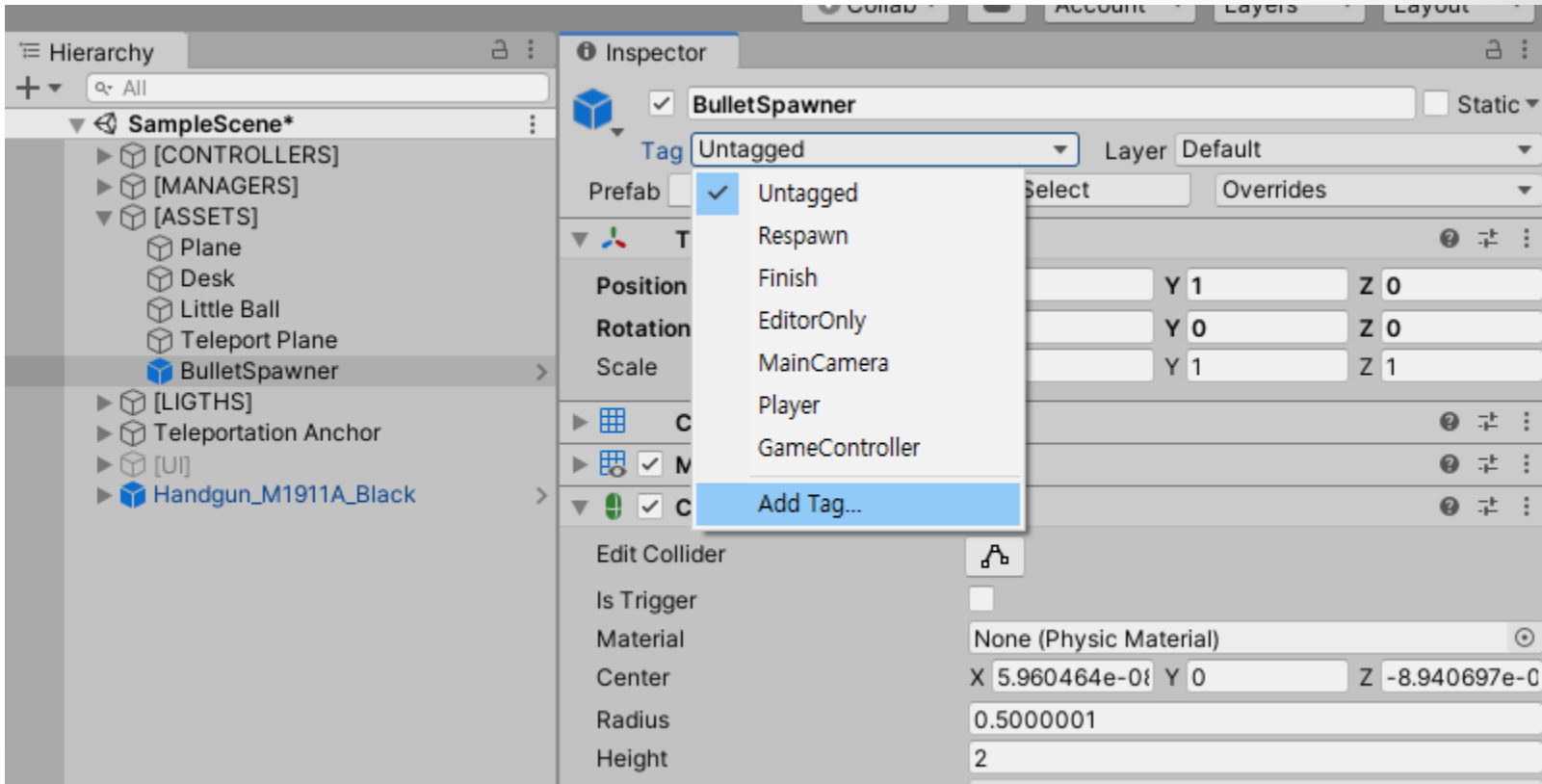
전투 시스템 만들기!

- BulletSpawner 의 Fire Clip을 hi action으로 설정 (약간 멀리서 총을 발사한다는 느낌으로..)



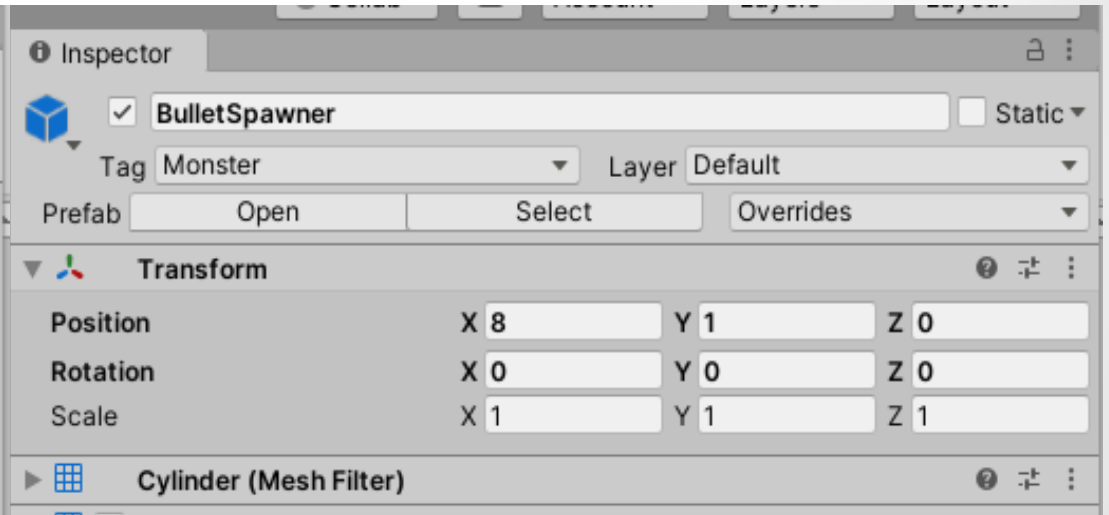
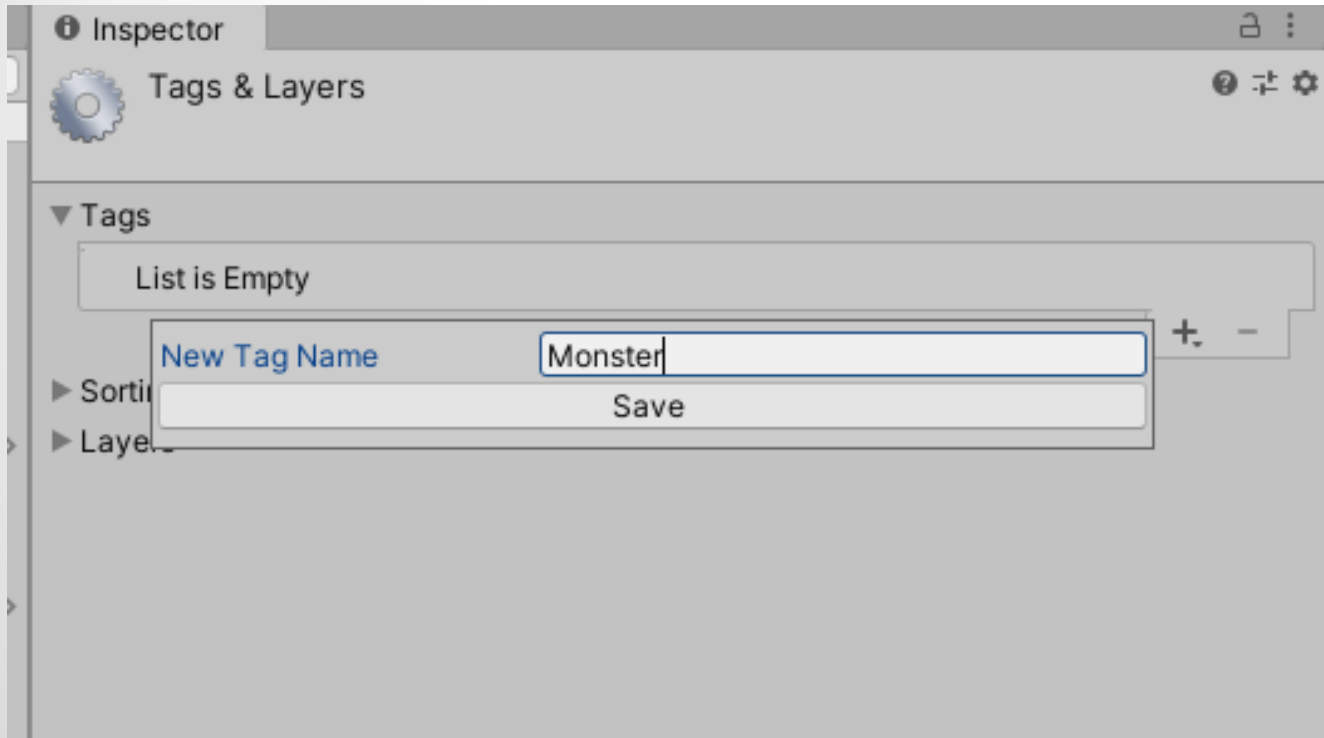
전투 시스템 만들기!

- 내가 쏜 총알이 적에게 맞았을때 처리하려면?
- 태그를 만들어서 관리를 하면 편리함!
- BulletSpawner 를 선택하고, Tag -> AddTag를 클릭!



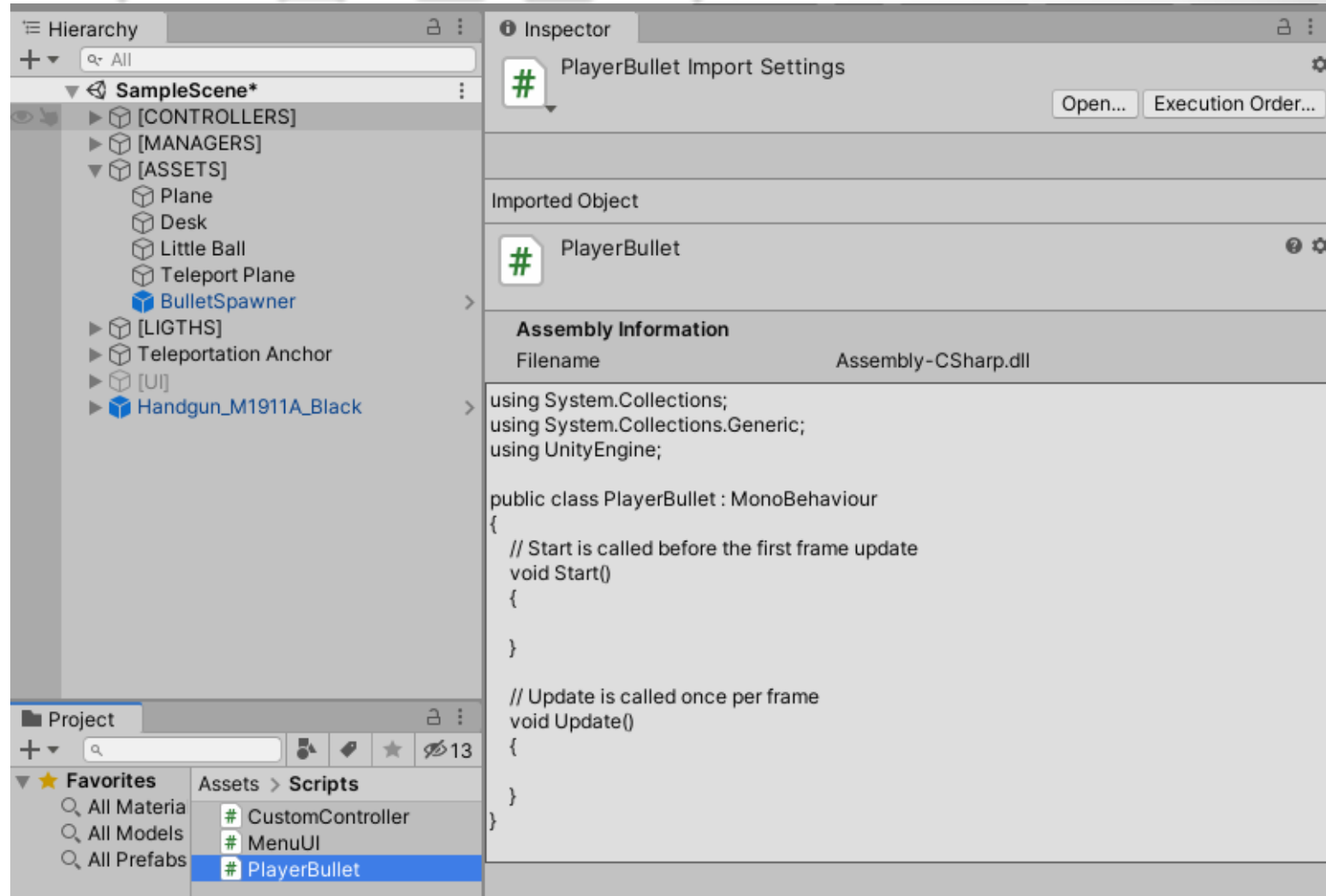
전투 시스템 만들기!

- +버튼을 눌러서 태그를 추가하고 Monster 라는 이름으로 생성
- 이후 다시 BulletSpawner 의 태그를 Monster 라고 설정



전투 시스템 만들기!

- 플레이어 총알의 제어를 위해 PlayerBullet.cs 파일 생성!



전투 시

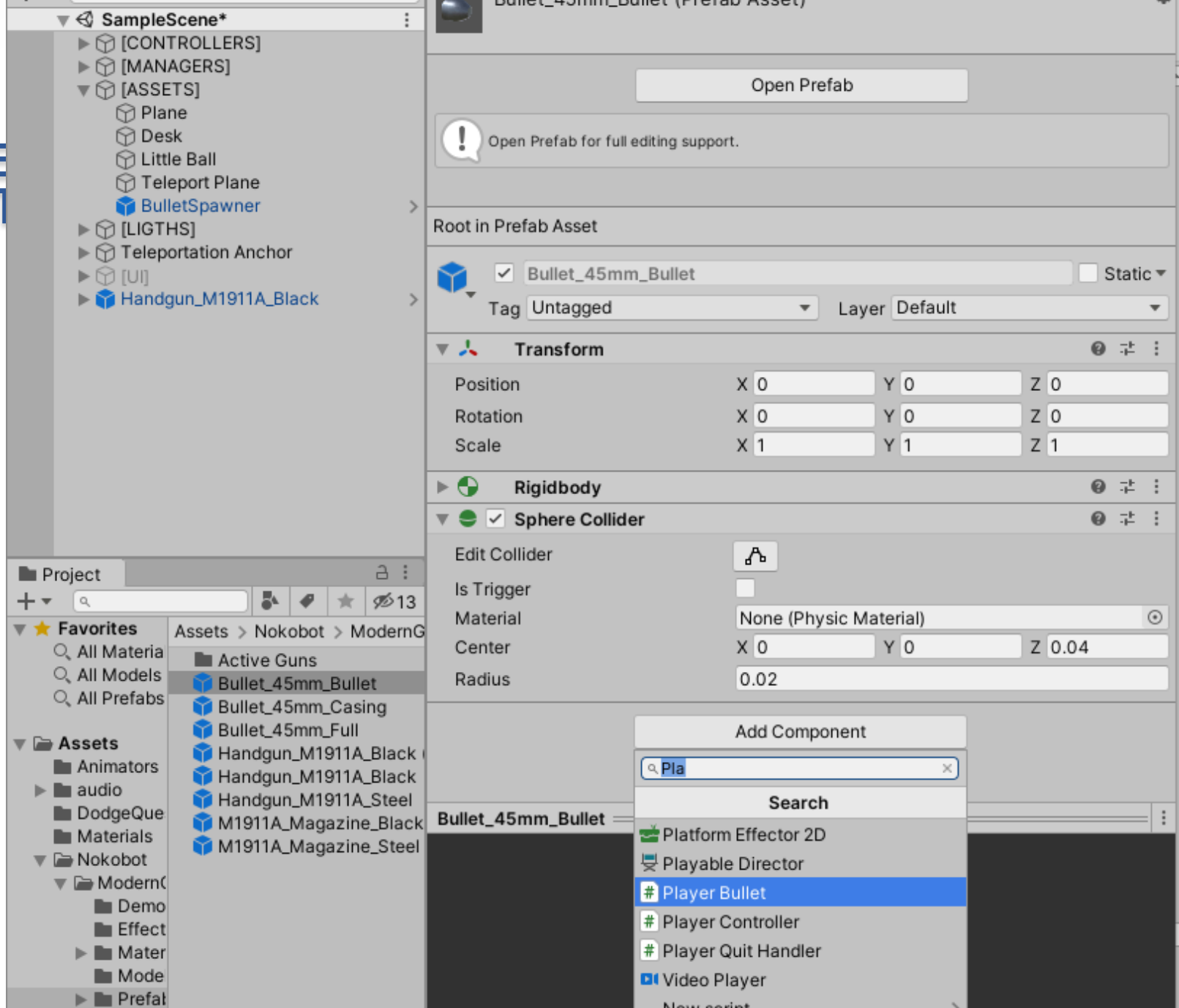
- PlayerBullet의 설정

- 기본 공격력 35.0f
- OnTriggerEnter 함수에서 Monster 인 경우 데미지를 주고, 총알 객체를 없앴!

```
Assembly CSharp> Documents> Quest2_UI> Assets> DodgeQuest> Bullet.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerBullet : MonoBehaviour
6 {
7     public float attackAmount = 35.0f;
8     // Start is called before the first frame update
9     void Start()
10    {
11        Destroy(gameObject, 5f); //5초후 제거!
12    }
13
14    // Update is called once per frame
15    void Update()
16    {
17    }
18
19
20    private void OnTriggerEnter(Collider other)
21    {
22        if (other.tag == "Monster")
23        {
24            BulletSpawner bulletmonster = other.GetComponent<BulletSpawner>();
25
26            if (bulletmonster != null)
27            {
28                bulletmonster.GetDamage(attackAmount);
29            }
30        }
31
32        Destroy(gameObject);
33    }
34 }
35
36
```

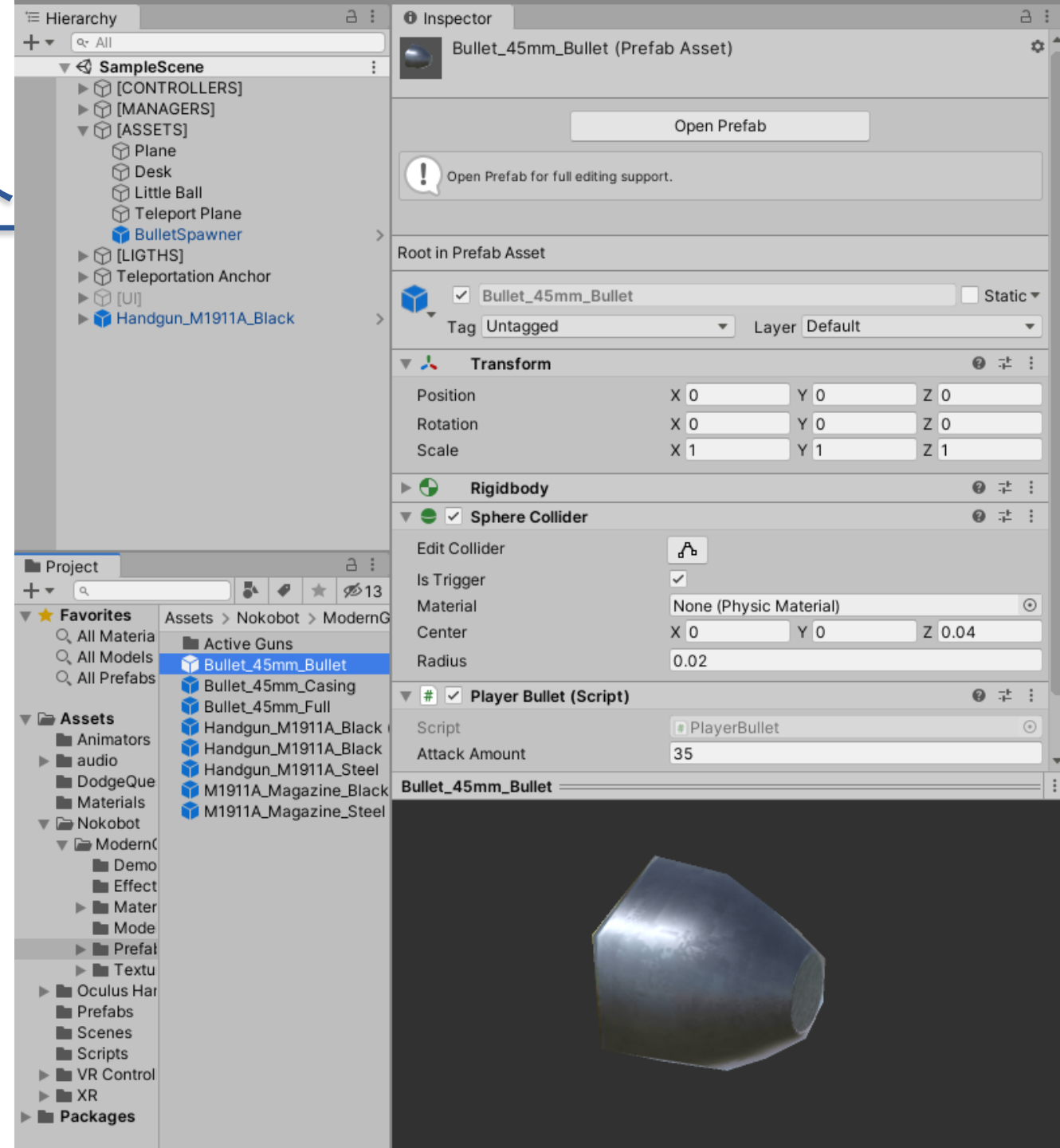
전투

- 다음의 총알 경로의 프리팹에 직접 PlayerBullet 컴포넌트 추가!



전투 시스템

- 설정된 값을 확인하고
ISTrigger 는 반드시 활성화



전투 시

- 결과 확인!

