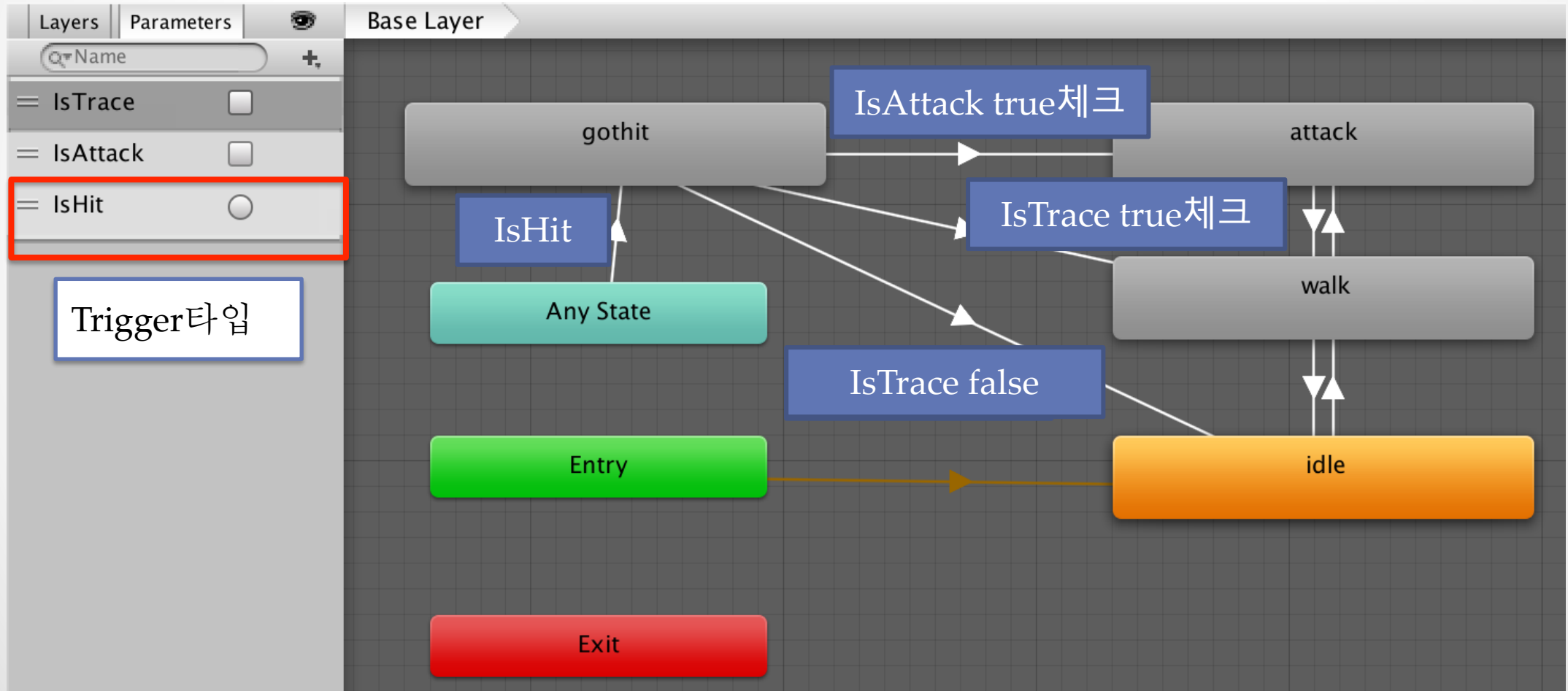


총싸움 전투 마지막

1. 이준

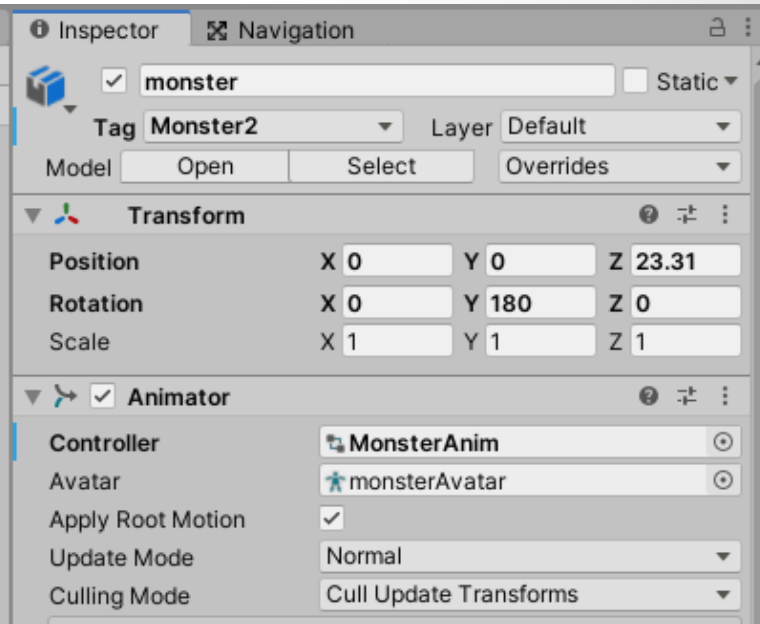
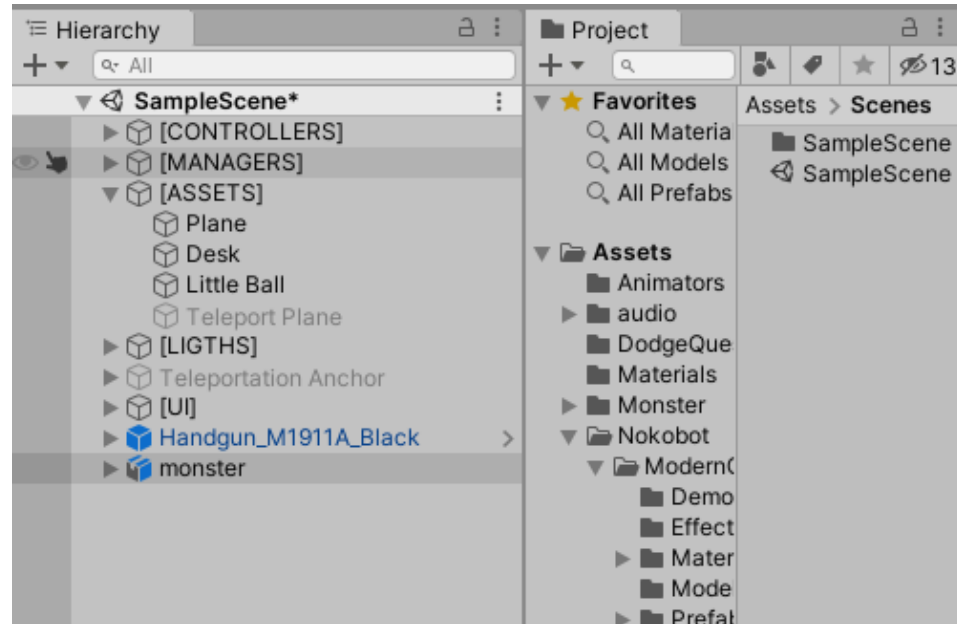
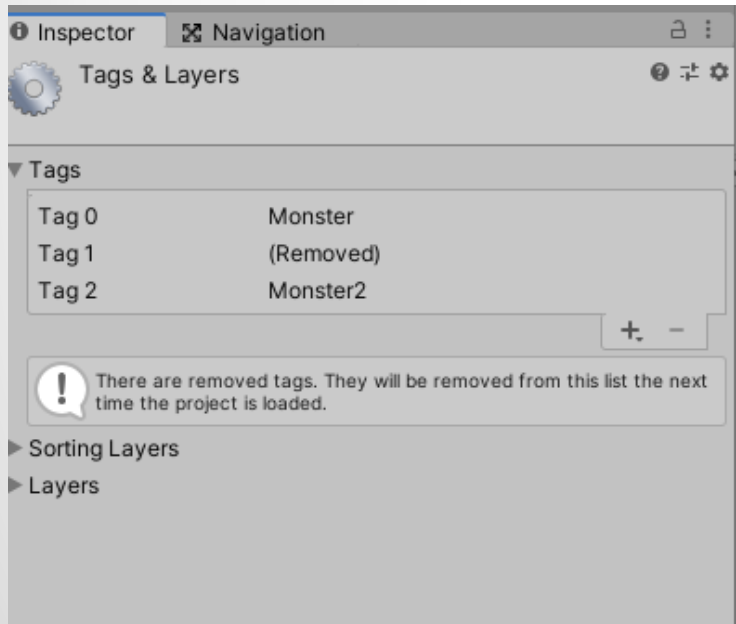
외계인 몬스터와 전투

- 몬스터가 총알에 맞았을 때의 애니메이션 설정



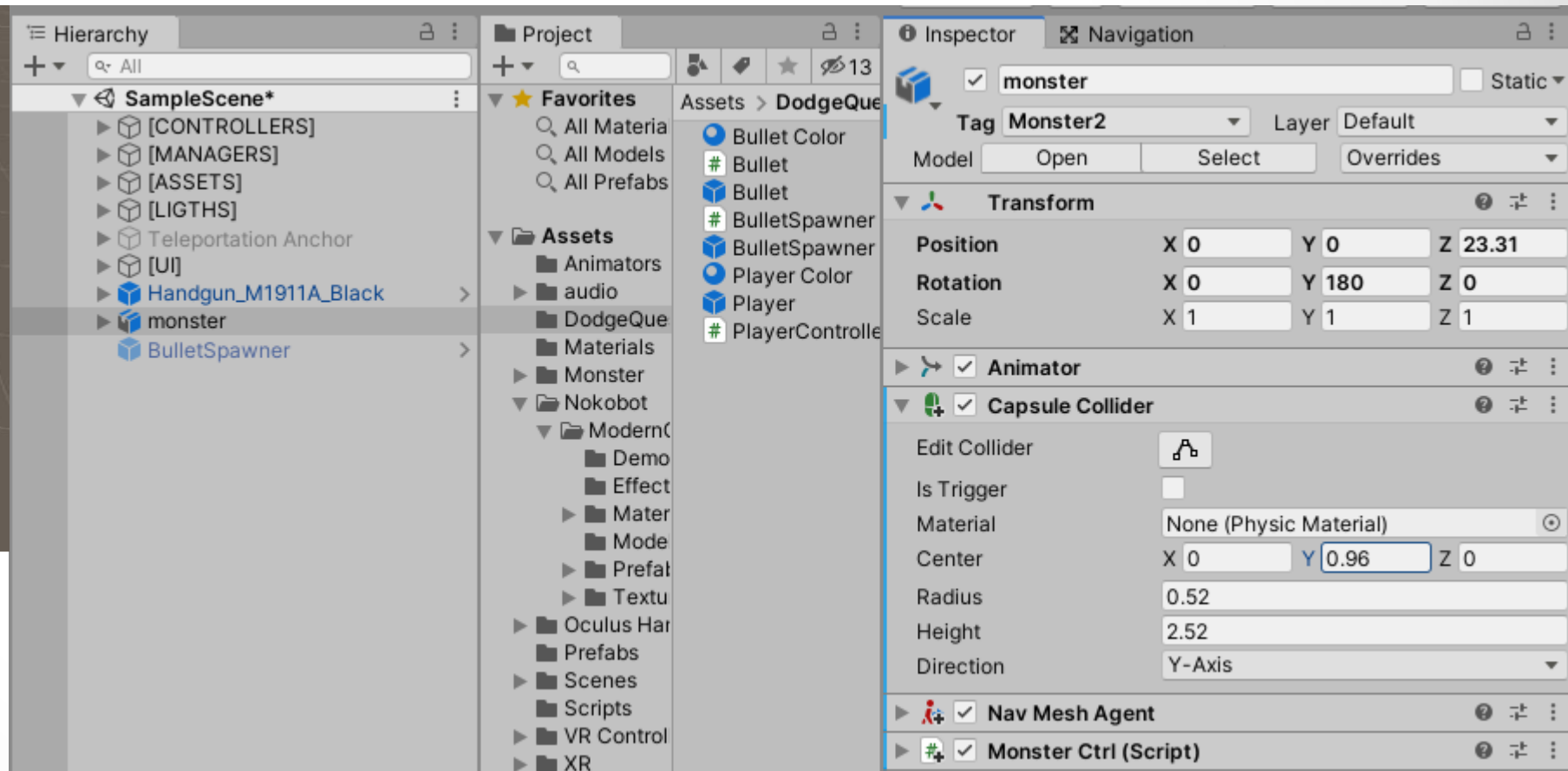
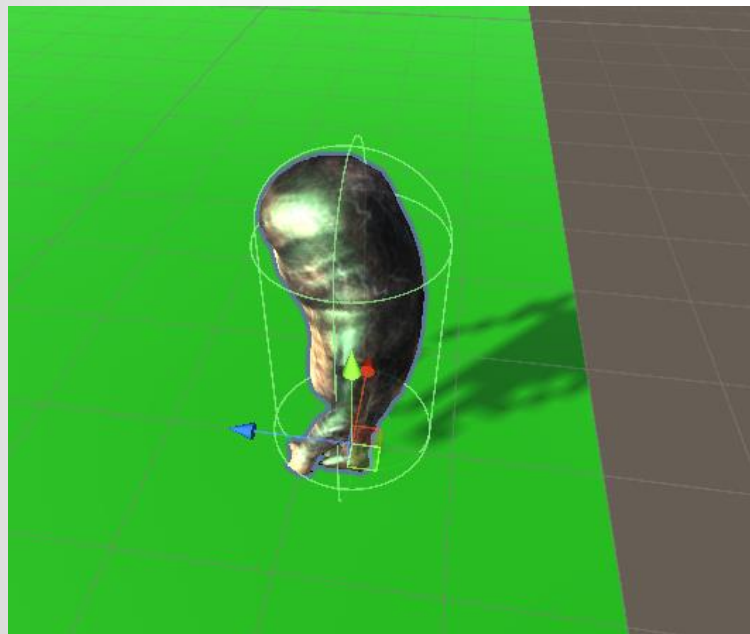
외계인 몬스터와 전투

- Tag 에서 외계인 몬스터를 지칭하는 Monster2 이름으로 추가
- Monster 의 태그를 Monster2로 변경



외계인 몬스터와 전투

- Monster 객체에 CapsuleCollider를 추가!



외계인 몬스터와 전투

■ MonsterCtrl.cs 에 GetDamage() 함수 추가

▶ 몬스터가 총알에 맞았을 경우에 대해 코드 구현

```
public void GetDamage(float amountt)
{
    hp -= (int) (amountt / 2.0f); // 외계인은 데미지를 절반으로 줄이는 특수 능력
    animator.SetTrigger("IsHit");
}
```

▶ 다음의 체력 변수를 선언해줘야합니다

```
private int hp = 100;
```

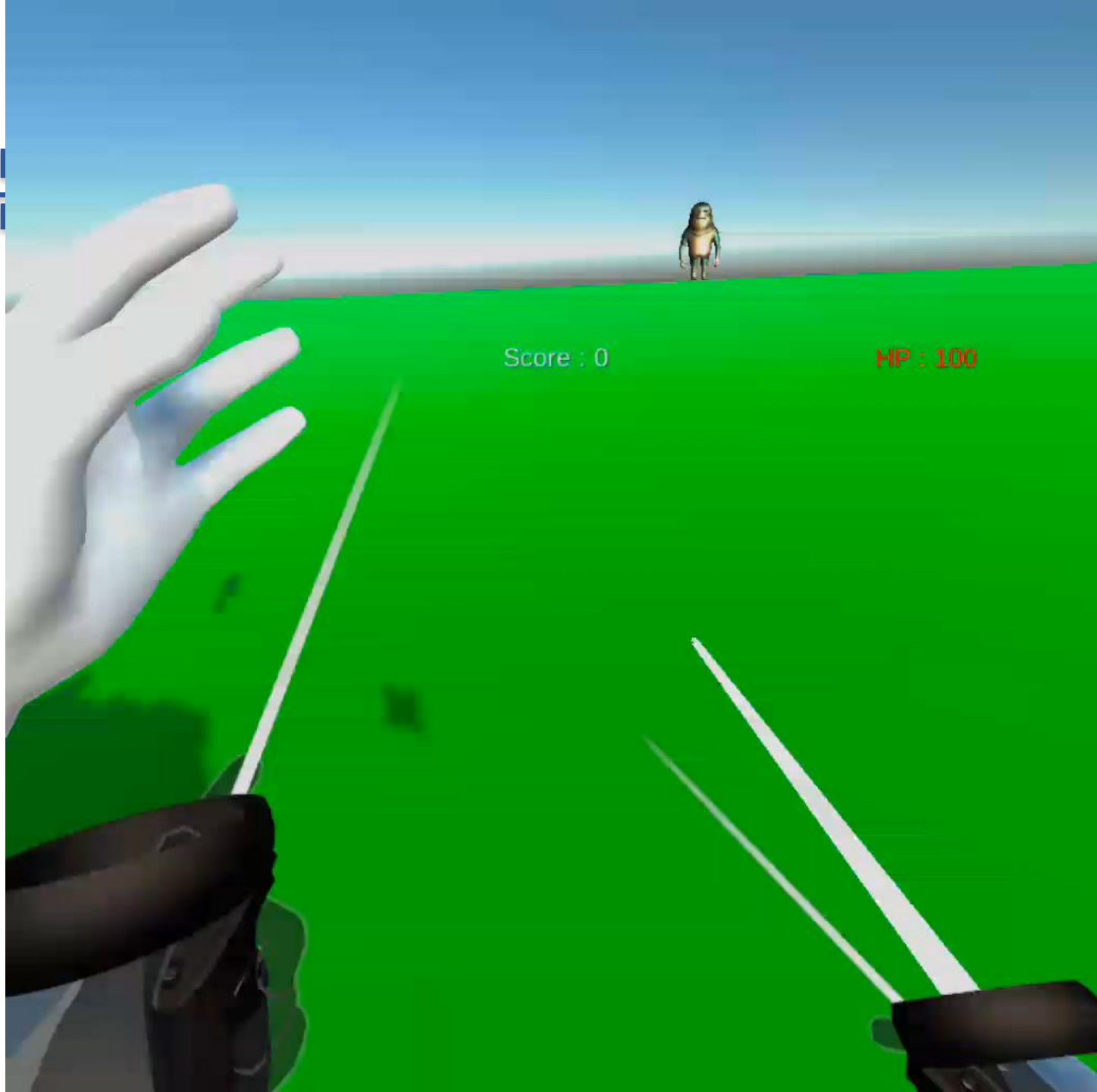
외계인 몬스터와 전투

- PlayerBullet.cs 파일의 OnTriggerEnter 함수에 외계인 몬스터 처리를 넣어줘야함!

```
20 private void OnTriggerEnter(Collider other)
21 {
22     if (other.tag == "Monster") // BulletSpawner
23     {
24         BulletSpawner bulletmonster = other.GetComponent<BulletSpawner>();
25
26         if (bulletmonster != null)
27         {
28             bulletmonster.GetDamage(attackAmount);
29             Destroy(gameObject);
30         }
31     else if(other.tag == "Monster2") // Alien Monster
32     {
33         MonsterCtrl alien = other.GetComponent<MonsterCtrl>();
34
35         if (alien != null)
36         {
37             alien.GetDamage(attackAmount);
38         }
39         Destroy(gameObject);
40     }
41
42
43 }
```

외계인

- 이제 데미지 처리 게임 매니저 처리를 해보자!



외계인 몬스터와 전투

■ 플레이어가 공격 데미지를 입어 생명력이 줄어드는 로직 구현

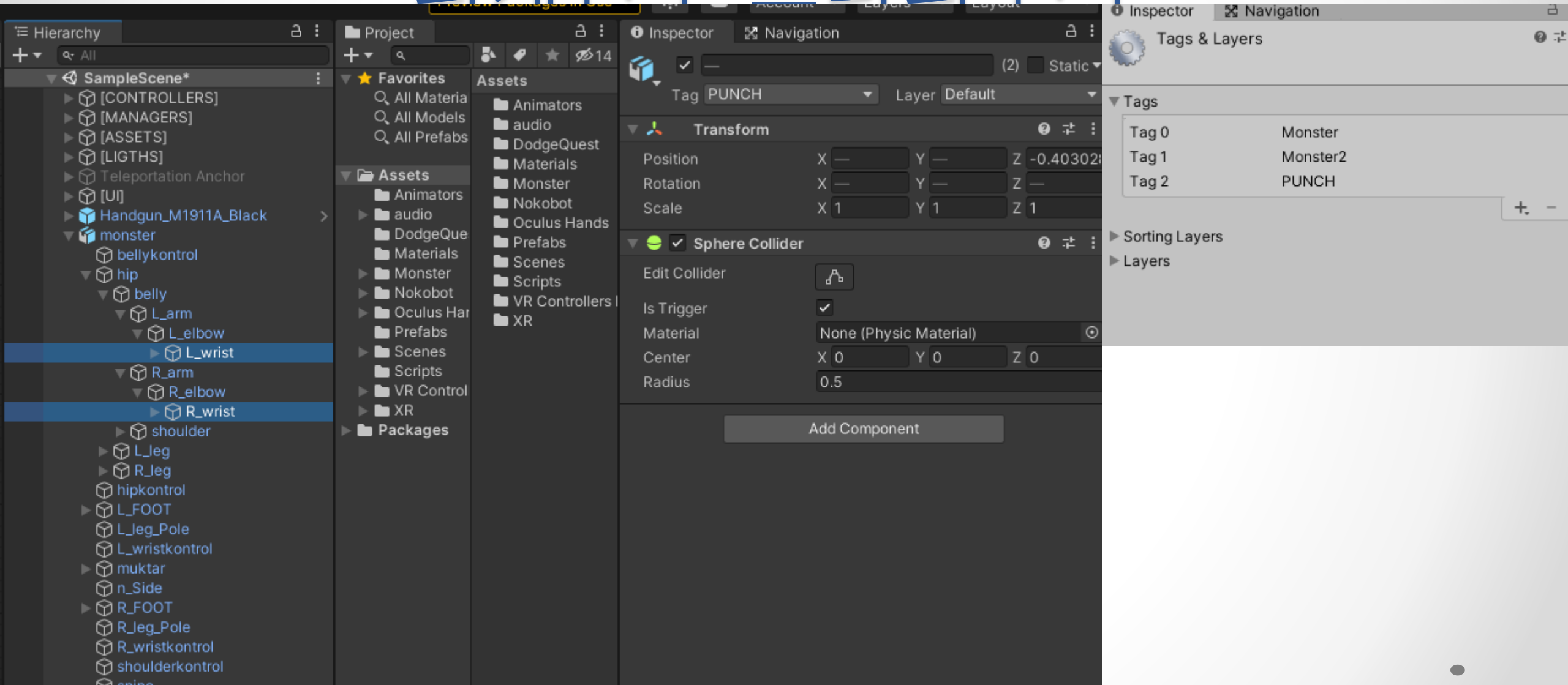
- ▶ 몬스터 양쪽 손에 Collider를 추가하여 몬스터가 공격 애니메이션을 수행 할 때 플레이어의 Collider와 충돌 시켜 데미지를 입히는 방식

■ 구현

- ▶ 씬 뷰의 Monster를 선택 한후 L_wrist / R_wrist를 동시에 선택 한 후 Sphere Collider와 Rigidbody 컴포넌트 두개를 동시에 추가한다.
 - 2개를 동시에 선택한 후 추가하면 똑같이 동시에 컴포넌트가 추가됨
- ▶ 2개를 동시 지정한 한 PUNCH태그를 만든 후에 태그 지정
- ▶ 2개를 동시 지정한 후 Use Gravity/ Is Kinematic / Is Trigger 속성을 다음과 같이 지정

컴포넌트	속성	설정값
Sphere Collider	IsTrigger	TRUE
	Use Gravity	FALSE
	Is Kinematic	TRUE

외계인 몬스터와 전투



외계인 몬스터와 전투

- 플레이어의 리igidBody 변수들은 다음과 같이 주석 처리합니다~

```
PlayerController.cs
Assembly-CSharp
PlayerController

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UIElements;
5
6  public class PlayerController : MonoBehaviour
7  {
8      //private Rigidbody playerRigidbody;
9      public float speed = 8f;
10     public float hp = 100.0f;
11     // Start is called before the first frame update
12     void Start()
13     {
14         //playerRigidbody = GetComponent<Rigidbody>();
15     }
16
17     void Update()
18     {
19         //float xInput = Input.GetAxis("Horizontal");
20         //float zInput = Input.GetAxis("Vertical");
21
22         //float xSpeed = xInput * speed;
23         //float zSpeed = zInput * speed;
24
25         //Vector3 newVelocity = new Vector3(xSpeed, 0f, zSpeed);
26         //playerRigidbody.velocity = newVelocity;
27     }
```

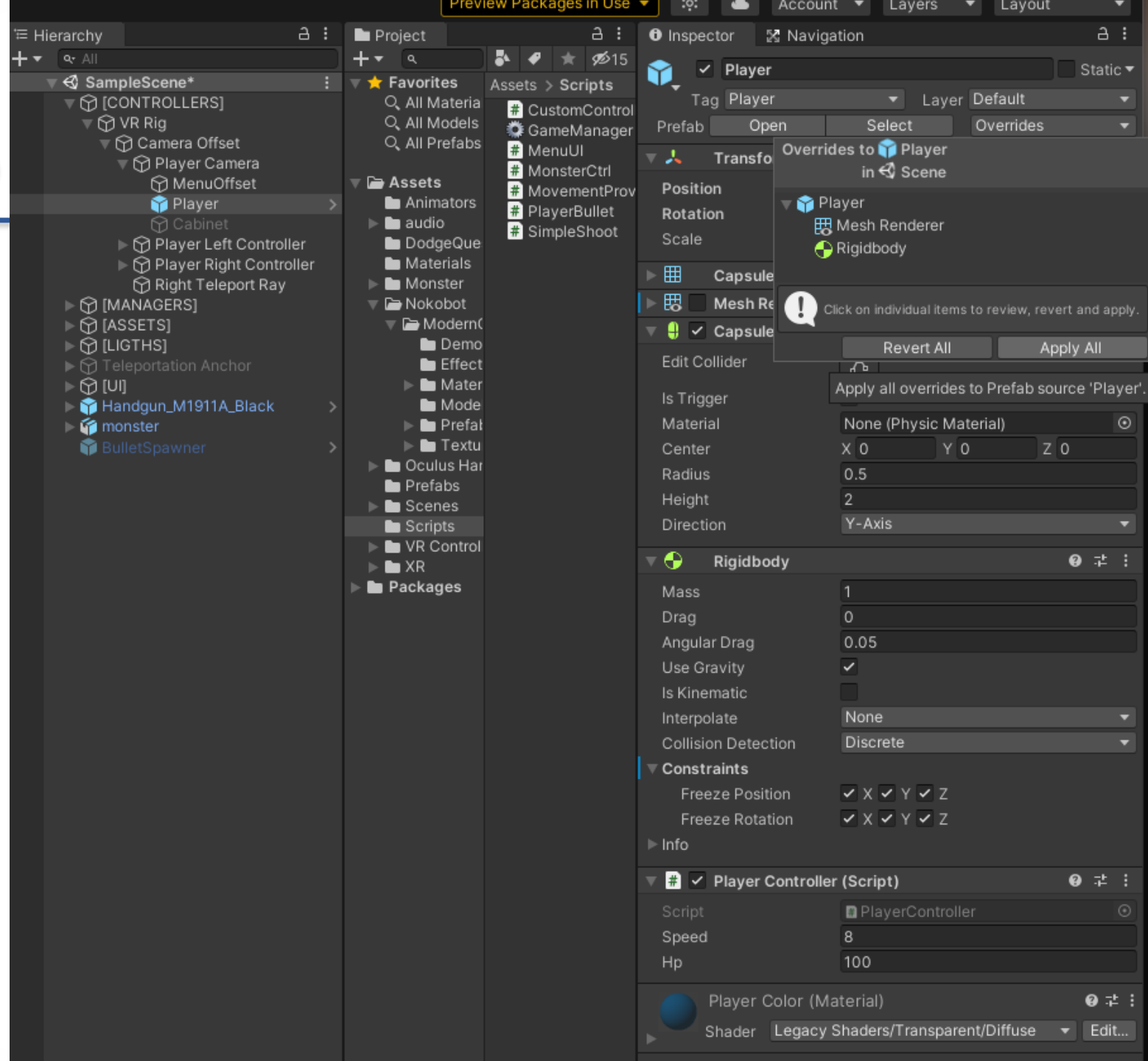
외계인 몬스터와 전투

- 이제 PUNCH 공격을 받는 대상인 PlayerCtroller.cs 에서
- TriggerEnter 함수를 추가하여 태그가 PUNCH 인 경우 GetDamage 호출!

```
private void OnTriggerEnter(Collider other)
{
    if(other.gameObject.tag == "PUNCH")
    {
        GetDamage(10.0f);
    }
}
```

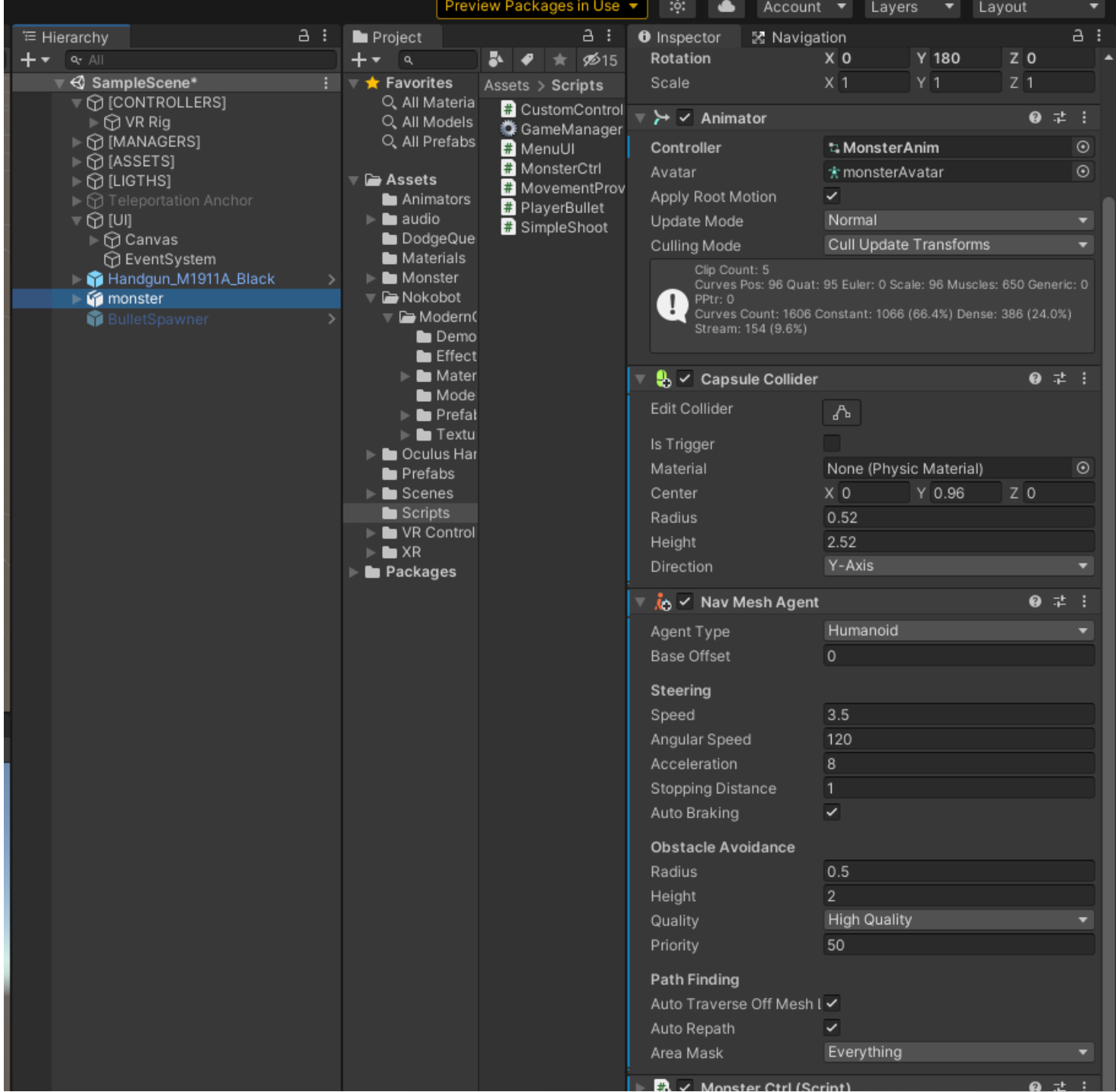
외계인

- Player 객체는
리igidBody는
제약조건을 다음과 같이
주면 됨
- 콜라이더 두개다 트리거
모드인경우
리igidBody가 필요
(충돌체킹용)



외계인

- 몬스터에서 Nav Mesh Agent 의 Stopping Distance를 1.5로 설정 (그래야 플레이어 객체가 맞을수 있습니다~)



외계인 몬스터와 전투

- 플레이어가 죽으면 몬스터는 공격행동을 멈춰야함! Idle로 변경!
- 이를 위해 MonsterCtrl.cs 파일의 몬스터 상태 부분에 추가!

```
58 //일정한 간격으로 몬스터의 행동 상태를 체크하고 monsterState값 변경
59 IEnumerator CheckMonsterState()
60 {
61     while(!isDie)
62     {
63         //0.2초 동안 기다렸다가 다음으로 넘어감
64         yield return new WaitForSeconds(0.2f);
65
66         //몬스터와 플레이어 사이의 거리 측정
67         float dist = Vector3.Distance(playerTr.position , monsterTr.position);
68
69         if (dist <= attackDist && !FindObjectOfType<GameManager>().isGameOver) //공격거리 범위 이내로 들어왔는지 확인
70         {
71             monsterState = MonsterState.attack;
72         }
73         else if (dist <= traceDist) //추적거리 범위 이내로 들어왔는지 확인
74         {
75             monsterState = MonsterState.trace; //몬스터의 상태를 추적으로 설정
76         }
77         else
78         {
79             monsterState = MonsterState.idle; //몬스터의 상태를 idle모드로 설정
80         }
81     }
82 }
```

외계인 몬스터

- 플레이어가 죽으면 몬스터는 공격행동을 멈춰야함! Idle로 변경!
- 이를 위해 MonsterCtrl.cs 파일의 몬스터 상태 부분에 추가!



외계인 몬스터와 전투

- 몬스터가 죽는 애니메이션 구현 (MonsterCtrl.cs)
 - 멤버 변수 추가

```
controller.cs  MenuUI.cs  MonsterCtrl.cs  PlayerController.cs
bly-CSharp

using UnityEngine;
using System.Collections;

public class MonsterCtrl : MonoBehaviour {
    //몬스터의 상태 정보가 있는 Enumerable 변수 선언
    public enum MonsterState { idle, trace, attack, die };
    //몬스터의 현재 상태 정보를 저장할 Enum 변수
    public MonsterState monsterState = MonsterState.idle;

    //속도 향상을 위해 각종 컴포넌트를 변수에 할당
    private Transform monsterTr;
    private Transform playerTr;
    private UnityEngine.AI.NavMeshAgent nvAgent;
    private Animator animator;

    //추적 사정거리
    public float traceDist = 10.0f;
    //공격 사정거리
    public float attackDist = 2.0f;

    //몬스터의 사망 여부
    private bool isDie = false;

    //몬스터 생명 변수
    private int hp = 100;
```


외계인 몬스터와 전투

- 몬스터의 HP를 깎고 나서 0 밑이 되면 몬스터를 죽임!
- 몬스터 컨트롤러의 GetDamage 수정

```
public void GetDamage(float amountt)
{
    hp -= (int) (amountt / 2.0f); // 외계인은 데미지를 절반으로 줄이는 특수 능력
    animator.SetTrigger("IsHit");

    if (hp <= 0)
        MonsterDie();
}
```

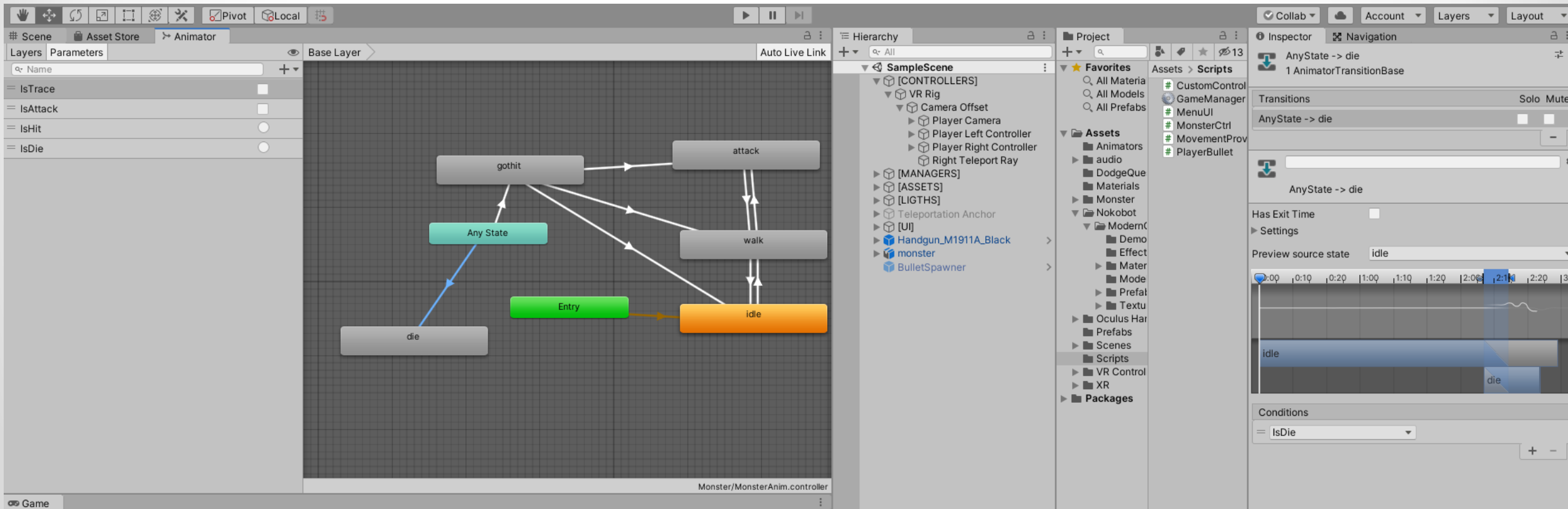
외계인 몬스터와 전투

- 몬스터가 죽는 MonsterDie 함수 추가

```
134 //몬스터 사망 시 처리 루틴
135 void MonsterDie()
136 {
137     if (isDie == true) return;
138
139     //모든 코루틴을 정지
140     StopAllCoroutines();
141     isDie = true;
142     monsterState = MonsterState.die;
143     nvAgent.isStopped = true;
144     animator.SetTrigger("IsDie");
145
146     //몬스터에 추가된 Collider를 비활성화
147     gameObject.GetComponentInChildren<CapsuleCollider>().enabled = false;
148     foreach(Collider coll in gameObject.GetComponentsInChildren<SphereCollider>())
149     {
150         coll.enabled = false;
151     }
152     // 외계인은 2점!
153     FindObjectOfType<GameManager>().GetScored(2);
154
155 }
```

외계인 몬스터와 전투

die 애니메이션 추가 및 IsDie 트리거 추가!



외계인 몬스터와

