

총싸움 전투 마지막

1. 이준

외계인 몬스터와 전투

■ 인공지능 구현 로직

- ▶ idle/trace/attack/die 4개의 몬스터 상태로 정의

Start()

StartCoroutine(this.CheckMonsterState());
StartCoroutine(this.MonsterAction());

IEnumerator CheckMonsterState()
몬스터 상태 체크

```
if(플레이어와의 거리 <= 공격 사정거리)
    몬스터 상태 = attack
if(플레이어와의 거리 <= 추적 사정거리)
    몬스터 상태 = trace
else
    몬스터 상태 = idle
```

0.2초 동안 기다린 후 호출 (0.2초 간격)

IEnumerator MonsterAction()
몬스터 행동 수행

```
switch (몬스터 상태)
    case idle :
        trace 중지
        idle 애니메이션 수행
    case trace :
        공격 중지
        네비게이션 추적 재시작
        walk 애니메이션 수행
    case attack :
        추적 중지
        attack 애니메이션 수행
```

외계인 몬스터와 전투

■ MonsterCtrl.cs 파일 작성

```
using UnityEngine;
using System.Collections;

public class MonsterCtrl : MonoBehaviour {

    // 몬스터 상태정보가 있는 Enumerable 변수 선언
    public enum MonsterState {idle, trace, attack, die};
    // 몬스터의 현재 상태 정보를 저장 할 Enum 변수
    public MonsterState monsterState = MonsterState.idle;

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

    public float traceDist = 10.0f;    // 추적 사정거리
    public float attackDist = 2.0f;    // 공격 사정거리
    private bool isDie = false;        // 몬스터 사망 여부
```

외계인 몬스터와 전투

■ MonsterCtrl.cs 파일 작성

```
void Start () {  
    monsterTr = this.gameObject.GetComponent<Transform>();  
    playerTr = GameObject.FindWithTag("Player").GetComponent<Transform>();  
    nvAgent = this.gameObject.GetComponent<NavMeshAgent>();  
    animator = this.gameObject.GetComponent<Animator>();  
  
    // 몬스터의 Transform 할당  
    // 추적대상인 Player의 Transform 할당  
    // NavMeshAgent 컴포넌트 할당  
    // Animator 컴포넌트 할당  
  
    // 추적대상의 위치를 설정하면 바로 추적 시작  
    // nvAgent.destination = playerTr.position;  
  
    StartCoroutine(this.CheckMonsterState()); // 일정 간격으로 몬스터의 행동 상태를 체크하는 코루틴 함수 실행  
    StartCoroutine(this.MonsterAction());      // 몬스터의 상태에 따라 동작하는 루틴을 실행하는 코루틴 함수 실행  
}
```

외계인 몬스터와 전투

■ MonsterCtrl.cs 파일 작성

```
// 일정한 간격으로 몬스터의 행동 상태를 체크하고 monsterState의 값 변경
IEnumerator CheckMonsterState() {
    while(!isDie) {
        // 0.2초 동안 기다렸다가 다음으로 넘어감
        yield return new WaitForSeconds(0.2f);

        //몬스터와 플레이어 사이의 거리 측정
        float dist = Vector3.Distance(playerTr.position, monsterTr.position);

        if(dist <= attackDist){           // 공격거리 범위 이내로 들어왔는지 확인
            monsterState = MonsterState.attack;
        }
        else if(dist <= traceDist) {      //추적거리 범위 이내로 들어왔는지 확인
            monsterState = MonsterState.trace; // 몬스터 상태를 추적으로 설정
        }
        else {
            monsterState = MonsterState.idle; // 몬스터 상태를 idle모드로 설정
        }
    }
}
```

외계인 몬스터와 전투

■ MonsterCtrl.cs 파일 작성

```
// 몬스터의 상태값에 따라 적절한 동작을 수행하는 함수
IEnumerator MonsterAction() {
    while (!isDie) {
        switch(monsterState){
            // idle 상태
            case MonsterState.idle:
                nvAgent. isStopped = true;
                animator.SetBool("IsTrace", false); // Animator의 IsTrace 변수를 false로 설정
                break;
            // 추적 상태
            case MonsterState.trace:
                nvAgent.destination = playerTr.position; // 추적 대상의 위치를 넘겨줌
                nvAgent. isStopped = false;
                animator.SetBool("IsTrace", true); // Animator의 IsTrace 변수값을 true로 설정
                break;
            // 공격 상태
            case MonsterState.attack:
                break;
        }
        yield return null;
    }
}
```

Project

Inspector

Navigation

+

▼

Q

13

Scene*

CONTROLLERS]

NAGERS]

SETS]

THS]

portation Anchor

dgun_M1911A_Black

ster

★ Favorites

Q All Materia

Q All Models

Q All Prefabs

Assets

Animators

audio

DodgeQue

Materials

Monster

Nokobot

Oculus Har

Prefabs

Scenes

Scripts

VR Control

XR

Packages

Assets > Scripts

CustomControl

GameManager

MenuUI

MonsterCtrl

MovementProv

PlayerBullet

monster

Static

Tag Untagged

Layer Default

Model Open

Select

Overrides

Transform

Position X 0 Y 0 Z 23.31

Rotation X 0 Y 180 Z 0

Scale X 1 Y 1 Z 1

Animator

Controller MonsterAnim

Avatar monsterAvatar

Apply Root Motion

Update Mode Normal

Culling Mode Cull Update Transforms

Clip Count: 2

Curves Pos: 42 Quat: 42 Euler: 0 Scale: 42 Muscles: 260 Generic: 0

PPtr: 0

Curves Count: 680 Constant: 457 (67.2%) Dense: 76 (11.2%)

Stream: 147 (21.6%)

Nav Mesh Agent

Agent Type Humanoid

Base Offset 0

Steering

Speed 3.5

Angular Speed 120

Acceleration 8

Stopping Distance 2

Auto Braking

Obstacle Avoidance

Radius 0.5

Height 2

Quality High Quality

Priority 50

Path Finding

Auto Traverse Off Mesh I

Auto Repath

Area Mask Mixed...

Monster Ctrl (Script)

Script MonsterCtrl

Monster State Idle

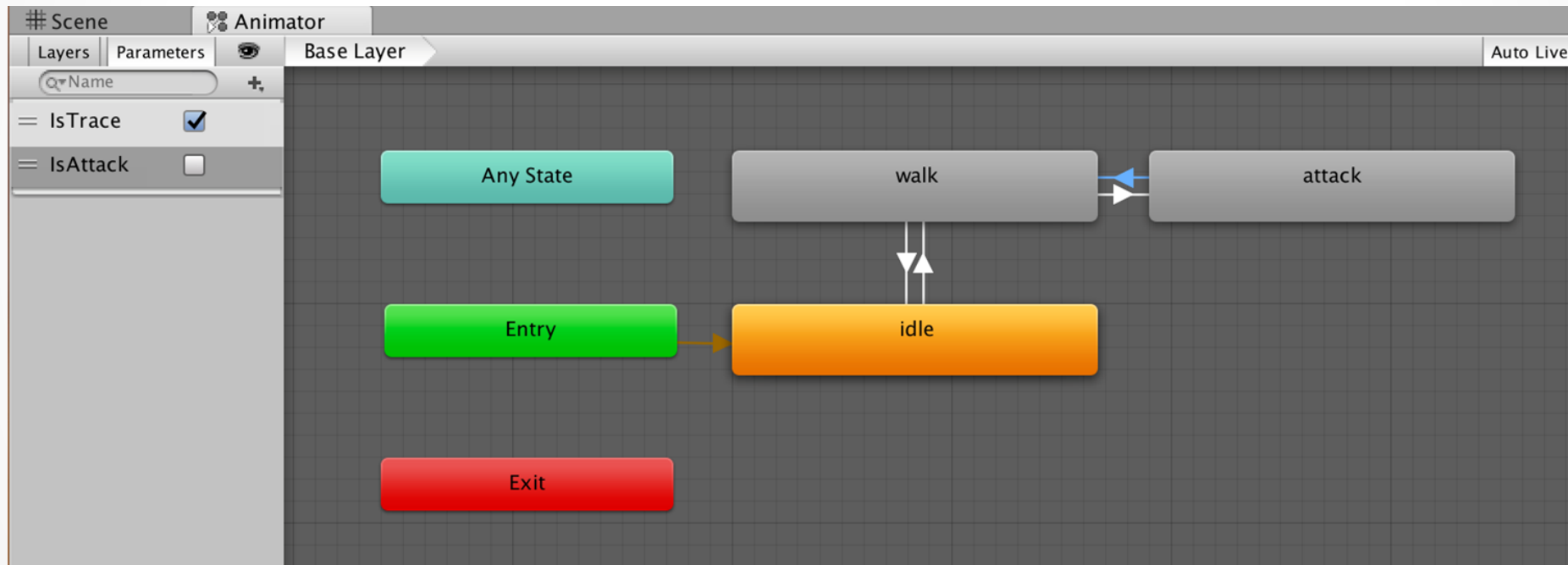
Trace Dist 10

Attack Dist 0.01



외계인 몬스터와 전투

- Animator 뷰에서 attack 애니메이션 클립을 배치 한 후 walk 스테이트와 양방향 Transition을 연결함



Transition	Conditions	설정값	Has Exit Time옵션
walk -> attack	IsAttack	TRUE	UnCheck
attack->walk	IsAttack	FALSE	UnCheck

외계인 몬스터와 전투

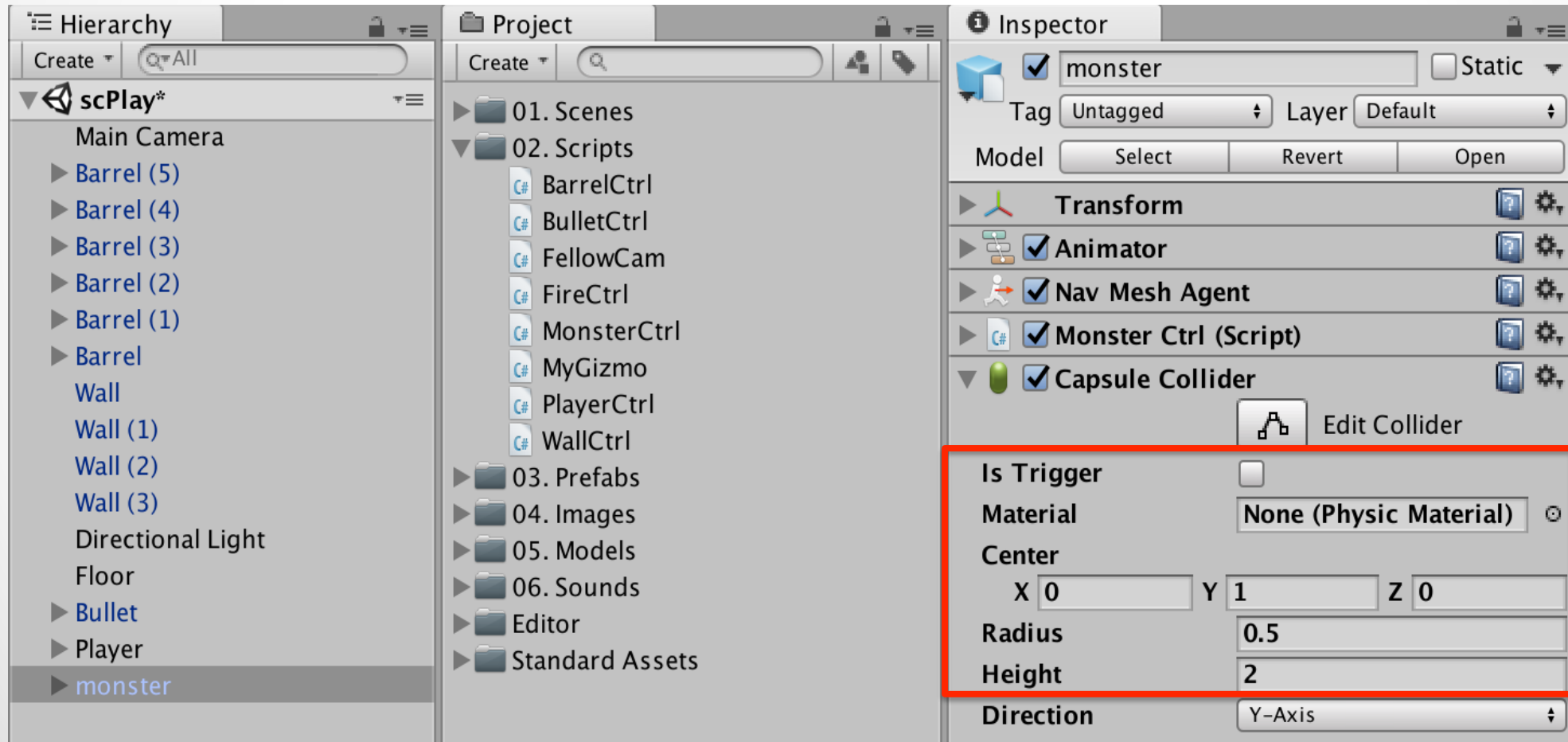
■ MonsterCtrl.cs 에서 Attack 수정

```
// 몬스터의 상태값에 따라 적절한 동작을 수행하는 함수
IEnumerator MonsterAction() {
    while (!isDie) {
        switch(monsterState){
            // idle 상태
            case MonsterState.idle:
                nvAgent. isStopped = true;
                animator.SetBool("IsTrace", false); // Animator의 IsTrace 변수를 false로 설정
                break;
            // 추적 상태
            case MonsterState.trace:
                nvAgent.destination = playerTr.position; // 추적 대상의 위치를 넘겨줌
                nvAgent. isStopped = false;
                animator.SetBool("IsAttack", false); // Animator의 IsAttack변수를 false로 설정 (공격하지 않음))
                animator.SetBool("IsTrace", true); // Animator의 IsTrace 변수값을 true로 설정 (따라가야 함:Walk동작)
                break;
            // 공격 상태
            case MonsterState.attack:
                nvAgent isStopped = true;
                animator.SetBool("IsAttack", true); // IsAttack을 true로 설정해 attack state로 전이
                break;
        }
        yield return null;
    }
}
```

외계인 몬스터와 전투

■ 플레이어가 발사한 총알이 몬스터에 명중했을 때 타격을 입는 리액션 구현

▶ Monster를 선택한 후 Capsule Collider를 추가 함



외계인

- 이제 몬스터가 공격하는 부분들을 볼 수 있음!

