

# 총싸움 전투 마지막

## 1. 이준

# 외계인 몬스터와 전투

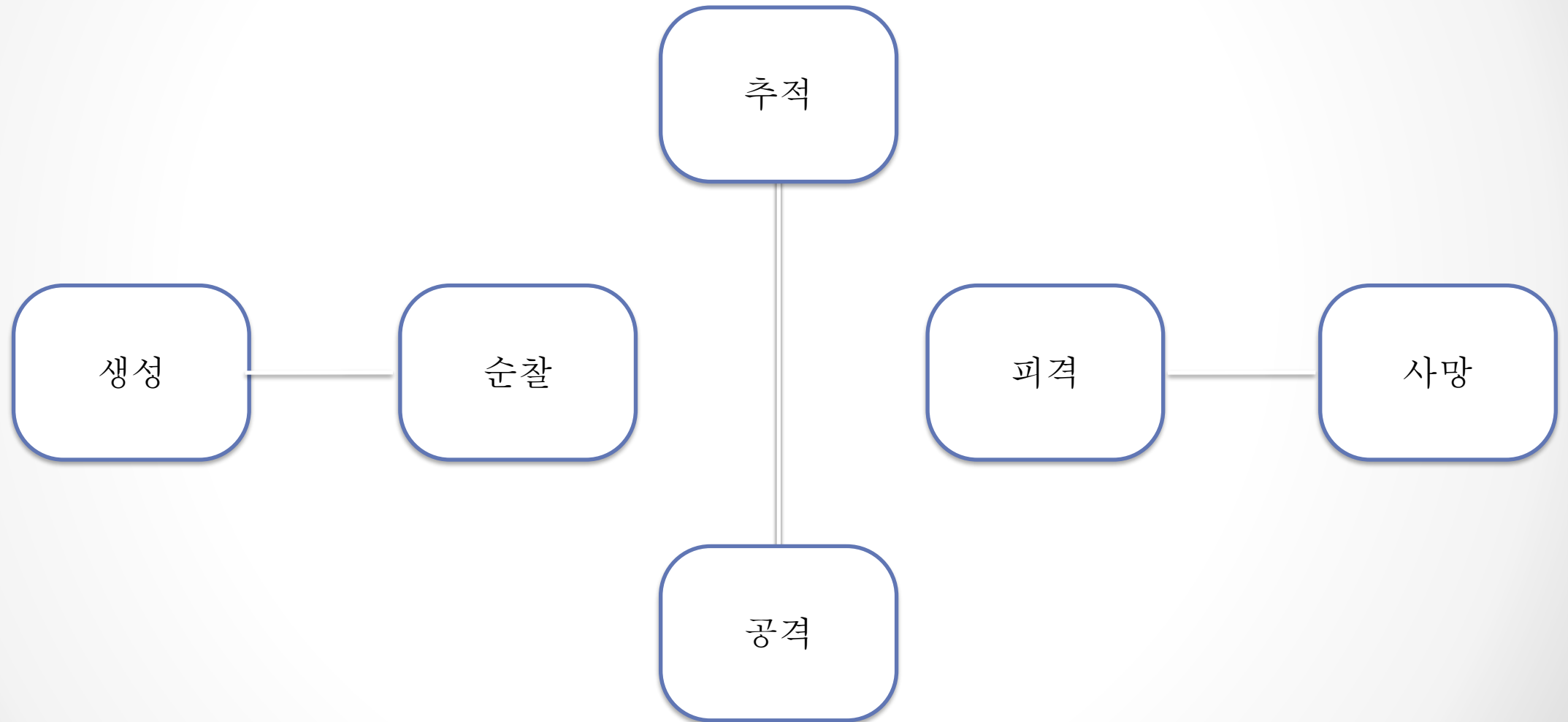
- **적 캐릭터 제작**

- ▶ 자신만의 인공지능(AI)를 가지고 있어야 현실감 있는 게임

- **NPC의 인공지능 구현**

- ▶ FSM(Finite State Machine : 유한상태머신)
  - 적 캐릭터가 스스로 알아서 주변환경에 적응, 들어오는 반응에 적절하게 반작용 하는 상태
  - 생성되면 주변 순찰 - 플레이어를 만나면 추적 시작 - 사정거리 내에 근접하면 공격 시작 - 데미지를 입고 생명이 다하면 사망

# 외계인 몬스터와 전투



# 외계인 몬스터와 전투

## ■ 메카닉

- ▶ 유니티의 애니메이션 상태머신 시스템, 애니메이션 미들웨어 엔진
  - FSM 상태에 따라 적절한 애니메이션 조작을 메카닉을 이용하여 구현함
  - 시각적인 설계방식을 제공함

## ■ Humanoid Type vs Generic Type

- ▶ Humanoid : 팔(2개), 다리(2개), 머리(1개)를 가진 인간형 캐릭터 ->  
리타겟팅(ReTargetting) 시스템을 사용하여 애니메이션을 공유 할 수 있음
  - 예) 오크/ 엘프/ 사람 ... 같은 애니메이션 공유
- ▶ Generic : 인간형을 제외한 4족 보행동물, 슬라임과 같은 기타 메시등을 일컬음
  - 리타겟팅 기능이 제공되지 않고 메카닉을 이용하여 애니메이션 처리

# 외계인 몬스터와 전투

- **적 캐릭터 제작**

- ▶ 자신만의 인공지능(AI)를 가지고 있어야 현실감 있는 게임

- **NPC의 인공지능 구현**

- ▶ FSM(Finite State Machine : 유한상태머신)
    - 적 캐릭터가 스스로 알아서 주변환경에 적응, 들어오는 반응에 적절하게 반작용 하는 상태
    - 생성되면 주변 순찰 - 플레이어를 만나면 추적 시작 - 사정거리 내에 근접하면 공격 시작 - 데미지를 입고 생명이 다하면 사망

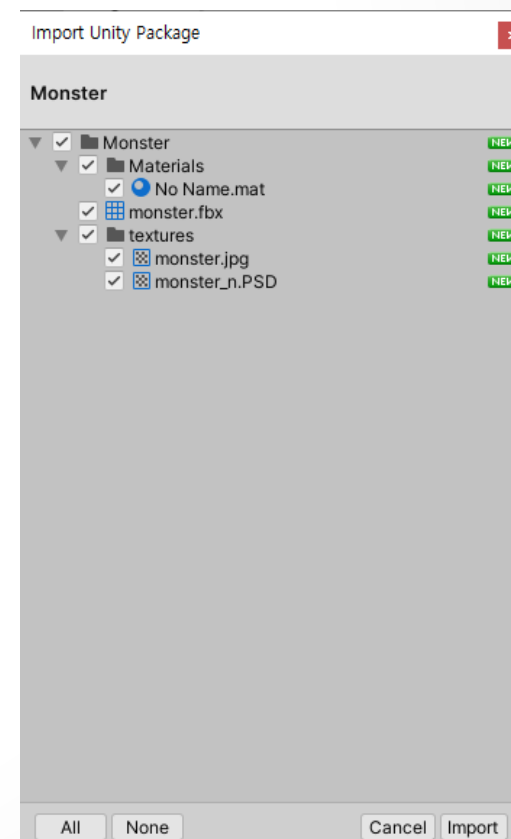
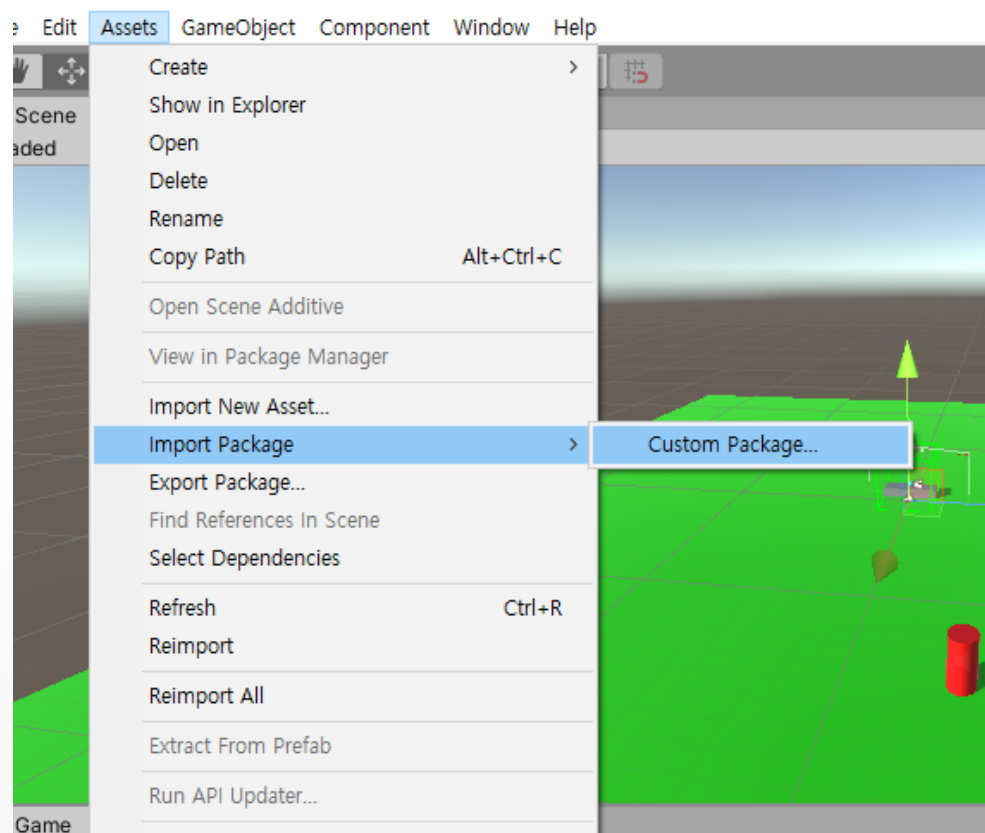
# 외계인 몬스터와 전투

- 모션 캡처 및 리타겟팅 사례

▶ <https://www.youtube.com/watch?v=rDpiujt0-m8>

# 외계인 몬스터와 전투

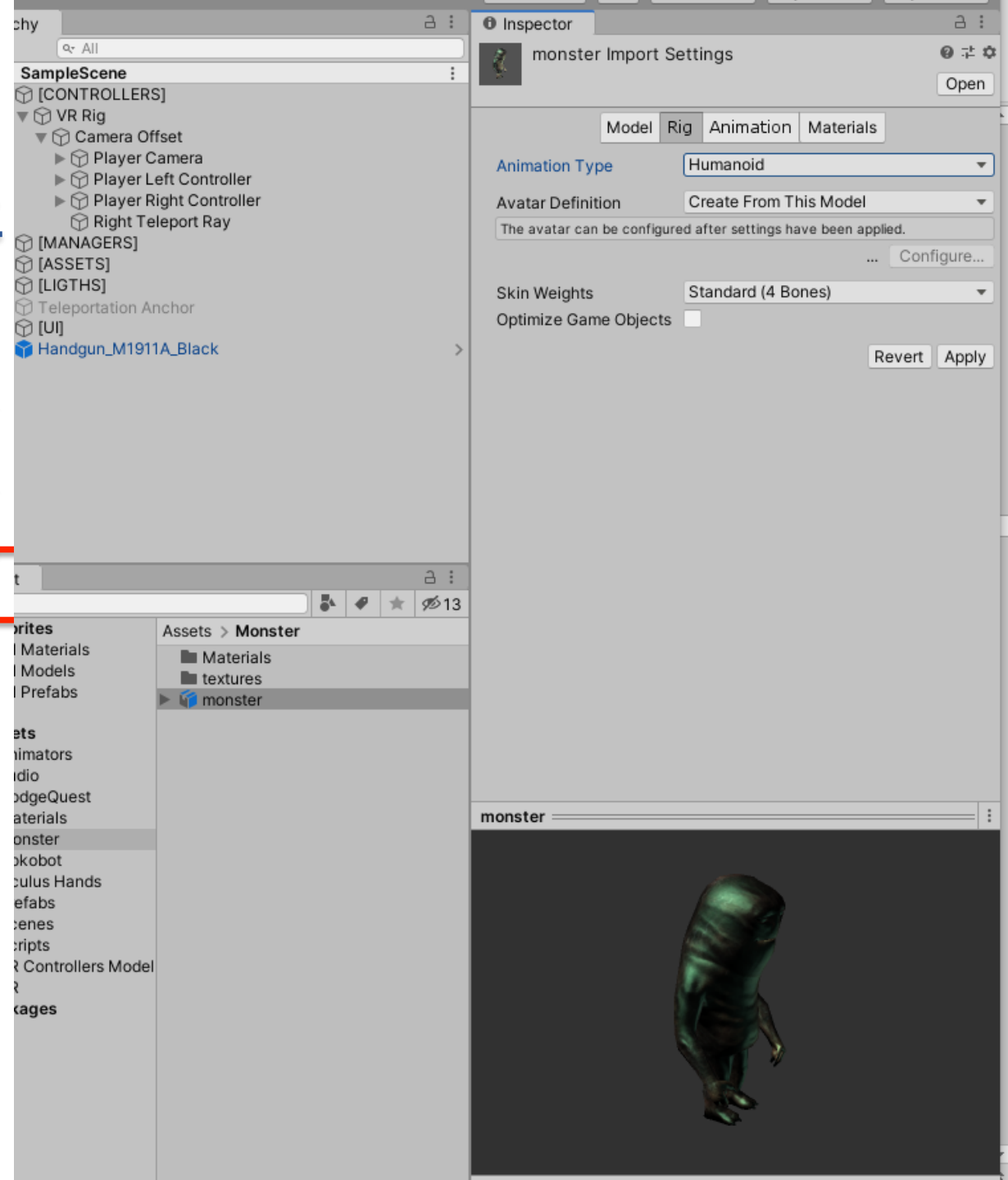
- Monster 패키지를  
임포트(<https://drive.google.com/file/d/1AM1leEMY9bhq7cSWPYTBhclIWYvCXV4Y/view?usp=sharing>)
- idle/walk/run/jump/fall/attack 등과 같은 애니메이션을 포함



# 외계인 몬스

- Monster 프리팹을 선택한후, Rig -> 메카닉 애니메이션으로 전환

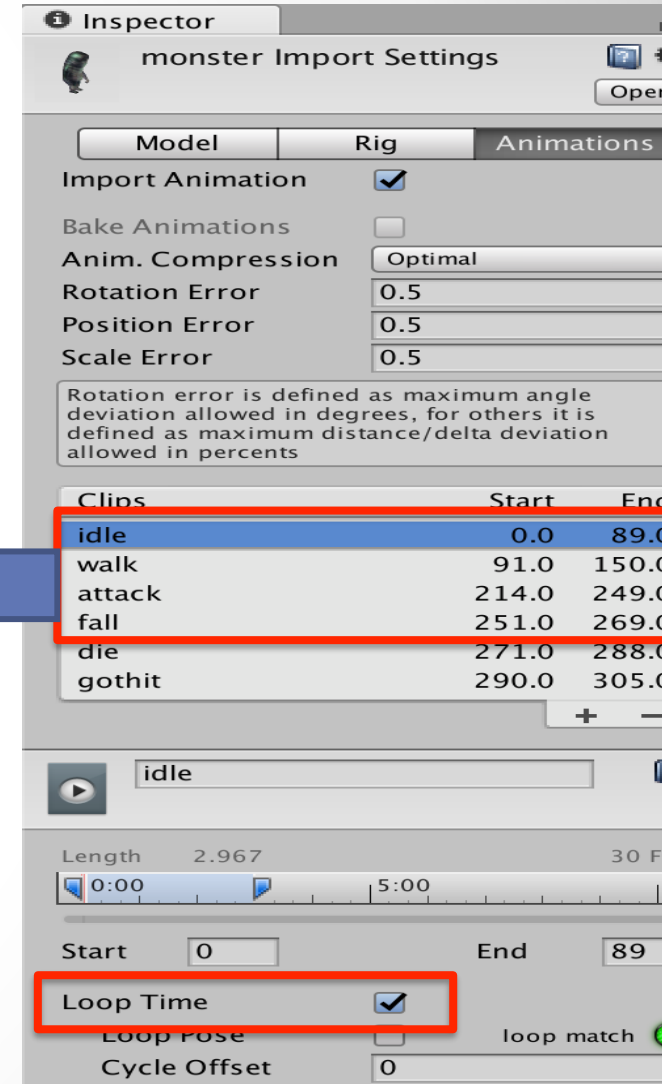
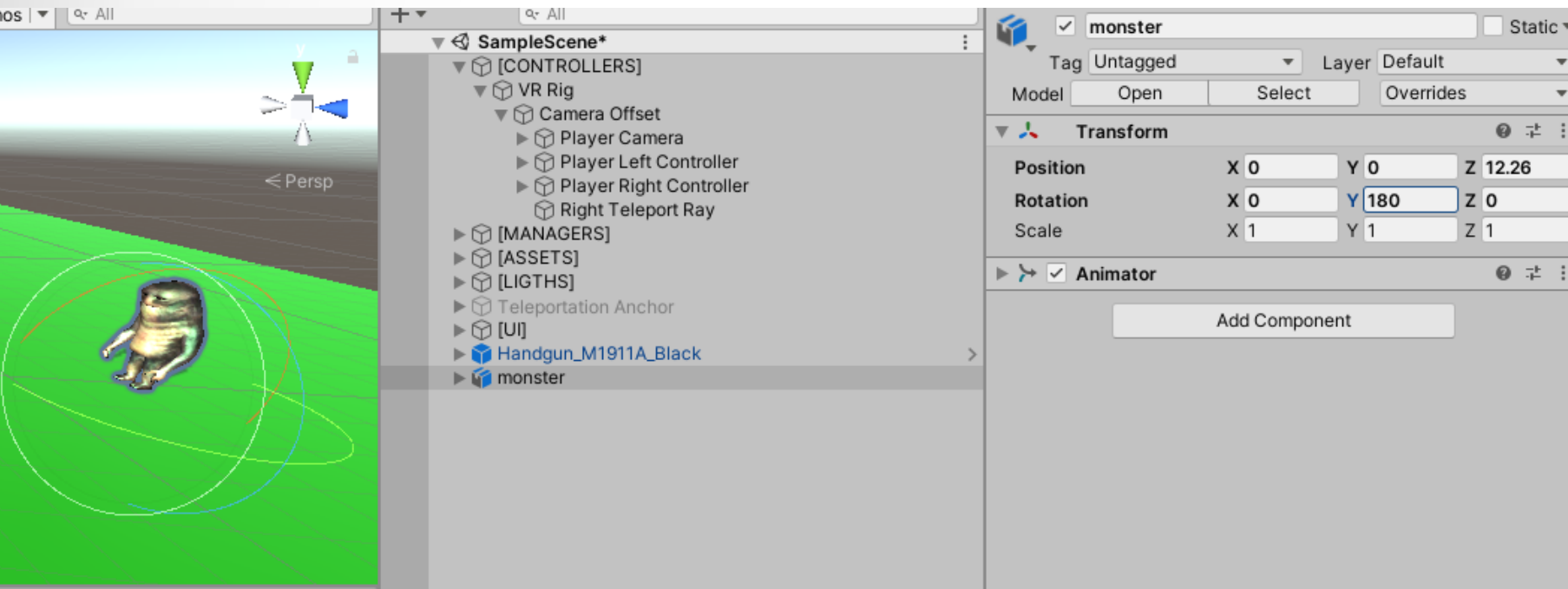
레거시	하위 호환성을 위해 유니티 4.X에서 사용했던 애니메이션
제너릭	메카닉 애니메이션(비인간형 모델), 리타겟팅 불가
휴머노이드	메카닉 애니메이션(인간형 모델), 리타겟팅 가능





# 외계인 몬스터와 전투

- 애니메이션 클립의 Loop Time 옵션 체크
  - 4개의 애니메이션 클립(idle/walk/attack/fall)만 Loop Time 체크
- Monster를 Scene으로 등록하여 배치

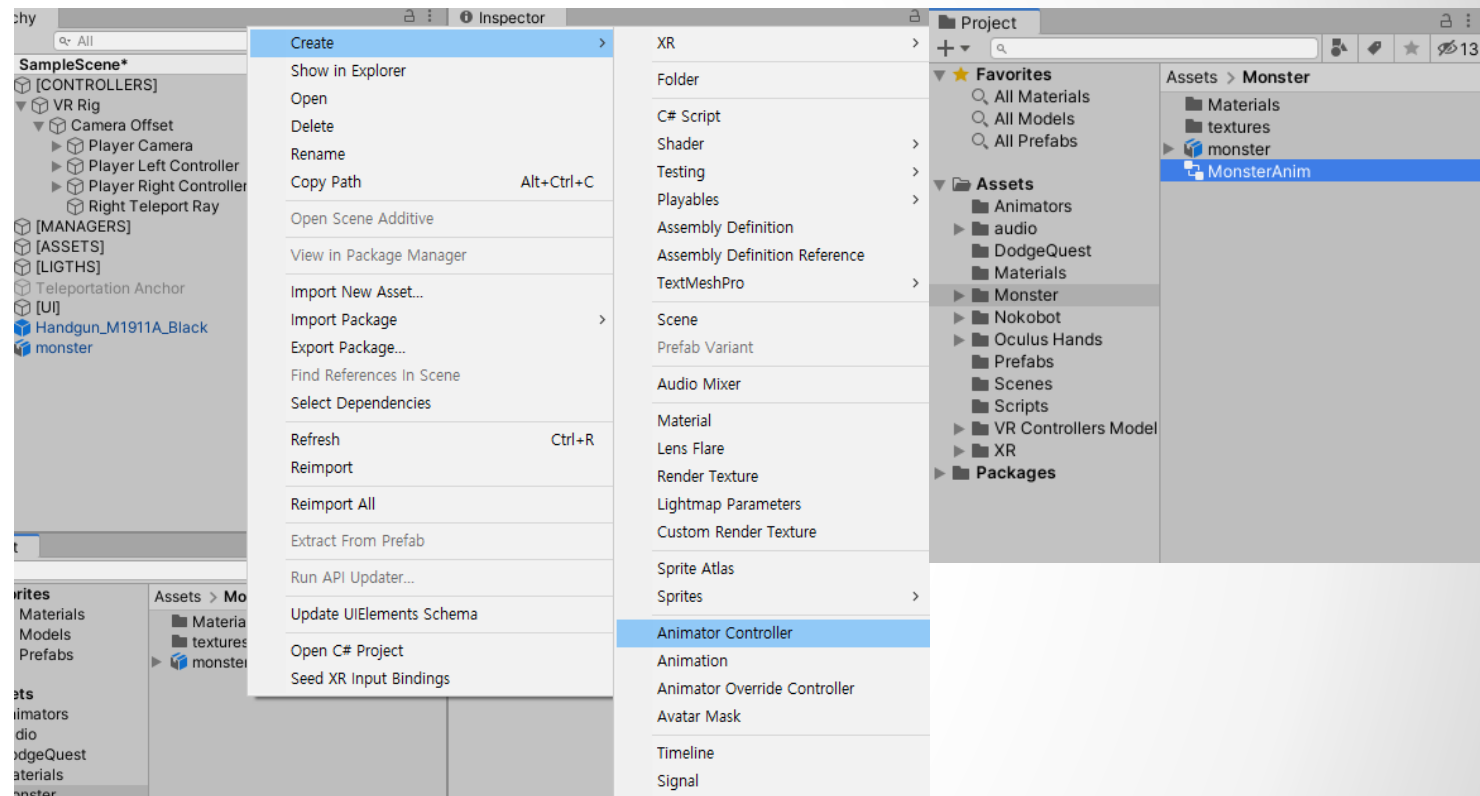


# 외계인 몬스터와 전투

## ■ 애니메이션 컨트롤러

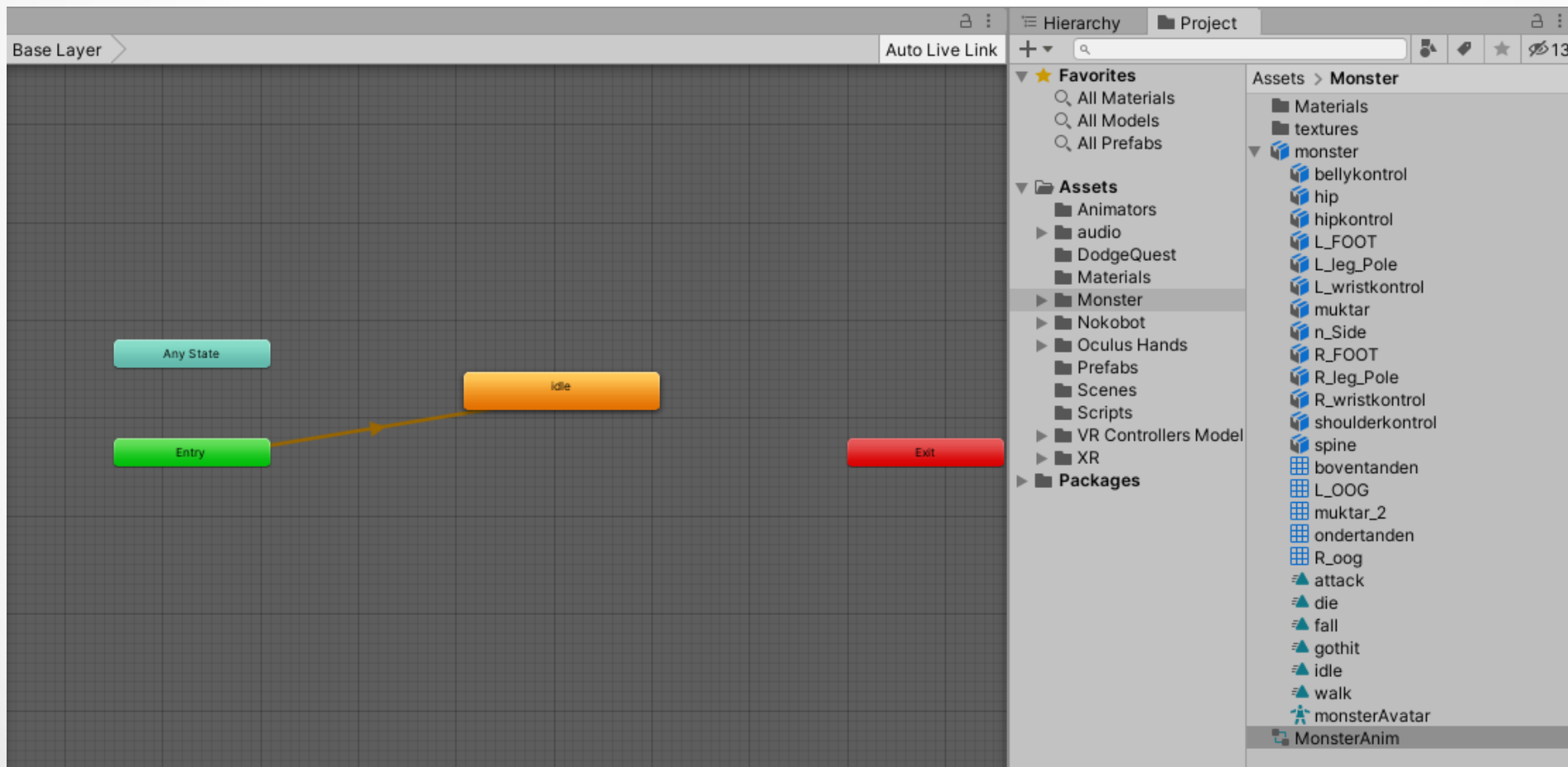
▶ 메카닉을 적용한 모델에 애니메이션을 적용하고, 특정 조건에 따라 다른 애니메이션 상태로 전이하는 규칙을 설계할 수 있다.

▶ Monster 폴더 아래에 Animation Controller를 생성하고 이름을 MonsterAnim으로 변경 한 후에 더블 클릭



# 외계인 몬스터와 전투

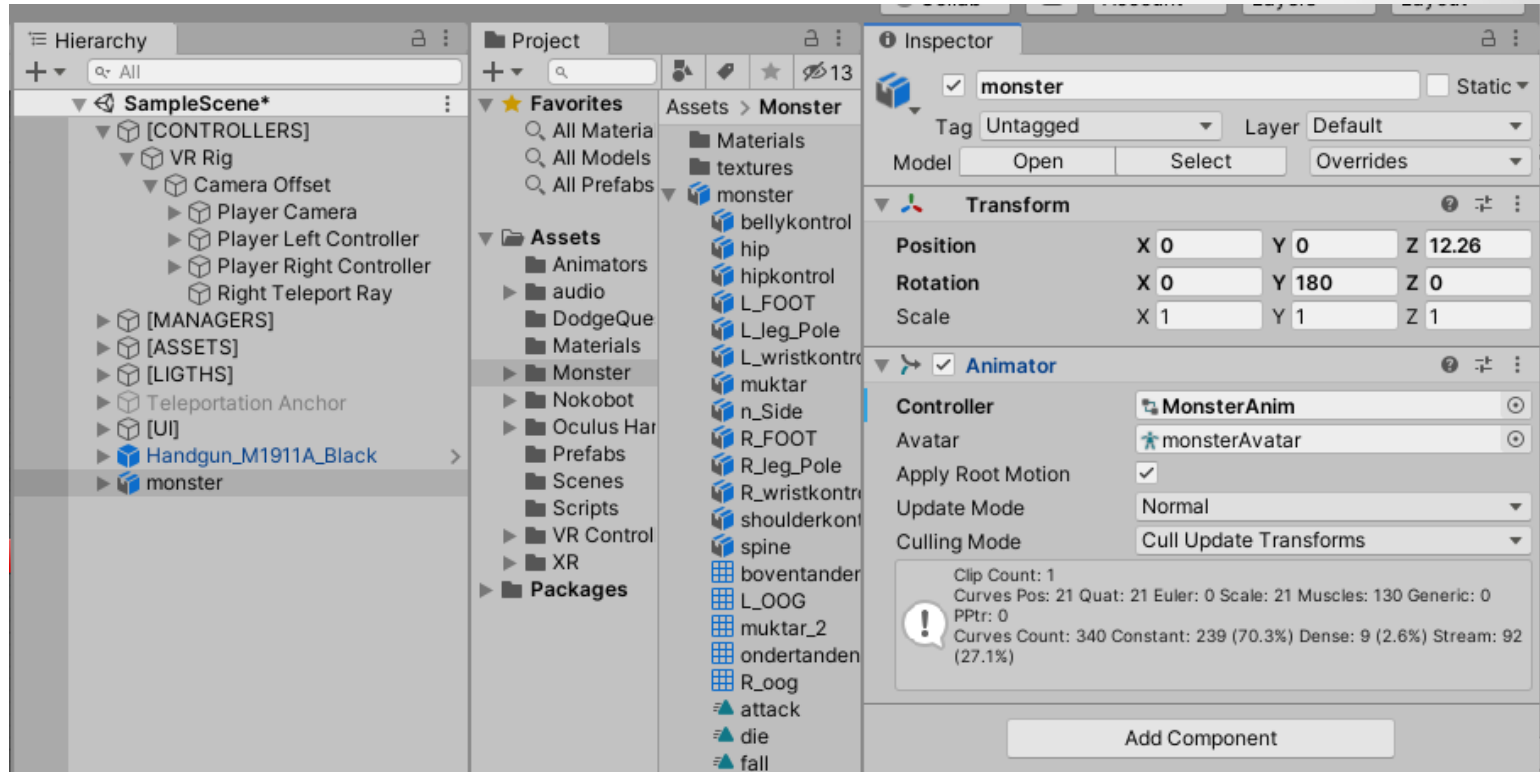
- 초기 애니메이션 상태 지정 : idle state



# 외계인 몬스터와 전투

## ■ Monster의 Animator 컴포넌트의 Controller 속성에 MonsterAnim 컨트롤러를 연결

- ▶ 현재 idle상태로 초기화 됨
- ▶ 실행하면 idle상태바가 움직이는것을 확인 할 수 있음



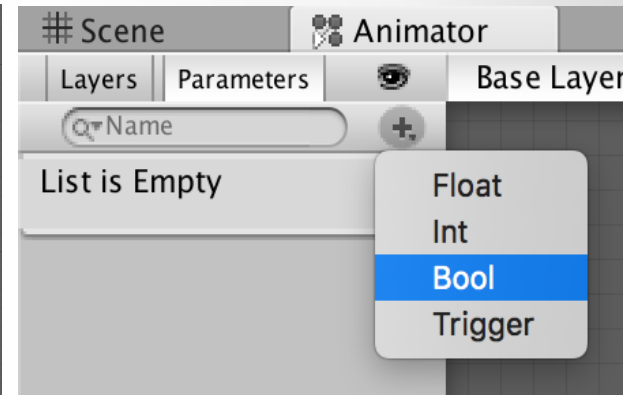
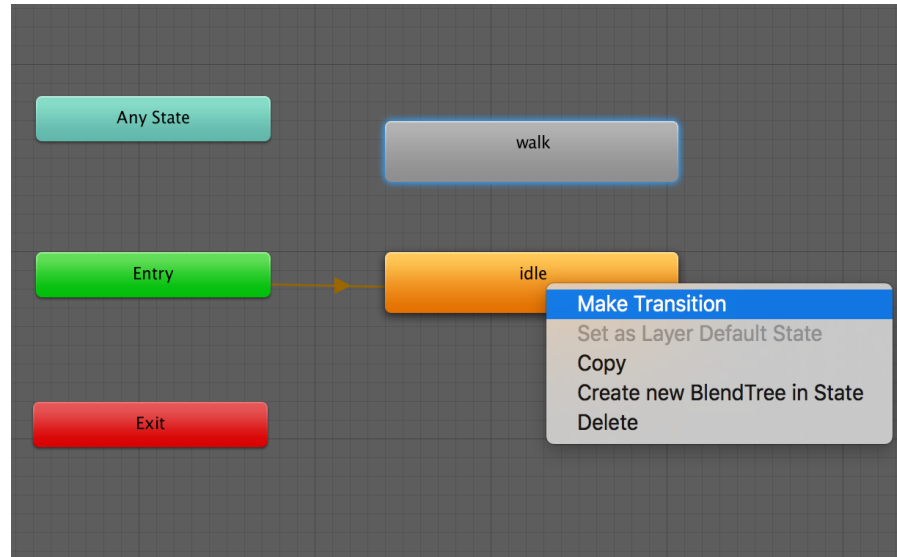
# 외계인 몬스터와 전투

## ■ 상태 전이 (idle -> Walk) 구현

▶ walk 상태 노드를 위치시킨 후 idle 노드에서 오른쪽 마우스 클릭, Make Transition 선택

▶ 왼쪽의 Parameter 탭에서 Bool 타입의 변수 생성(변수명 ; IsTrace)

- IsTrace (true) : idle -> walk 로 전이
- IsTrace (false) : walk -> idle 로 전이

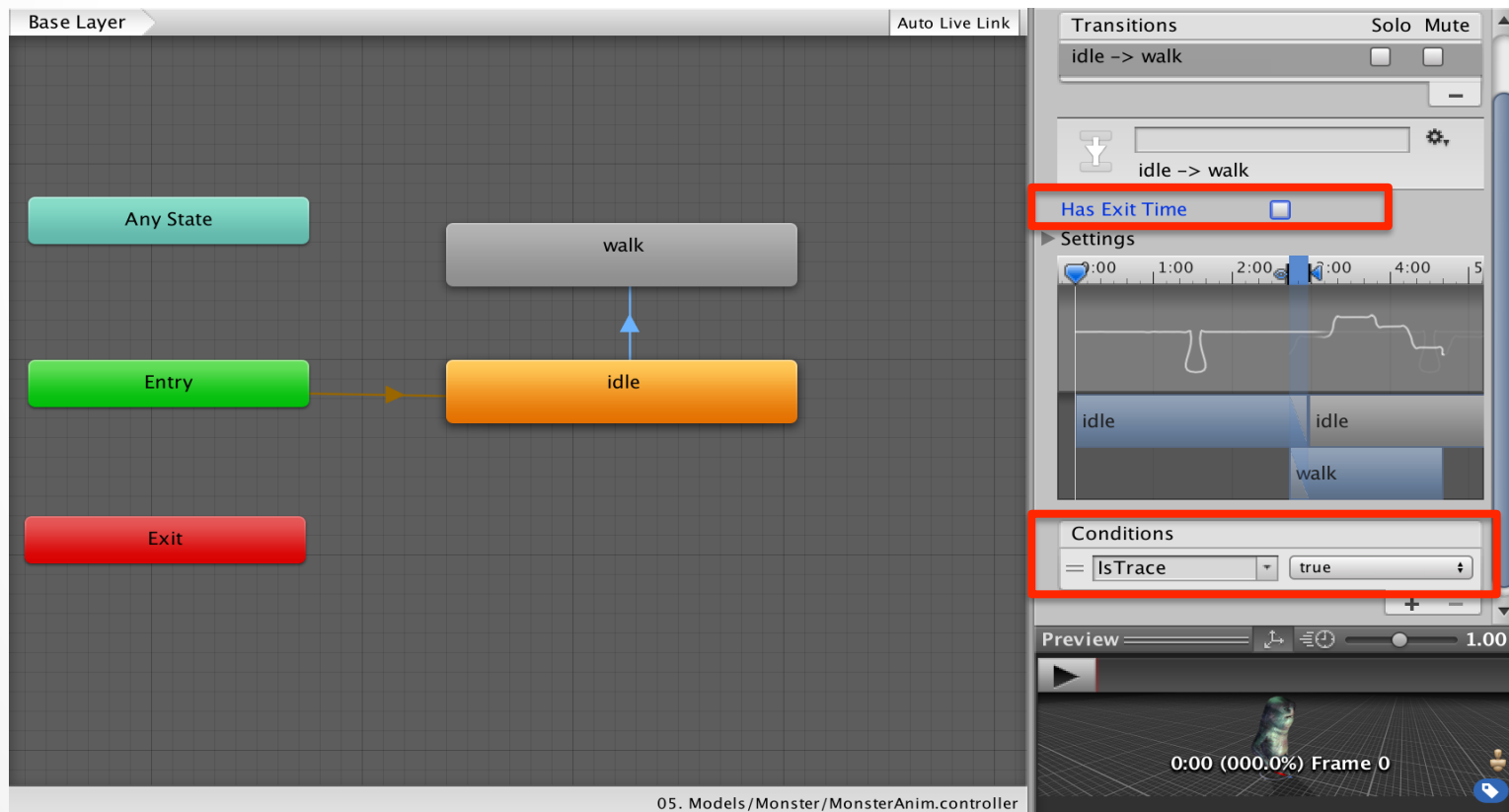


# 외계인 몬스터와 전투

## ■ Has Exit Time 속성

▶ 속성이 체크되면 즉각적인 상태 전이가 일어나지 않고 애니메이션이 다 끝난 다음 다음 상태로 변경 됨, 화살표를 클릭하면 Inspector창에서 확인 가능함

## ■ Conditions속성에서 IsTrace 변수에 true로 설정



# 외계인 몬스터와 전투

- 상태 전이 (Walk->idle) 구현

The image displays the Unity Animator interface for a monster character. The main workspace shows a state machine with three states: 'Any State' (teal), 'walk' (grey), and 'idle' (orange). A green 'Entry' state points to the 'idle' state. A red 'Exit' state is also present. A transition arrow points from the 'walk' state to the 'idle' state. The right-hand panel shows the 'Transitions' list with 'walk -> idle' selected. Below this, the 'Settings' section is visible, featuring a timeline and a 'Conditions' section. The 'Conditions' section is highlighted with a red box, showing the condition 'IsTrace' set to 'false'. The 'Preview' window at the bottom shows a 3D view of the monster character in a grid environment, with a timestamp of '0:00 (000.0%) Frame 0'.

Base Layer Auto Live Link

Any State

walk

idle

Entry

Exit

Transitions

walk -> idle

Has Exit Time

Settings

Conditions

= IsTrace false

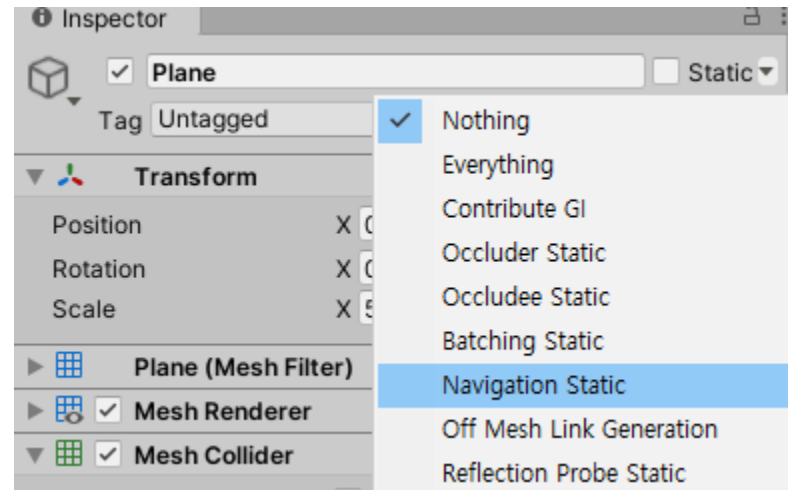
Preview

0:00 (000.0%) Frame 0

05. Models/Monster/MonsterAnim.controller

# 외계인 몬스터와 전투

- 몬스터가 플레이어를 쫓아 오려면?
- 네비게이션 기능
  - ▶ 기존의 길찾기 알고리즘(Path Finding, A\*)을 대체, 유니티에서 독자적으로 제공
  - ▶ 3D 메쉬정보를 분석해 걸어갈수 있는 영역과 지나갈 수 없는 영역으로 구분한 정보를 바탕으로 런타임시 그 정보를 활용하여 최단거리를 계산해 추적할 수 있게 하는 기능
- 네비게이션 설정 및 베이킹 (미리 만들어 놓음)
  - ▶ 네비게이션을 베이킹 하려면 Floor의 Static 옵션을 Navigation Static으로 바꿈(모두 static 으로 바꾸는 것이 아님)

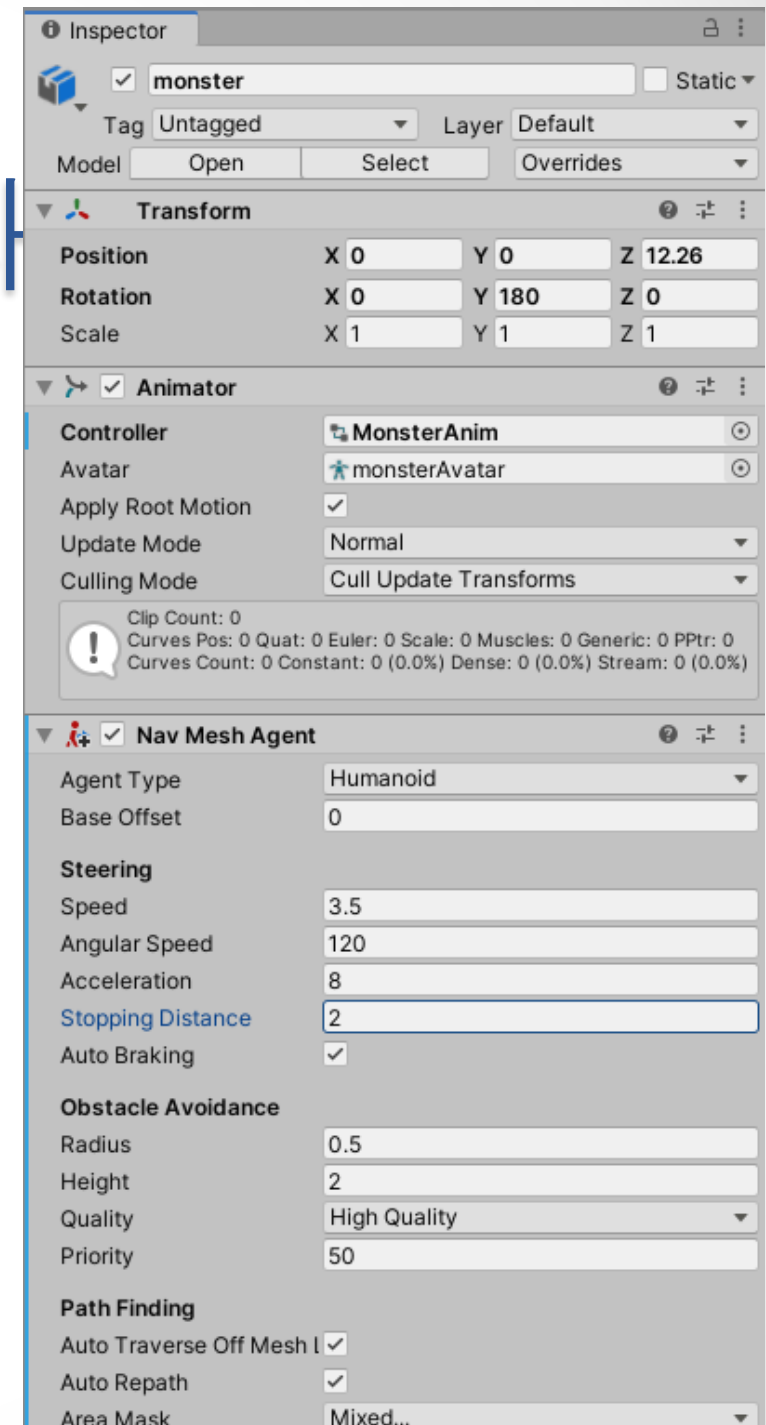




# 외계인 몬스터와

## ■ Nav Mesh Agent 컴포넌트

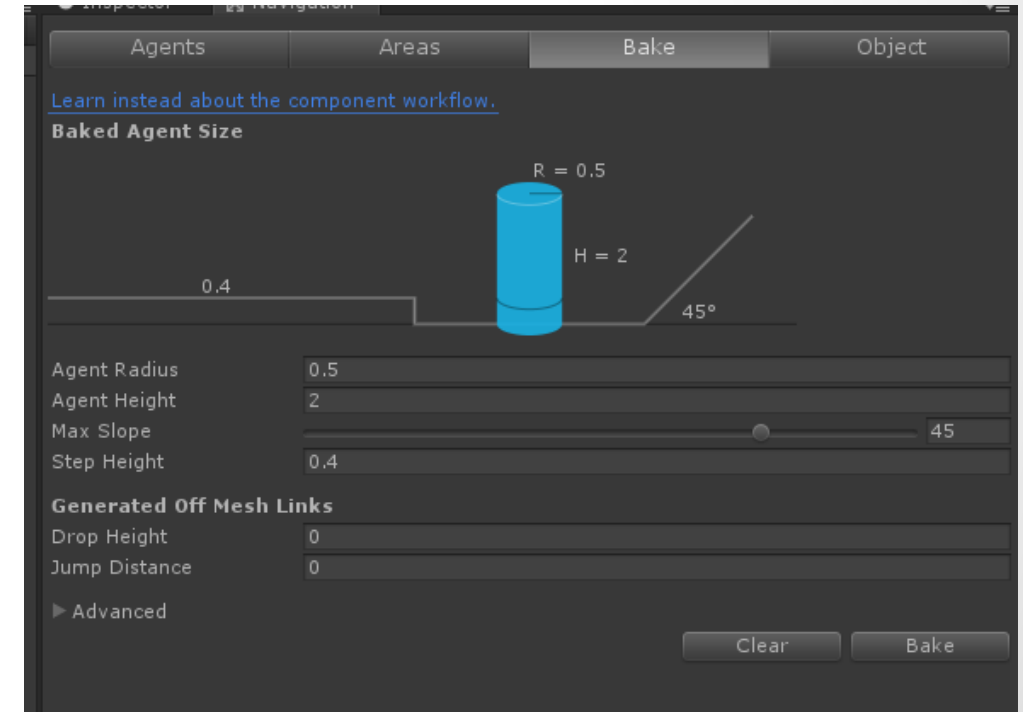
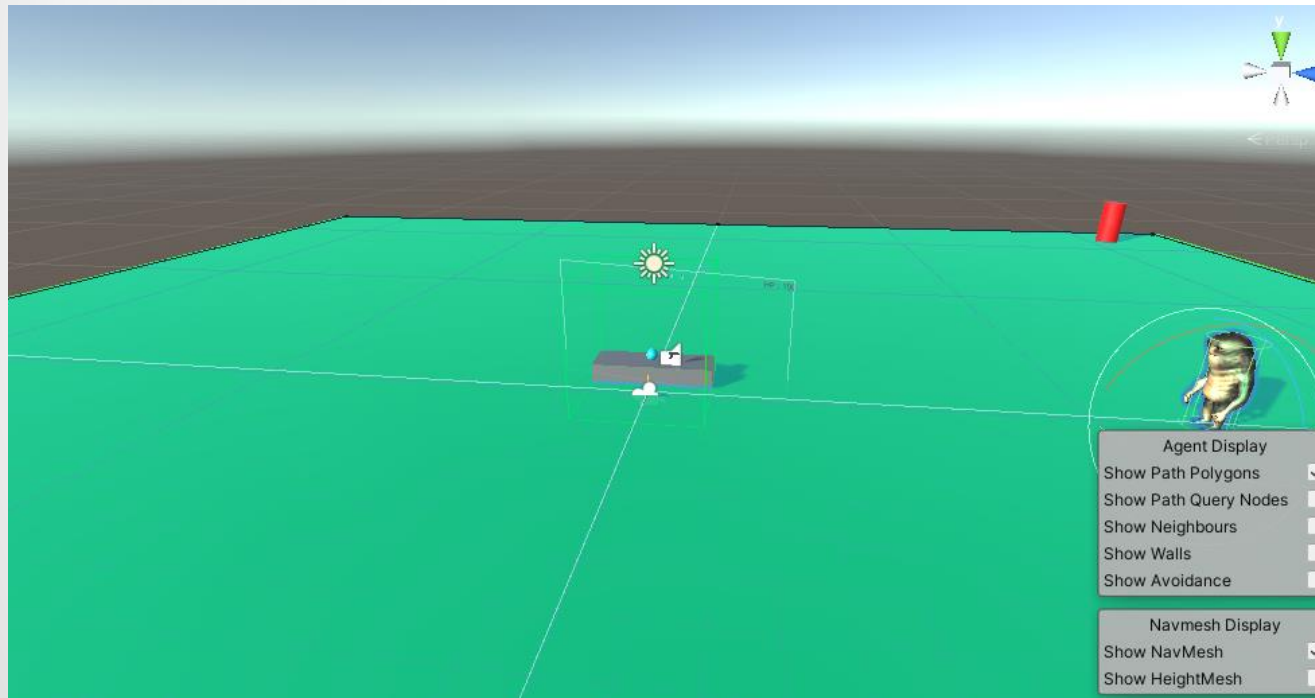
- ▶ 실제로 플레이어를 추적할 수 있게 하는 컴포넌트
- ▶ 베이킹 된 네비게이션 메쉬를 분석하여 장애물을 회피하고 목표물까지 최단거리를 계산하고 추적하게 할 수 있다.
- ▶ 메뉴-Component-Navigation-Nav Mesh Agent 를 선택한 후 Monster에 적용
- ▶ Stopping Distance
  - 추적하다가 정지하는 근접거리, 플레이어와의 거리, 0 이면 플레이어와 겹치게 됨



# 외계인 몬스터와 전투

## ■ 네비게이션 뷰 사용

- ▶ 윈도우-AI-Navigation 선택
- ▶ Bake를 꼭 해야 함!



■ **MonsterCtrl.cs** 스크립트  
생성

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.AI;

public class MonsterCtrl : MonoBehaviour {
    private Transform monsterTr;
    private Transform playerTr;
    private UnityEngine.AI.NavMeshAgent nvAgent;
    private Animator animator;

    void Start () {
        monsterTr = this.gameObject.GetComponent<Transform>();
        playerTr = GameObject.FindWithTag( "Player ").GetComponent<Transform>();
        nvAgent = this.gameObject.GetComponent<UnityEngine.AI.NavMeshAgent>();
        animator = GetComponent<Animator>();
        nvAgent.destination = playerTr.position;
    }

    void Update()
    {
        nvAgent.destination = playerTr.position;
        animator.SetBool("IsTrace", true);
    }
}
```

# 외계인

- 너무 빠르게 쫓아와서 피하기가 쉽지 않음!
- 로밍 거리 제한 필요

