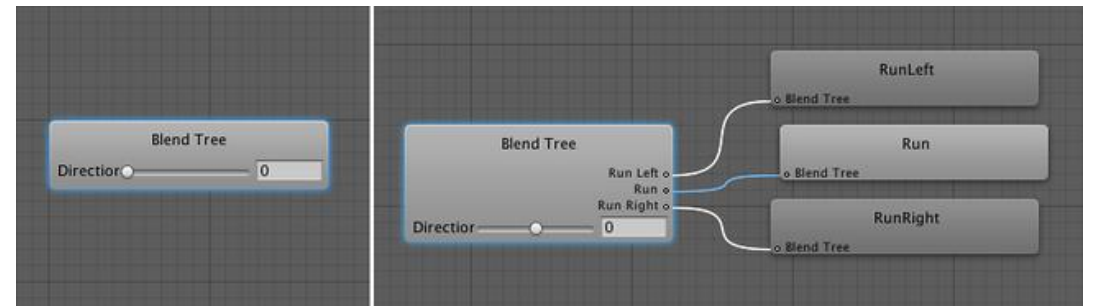
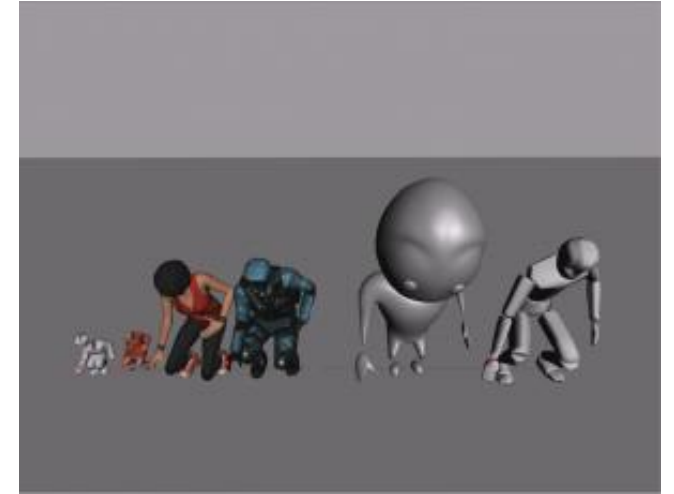


# 메카닉 애니메이션

---

# 메카닉 애니메이션 시스템

- 많은 기능을 상태머신 기반의 에디터를 이용하여 처리
- 모션 리타겟팅(Motion Retargeting)
  - 캐릭터에 적용된 애니메이션을 다른 캐릭터에 적용하는 기술
  - 기존 애니메이션을 아바타로 정규화 하여 애니메이션을 재 활용
  - 캐릭터의 종류가 다양할 경우 적은 애니메이션 클립으로 다양한 연출 가능
- 블렌드 트리
  - 다수의 트랜지션에 따른 기능을 자동으로 처리

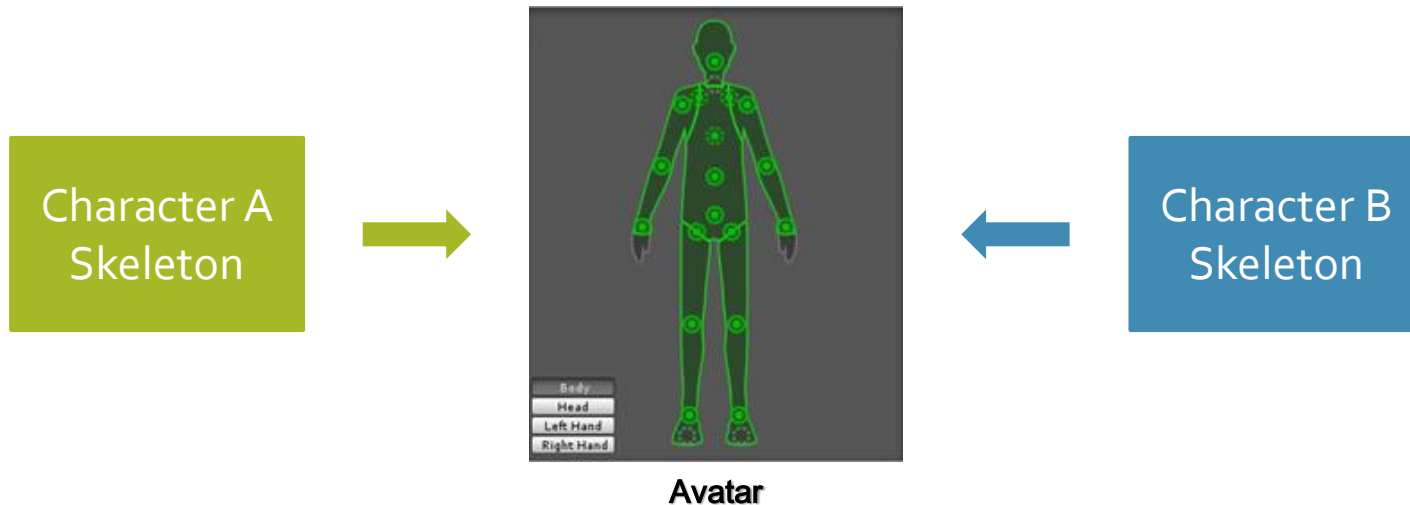


# 아바타 시스템

- 아바타(Avatar)

- 3d 프로그램에서 만든 Skeleton 데이터를 정규화 하기 위한 유니티 고유의 Skeleton
- 모션 리타겟팅을 위한 핵심 요소

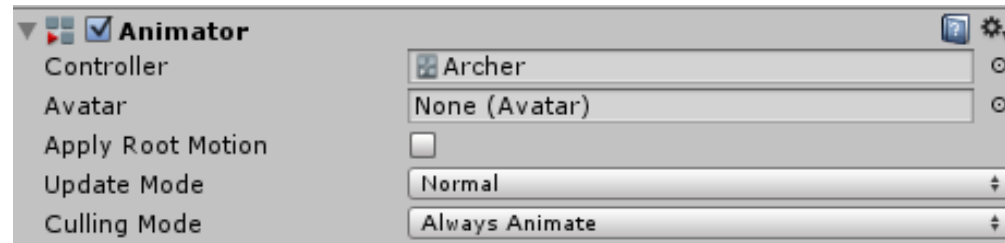
- 정규화가 끝난 캐릭터들은 애니메이션 클립의 공유 가능



# 메카닉 캐릭터 애니메이션 절차

- ① 캐릭터의 제작
- ② 상태머신 설계
- ③ 상태머신 구현
- ④ 스크립트 작성

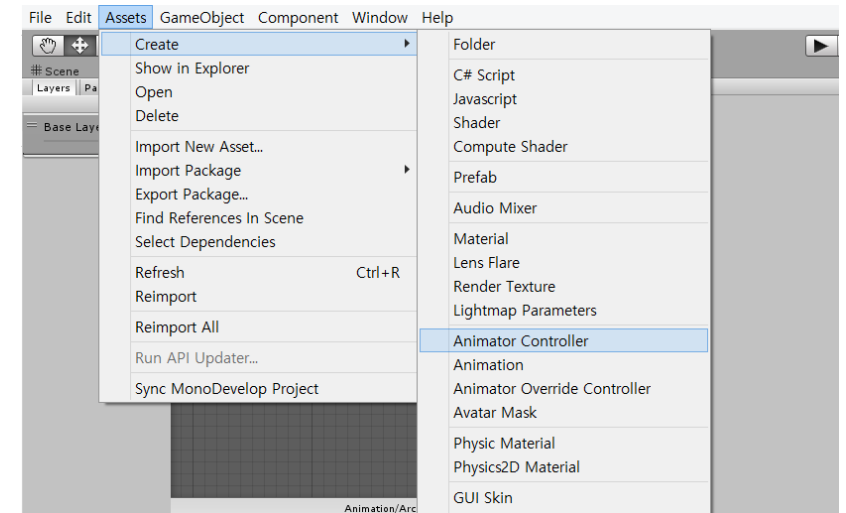
# 애니메이터 컴포넌트



- 애니메이터(Animator)
  - 게임 오브젝트가 애니메이션을 사용하기 위한 기능을 제공 하는 컴포넌트
  - 애니메이션 클립을 제어하기 위한 애니메이터 컨트롤러가 필요
  - 주요 설정 옵션
    - Controller – 게임 오브젝트의 애니메이션을 제어하기 위한 애니메이터 컨트롤러 설정
    - Avatar – 인간형 캐릭터를 사용할 경우 캐릭터의 Rigging 정보를 담은 데이터를 설정
    - Apply Root Motion – 애니메이션이나 스크립트로 부터 캐릭터의 위치와 회전 정보를 제어
    - Update Mode – 애니메이터를 갱신할 때 사용할 시간 척도를 선택
    - Culling Mode – 애니메이션들의 컬링 모드를 설정

# 애니메이터 컨트롤러

- 애니메이터 컨트롤러(Animator Controller)
  - 메카닉을 이용하여 애니메이션을 제어하기 위한 정보를 담고있는 애셋
  - 애니메이션 클립 간의 상태 전이 제어
  - 상태 전이 발생을 위한 매개변수 정의
  - 애니메이션 레이어 설정
- 애니메이션 컨트롤러 생성
  - 게임 오브젝트 선택 후 애니메이션 클립 생성 시 자동 생성
  - [메뉴 > Asset > Create > Animator Controller]
  - 프로젝트 뷰 빈 공간 우클릭 후 [Create > Animator Controller]

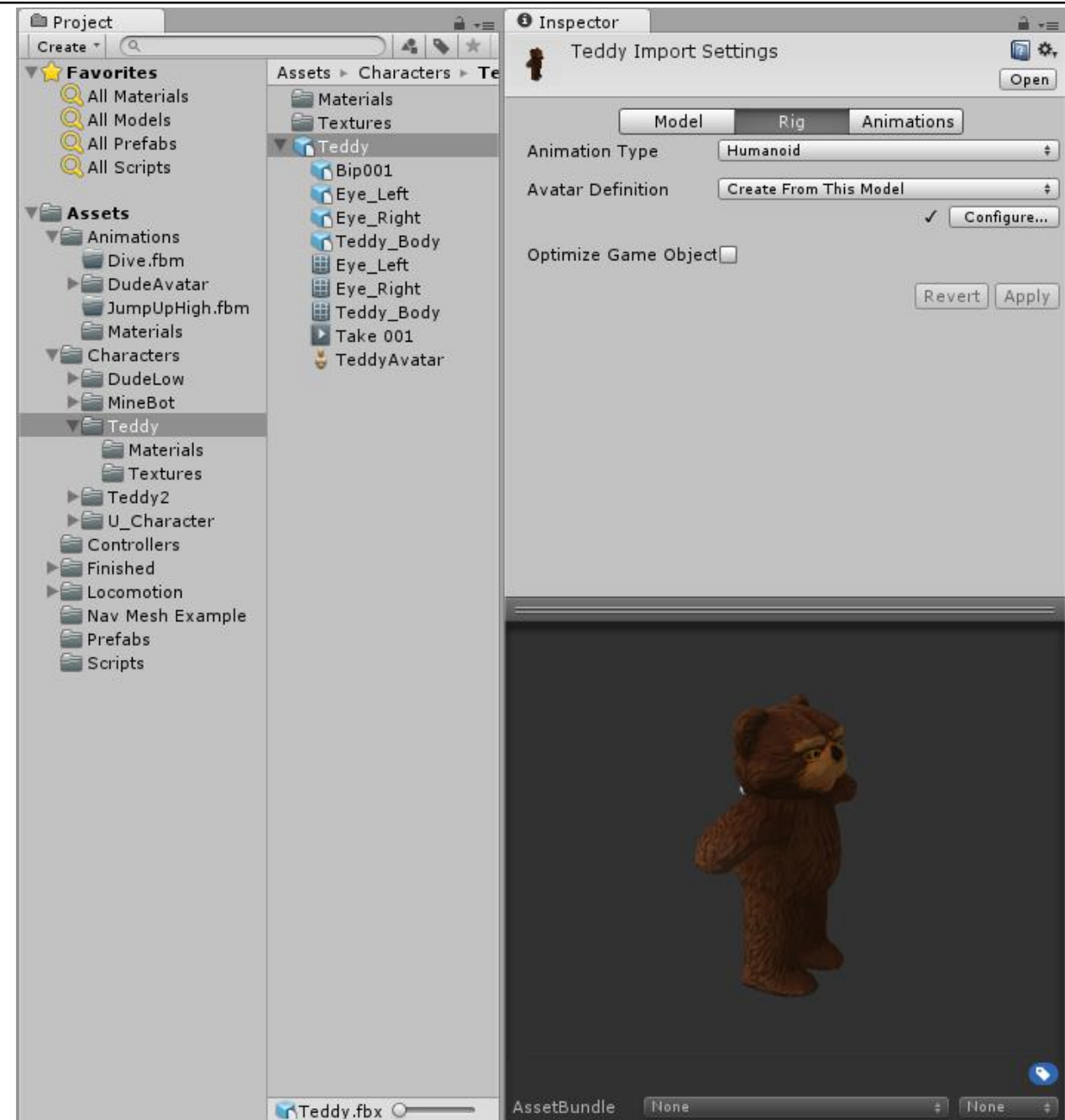


# 캐릭터의 사용

- FBX의 관절 구조를 읽는 방식에 따라
  - Legacy 방식
    - 캐릭터를 제작할 때 사용했던 기본 관절 구조를 그대로 사용
    - 메카닉 적용 불가
  - **Humanoid 방식**
    - 관절 구조를 유니티에서 제공하는 Humanoid 기반의 관절 구조와 연동시켜 사용
    - 메카닉 적용가능
      - 상태 머신, 모션 리타게팅

# FBX 파일 살펴보기

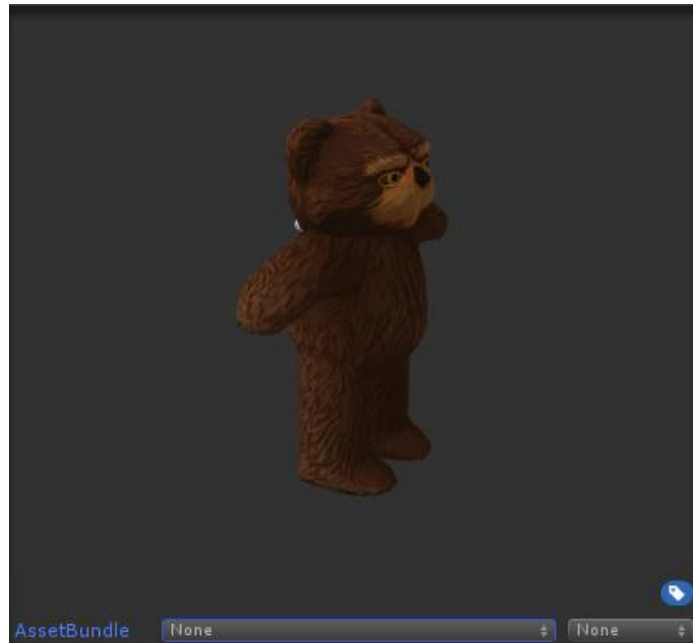
- FBX 캐릭터 데이터의 구성
  - Model
  - Rig
  - Animations





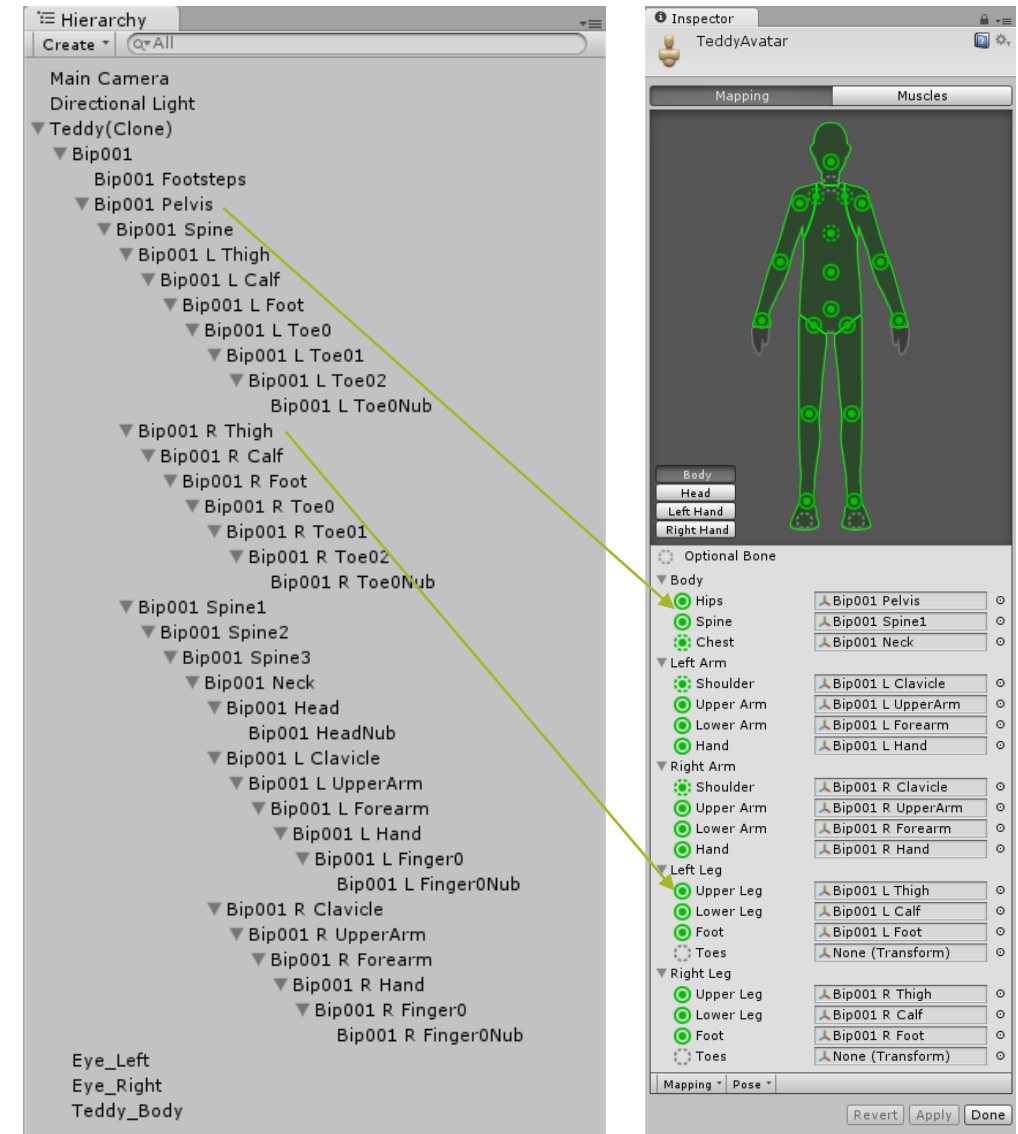
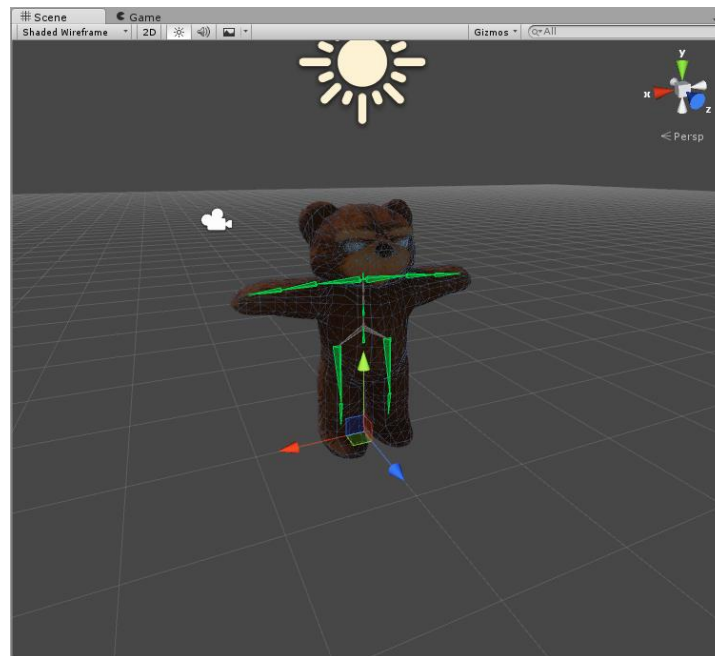
# Model

- 캐릭터의 외형을 구성하는 메쉬, 노멀, 재질 등의 요소를 설정
  - Scale Factor – 캐릭터의 모델의 크기 설정



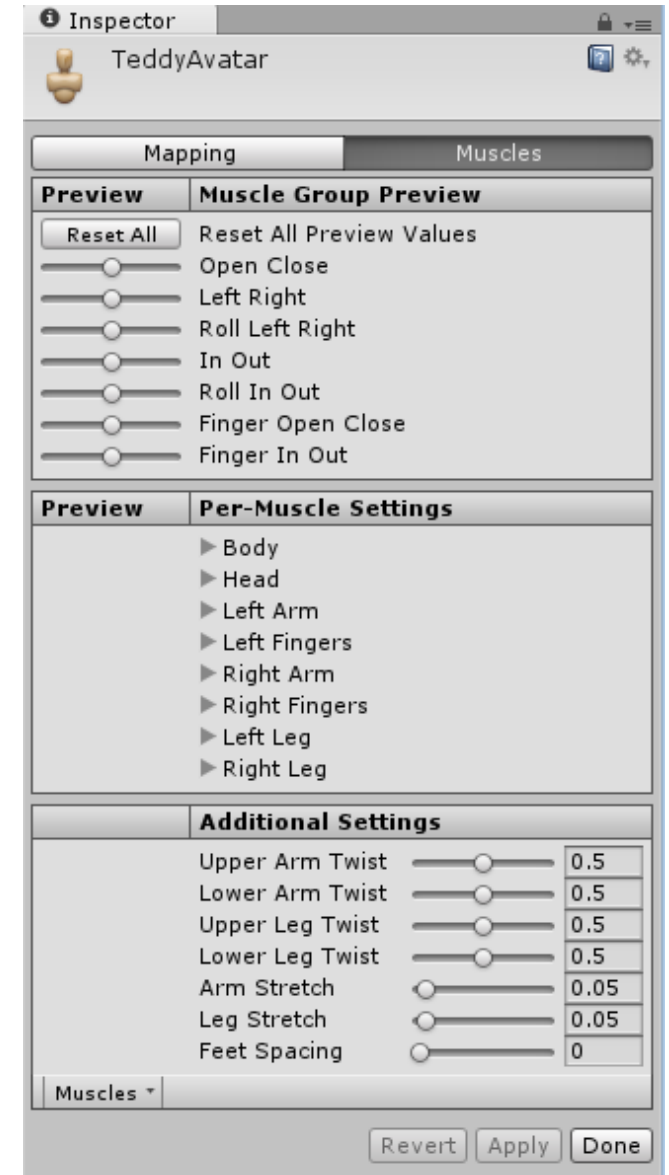
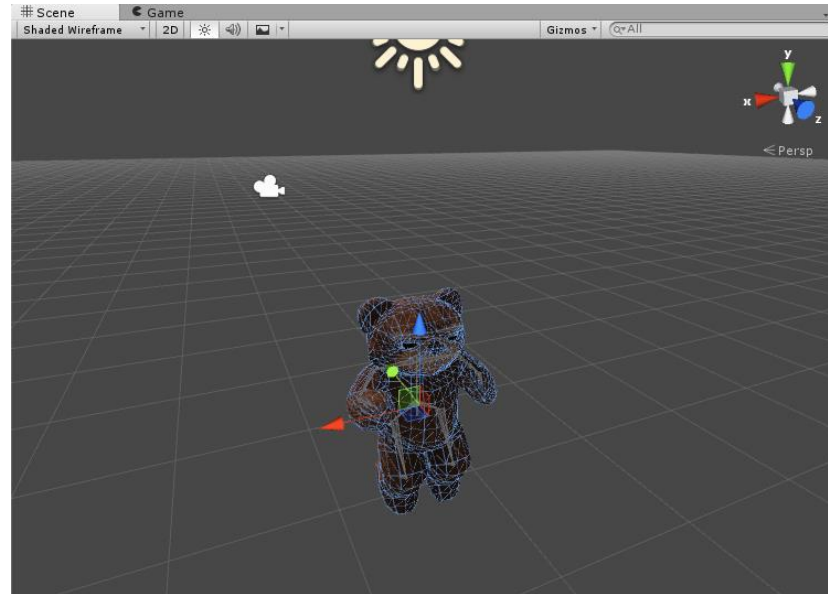
# Rig

- Mapping
  - FBX 파일의 캐릭터 관절 구조를 Unity의 Avatar에 연결



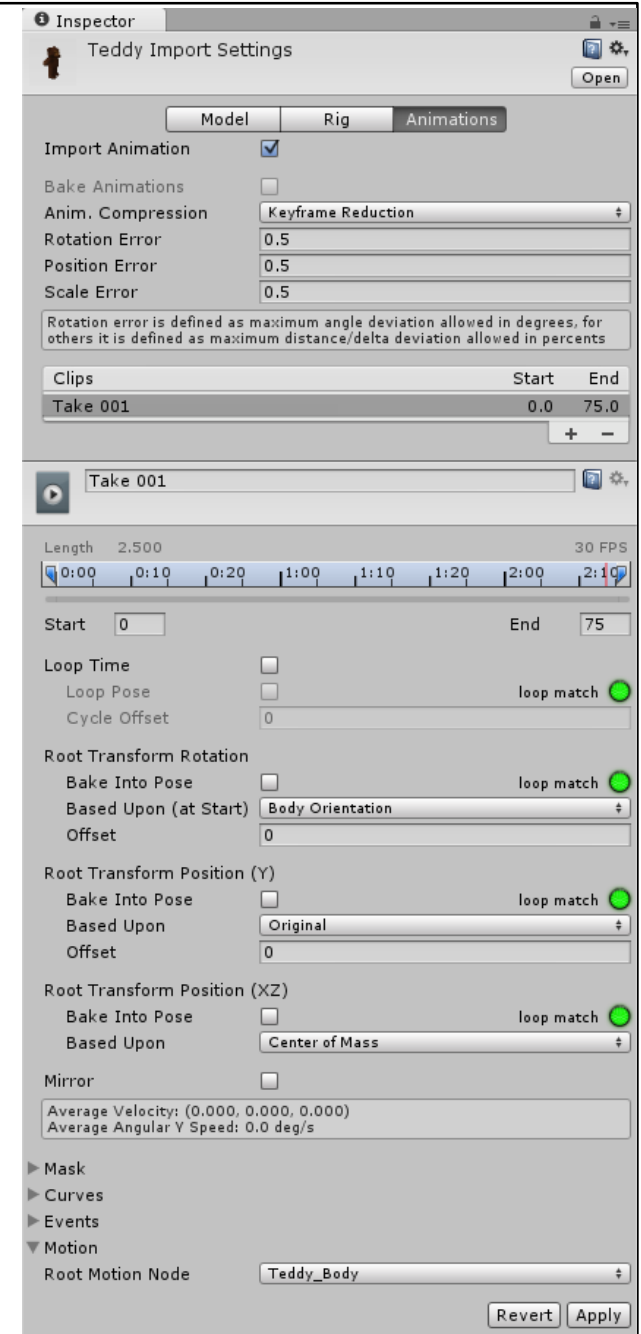
# Rig

- Muscles
  - 맵핑 후 아바타 시스템에 의해 캐릭터가 Rigging 상태 확인



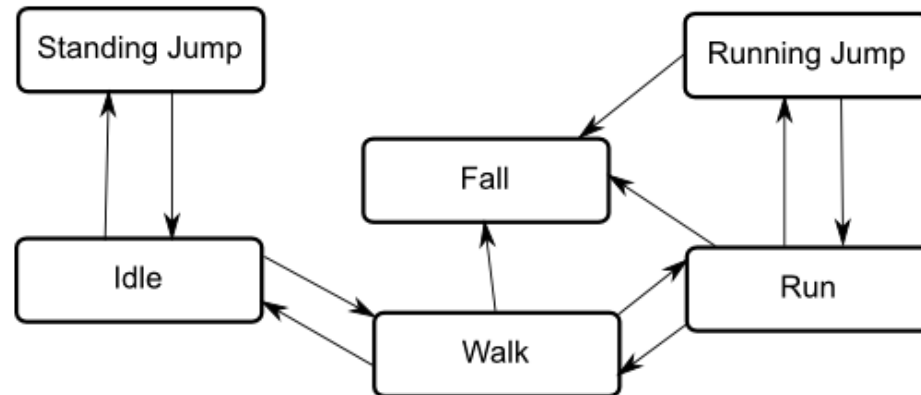
# Animation

- 애니메이션 동작 관련 데이터 설정
  - 애니메이션 클립 추가 삭제
  - 마스크 수정
  - 이벤트 추가
  - 루트 모션 설정



# 상태 전이 다이어그램

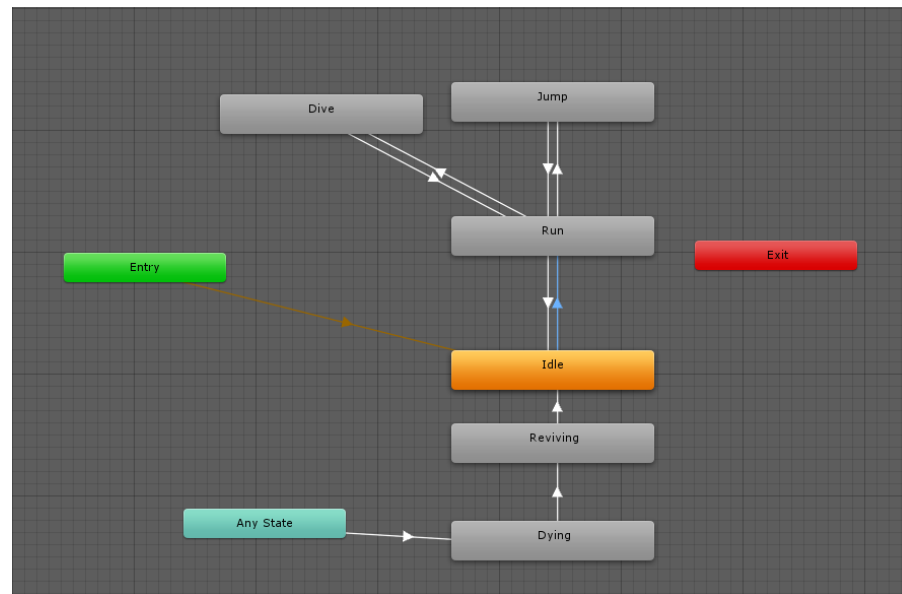
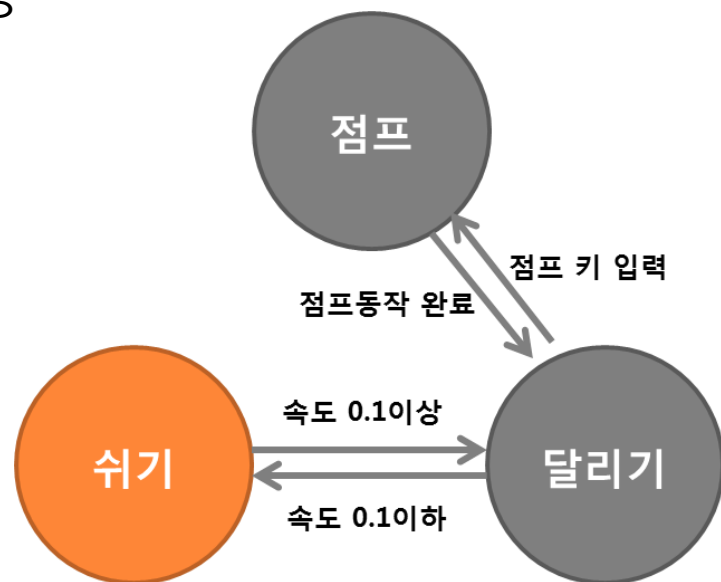
- 동작들 사이의 상태 변화를 표현



# 상태머신의 구현

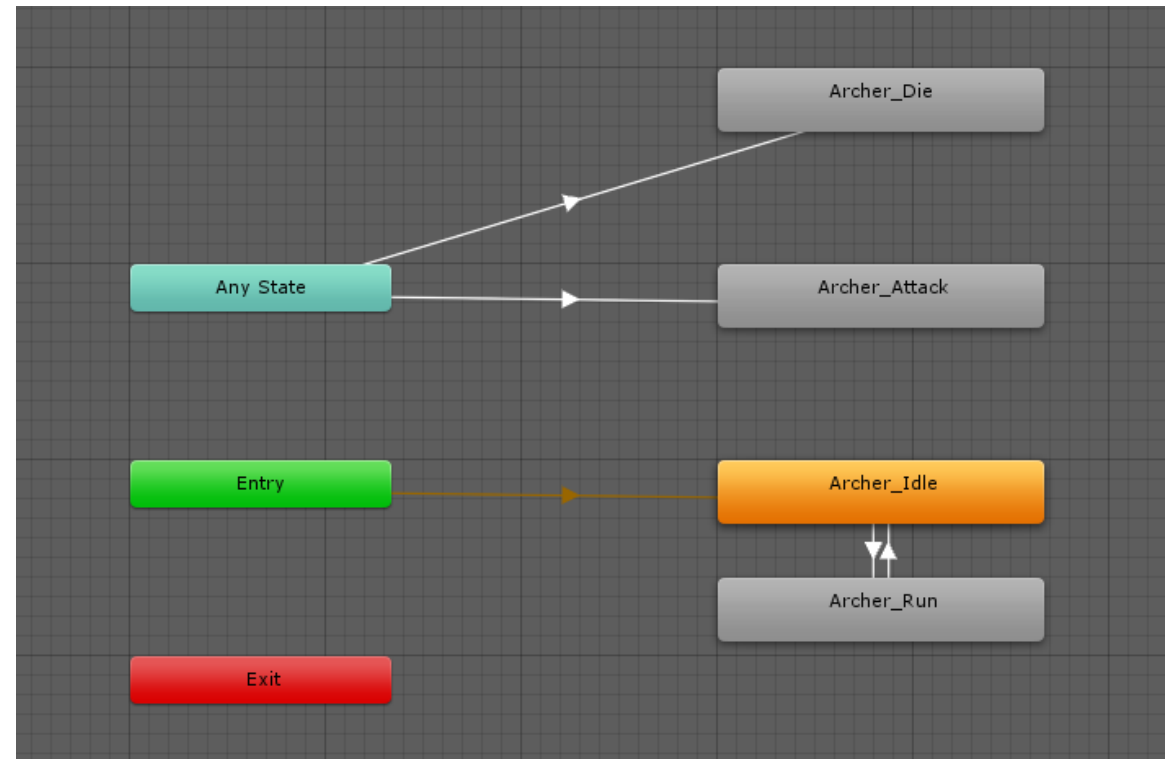
- 컨트롤러 작성

- 노드 생성
- 에지 생성
- 조건 정의



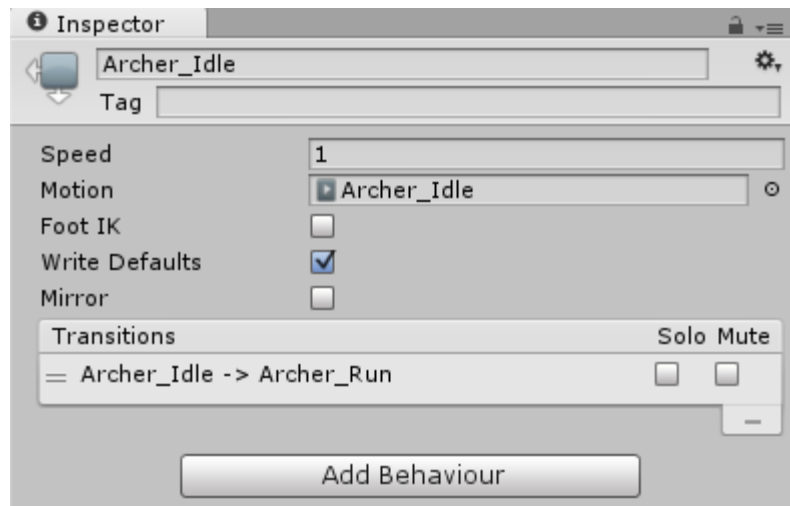
# 상태 전이 다이어그램의 요소

- 애니메이션 상태
  - Entry – 애니메이션 시작 상태
  - Exit – 애니메이션 종료 상태
  - Any State – 애니메이션 재생 상태
  - 주황색 노드 – 최초 재생 애니메이션 상태
  - 회색 노드 – 일반 애니메이션 상태



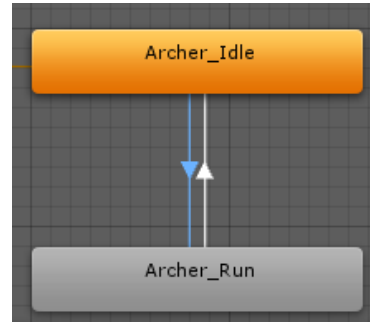
# 애니메이션 상태

- 애니메이션 상태(Animation States)
- 주요 설정 옵션
  - Speed – 애니메이션 클립의 재생속도 초기 값
  - Motion – 현재 상태에 재생할 애니메이션 클립
  - Foot IK – 발의 Inverse Kinematics 적용 여부
  - Write Defaults – 애니메이션 클립 속성들의 초기값 사용 여부
  - Mirror – 애니메이션 클립의 좌우 대칭
  - Transitions – 현재 상태에서부터 시작하는 전이들
  - Add Behaviour - 애니메이션 재생 도중 발생하는 이벤트들을 처리하는 스크립트 설정





# 상태 전이



- 상태 전이(State Transition)

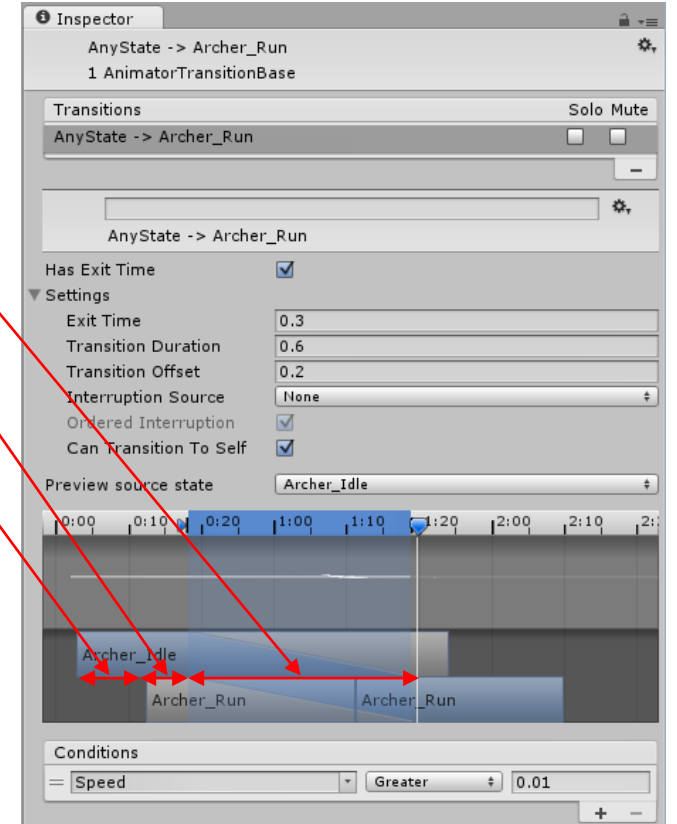
- 한 애니메이션에서 다른 애니메이션으로 바뀌는 과정
- 상태 전이는 특정 조건을 만족해야 발생
- 주요 설정 옵션

- Has Exit Time – 애니메이션 클립이 특정 시간에 전이가 일어나도록 할 것인지 설정
- Exit Time – 전이가 일어나기 시작하는 시점
- Transition Duration – 전이가 일어나는 시간 간격
- Transition Offset – 전이 시점 보정
- Interruption Source – 전이가 일어나는 동안 다른 외부 요인에 영향을 받을 수 있는지 여부 설정
- Conditions – 상태 전이를 발생시키는 매개변수

Transition Duration

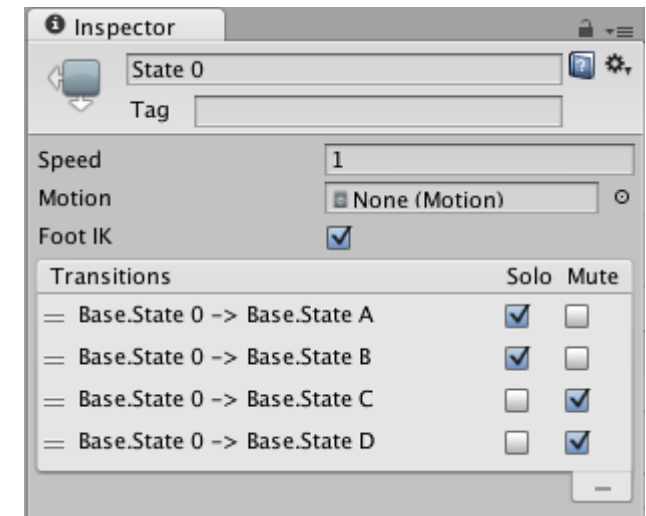
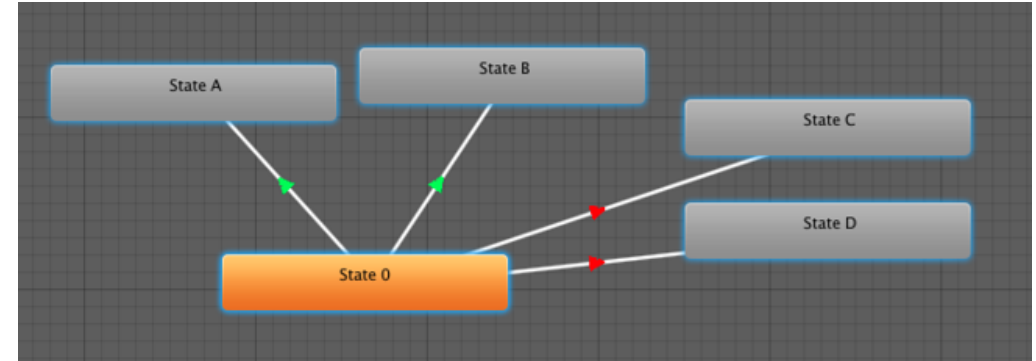
Transition Offset

Exit Time



# 상태 전이

- Solo & Mute
  - Solo
    - 해당 전이를 사용함
    - 여러 전이가 있을 경우 하나의 트랜지션만 Solo로 설정
  - Mute
    - 해당 전이를 사용하지 않음
- 선택 시 화살표 색상 Solo(녹색), Mute(빨간색) 표시
- 둘다 활성화 될 경우 Mute 수행



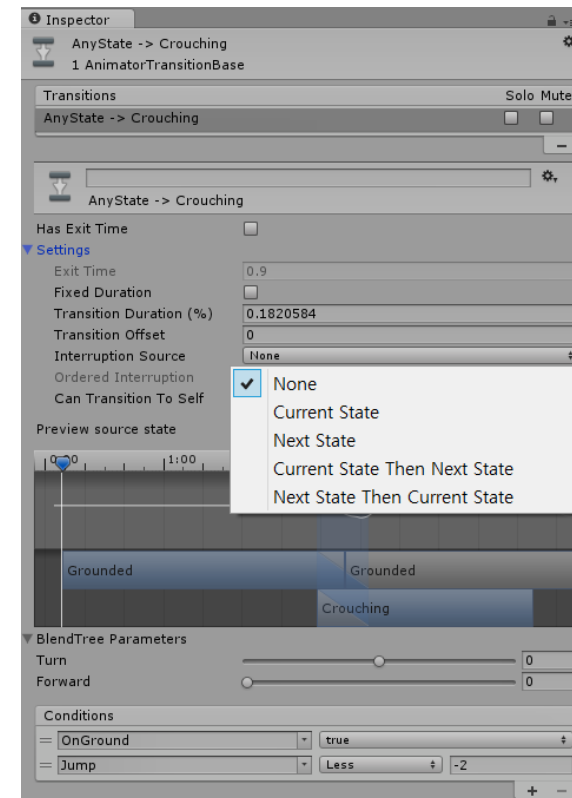
# 상태 전이

- **Has Exit Time 설정**

- 전이 조건을 충족하더라도 현재 재생 중인 애니메이션을 모두 재생한 후 다음으로 넘어감

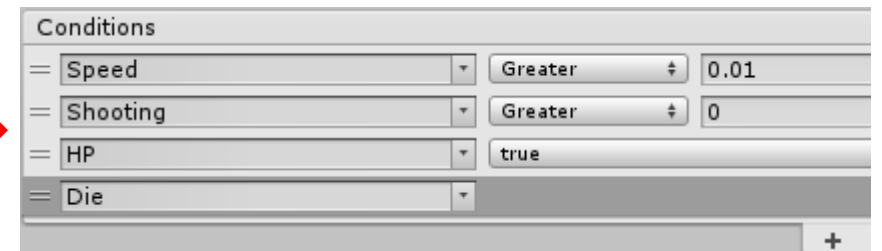
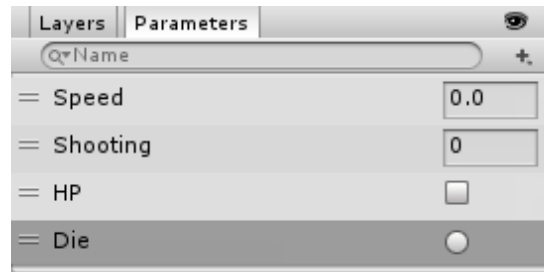
- **Interruption Source 설정**

- 트랜지션이 일어나는 도중 다른 트랜지션에 의해 해당 트랜지션의 중단 여부를 설정
- None : 해당 전이는 어떠한 상황에서도 간섭을 받지 않음
- Current State : 해당 전이는 현재 노드에서 다른 트랜지션에 의해 중단 될 수 있다.
- Next State : 해당 전이는 다음 노드 재생 중 다른 트랜지션에 의해 중단 될 수 있다.
- Current State Then Next State : 해당 트랜지션은 모든 다른 트랜지션에 의해 중단될 수 있다. 단 현재 노드와 연결된 전이가 우선 발생한다.
- Next State Then Current State : 해당 트랜지션은 모든 다른 트랜지션에 의해 중단될 수 있다. 단 다음 노드와 연결된 전이가 우선 발생한다



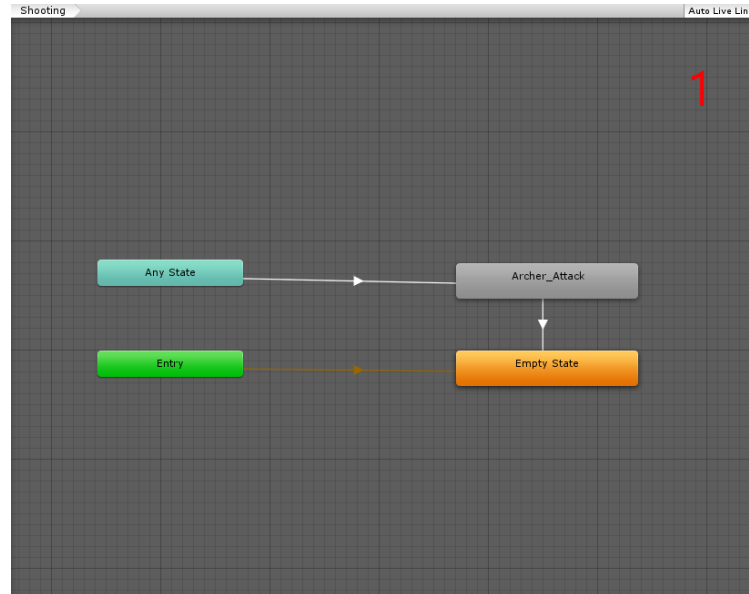
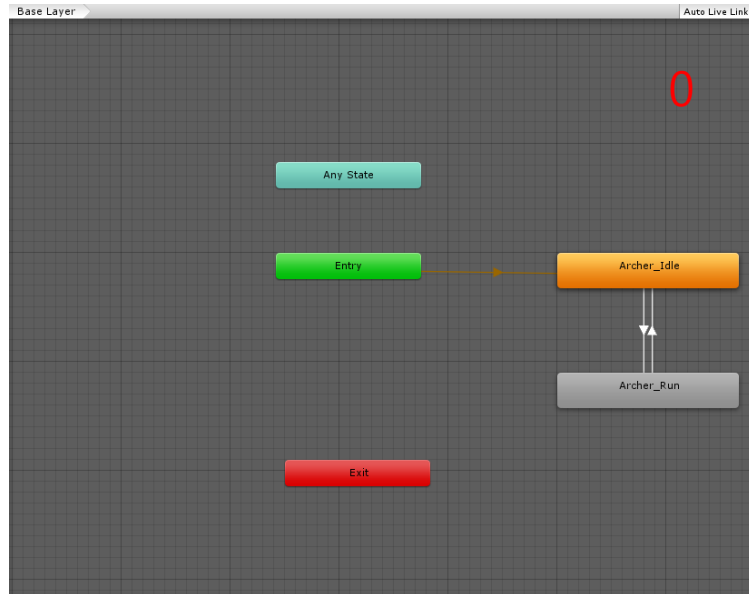
# 상태 전이 매개변수 정의

- 상태 전이를 발생시키는 조건 변수
  - 스크립트에서 설정한 매개변수 값을 변경하여 상태 전이를 발생시킴
- 매개변수 유형
  - Float – Greater or Less
  - Int – Greater or Less
  - Bool – true or false
  - Trigger

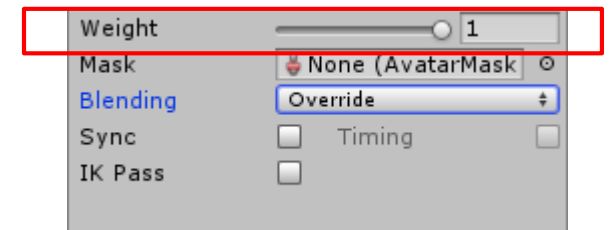


# 애니메이션 레이어

- 3D 캐릭터에서 부위 별로 다른 애니메이션을 적용하거나 애니메이션 상태에 우선 순위를 주어 보다 정교한 애니메이션을 생성한다.

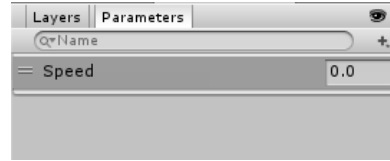


가중치가 높은 레이어가 동작

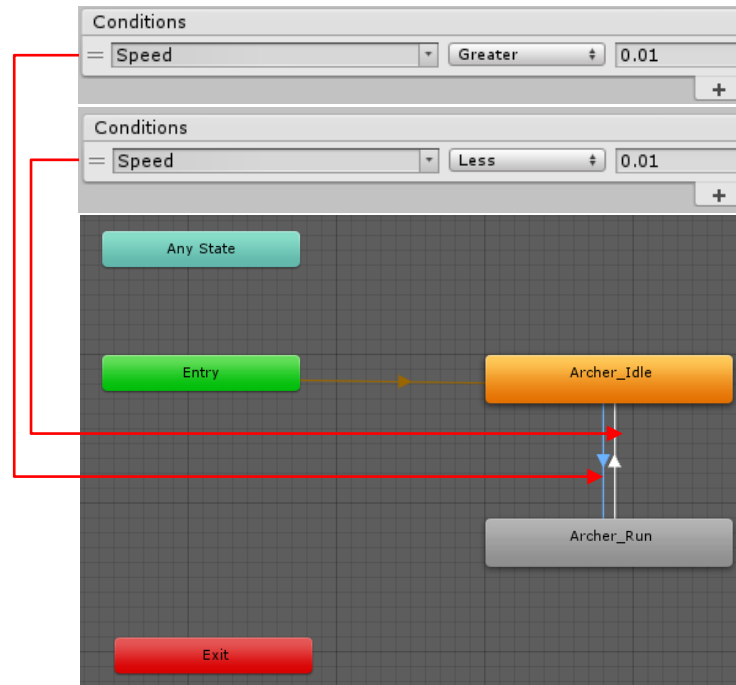


# 스크립트의 작성

- 애니메이션 이동 매개변수 정의



- 전이 조건 설정



- 스크립트 작성

```
public class ArcherControl : MonoBehaviour {  
    private Animator mAnimator;  
  
    public enum Status  
    {  
        Idle,  
        Run  
    }  
  
    public Status mStatus = Status.Idle;  
  
    void Start () {  
        mAnimator = gameObject.GetComponent<Animator> ();  
    }  
  
    void Update () {  
        float speed = Mathf.Abs (Input.GetAxis ("Horizontal"));  
        SetStatus(Status.Run, speed);  
    }  
  
    public void SetStatus(Status status, float param)  
    {  
        switch (status) {  
            case Status.Idle:  
                mAnimator.SetFloat ("Speed", 0.0f);  
                break;  
  
            case Status.Run:  
                mAnimator.SetFloat ("Speed", param);  
                break;  
        }  
    }  
}
```

# 실습(Mecanim Example Scenes)

- 애셋 스토어에서 다운로드(Ctrl+9)

