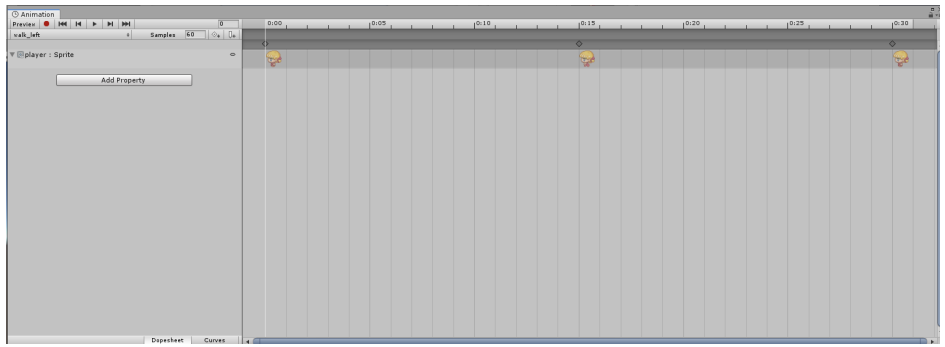
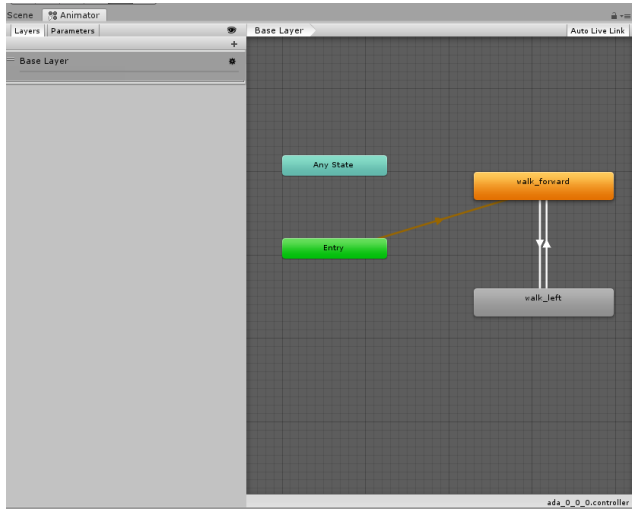
Unity3D Script – 2D Animation

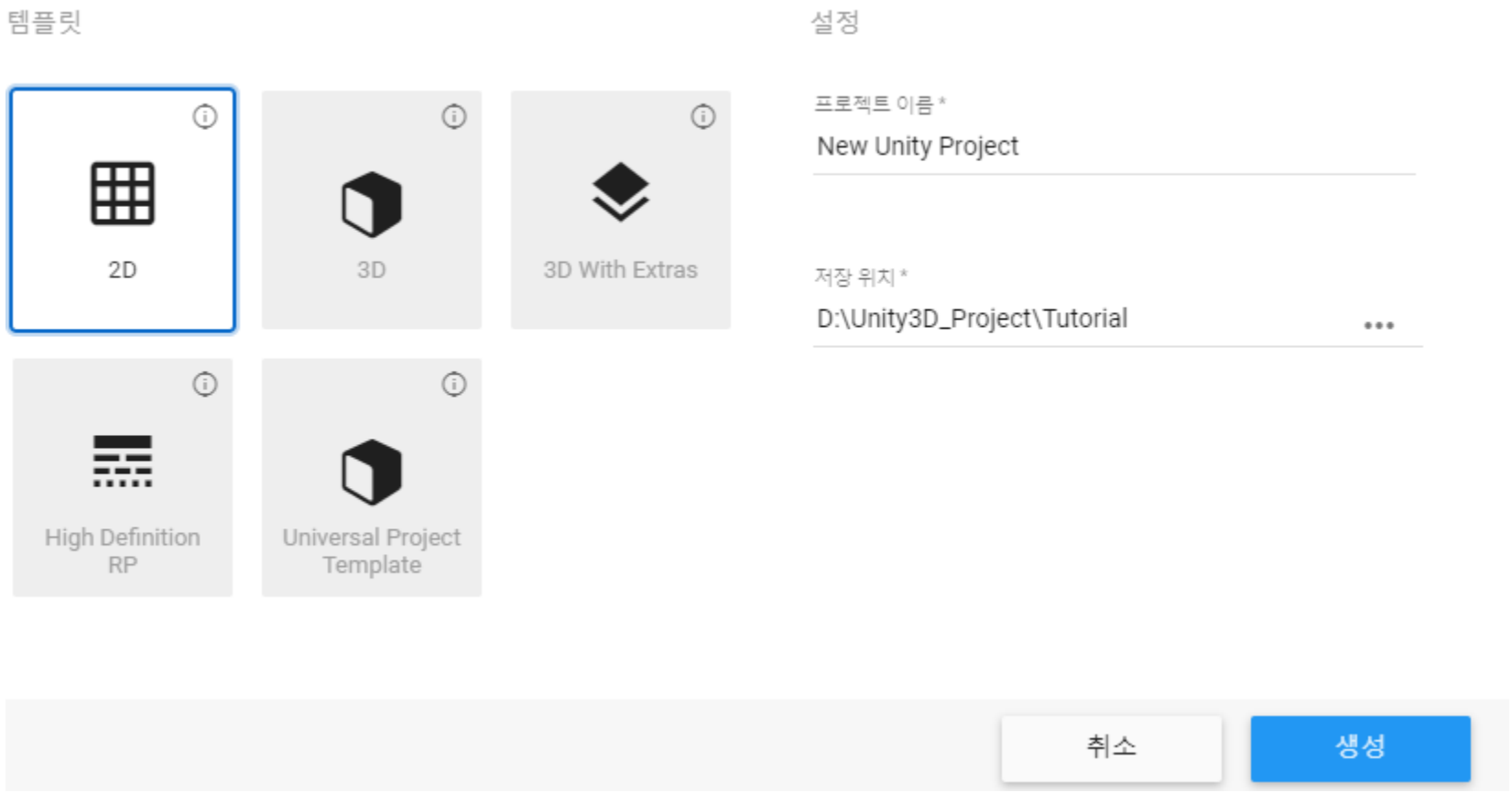
2D Sprite Process

- 소스 이미지 로드
- 스프라이트 생성 (Sprite Editor)
- anim 파일 생성 (Animation)
- controller 상태 전이도 생성 (Animator)
- 애니메이션 스크립트





2D 프로젝트 생성


템플릿
2D
3D
3D With Extras
High Definition RP
Universal Project Template
설정
프로젝트 이름 *
New Unity Project
저장 위치 *
D:\Unity3D_Project\Tutorial
취소
생성

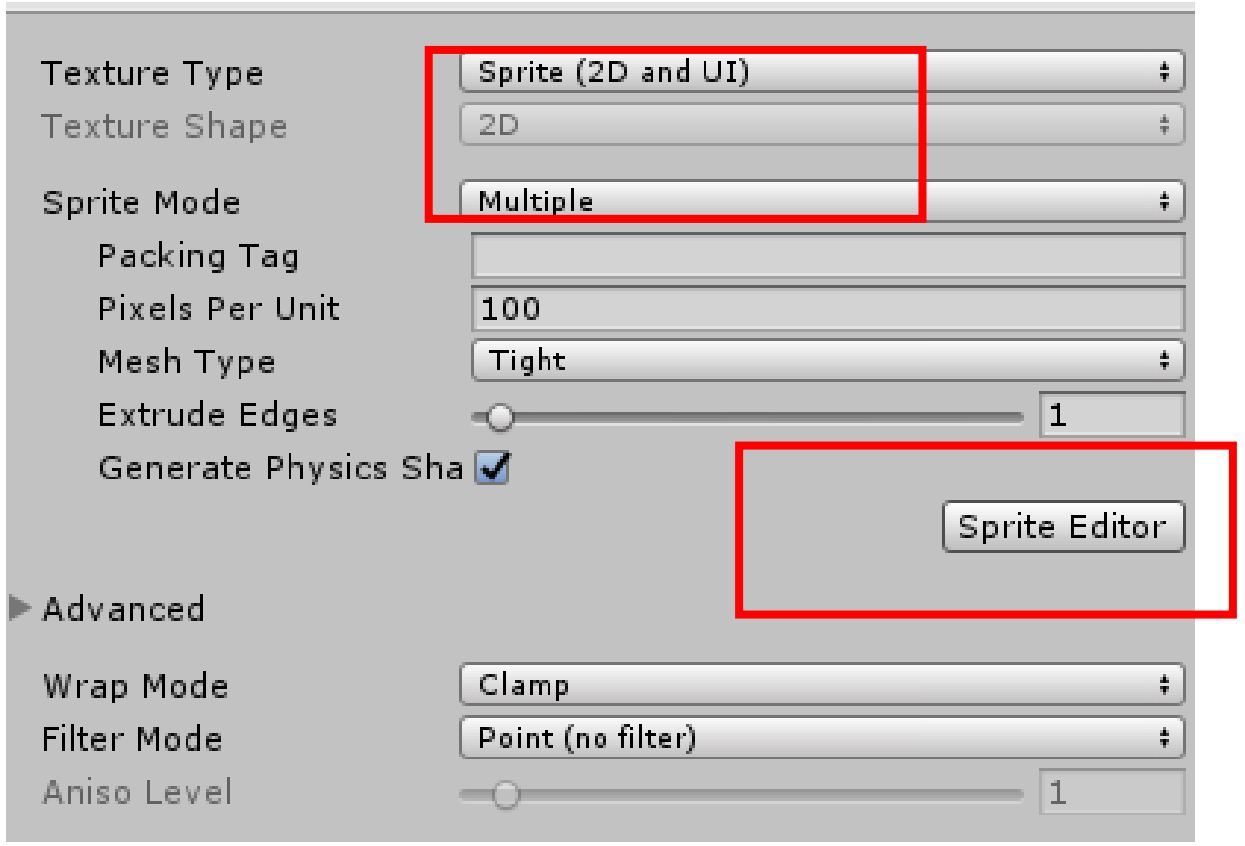
스프라이트 준비

아틀라스맵: 여러 개의 스프라이트를 정렬해서 하나의 텍스쳐에 모아둠
Hierarchy에 드래그 하여 게임 내 객체로 생성
리소스들: https://www.spriters-resource.com/

스프라이트 가공

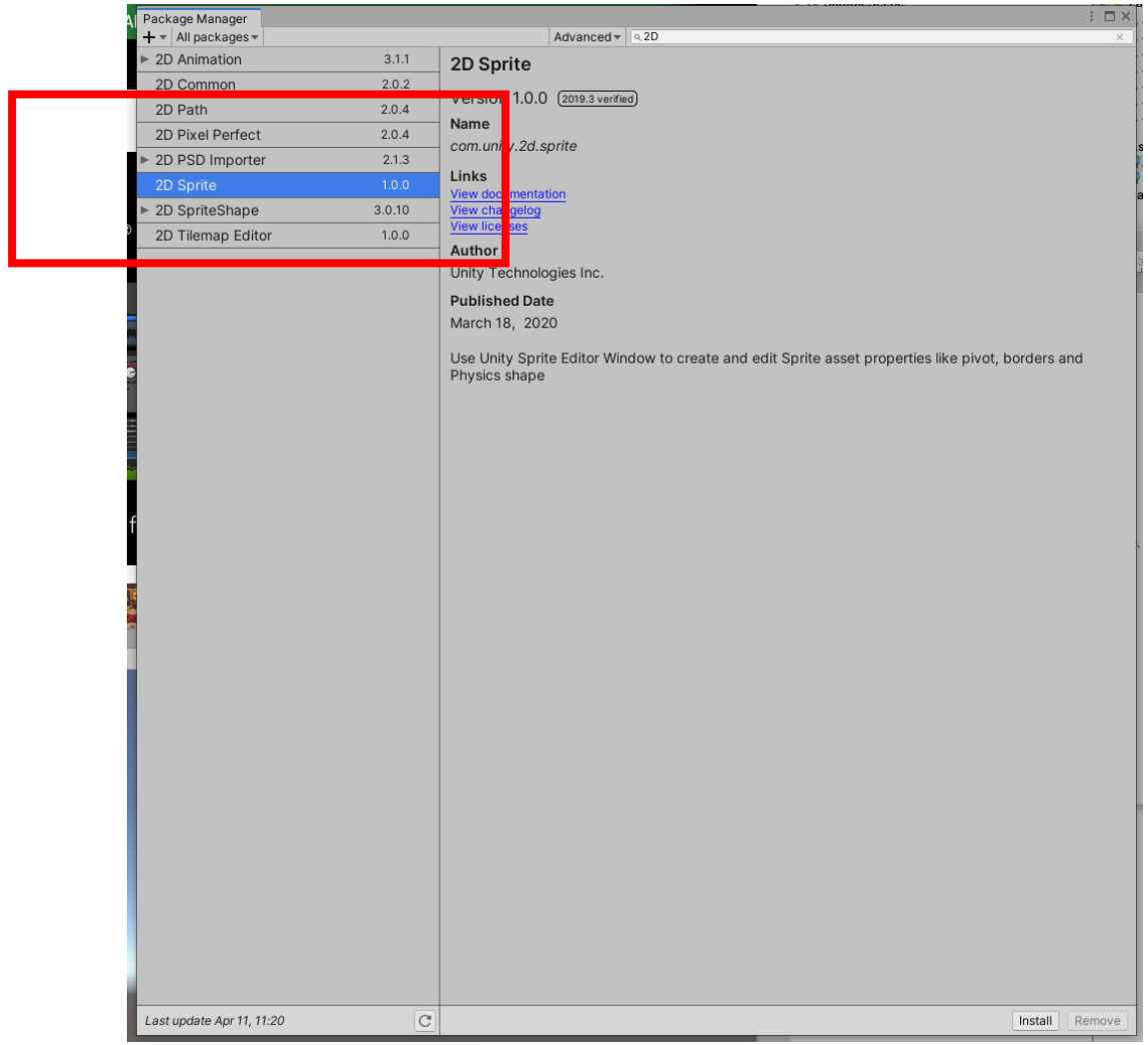
아틀라스 맵에서 개별 스프라이트로 분리해 냄

Texture Type: Sprite
Sprite Mode: Multiple
Filter Mode: Bilinear



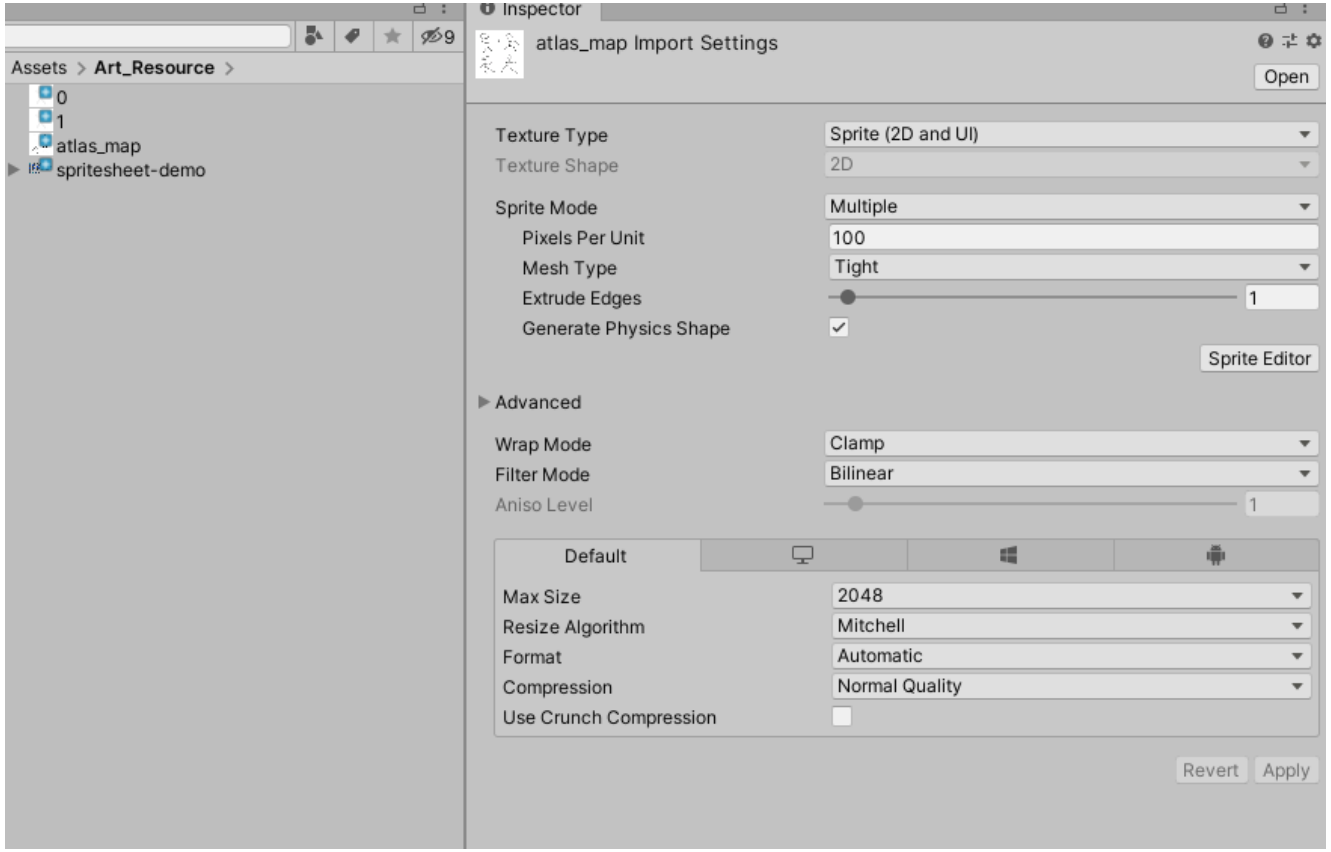
스프라이트 에디터 (Sprite Editor)

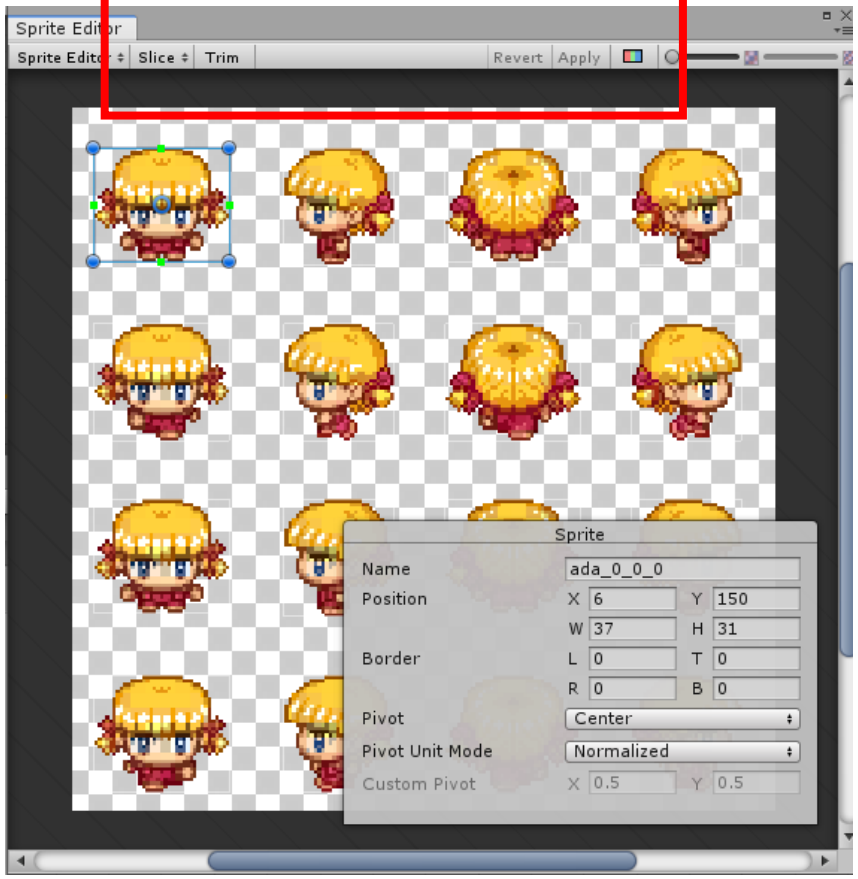
Package Manager 에서 2D Sprite Install



스프라이트 에디터 (Sprite Editor)

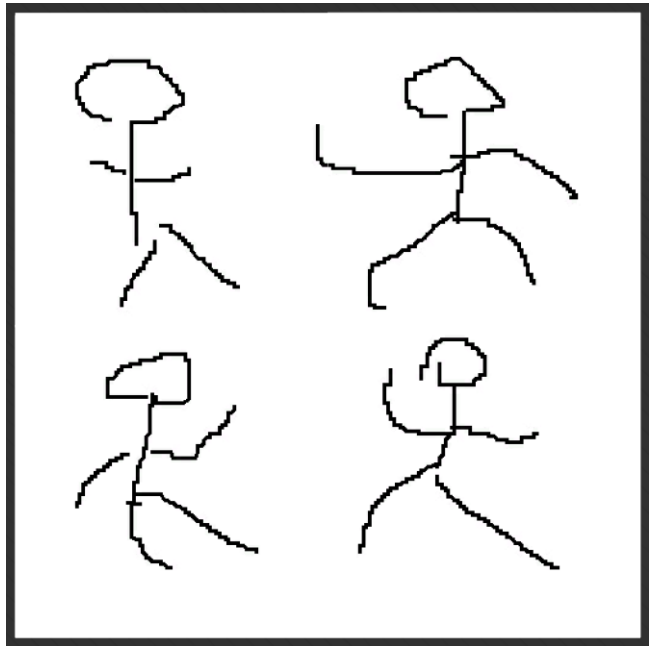
이미지를 선택한 상태에서 Sprite Editor 버튼을 누른다.
Slice 누르면 자동으로 잘림, 행/열 개수로도 자동 slice 가능
위치, 크기, 피벗 등을 조절 후 Apply 누르면 잘라진 sprite 생성





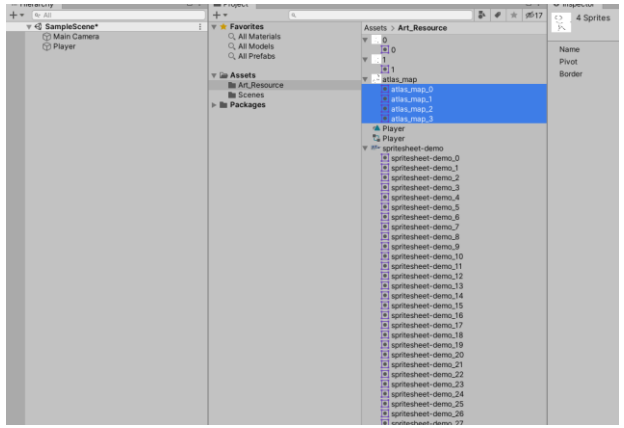
스프라이트 에디터 (Sprite Editor)

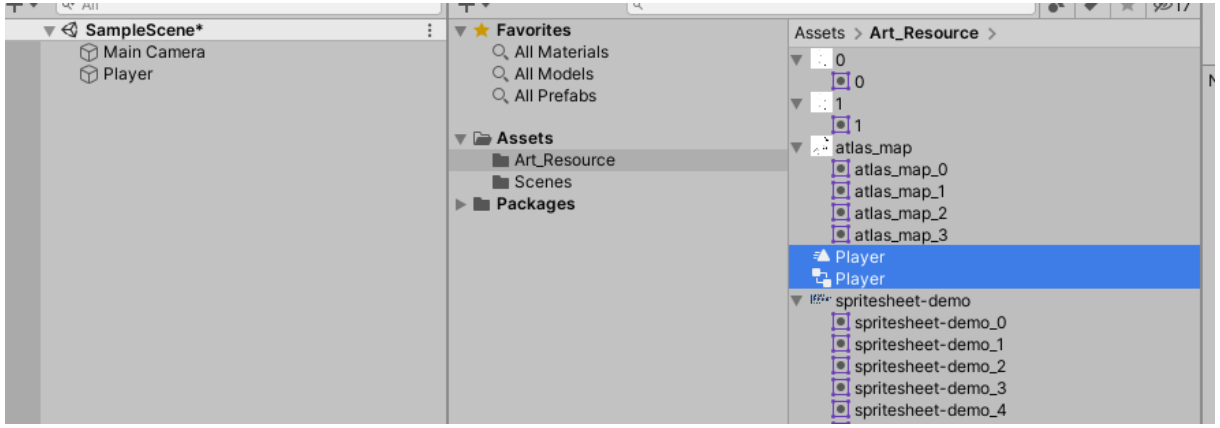
Sprite 생성 방법
1) 모션 1개당 1개의 이미지일 경우 → 1개씩 Single Sprite로 변환 시키기
2) 다수의 모션이 하나의 이미지에 들어 있을 경우 (아틀라스 맵)
→ Multiple Sprite 속성으로 Sprite Editor에서 생성



Player object / anim 생성

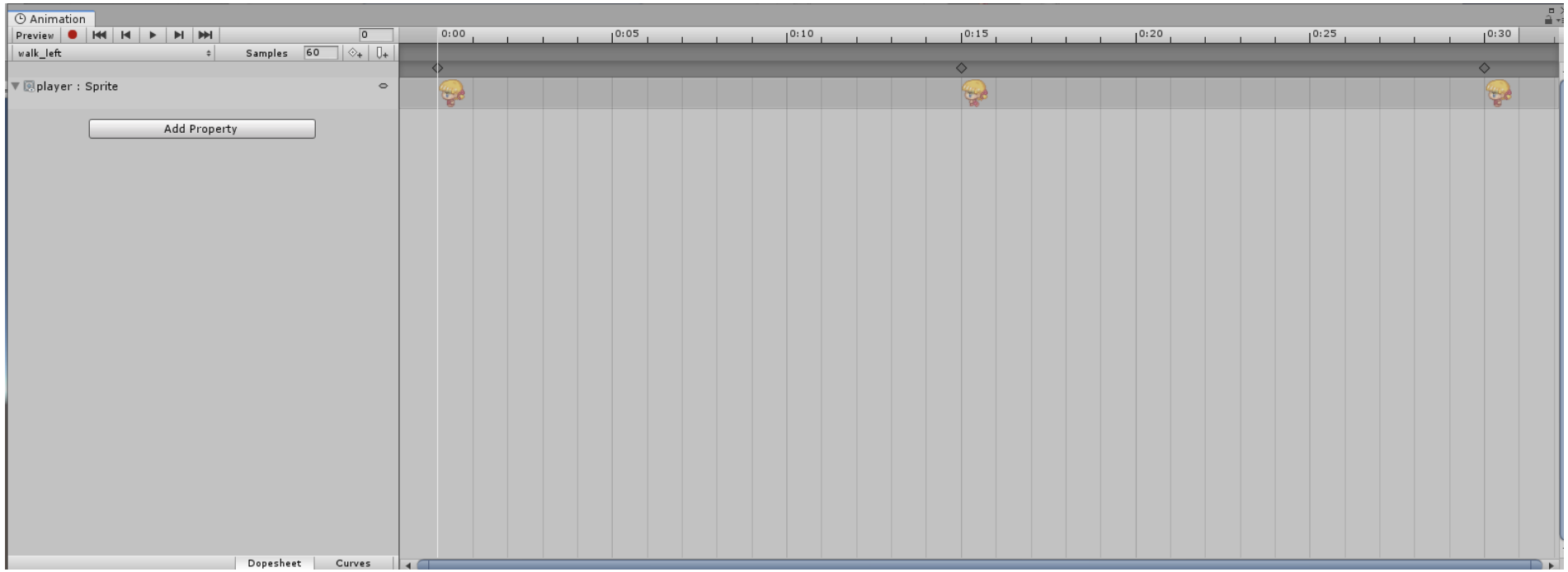
- Empty object를 만들고, 사용한 스프라이트들을 모두 선택한 뒤, Drag & Drop
- .anim 파일명 입력하면 anim 파일이 생성됨 (Sprite Render와 Animator가 자동으로 생성됨)
- Empty object를 선택한 상태에서 window → animation → animation 하면 animation 창이 팝업





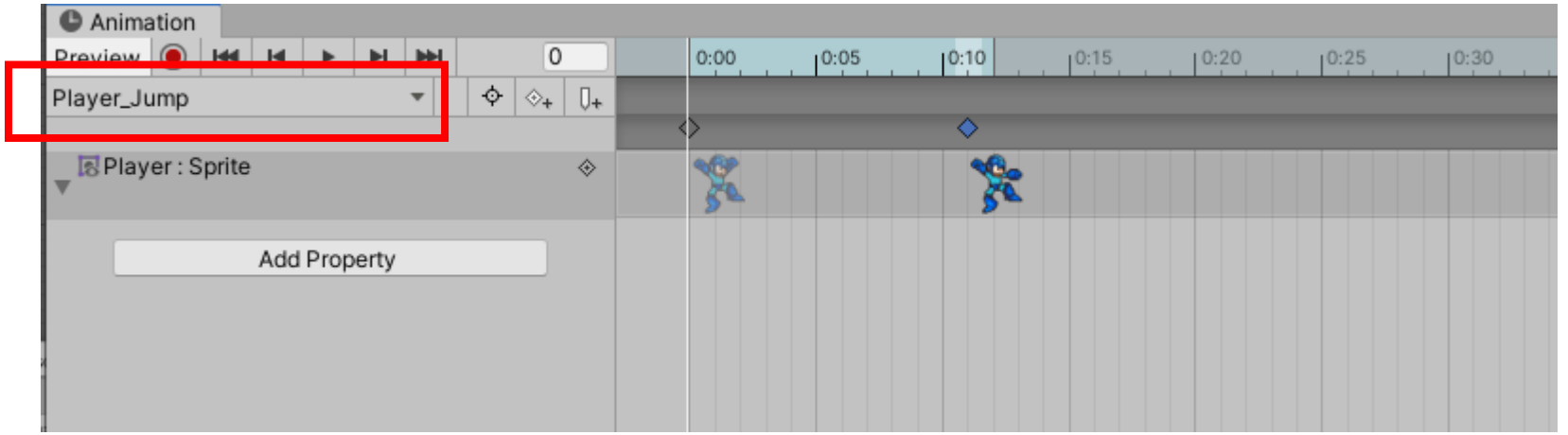
애니메이션 (Animation)

스프라이트 오브젝트를 선택 후, 애니메이션 이름 있는 곳에서 오른쪽 버튼 Create
New clip
키 프레임을 지정하여 애니메이션 클립(.anim 파일)을 생성함
일반적으로 4/8/16 방향, 개별 전투 애니메이션을 클립을 다 생성해 둠



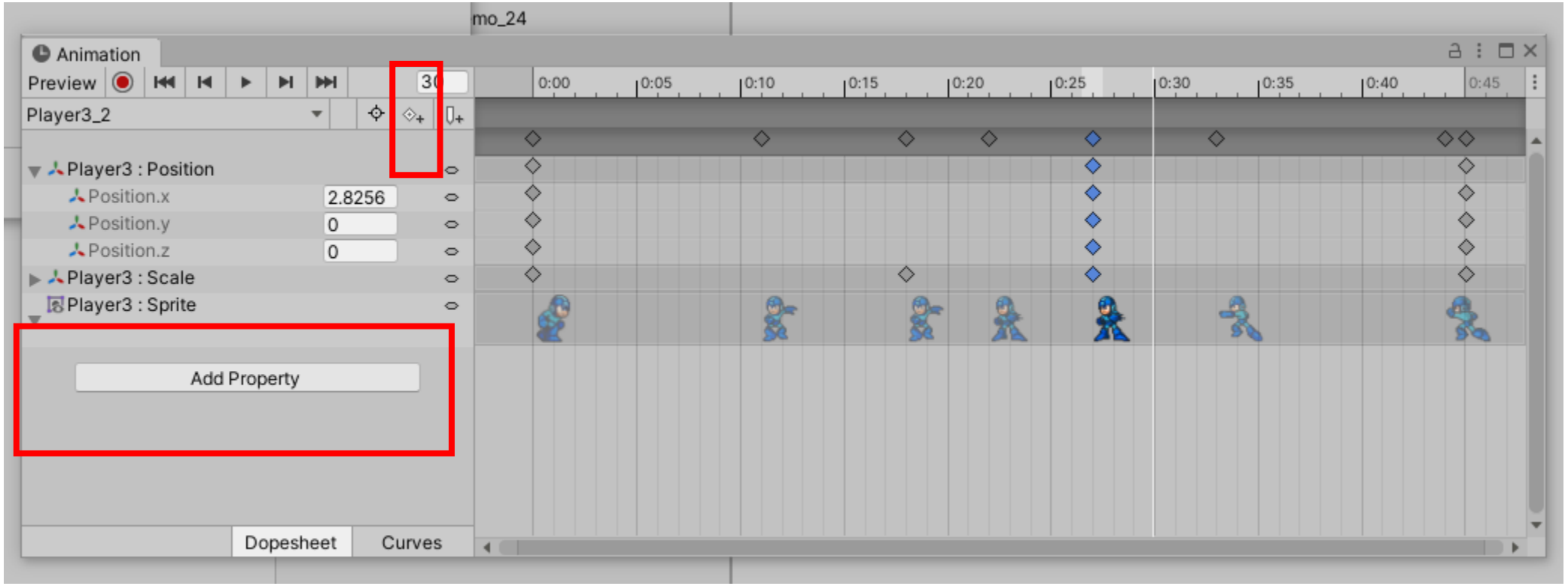
애니메이션 (Animation)

왼쪽 후, 버튼 Create New clip



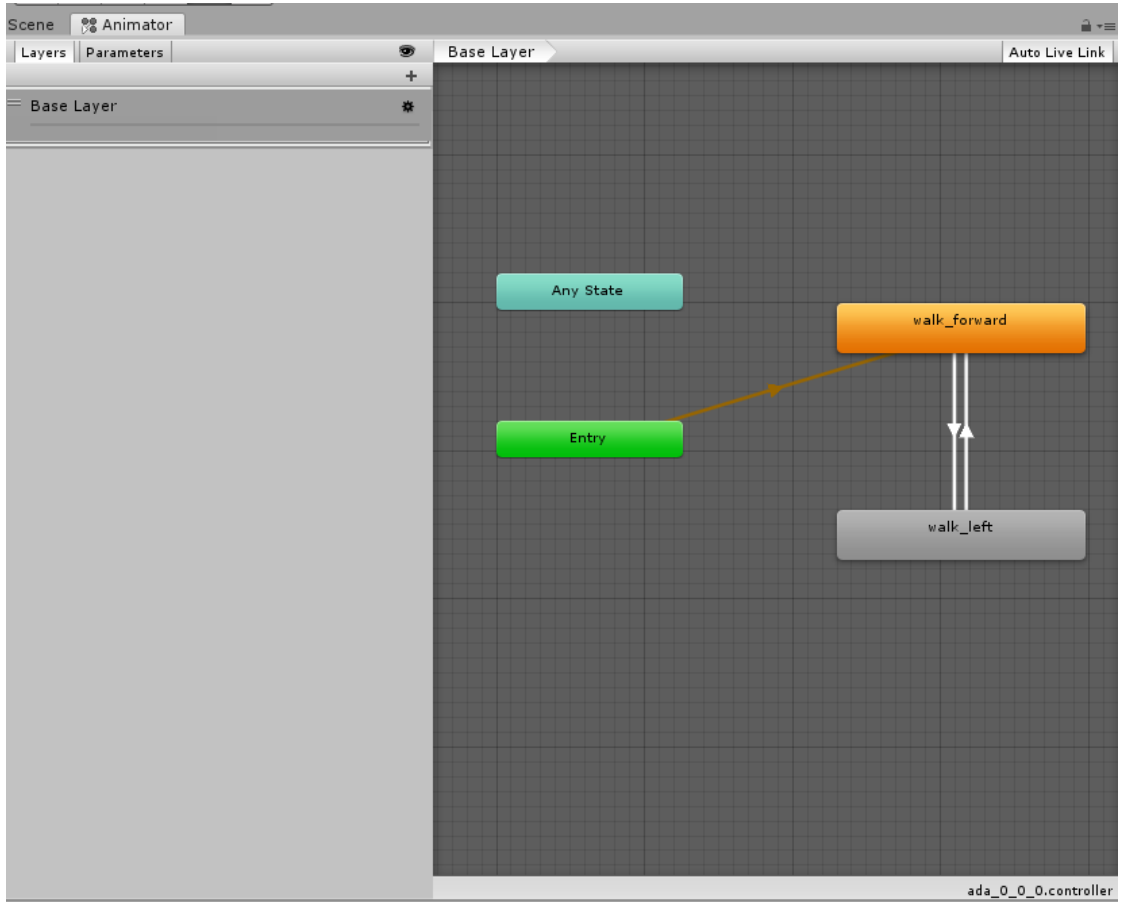
애니메이션 (Animation)

Add Property → Transform, Position 후, Key frame 추가해 보기



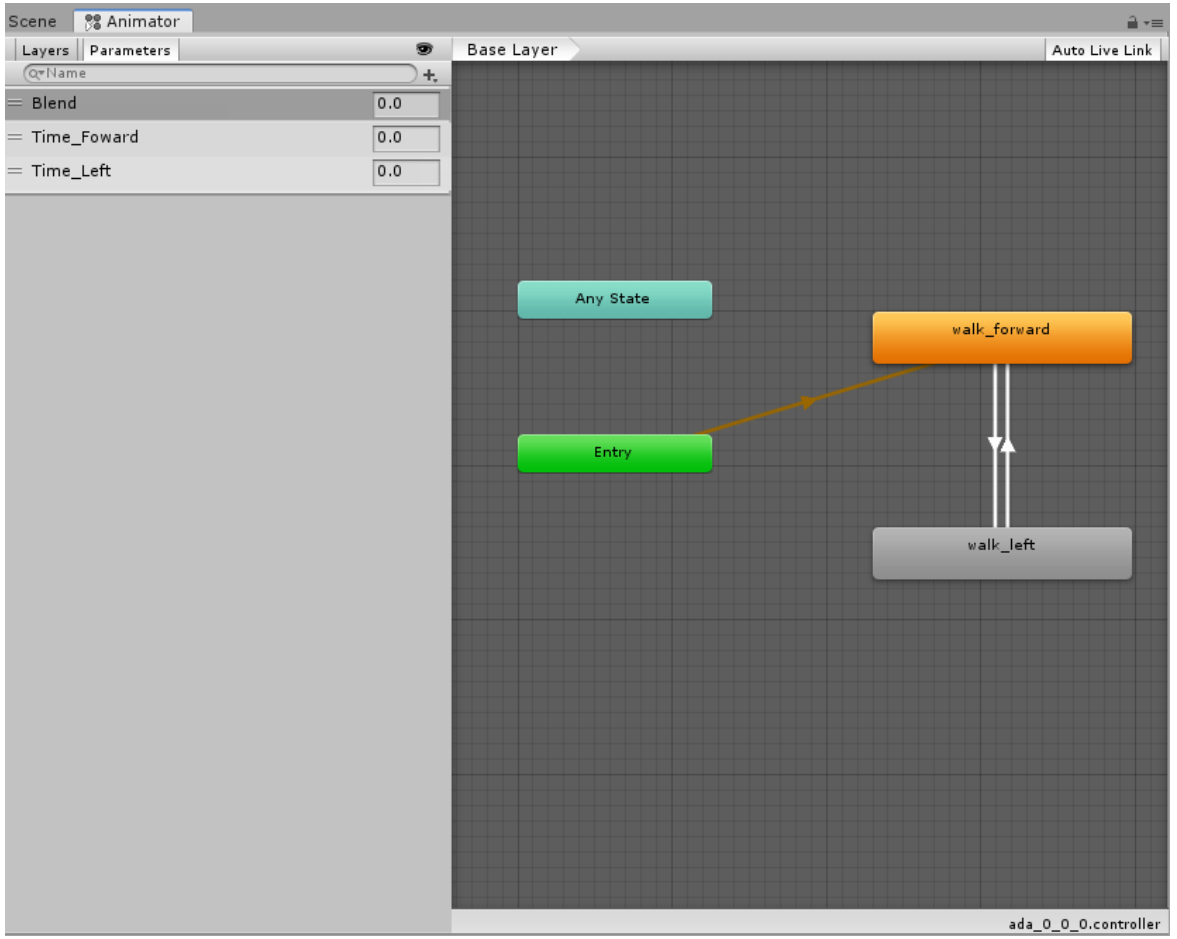
애니메이터 (Animator) – Layers

애니메이션 클립(.anim 파일) 간의 상태 전이를 지정함 (상태 다이어그램)
모든 애니메이션 상태들을 표시하고 이들 간의 전이 조건을 지정함
마우스 오른쪽 버튼 Make Transition 으로 전이 추가



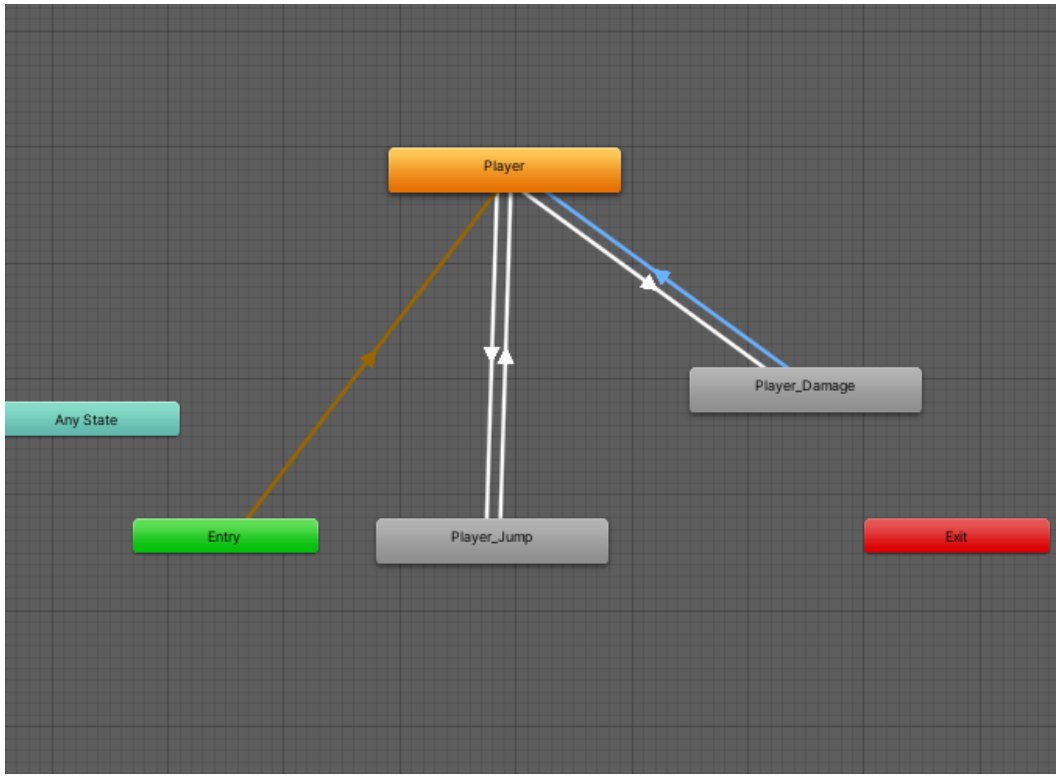
애니메이터 (Animator) – Parameters

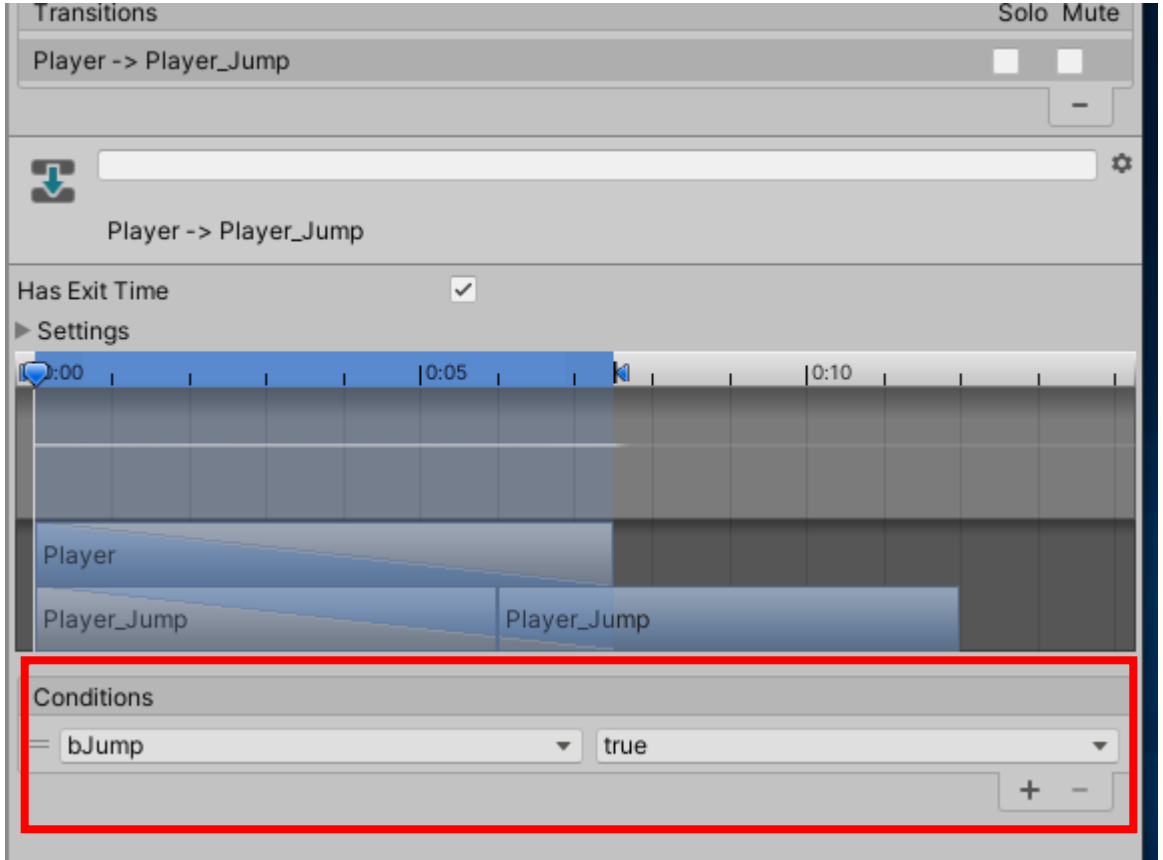
전이 조건에서 사용할 모든 파라미터를 정의함
이곳에 지정해야 아래 스크립트에서 해당 파라미터를 접근하여 흐름을 제어할 수 있음



애니메이터 (Animator) – 전이조건입력

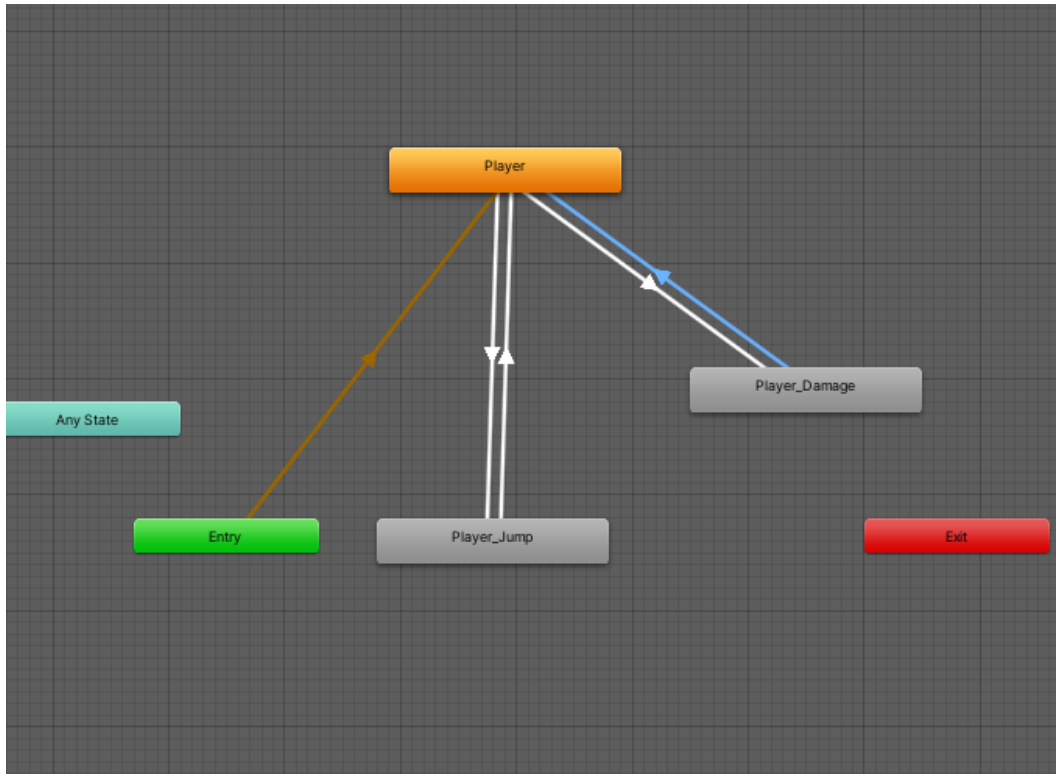
Transition 라인을 선택한 뒤, inspector에서 transition condition으로 사용한 변수 설정

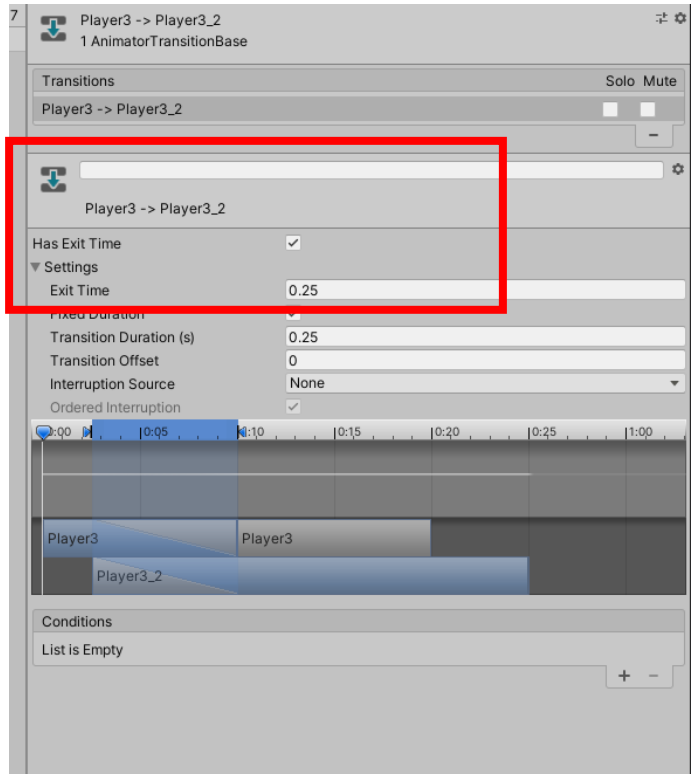




애니메이터 (Animator) – 전이조건입력

Has Exit Time 체크 후, 시간 입력하면 시간이 지나면 원래대로 돌아옴 (돌아오는 Transition 추가 필요)





애니메이션 스크립트

전이 조건에서 사용할 모든 파라미터를 정의함

```
protected Animator animator ;

void Start () {
animator = GetComponent < Animator > ();
}

animator.SetBool("bJump", true);
```

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

public class Player_Manager : MonoBehaviour
{
    protected Animator animator;
    private int a;

    // Start is called before the first frame update
    void Start()
    {
        a = 0;
        animator = GetComponent<Animator>();

    }

    // Update is called once per frame
    void Update()
    {
        a = a + 1;

        if (a > 300 && a < 600)
        {
            animator.SetBool("bJump", true);


        }else if (a > 600)
        {

            animator.SetBool("bJump", false);


        }

    }
}
```

애니메이션 스크립트

전이 조건에서 사용할 모든 파라미터를 정의함

```
protected Animator animator ;
```

```
void Start () {
animator = GetComponent < Animator > ();
}
```

```
animator.SetFloat( "Time_Left" , b );
```

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

0 references
public class PlayerControl : MonoBehaviour
    {

    6 references
    protected Animator animator ;
    6 references
    private int a = 0;
    6 references
    private int b = 0;
    6 references
    private int status = 0;

    0 references
    void Start () {
        animator = GetComponent < Animator > ();
    }
    0 references
    void Update () {
        if (status == 0)
        {
            a = a + 1;
            if (a > 300)
            {
                animator.SetFloat( "Time_Foward" , a );
                animator.SetFloat( "Time_Left" , 0 );
                a = 0;
                status = 1;
            }
        }else if (status == 1)
        {

            b = b + 1;
            if (b > 300)
            {
                animator.SetFloat( "Time_Left" , b );
                animator.SetFloat( "Time_Foward" , 0 );
                b = 0;
                status = 0;
            }
        }
        if ( animator )
        {
            if (status == 0)
            {
                Debug.Log("animator foward running" + a);
            }else if (status == 1)
            {
                Debug.Log("animator left running" + b);
            }


        }


    }
}
```

여러 개의 캐릭터 만들어서 상태 정의