Unity – Player Animation

NHN NEXT
서형석

Unity

- Mecanim 구현 순서

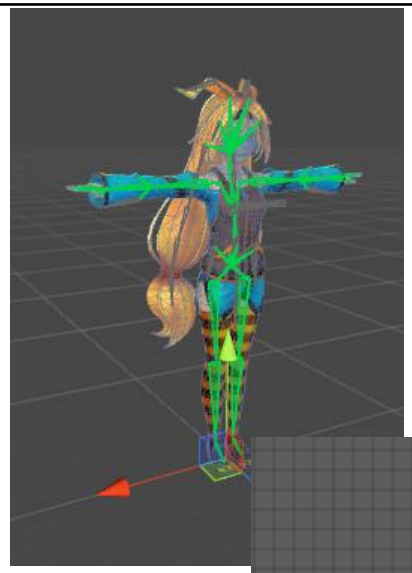
Avatar 설정

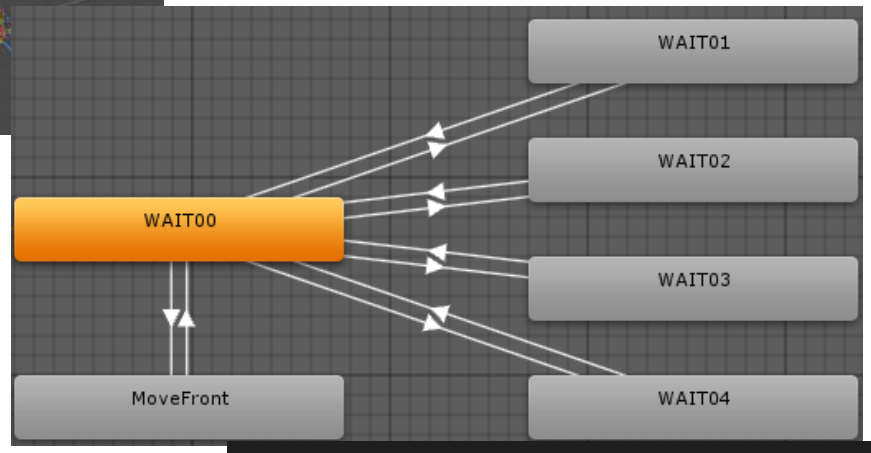
↓

Animator 설정

↓

프로그래밍






```
if (Input.GetKeyDown(KeyCode.Alpha1))
{
    anim.SetInteger("randomidle", 1);
}
```

Unity

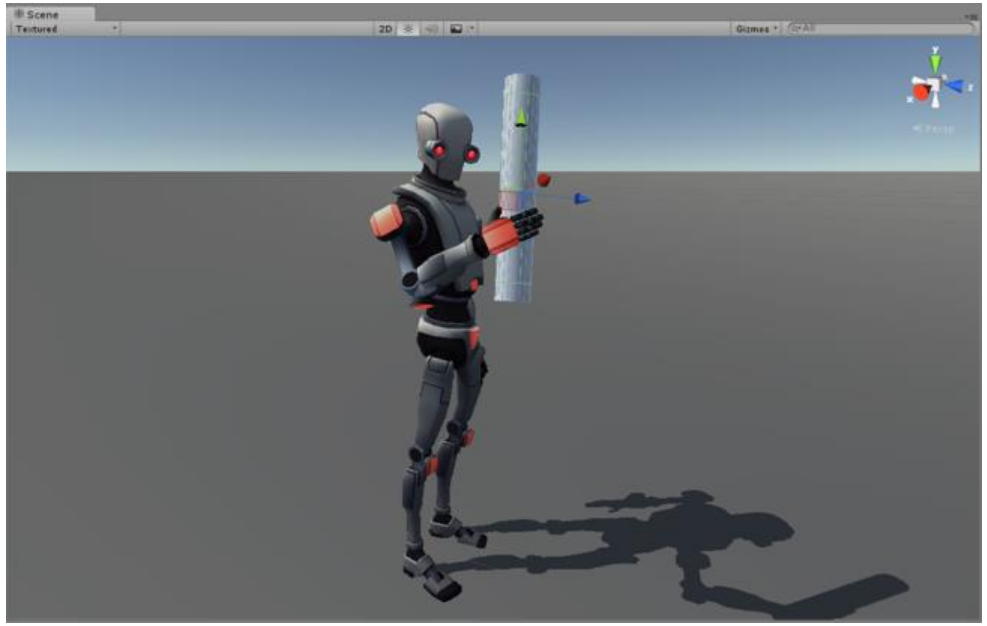
- Mecanim

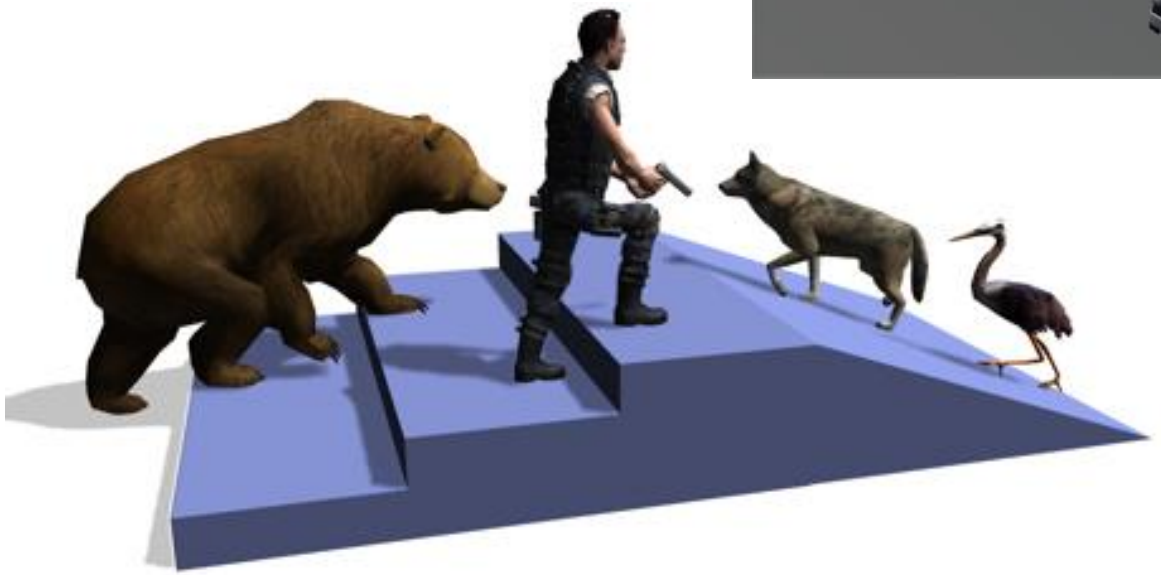
  애니메이션 Retargeting을 지원하는 시스템
  즉, 애니메이션 공유가 가능

  당연히 인간형 캐릭터만 가능

Unity

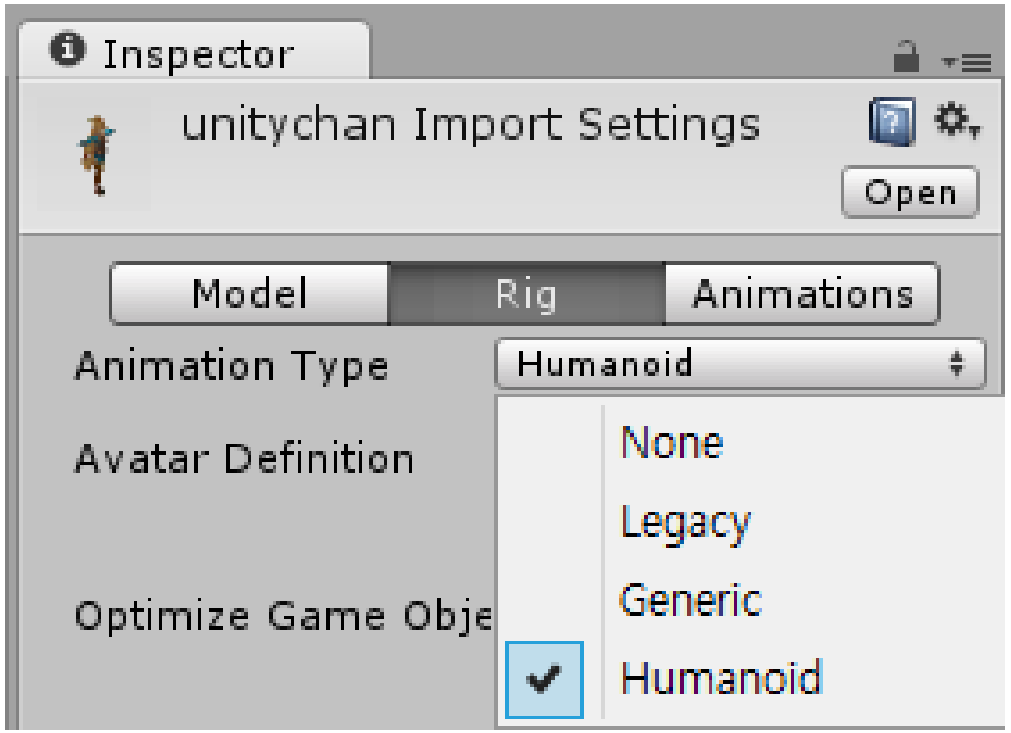
- 프로그래밍
  - 상태 전이(모션 제어)
  - IK 구현 등.





Unity

- Rig 설정


Inspector
unitychan Import Settings
Open
Model
Rig
Animations
Animation Type
Humanoid
Avatar Definition
Optimize Game Obje
None
Legacy
Generic
Humanoid

Unity

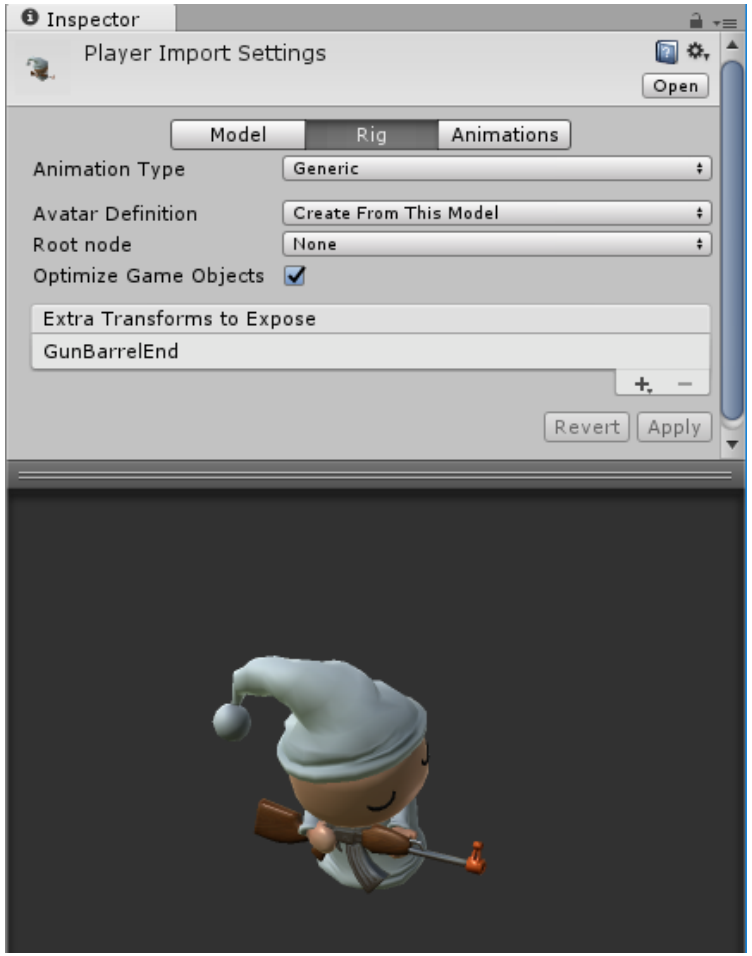
- Rig 설정

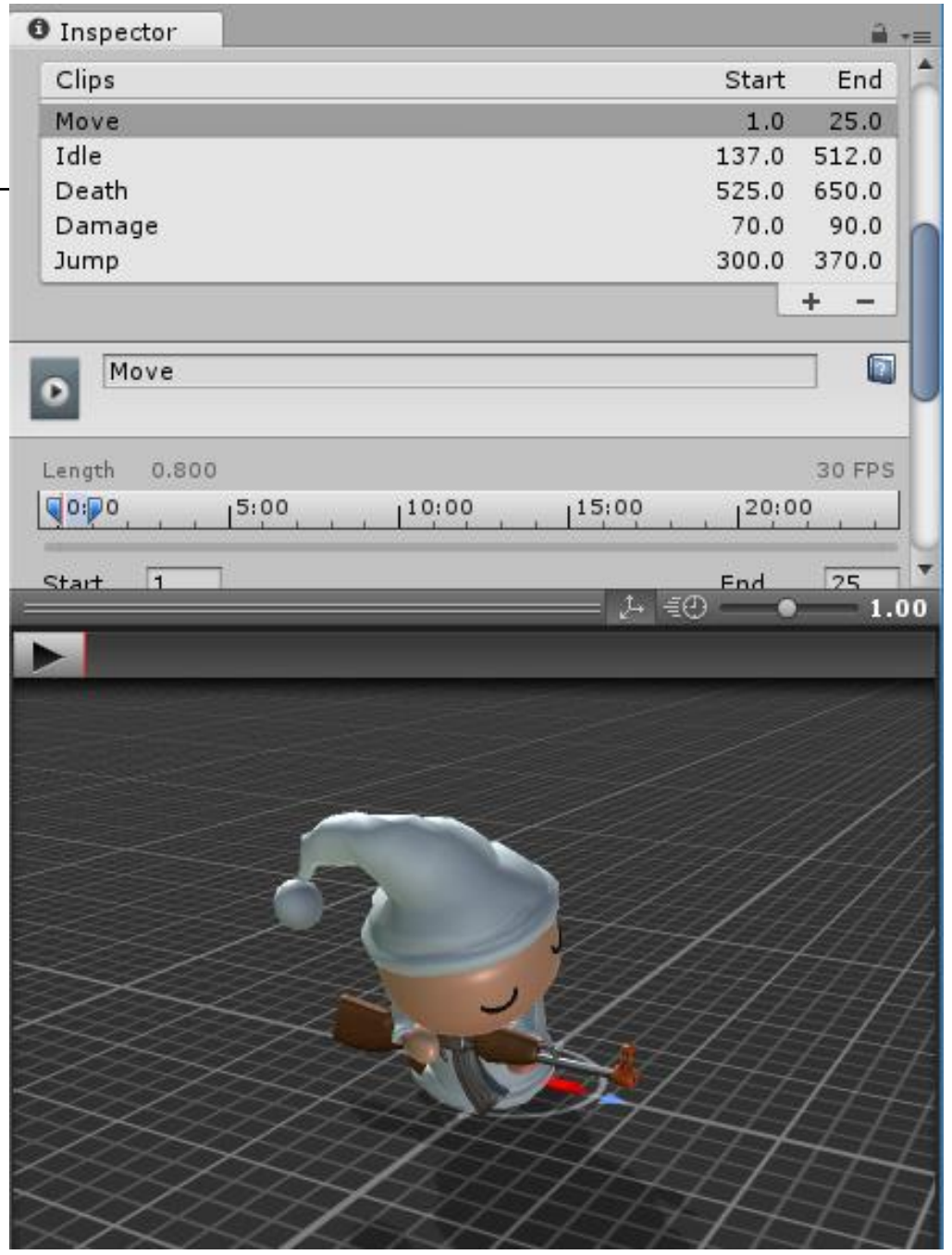
Generic

Humanoid

Unity

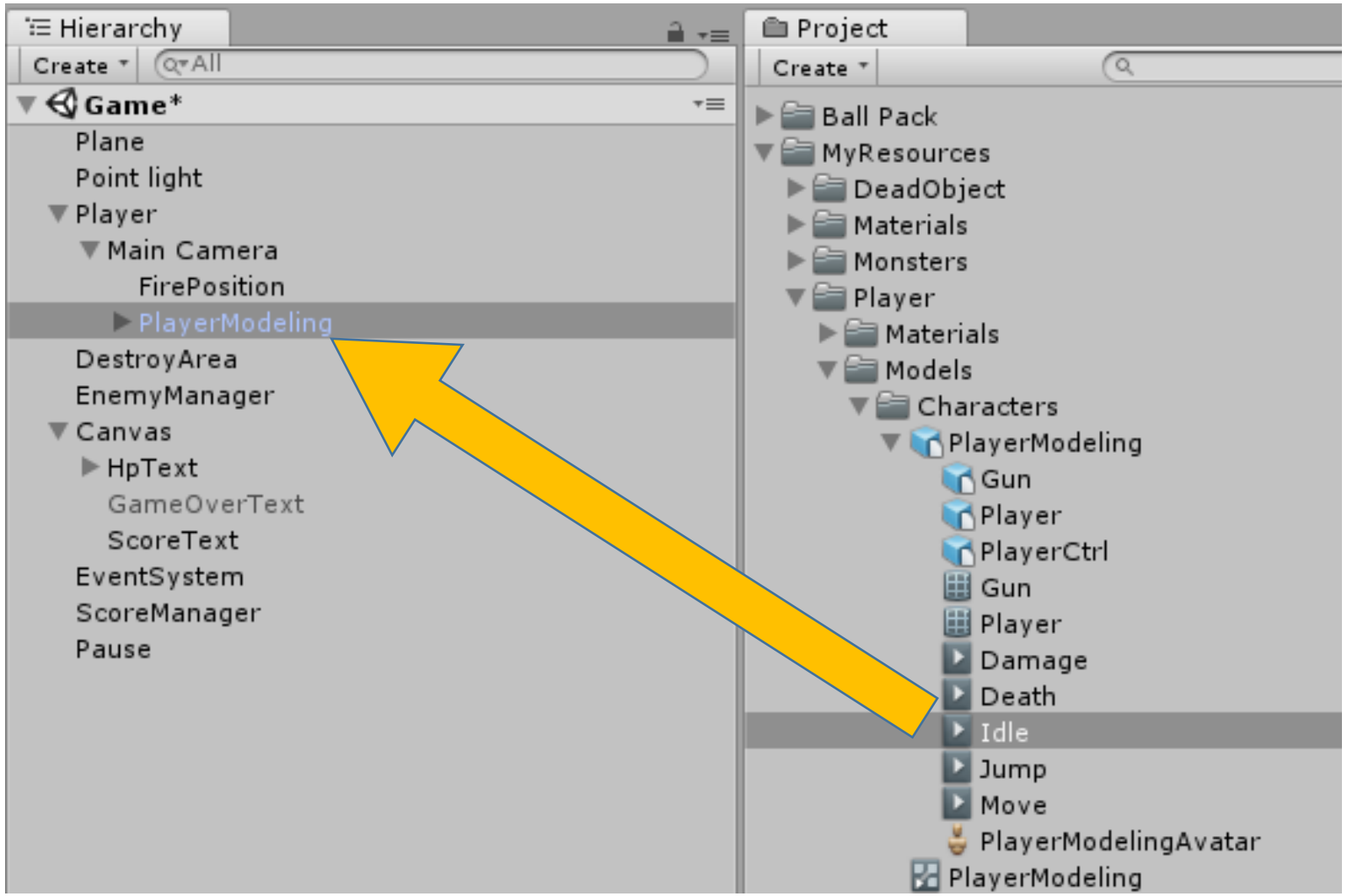
- Rig 설정 및 애니메이션 추가





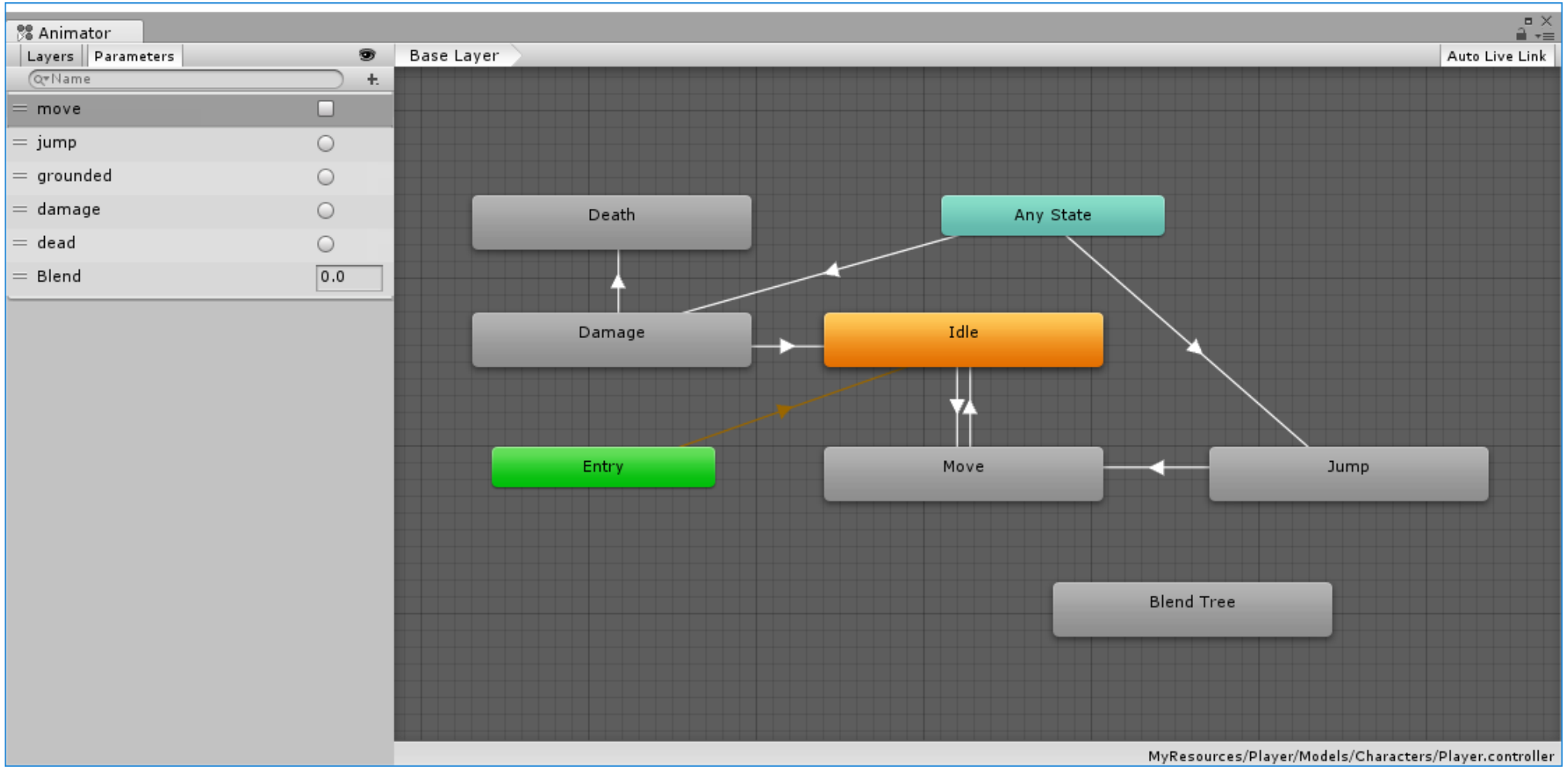
Unity

- Animator 생성



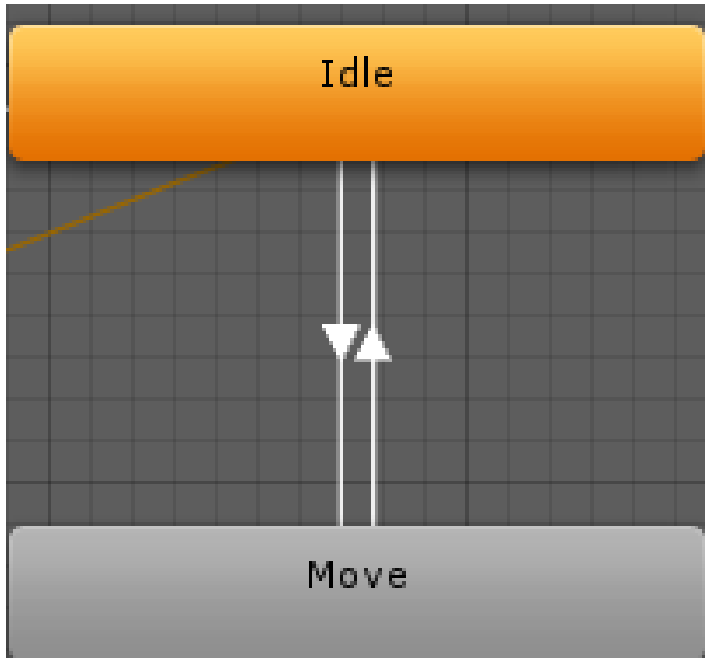
Unity

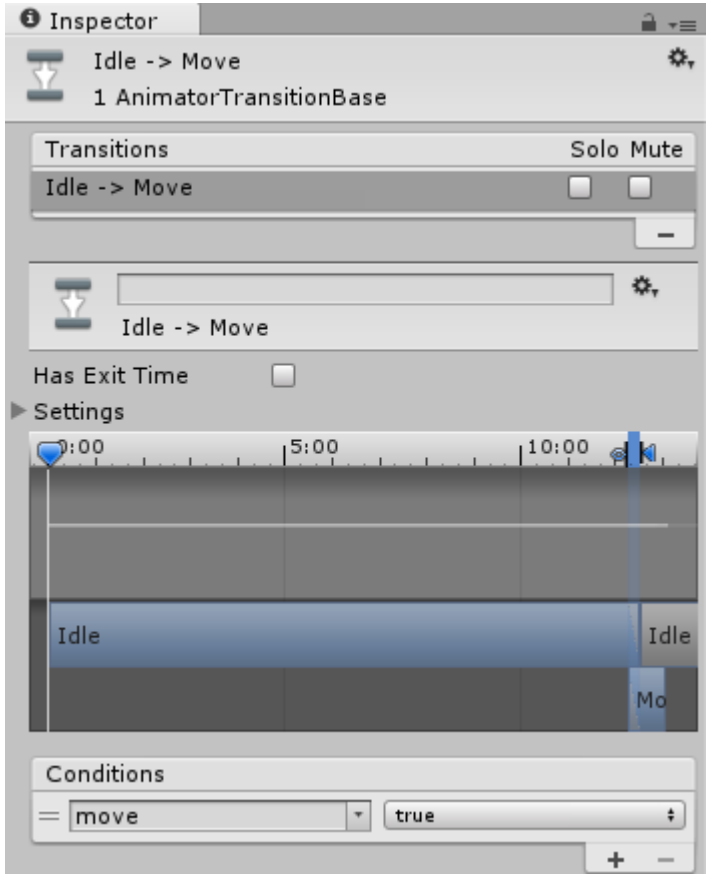
- Animator

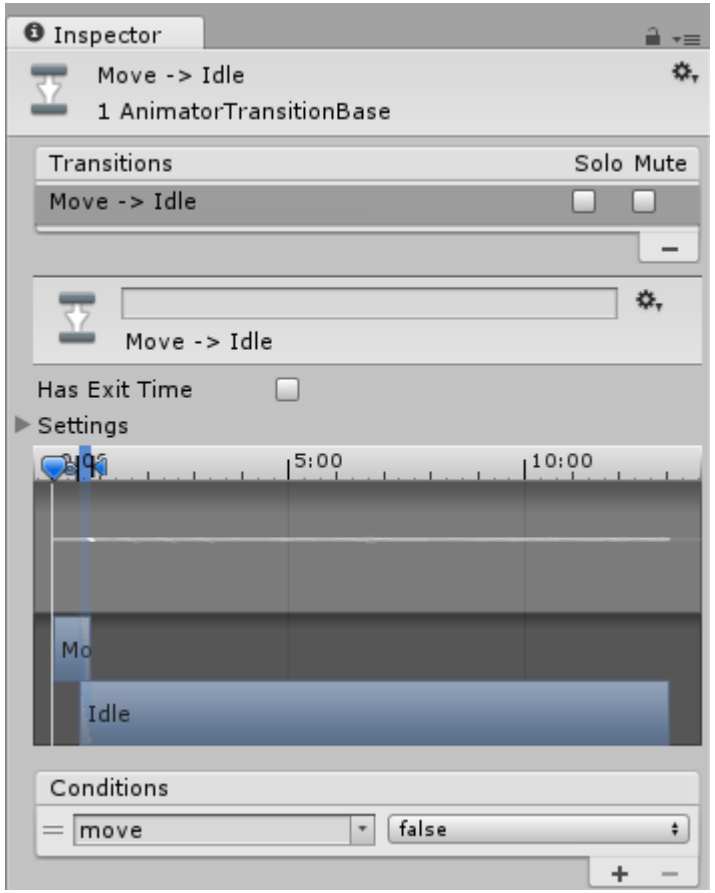


Unity

- Transition 설정

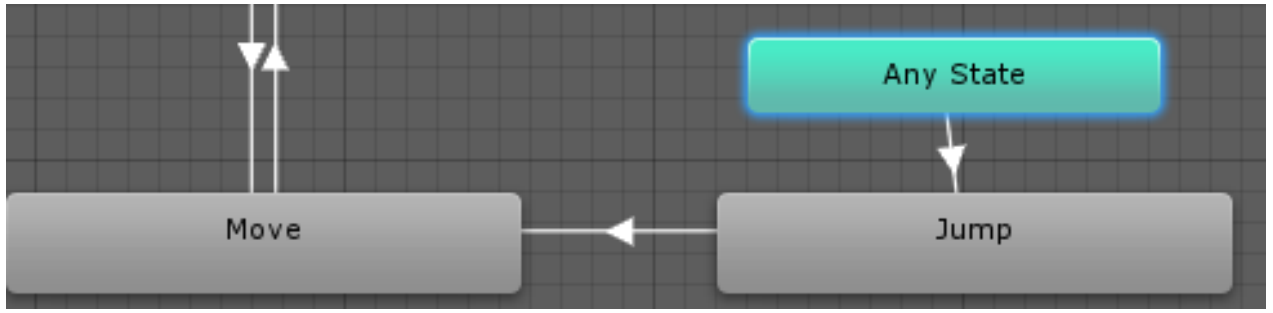


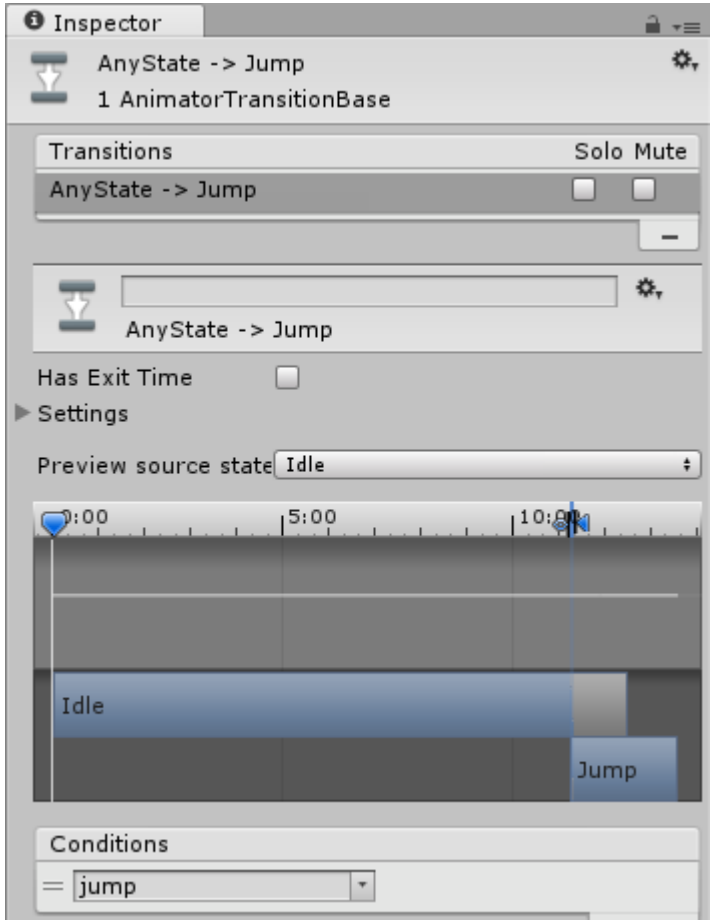


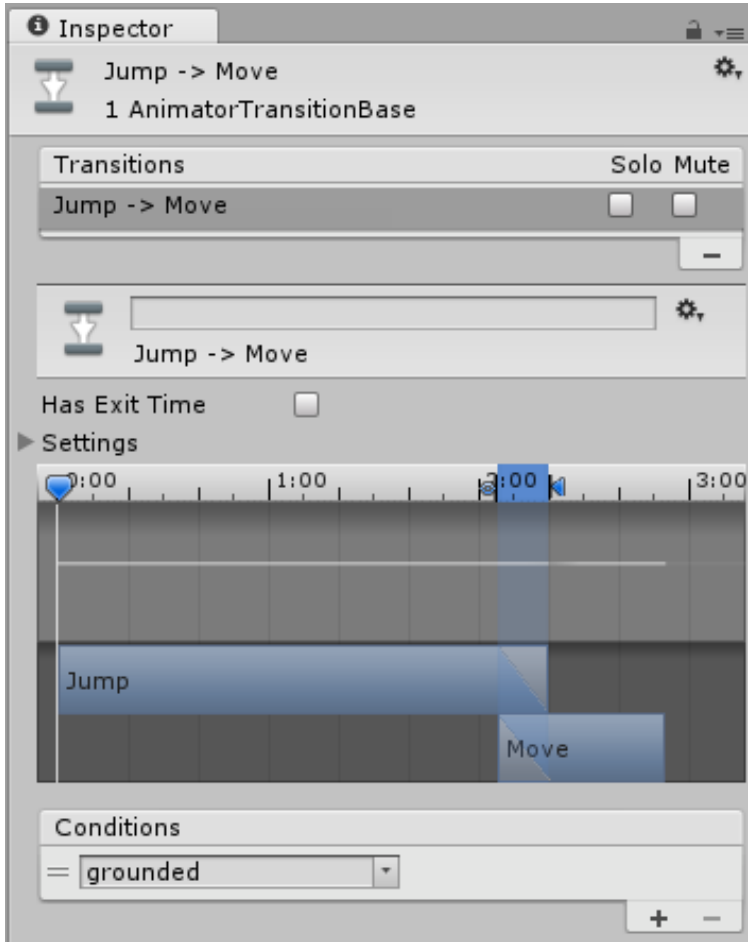


Unity

- Transition 설정

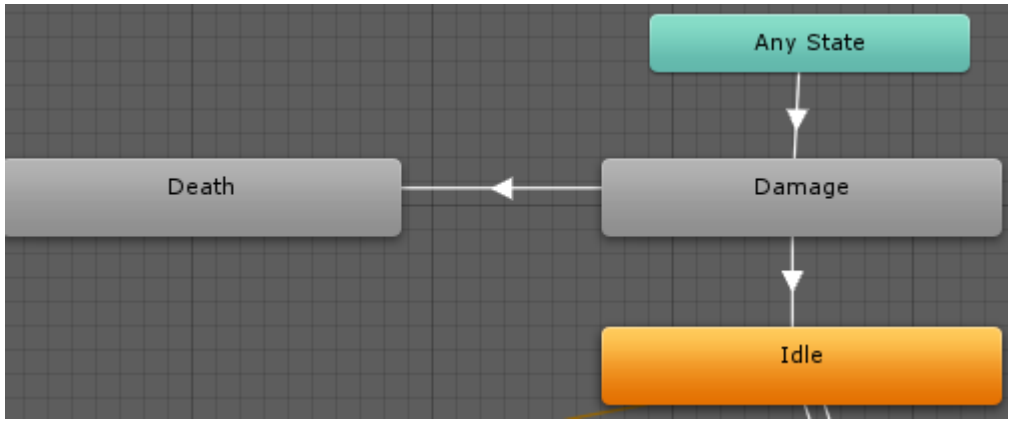


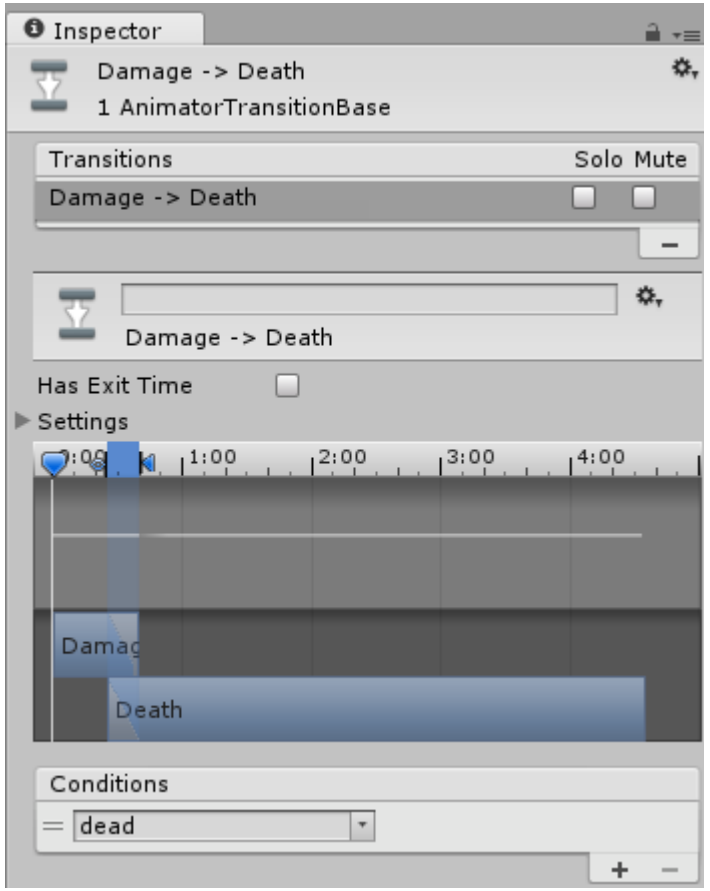


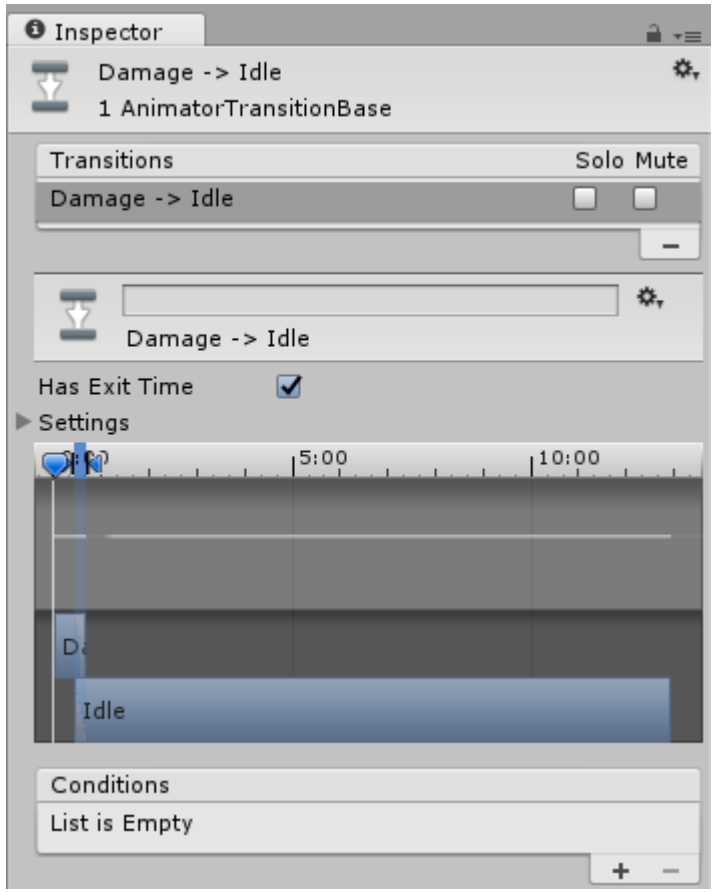


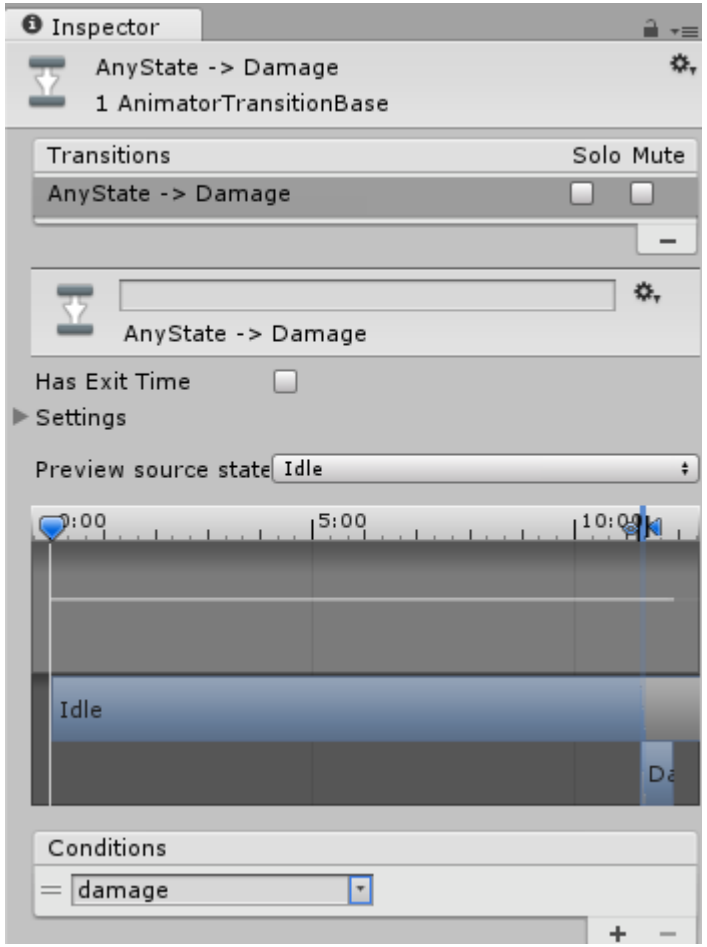
Unity

- Transition 설정









```
public class PlayerAnim : MonoBehaviour
{
    public   Animator   animator;

    public void Idle()
    {
        animator.SetBool("move", false);
    }

    public void Move()
    {
        animator.SetBool("move", true);
    }

    public void Jump()
    {
        animator.SetTrigger("jump");
    }

    public void Grounded()
    {
        animator.SetTrigger("grounded");
    }

    public void Damage()
    {
        animator.SetTrigger("damage");
    }

    public void Dead()
    {
        animator.SetTrigger("dead");
    }
}
```

```
public class PlayerMove : MonoBehaviour
{
    public PlayerAnim playerAnim;

    void Update()
    {
        float  x = Input.GetAxis(  “Horizontal”  );
        float  z = Input.GetAxis(  “Vertical”  );

      if (x != 0.0f || z != 0.0f)
      {
          playerAnim.Move();
      }
      else
      {
          playerAnim.Idle();
      }

      ......
    } // End of Update
}
```

Unity

```
public class PlayerMove : MonoBehaviour
{
    void Update()
    {
        ......
        if( characterController.isGrounded == true )
        {
            yVelocity = 0.0f;
            playerAnim.Grounded();
        }

        if(  Input.GetButtonDown( “Jump “)  )
        {
            yVelocity = jumpSpeed;
            playerAnim.Jump();
        }
        ......
    } // End of Update
}
```

```
public class PlayerMove : MonoBehaviour
{
   bool    jumped = false;

   void Update()
   {
        ......
       if( characterController.isGrounded == true )
       {
           yVelocity = 0.0f;
           if (jumped == true)
          {
             jumped = false;
             playerAnim.Grounded();
          }
       }

      if(  Input.GetButtonDown( “Jump “)  )
       {
           yVelocity = jumpSpeed;
           if (jumped == false)
          {
              jumped = true;
             playerAnim.Jump();
          }
       }
       ......
```

Unity

- Damage / Dead 처리

```
public class PlayerState : MonoBehaviour
{
    public    PlayerAnim    playerAnim;
```

```
    public void DamageByEnemy()
    {
        if( isDead )
            return;

        ......

        cameraShake.PlayCameraShake();
        playerAnim.Damage();

        if (healthPoint <= 0)
        {
            playerAnim.Dead();
            isDead = true;
            ...
        }
    }
```