3D콘텐츠 이론 및 활용

14주(1). 애니메이션 블렌딩

- Mecanim BlendTree
- Retargeting animations
- Animation Layer

학습개요

학습목표

- BlendTree를 이해하고 응용할 수 있다.
- 캐릭터 컨트롤러를 다른 캐릭터에 활용할 수 있다.
- 캐릭터 애니메이션을 다른 캐릭터에 적용할 수 있다.
- 아바타 마스크를 통해 원하는 동작만 추출해서 활용할 수 있다.

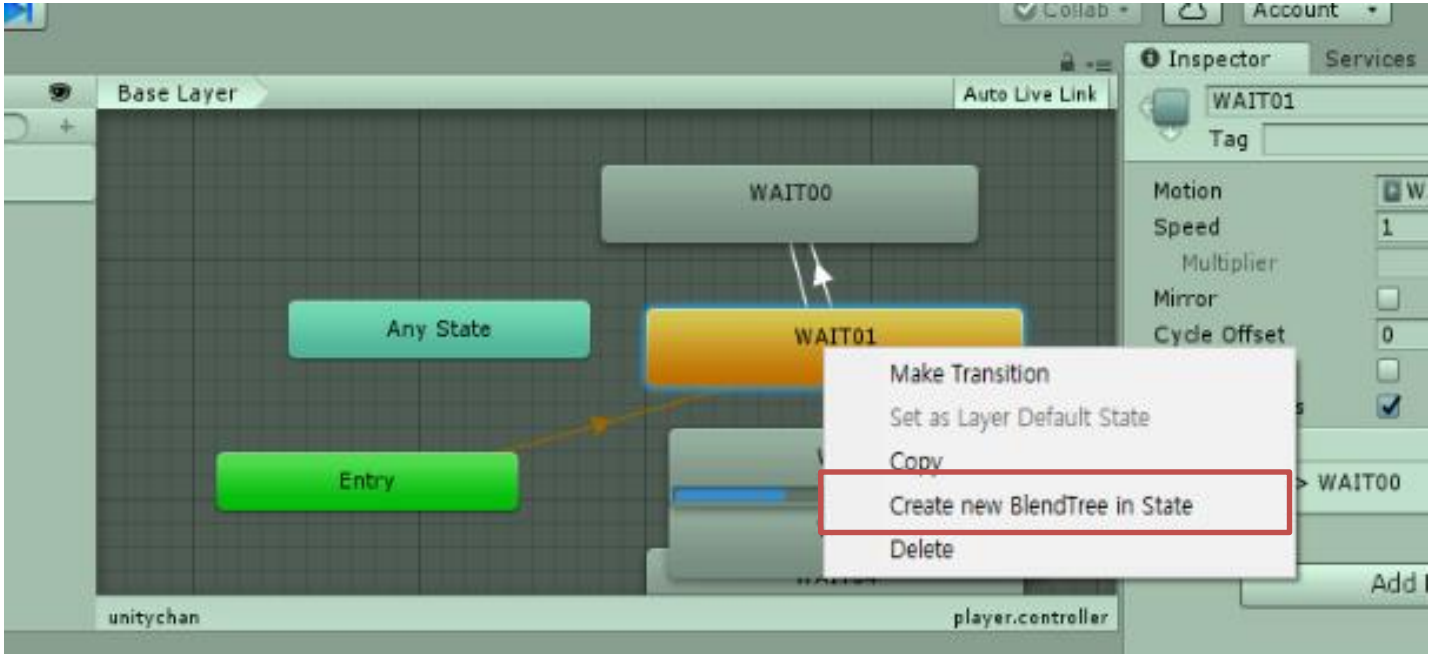
학습내용

- Play(), BlendTree
- animation retargeting
- Animation Layer
- Avatar Mask

1. 애니메이션 재생

1) Base Layer 구성

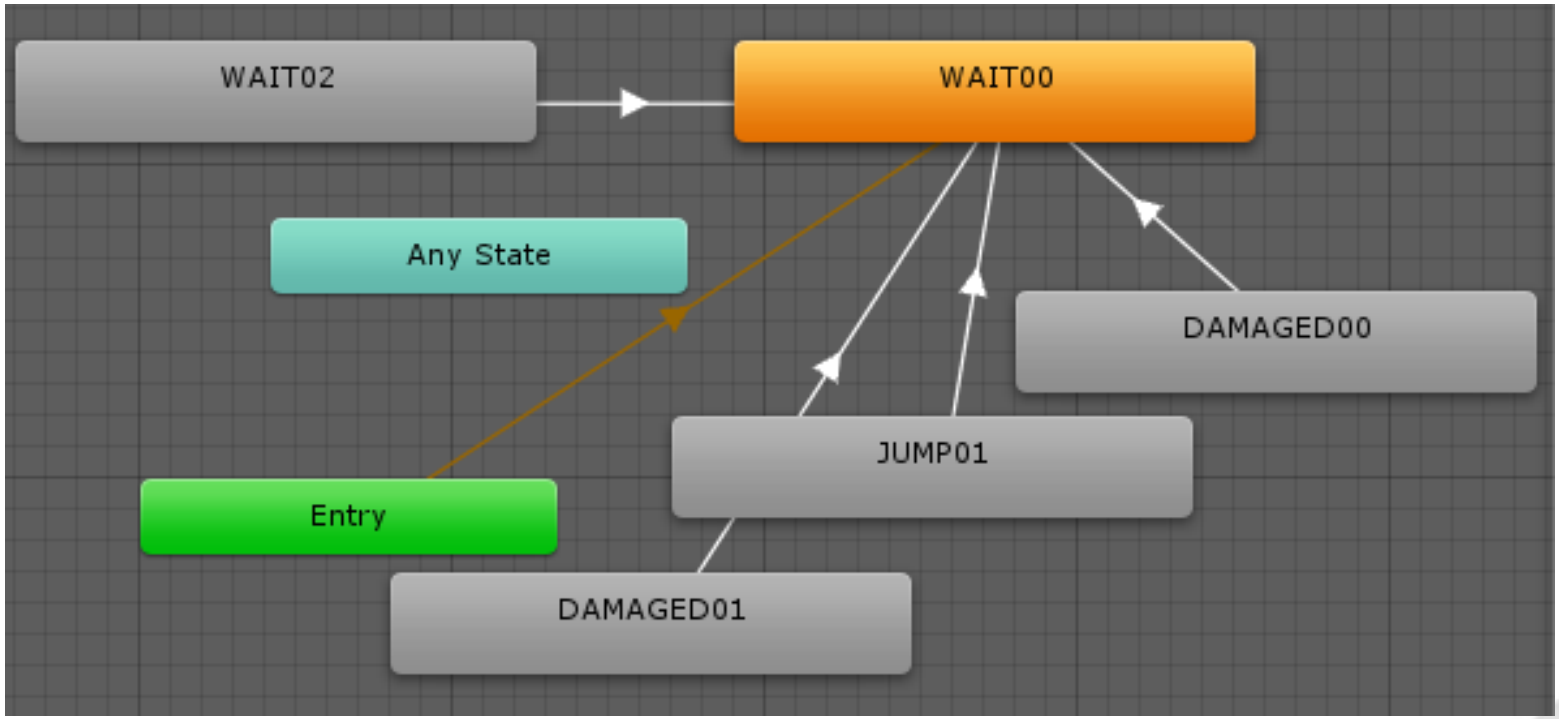
- Make Transition : 상태전환
- Set as Layer Default State : 디폴트 애니메이션
- Create new BlendTree in State : 새로운 BlendTree 생성



1. 애니메이션 재생

2) Play() 함수

- 애니메이션을 블렌딩 없이 재생
- Play() 함수는 animation 파라미터에 명시된 이름의 애니메이션을 시작하거나 기본 애니메이션을 재생
- Play() 함수 적용 예



1. 애니메이션 재생

3) Play() 함수 적용 예

```
public class player : MonoBehaviour
{
    public Animator anim;
    // Use this for initialization
    void Start ()
    {
        anim = GetComponent<Animator> ();
    }

    void Update ()
    {
        if (Input.GetKeyDown(KeyCode.Space)) {
            anim.Play ("WAIT01");
        }
        if (Input.GetKeyDown ("1")) {
            anim.Play ("WAIT02", -1, 0.5f);//
        }
```

1. 애니메이션 재생

3) Play() 함수 적용 예

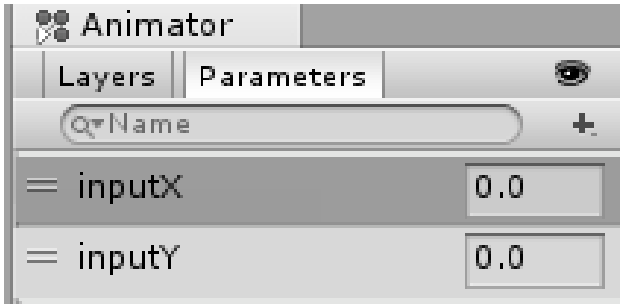
- 마우스 클릭 시 데미지 랜덤 처리

```
if (Input.GetMouseButtonDown (0)) {
    int random_n = Random.Range (0, 2);

    if (random_n == 0) {
        anim.Play ("DAMAGED00");
    } else {
        anim.Play ("DAMAGED01");
    }

}
```

2. Animation Blend

비슷하지만 방향성이 다른 동작, 혹은 비슷하지만 속도가 다른 동작을 섞어서 사용

1) Create new BlendTree in State

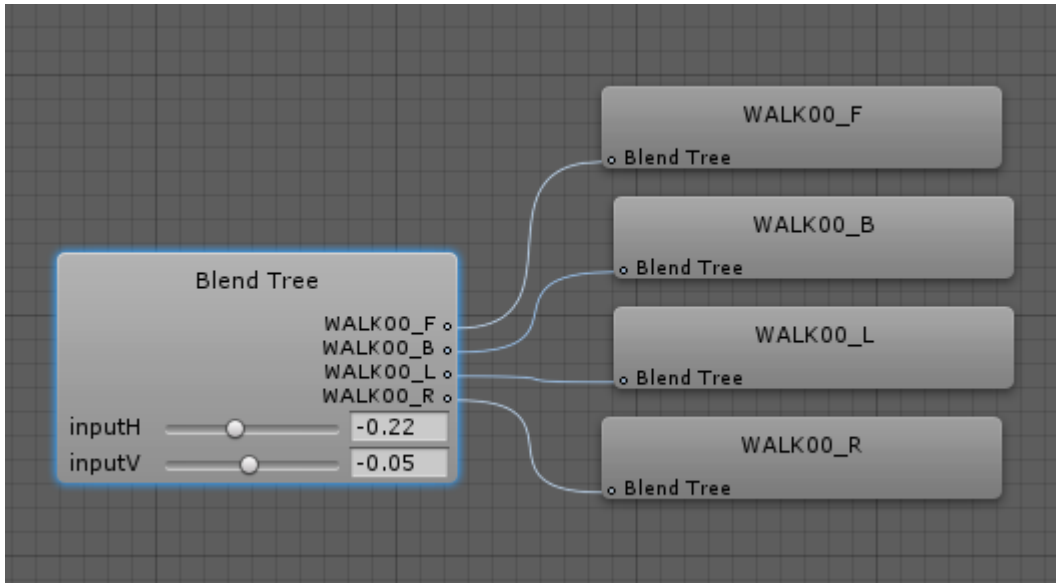


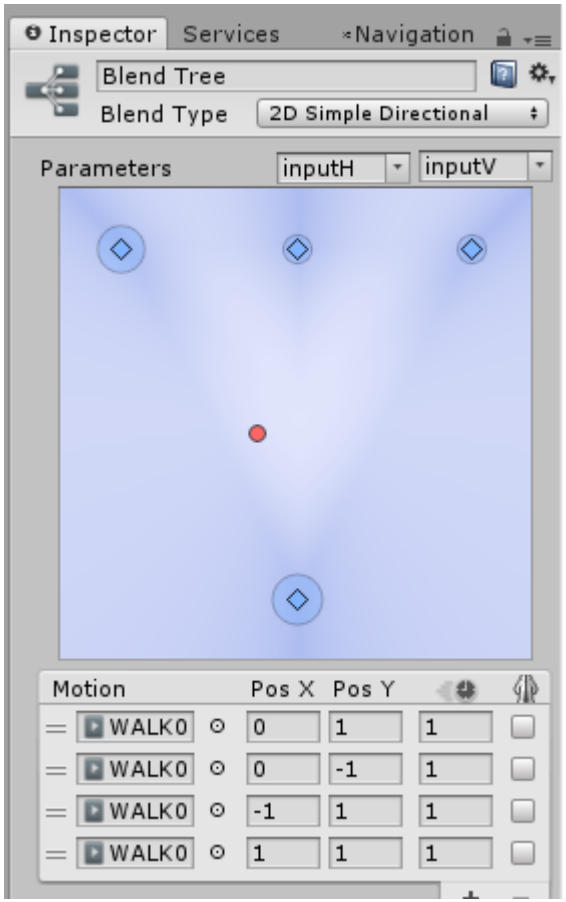

```
public class Player : MonoBehaviour
{
    public Animator anim;
    float inputX = 0;
    float inputY = 0;
    void Start ()
    {
        anim = GetComponent<Animator> ();
    }
    void Update ()
    {
        inputX = Input.GetAxis ("Horizontal");
        inputY = Input.GetAxis ("Vertical");
        anim.SetFloat ("inputX", inputX);
        anim.SetFloat ("inputY", inputY);
    }
}
```

2. Animation Blend

1) Create new BlendTree in State

- Walk 레이어 설정

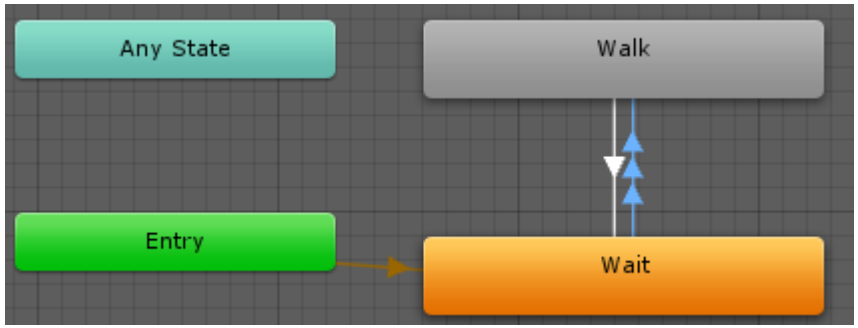


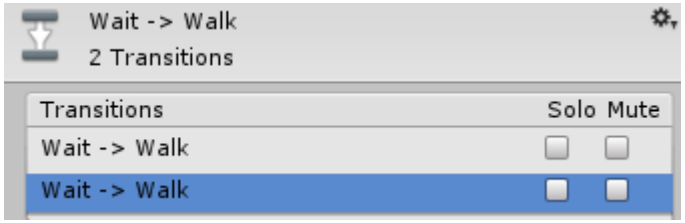


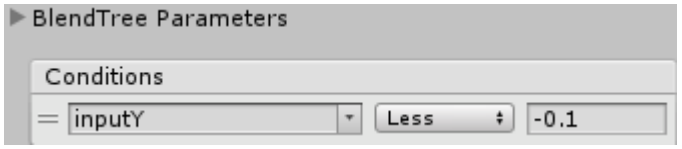
2. Animation Blend

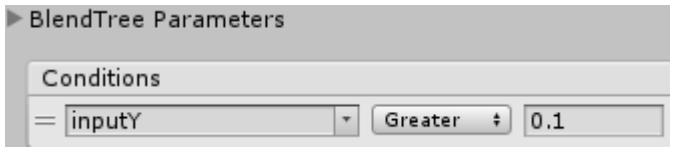
1) Create new BlendTree in State

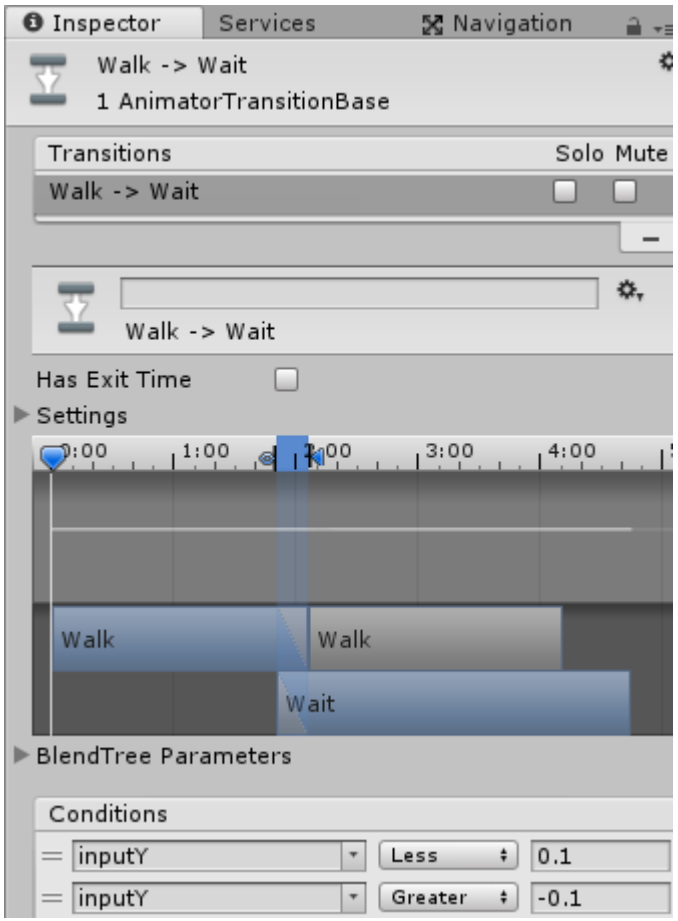
- 상태 전환 설정







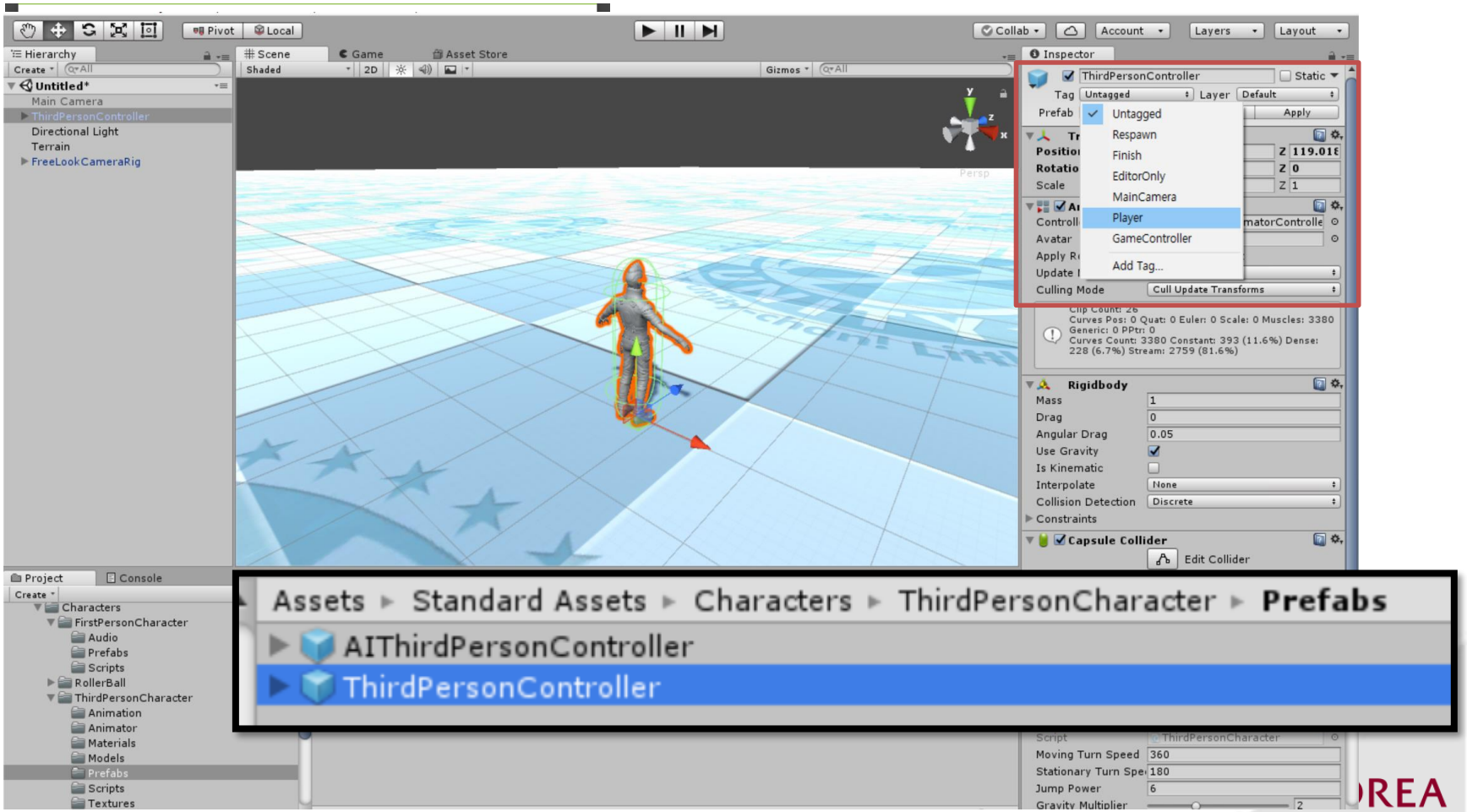




3. Retargeting animations

1) Ethan 캐릭터 import

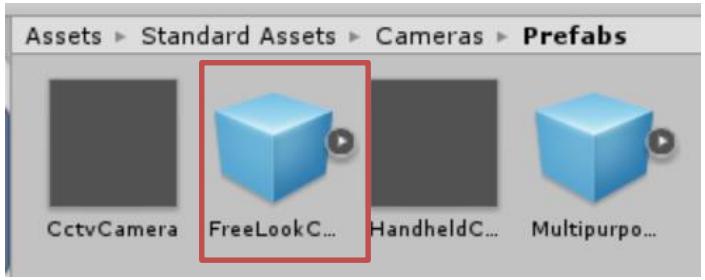
- ₩Assets₩Standard Assets₩Characters₩ThirdPersonCharacter₩Prefabs



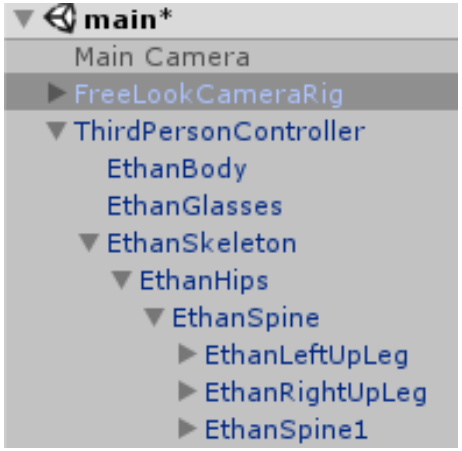
3. Retargeting animations

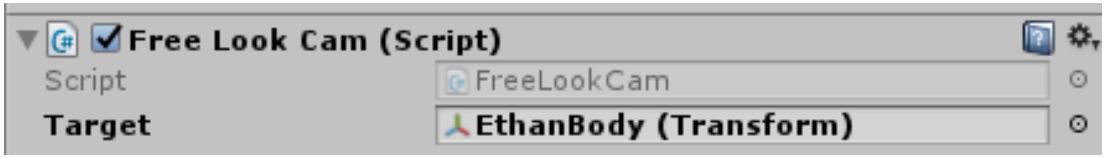
2) import Cameras

- ₩Assets₩Standard Assets₩Cameras₩Prefabs



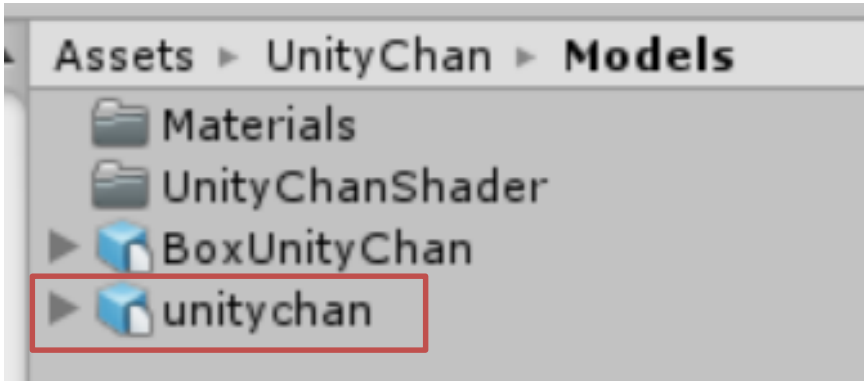
- 동적움직임에 따라 캐릭터를 추적하도록 Free Look Cam 스크립트에서 추적 대상을 지정
- 메인 카메라는 해제



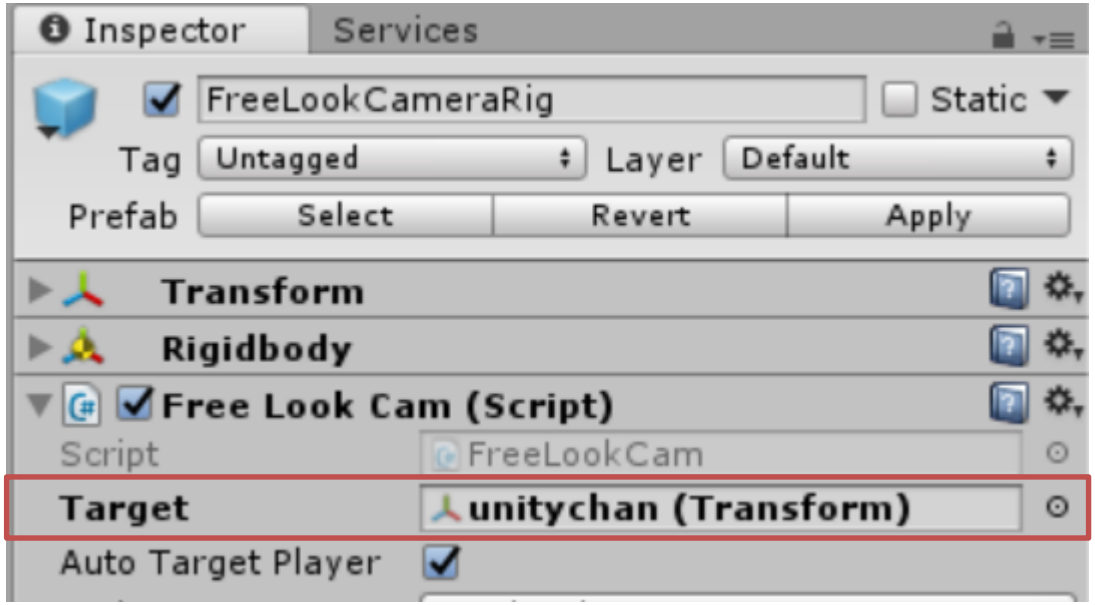


3. Retargeting animations

3) 유니티짱 캐릭터 import

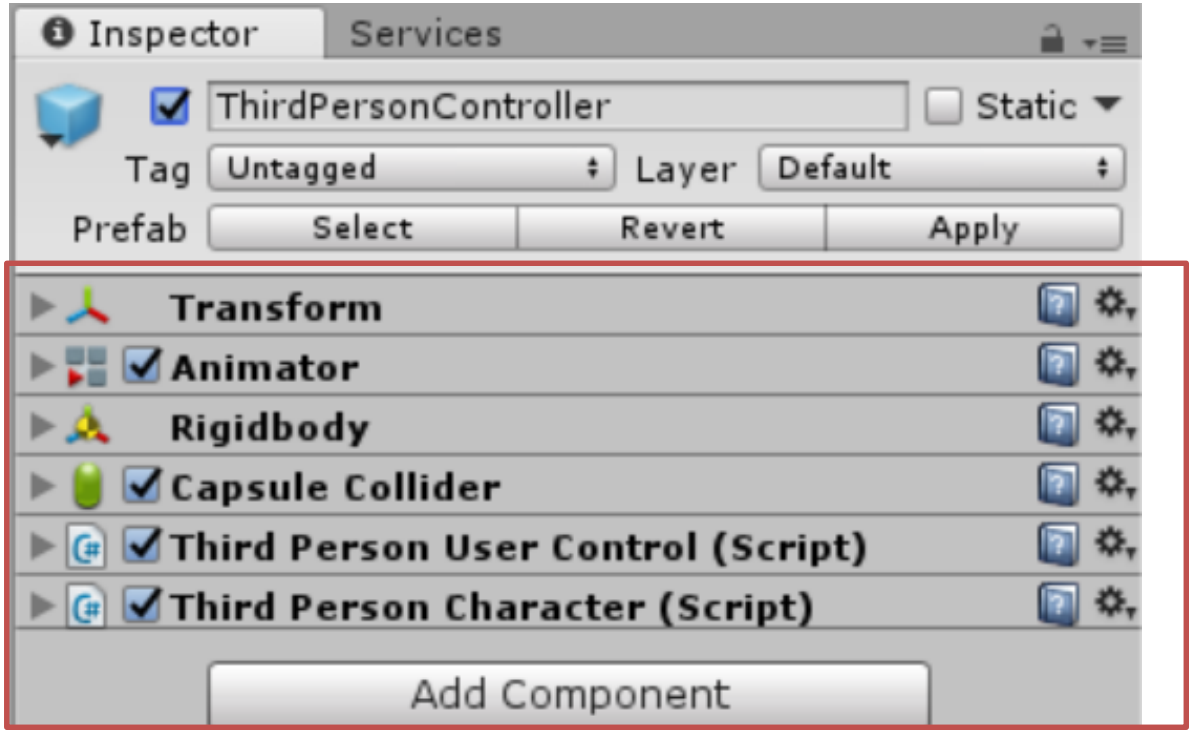


4) 카메라 타켓 변경



3. Retargeting animations

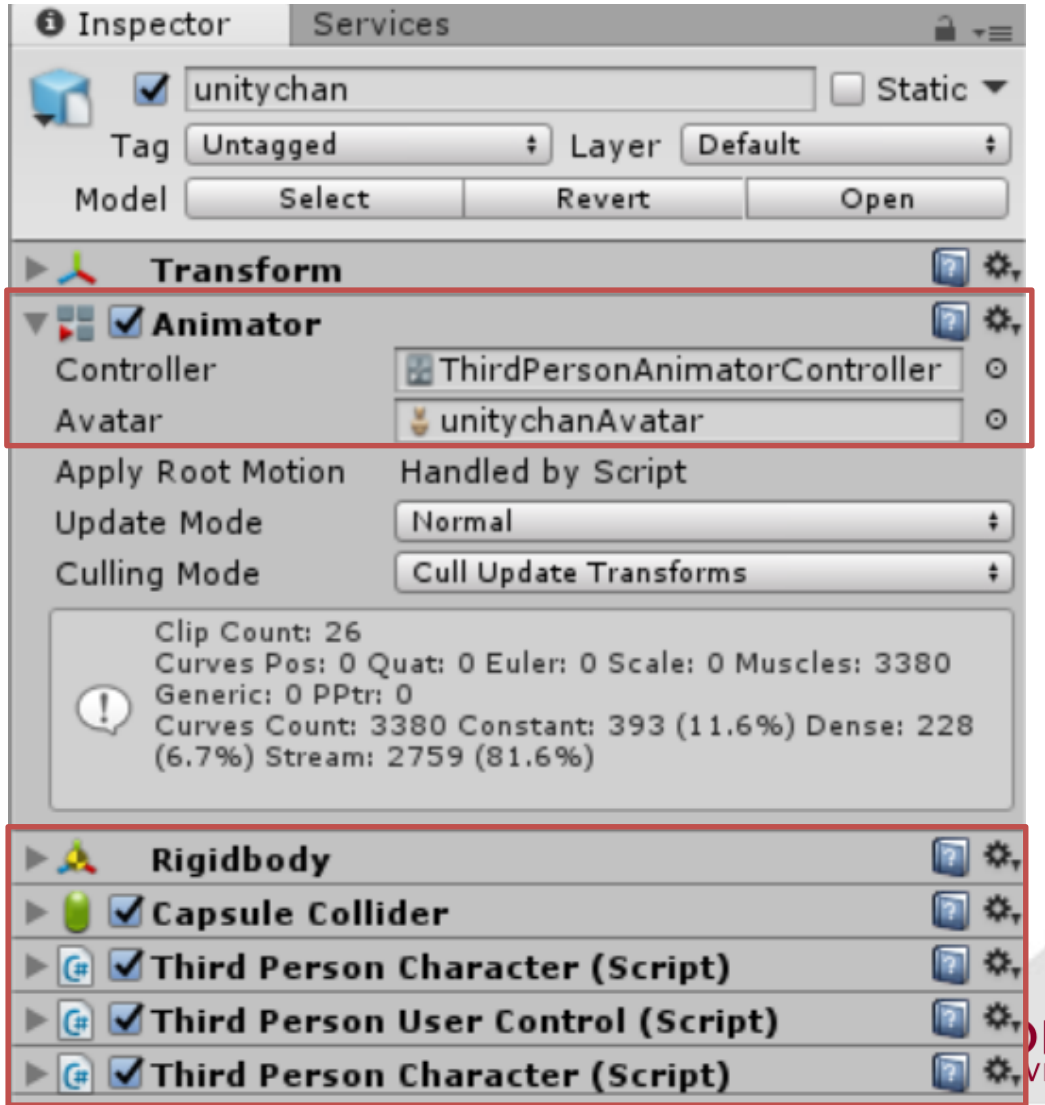
5) Ethan 캐릭터의 컴포넌트를 Unity-chan 캐릭터에 복재



3. Retargeting animations

5) Ethan 캐릭터의 컴포넌트를 Unity-chan 캐릭터에 복제

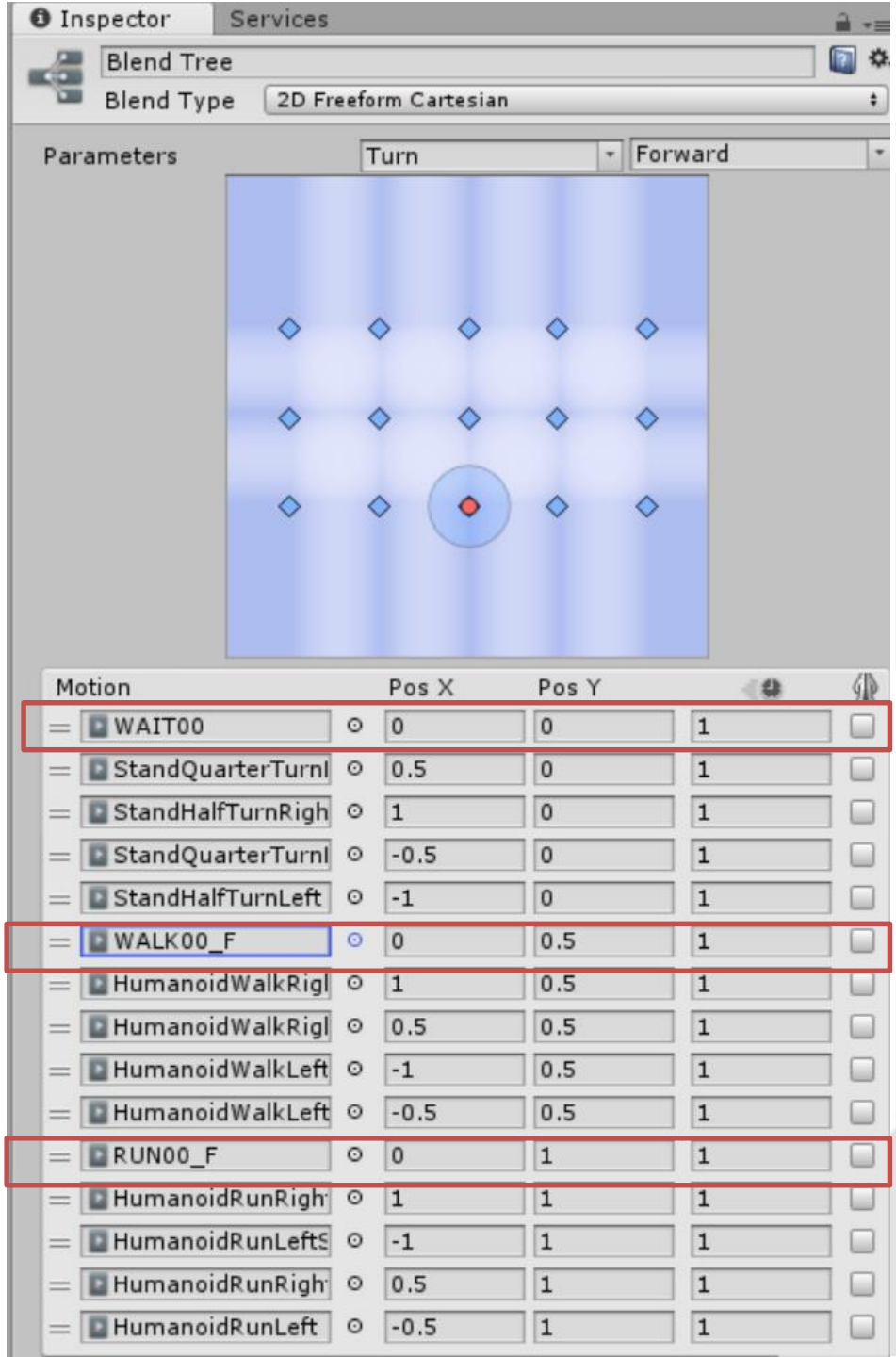
- Animator Controller
- Rigidbody
- Capsule collider
- Third Person Character
- Third Person User control
- Third Person character



3. Retargeting animations

6) Unity-chan 모션으로 수정

- 부자연스러운 Ethan의 모션을 유니티짱 모션으로 수정



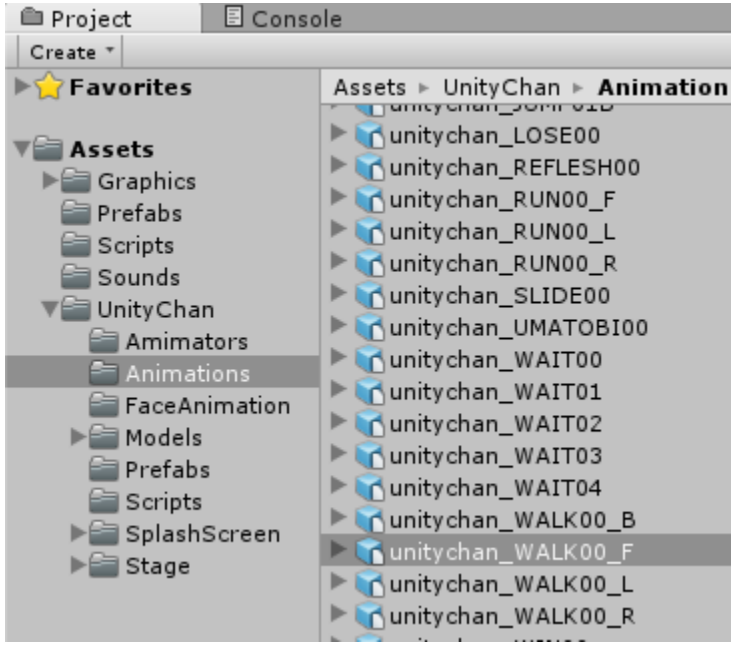
4. Animation Layer

아바타 마스크를 사용하여 애니메이션 위에 일정 부분에 애니메이션을 덮어씌우는 데 사용

1) 이동 모션 선택

- 기본으로 제공되는 모션외에 더 필요한 경우
- 사격모션과 같이 없는 모션은 새로 만들어서 사용해야 함

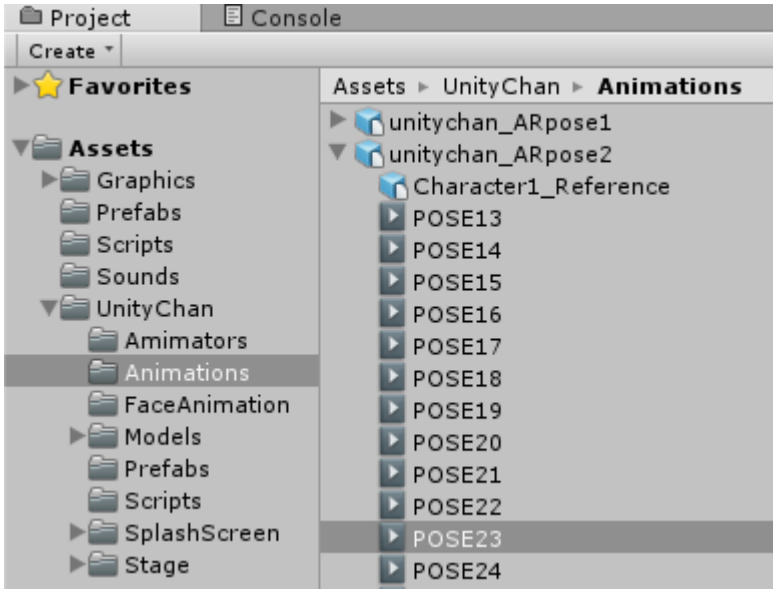
예) 앞으로 달리고, 뒤로 걷고, 정지상태 모습, 상체 총 쏘기 동작의 조합으로 앞으로 가면서 총쏘고, 뒤로 가며 총쏘기, 정지상태 총쏘기 모션 가능

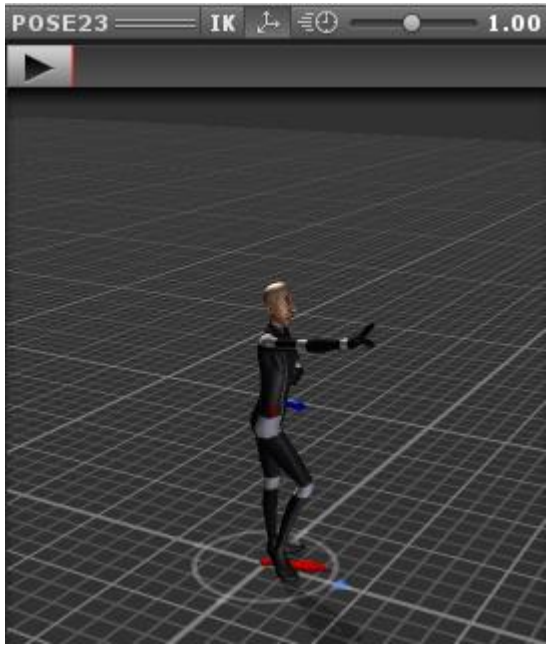


4. Animation Layer

2) 사격 모션 선택

- 기본 제공되는 포즈에서 사격 자세와 유사한 모션을 검색.
- Pose23번 모양이 총 쏘는 자세와 가장 유사하다. V자 손가락으로 팔을 내밀고 있는 자세
- 정지 자세에서만 총을 쏘는 포즈에 적합

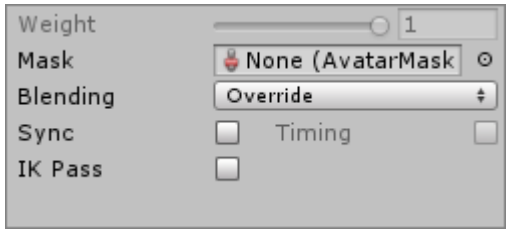




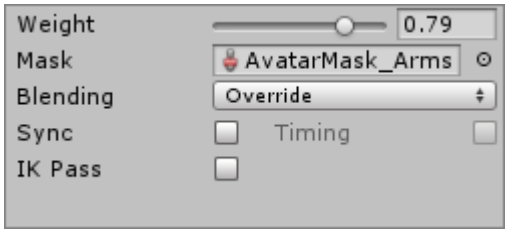
4. Animation Layer

3) 메카님 레이어 설정

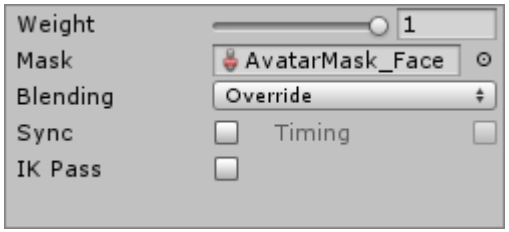
- 달리면서 총을 쏘는 자세를 구현하려면 인체 모델에 여러 모션(얼굴표정, 상체자세, 하체자세)을 합성해야 함
- 메카님에서 사용하는 애니메이션을 이용한 모션 만들기
  - Animator Controller [UnityChanGun] 을 생성하고 더블클릭
  - Animator Layers를 하체(base), 상체(팔), 얼굴 레이어로 나누어 생성



Base Layer



Arms Layer

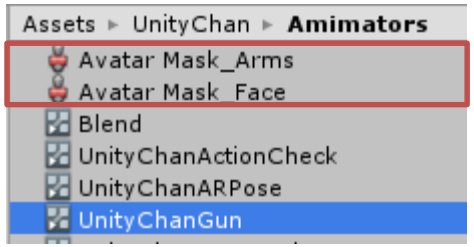


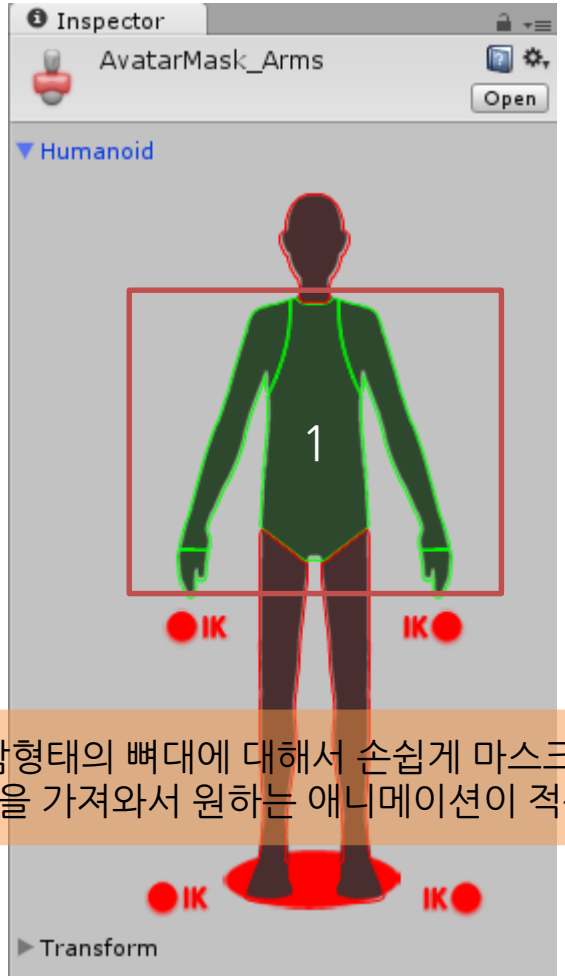
Face Layer

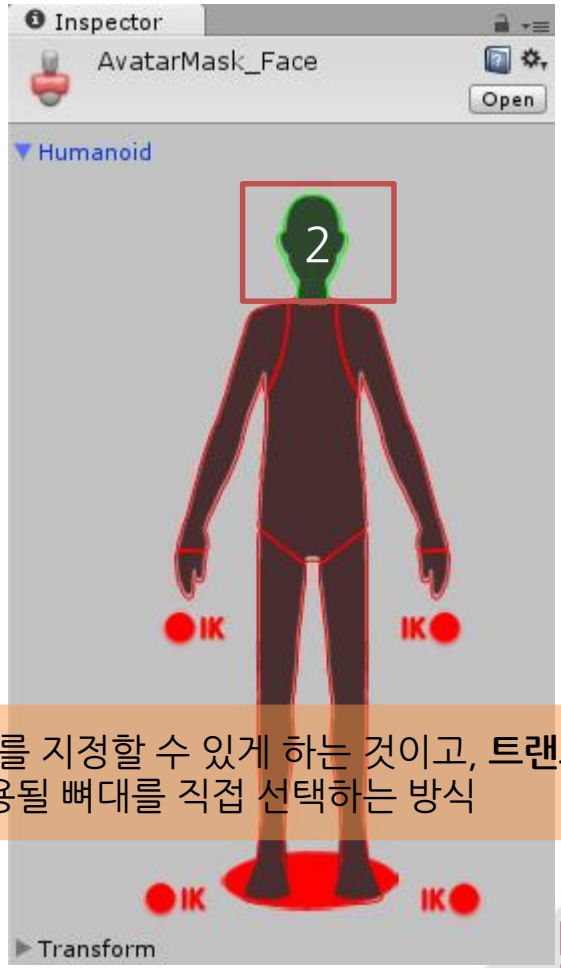
4. Animation Layer

4) 마스크를 이용한 애니메이션 처리.

- 아바타 마스크 2개 생성
- 팔과 얼굴 레이어 각각을 더블클릭하여 마스크하기





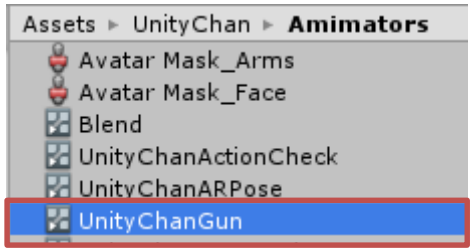


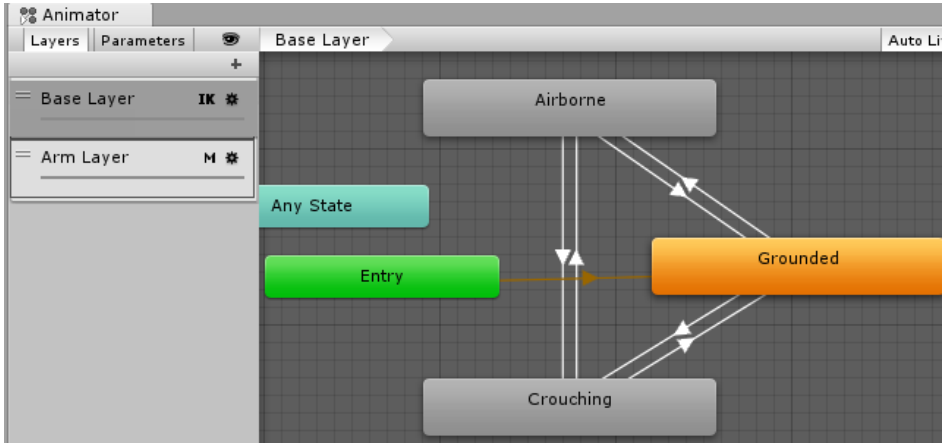
휴머노이드 방식은 일반적인 사람형태의 뼈대에 대해서 손쉽게 마스크를 지정할 수 있게 하는 것이고, **트랜스폼 방식**은 직접 만든 특별한 형태의 본을 가져와서 원하는 애니메이션이 적용될 뼈대를 직접 선택하는 방식

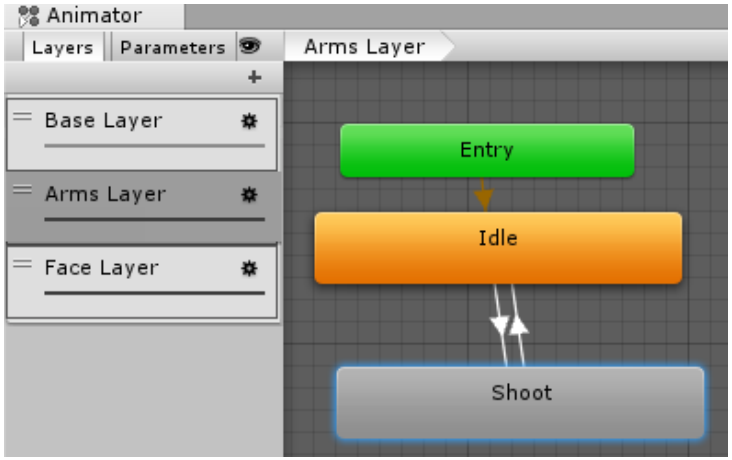
4. Animation Layer

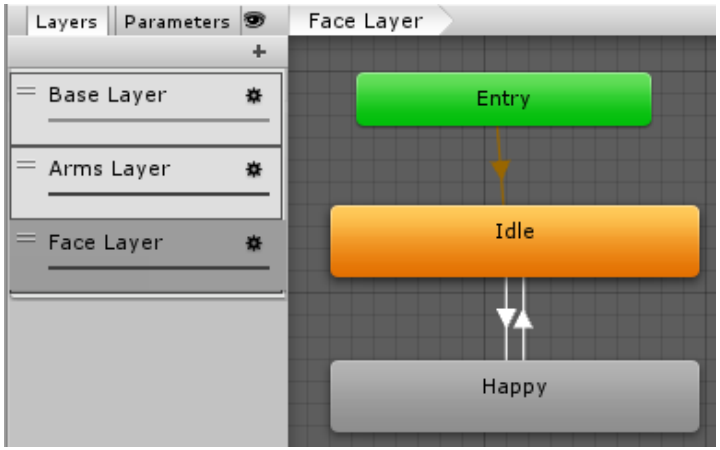
5) 상태전환 설정

- BaseLayer는 하체, 다리부분 모션으로 구성, Arms Layer는 정지상태와 슈팅상태, Face Layer는 행복한 얼굴과 평상시 얼굴로 구성함.





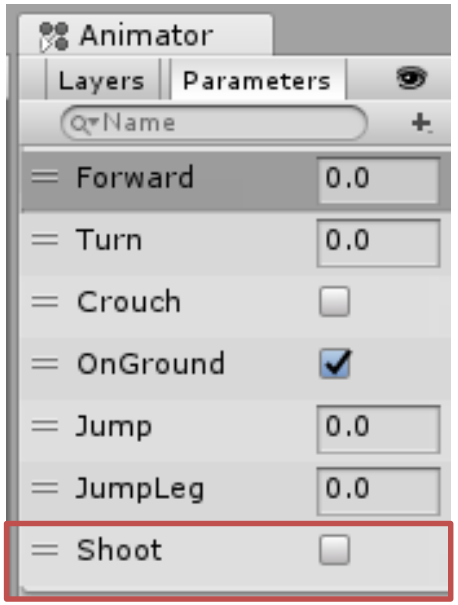




4. Animation Layer

6) 파라메터 설정하기

- 모션 상태를 다른 모션 상태로 변경시키는 파라메터
  - Ethan 캐릭터가 사용하는 컨트롤러를 그대로 활용
  - Bool형 Shoot 변수는 사격 버튼이 눌렀다는 사실을 알려주는 파라메터로 상체와 얼굴을 움직이는 스위치 역할을 수행하는 데 필요



4. Animation Layer

6) PlayerShoot.cs 스크립트 작성 (1/3)

- 마우스 클릭에 의해 슈팅이 되었을 경우 발사처리를 위한 스크립트

```
Animator  anim;
bool shootFlag;

public   GameObject   bulletObject = null;
//총알 프리팹
public   Transform    bulletStartPosition = null;
//총알 발사 위치

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

4. Animation Layer

6) PlayerShoot.cs 스크립트 작성 (2/3)

- 마우스 클릭에 의해 슈팅이 되었을 경우 발사처리를 위한 스크립트

```
private  void    Update ()
{
    if (Input.GetMouseButtonDown (0)) {  //사격 버튼(클릭) 체크
        shootFlag    = true;             //사격 처리

        //총알을 발사할 위치가 지정되어 있는지 여부를 검사
        if (null != bulletStartPosition) {
            //총알을 생성할 위치 지정
            Vector3 vecBulletPos = bulletStartPosition.position;
            //전진하는 방향으로 조금 진행
            vecBulletPos += (transform.rotation * Vector3.forward);
            //Y 높이를 적당히 올린다
            vecBulletPos.y = 1.0f;

  //총알을 생성(발사)
  Instantiate (bulletObject, vecBulletPos, transform.rotation);
```

4. Animation Layer

6) PlayerShoot.cs 스크립트 작성 (3/3)

- 마우스 클릭에 의해 슈팅이 되었을 경우 발사처리를 위한 스크립트

```
            }
        } else {
            //클릭되지 않은 경우 발사하지 않는다
            shootFlag    = false;
        }

        anim.SetBool ("Shoot", shootFlag);// 애니메이터에게 사격 알림

    }
```

4. Animation Layer

7) Bullet.cs 스크립트 작성

▪ 총알이 날아가는 스크립트

```
public class Bullet  : MonoBehaviour
{
    float bulletMoveSpeed = 7.0f;
    //1초 동안 총알이 나아가는 거리
    private        void    Update ()
    {
        //1초 동안 이동량
        Vector3 vecAddPos  = (Vector3.forward * bulletMoveSpeed);
        /* Vector3.forward는 new Vector3( 0f, 0f, 1f)와 같다
           Vector3에 transform.rotation을 곱하면 그 방향으로 향한다.
           이때 Vector3는 Z+ 방향을 정면으로 여긴다 */
        transform.position    += ((transform.rotation * vecAddPos)
 * Time.deltaTime);
        Destroy (gameObject, 5);
    }
}
```

4. Animation Layer

8) Bullet 프리펩 작성 및 스크립트 적용

- 인스펙트 설정