

이만택 - 2

UUserWidget 내에 AActor를 생성하지 않습니다.

액터는 게임 월드 내에 존재하는 오브젝트인 반면, 위젯은 서로 분리되어야 하는 사용자 인터페이스입니다.

CreateDefaultSubobject 는 주로 액터의 컴포넌트를 초기화하는 데 사용됩니다.

따라서 위젯 안에 액터를 생성하는 것이 덜 일반적입니다...

위젯이 생성하는 것은 일반적으로 UWidget에서 상속하는 객체입니다.

이것은 UUSMiniViewWidget에서 USceneCaptureComponent2D를 만들었던 이전 작업이 생각나는데, 원칙적으로 플레이어가 소유하고 텍스처만 전달하는 것이어야 합니다.

카메라 우선순위?

USceneCaptureComponent2D와 UCameraComponent는 모두 USceneComponent를 상속하므로,

UUserWidget 내에서 AActor을 생성 하진않음

액터는 게임 월드 내에 존재하는 객체이고 위젯은 사용자 인터페이스로 서로 분리되어야함.

CreateDefaultSubobject는 주로 액터의 컴포넌트를 초기화하는데 사용함

그래서 위젯 내에서 액터를 생성하는 일은 일반적이지 못한 일임..

위젯에서 생성하는 것들은 보통 UWidget을 상속받는 객체들

이걸 생각하면 이전 작업에서 UUSMiniViewWidget에 USceneCaptureComponent2D를 생성하였는데 원칙에 의하면 player 에서 가지고 있다가 텍스처만 넘겨주는 형태이어야 할듯

카메라 우선순위??

플레이어 컨트롤러->캐릭터->월드->기본카메라

아무것도 없는 빈 프로젝트로 디버깅해서 카메라를 알고싶었는데...

USceneCaptureComponent2D랑 UCameraComponent 둘다 USceneComponent를 상속받아서 사용하길래 장면에 관련된 컴포넌트인줄 알았는데 ULightComponentBase에도 사용되고...

찾아보니 위치, 회전, 크기 등을 사용하는 렌더링과 충돌 기능 없는 기본 컴포넌트

매쉬, 카메라, 파티클 등 공간적인 활동이 있으면 다 사용됨

USceneComponent 안에는 계층구조를 이루도록 변수들이 있음

```
1 TSharedPtr<USceneComponent> AttachParent;  
2 TArray<TSharedPtr<USceneComponent>> AttachChildren;
```

UPrimitiveComponent는 USceneComponent를 상속 받은 클래스로 렌더링과 충돌기능이 추가된 클래스

카메라

근데 카메라에 대한 우선순위에 대한 정보를 찾을 수 없었음...

로직중에 처음 월드에 컨트롤러가 생성 될 때 APlayerCameraManager가 생성 되는것을 발견

```
1 void APlayerController::SpawnPlayerCameraManager()
2 {
3     // servers and owning clients get cameras
4     // If no archetype specified, spawn an Engine.PlayerCameraManager. NOTE all games should specify an archety
5     FActorSpawnParameters SpawnInfo;
6     SpawnInfo.Owner = this;
7     SpawnInfo.Instigator = GetInstigator();
8     SpawnInfo.ObjectFlags |= RF_Transient; // We never want to save camera managers into a map
9     if (PlayerCameraManagerClass != NULL)
10    {
11        PlayerCameraManager = GetWorld()->SpawnActor<APlayerCameraManager>(PlayerCameraManagerClass, SpawnInfo);
12    }
13    else
14    {
15        PlayerCameraManager = GetWorld()->SpawnActor<APlayerCameraManager>(SpawnInfo);
16    }
17
18    if (PlayerCameraManager != NULL)
19    {
20        PlayerCameraManager->InitializeFor(this);
21    }
22    else
23    {
24        UE_LOG(LogPlayerController, Log, TEXT("Couldn't Spawn PlayerCameraManager for Player!!") );
25    }
26 }
27
```

그리고 APlayerCameraManager안에서는 UCameraModifier에서 우선순위 값을 비교함,,

```
1 void APlayerCameraManager::PostInitializeComponents()
2 {
3     Super::PostInitializeComponents();
4
5     // Setup default camera modifiers
6     TArray<TSubclassOf<UCameraModifier>> AllDefaultModifiers(DefaultModifiers);
7     TArray<ICameraModularFeature*> CameraModularFeatures = IModularFeatures::Get()
8         .GetModularFeatureImplementations<ICameraModularFeature>(ICameraModularFeature::GetModularFeatureName())
9     for (const ICameraModularFeature* CameraModularFeature : CameraModularFeatures)
10    {
11        if (ensure(CameraModularFeature))
12        {
13            CameraModularFeature->GetDefaultModifiers(AllDefaultModifiers);
14        }
15    }
16
17    if (AllDefaultModifiers.Num() > 0)
18    {
19        for (auto ModifierClass : AllDefaultModifiers)
```

```

20     {
21         // empty entries are not valid here, do work only for actual classes
22         if (ModifierClass)
23         {
24             UCameraModifier* const NewMod = AddNewCameraModifier(ModifierClass);
25
26             // cache ref to camera shake if this is it
27             UCameraModifier_CameraShake* const ShakeMod = Cast<UCameraModifier_CameraShake>(NewMod);
28             if (ShakeMod)
29             {
30                 CachedCameraShakeMod = ShakeMod;
31             }
32         }
33     }
34 }
35 }

```

```

1 UCameraModifier* APlayerCameraManager::AddNewCameraModifier(TSubclassOf<UCameraModifier> ModifierClass)
2 {
3     UCameraModifier* const NewMod = NewObject<UCameraModifier>(this, ModifierClass);
4     if (NewMod)
5     {
6         if (AddCameraModifierToList(NewMod) == true)
7         {
8             return NewMod;
9         }
10    }
11
12    return nullptr;
13 }

```

```

1 bool APlayerCameraManager::AddCameraModifierToList(UCameraModifier* NewModifier)
2 {
3     if (NewModifier)
4     {
5         // Look through current modifier list and find slot for this priority
6         int32 BestIdx = ModifierList.Num();
7         for (int32 ModifierIdx = 0; ModifierIdx < ModifierList.Num(); ModifierIdx++)
8         {
9             UCameraModifier* const M = ModifierList[ModifierIdx];
10            if (M)
11            {
12                if (M == NewModifier)
13                {
14                    // already in list, just bail
15                    return false;
16                }
17
18                // If priority of current index has passed or equaled ours - we have the insert location
19                if (NewModifier->Priority <= M->Priority)
20                {
21                    // Disallow addition of exclusive modifier if priority is already occupied
22                    if (NewModifier->bExclusive && NewModifier->Priority == M->Priority)
23                    {
24                        return false;
25                    }
26
27                    // Update best index

```

```

28         BestIdx = ModifierIdx;
29
30         break;
31     }
32 }
33 }
34
35 // Insert self into best index
36 ModifierList.InsertUninitialized(BestIdx, 1);
37 ModifierList[BestIdx] = NewModifier;
38
39 // Save camera
40 NewModifier->AddedToCamera(this);
41 return true;
42 }
43
44 return false;
45 }

```

그치만 UCameraComponent를 생성할 때 같이 생성이 안되는것으로 볼때 CreateDefaultSubobject<UCameraComponent>에서 연결되는것 같은데 연결 지점을 못찾음.

그렇지만 ULocalPlayer에서 PlayerCameraManager에 대한 호출을 계속 하고있음..

코드에 있는 NewModifier 는 화면 효과에 대한 우선순위로 카메라 우선순위가 아님

UCameraComponent를 붙이면 PlayerCameraManager와 연결되며 auto activate쪽을 살펴볼것



ULocalPlayer란 무엇인가..? [🔗](#)

[📄](#) ULocalPlayer [🔗](#)

- ULocalPlayer 는 로컬 플레이어의 정보와 상태를 관리하는 데 사용됩니다. 이는 플레이어의 입력 처리, 플레이어와 관련된 설정, 그리고 플레이어의 게임 세션 정보 등을 포함할 수 있습니다.
- ULocalPlayer 는 플레이어의 인터페이스와 직접적인 상호작용을 관리하는 데 중점을 둡니다. 예를 들어, 사용자 입력, 화면 설정, UI/UX 상호작용 등이 이에 해당합니다.
- 멀티플레이어 게임에서 각 로컬 플레이어는 자신만의 ULocalPlayer 인스턴스를 가질 수 있습니다.

[🔗](#) ACharacter [🔗](#)

- ACharacter 는 게임 내에서 플레이어나 NPC(Non-Player Character)가 조종할 수 있는 캐릭터를 구현하는 데 사용되는 클래스입니다. 이 클래스는 이동과 관련된 기능(예: 걷기, 뛰기, 점프하기 등)을 포함하고 있으며, 캐릭터의 물리적 상호작용과 애니메이션을 관리합니다.
- ACharacter 는 Unreal Engine의 Actor 시스템의 일부로, 게임 세계 내에서 시각적으로 표현되는 모든 엔티티의 기본이 되는 AActor 클래스를 상속받습니다.
- 캐릭터는 게임 플레이와 직접적인 상호작용을 위한 물리적인 존재를 나타냅니다. 예를 들어, 플레이어 캐릭터나 AI가 조종하는 캐릭터 등이 이에 해당합니다.

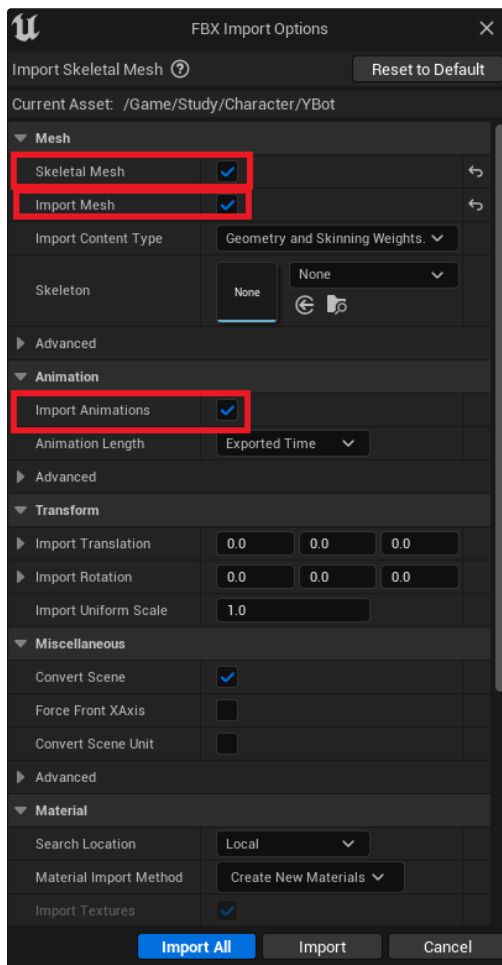
요약하자면, ULocalPlayer 는 플레이어의 세션과 인터페이스 관련 정보를 다루는 반면, Acharacter 는 게임 세계 내에서 플레이어 나 NPC의 물리적인 표현과 관련된 기능을 담당합니다. 두 클래스는 서로 다른 목적으로 설계되었으며, Unreal Engine 내에서 각각의 역할을 수행합니다.

애니메이션 사고싶었는데...  Mixamo 라는 사이트를 발견함

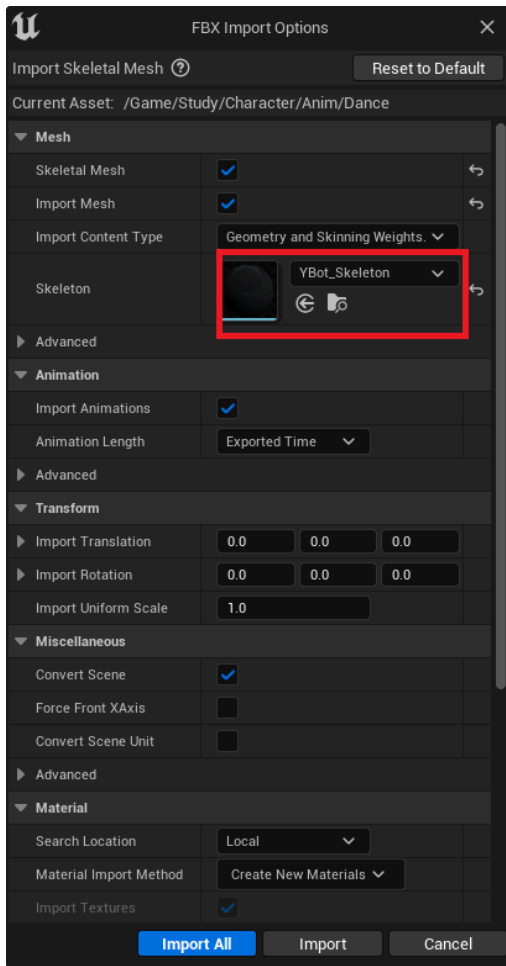
어도비가 인수했다함

계정 만들어서 다운받으면됨

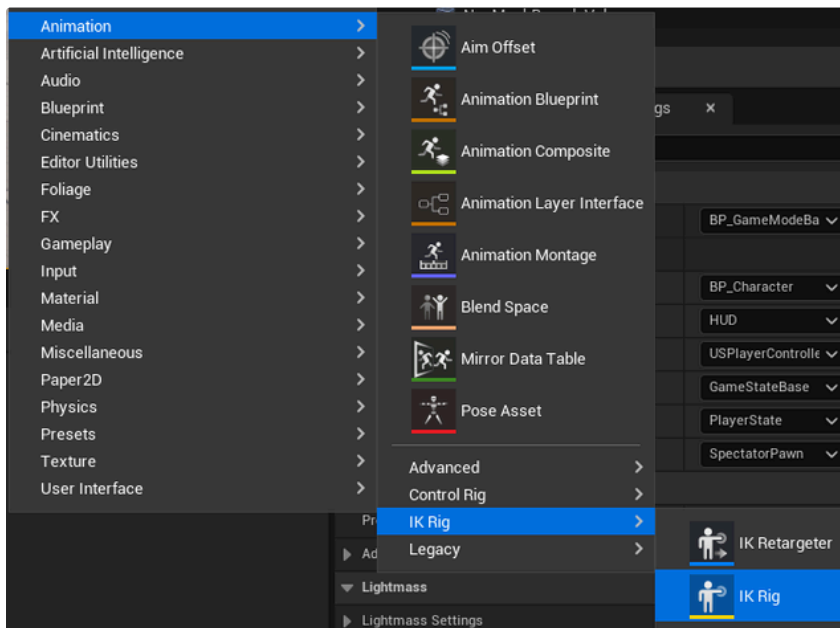
YBot을 다운받아서 content Browser에 드래그 하면 이창이 뜨는데 아래 3가지를 체크하고 Import All을 클릭



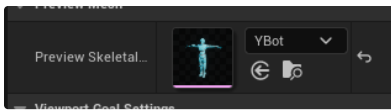
애니메이션도 다운받아서 드래그하면서 스켈레톤을 확인



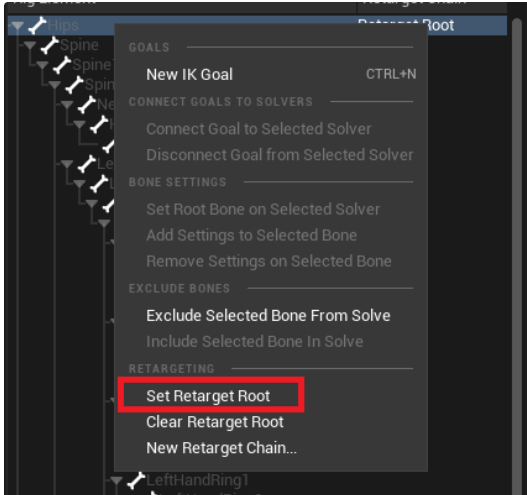
애니메이션 리타겟팅



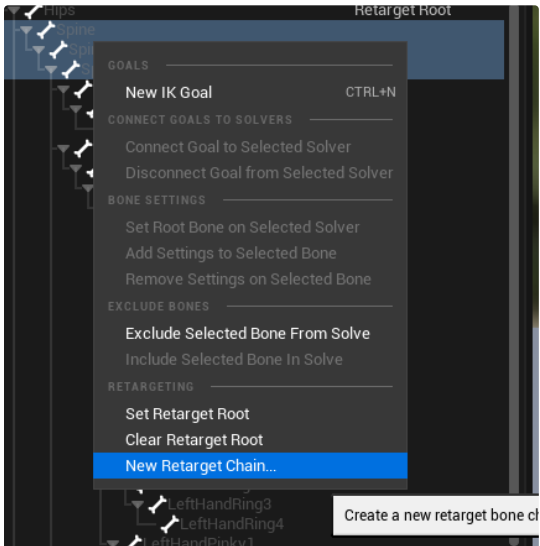
만든 IK Rig을 더블클릭하여 preview skeletal에 아까다 운받은 스켈레톤 매쉬 설정



가장 위에 있는 본을 루트로 설정



Spine를 모두 클릭후 New Retarget Chain 설정

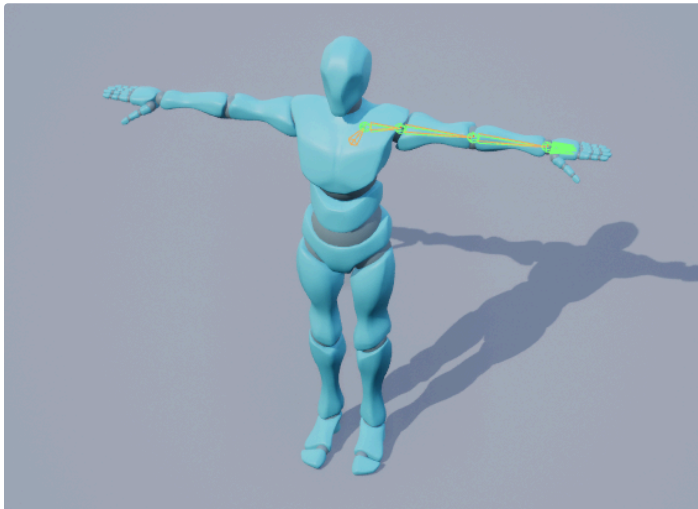


같은 방법으로 Neck과 Head를 클릭해서 New Retarget Chain 설정

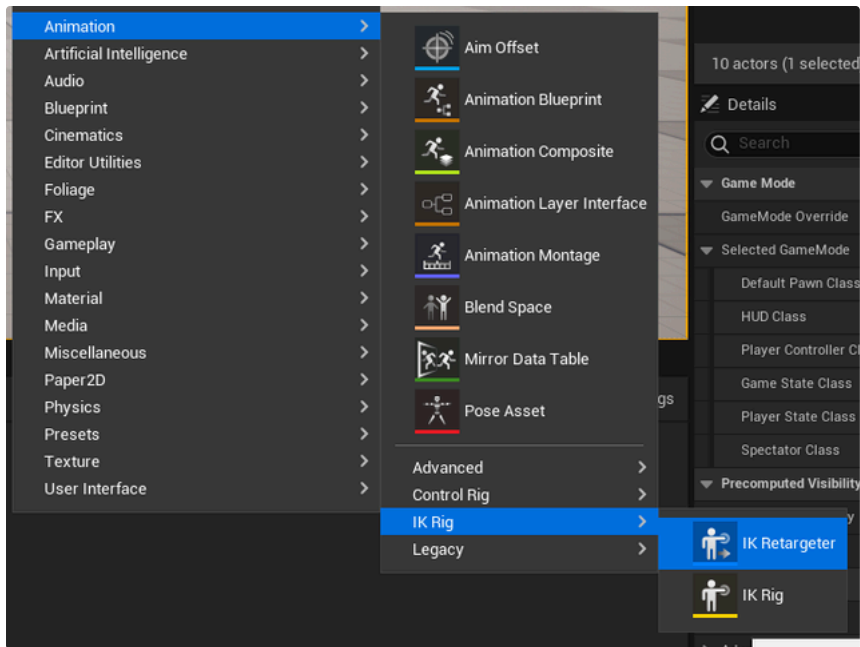
팔, 다리, 손가락도 동일하게 리타겟팅 설정

Rig Element	Retarget Chain
Hips	Retarget Root
Spine	Spine
Spine1	Spine
Spine2	Spine
Neck	Head
Head	Head
HeadTop_End	
LeftShoulder	LeftArm
LeftArm	LeftArm
LeftForeArm	LeftArm
LeftHand	LeftArm
LeftHandThumb1	LeftThumb
LeftHandThumb2	LeftThumb
LeftHandThumb3	LeftThumb
LeftHandThumb4	LeftThumb
LeftHandIndex1	LeftIndex
LeftHandIndex2	LeftIndex
LeftHandIndex3	LeftIndex
LeftHandIndex4	LeftIndex
LeftHandMiddle1	LeftMiddle
LeftHandMiddle2	LeftMiddle
LeftHandMiddle3	LeftMiddle
LeftHandMiddle4	LeftMiddle
LeftHandRing1	LeftRing
LeftHandRing2	LeftRing
LeftHandRing3	LeftRing
LeftHandRing4	LeftRing
LeftHandPinky1	LeftPinky
LeftHandPinky2	LeftPinky
LeftHandPinky3	LeftPinky
LeftHandPinky4	LeftPinky
RightShoulder	RightArm
RightArm	RightArm
RightForeArm	RightArm
RightHand	RightArm
RightHandThumb1	RightThumb
RightHandThumb2	RightThumb
RightHandThumb3	RightThumb
RightHandThumb4	RightThumb
RightHandIndex1	RightIndex
RightHandIndex2	RightIndex
RightHandIndex3	RightIndex

분리는 뼈가 이렇게 보여서 보면서 분리하면 됨

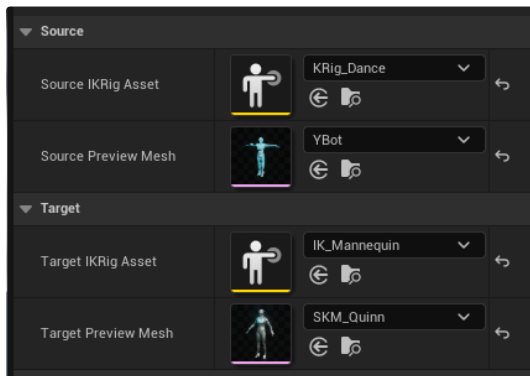


이제 리타겟팅 하기 위해서 리타게터를 만들

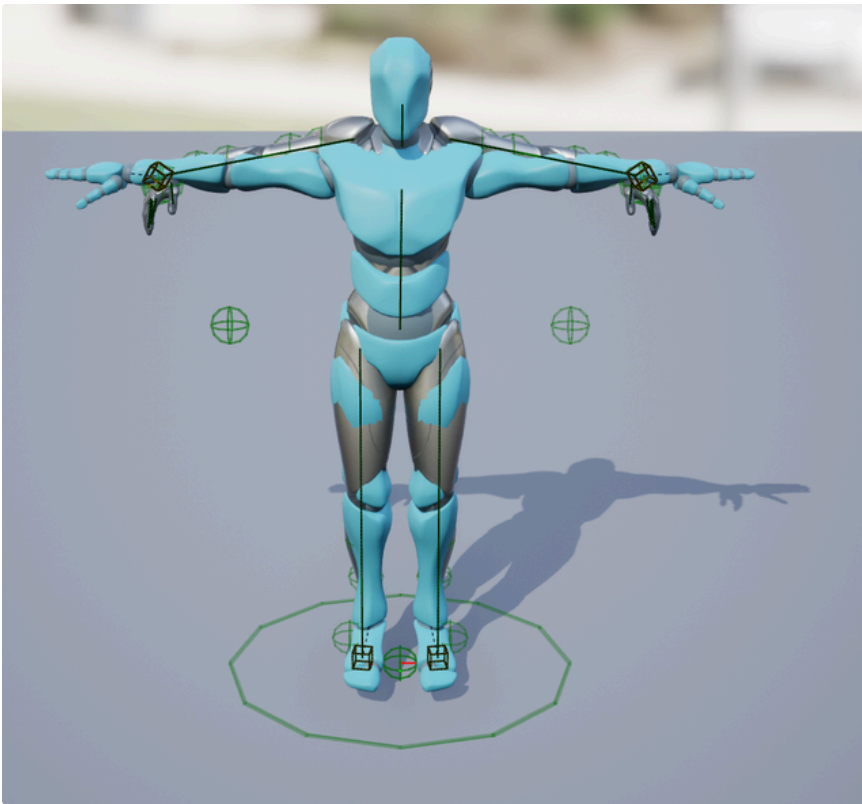


Source에는 만들고있는 리타겟팅

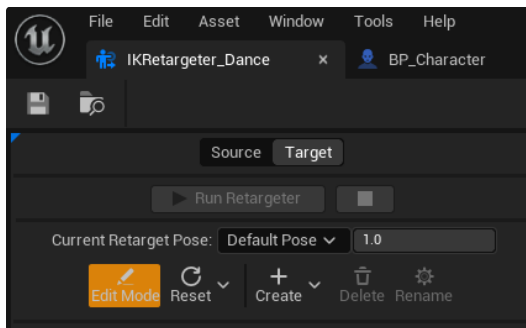
Target에는 애니메이션을 넣을 캐릭터



2개의 포즈가 다르기 때문에 오차를 줄이기 위해 위치를 조절함



Edit Mode로 들어감



UI 구성이 좀 다를 수 있음 찾아봤던 유튜브에선 다른 키를 눌렀음

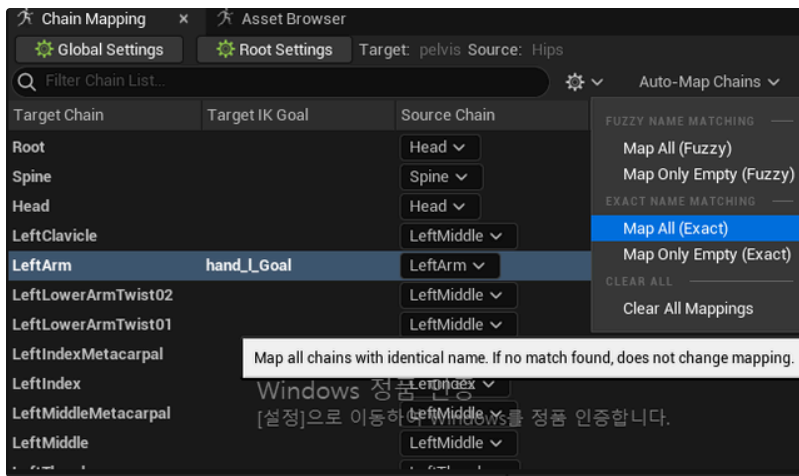
본같은게 사라지고 아래 그림처럼 나오면 됨



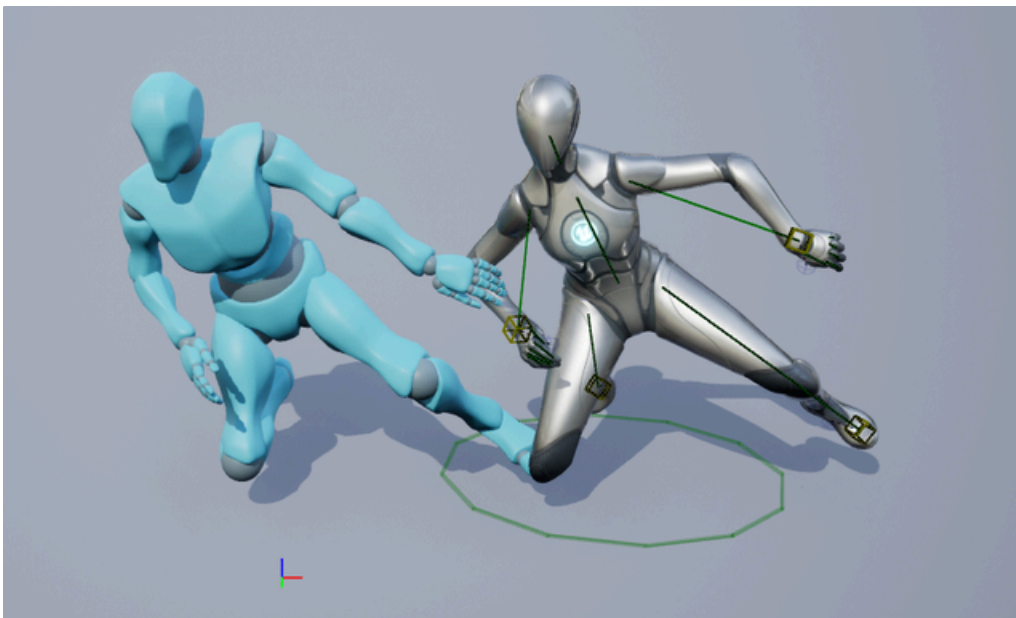
본을 잘 회전 시켜서 최대한 비슷하게 맞춰줌



chain Mpapping - Map All(exact) 클릭하면 끝



팔꿈치가 안맞는다.....



근데 팔꿈치가 안맞아서.. 다른 걸 찾아보니 5.4버전에서는 위에 과정 필요 없이 자동으로 해준다?

추가적인 이야기...

5.2버전이길래 5.3으로 버전업을 했더니.. 실패

```
1 C:\Program Files\Epic Games\UE_5.3\Engine\Source\Programs\AutomationTool\BuildGraph\BgScriptReader.cs(1640,37,164
2 3>"BuildGraph.Automation.csproj" 프로젝트를 빌드했습니다. - 실패
3 C:\Program Files\Epic Games\UE_5.3\Engine\Source\Programs\AutomationTool\Scripts\CheckForHacks.cs(87,26,87,45): e
```

[Error compiling the Automation Tool after updating Visual Studio today \(Unreal 5.3.2\)](#)

해결책을 보니 엔진코드 변경하라는데.. 이게 맞나..?

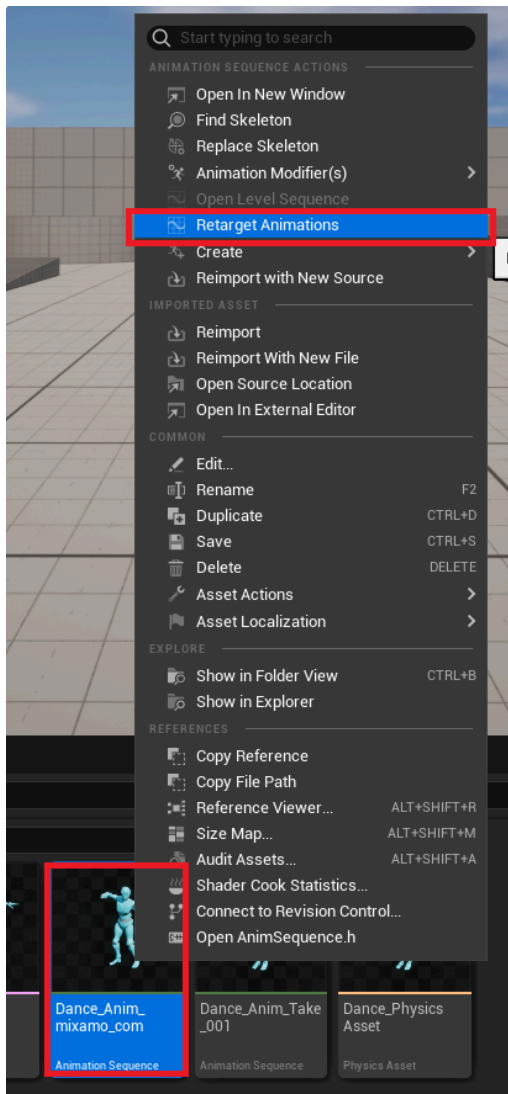
그리고 다음 문제..

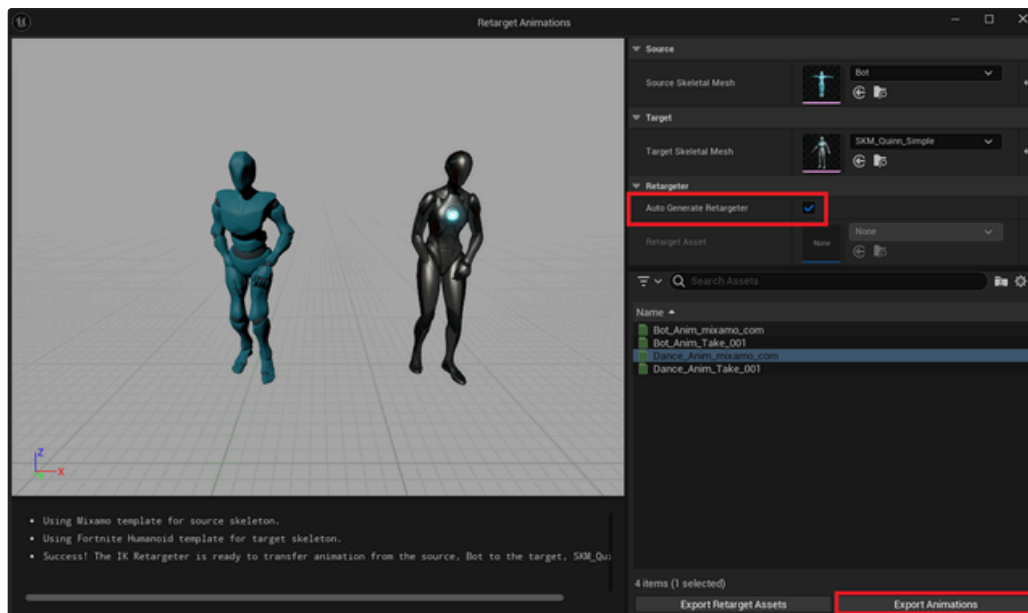
```
1 2>C:\Program Files\Epic Games\UE_5.3\Engine\Source\predefined C++ types (compiler internal)(420): error C2248: 'F
```

🤖 Can anyone help me with this Unreal 5 Error?

MSVC v143 - VS 2022 C++ x64/x86 build tools (v14.36-17.6) 설치필요

5.4 자동 리타겟팅





애니메이션 선택하고 익스포트 하면 끝 (진짜 딸깍)

MyProject5 Unreal Editor 2024 04 21 14 25 19



