


2단계_05. 캐릭터 콤보 액션

📅 수강일	@ 2023/06/30
➦ 이름	 전영재
🔍 멘토	Min-Kang Song, 현웅 최
👥 멘티	
✨ 작성 상태	Done
🕒 단계	
☑ 강의 시청 여부	<input checked="" type="checkbox"/>
☑ 이수 여부	<input type="checkbox"/>

Contents



캐릭터 콤보 액션

[1] 애니메이션 몽타주

[2] 콤보 공격

0n. 캐릭터 콤보 액션 강의 과제

캐릭터 콤보 액션



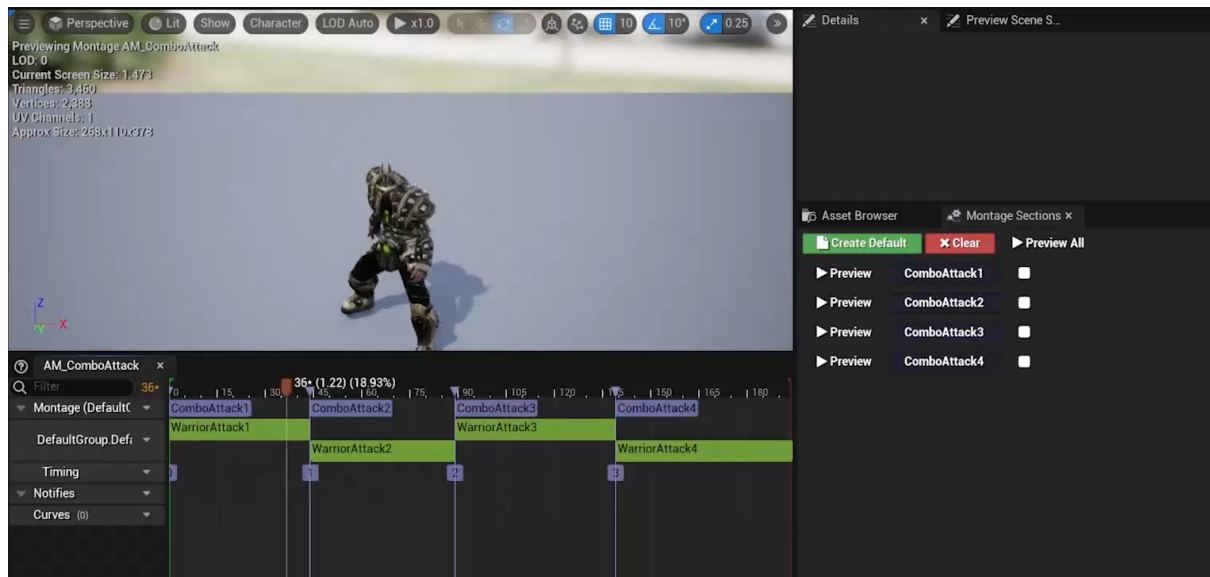
강의 목표

- 애니메이션 몽타주 시스템의 활용
- 데이터 애셋과 델리게이트를 활용한 콤보 공격의 구현

[1] 애니메이션 몽타주

몽타주 : 이미지 일부를 잘라내 한 화면에서 합성하는 기법

그러므로 애니메이션 몽타주는 애니메이션 클립을 합성해 원하는 애니메이션을 합성한다고 볼 수 있다.

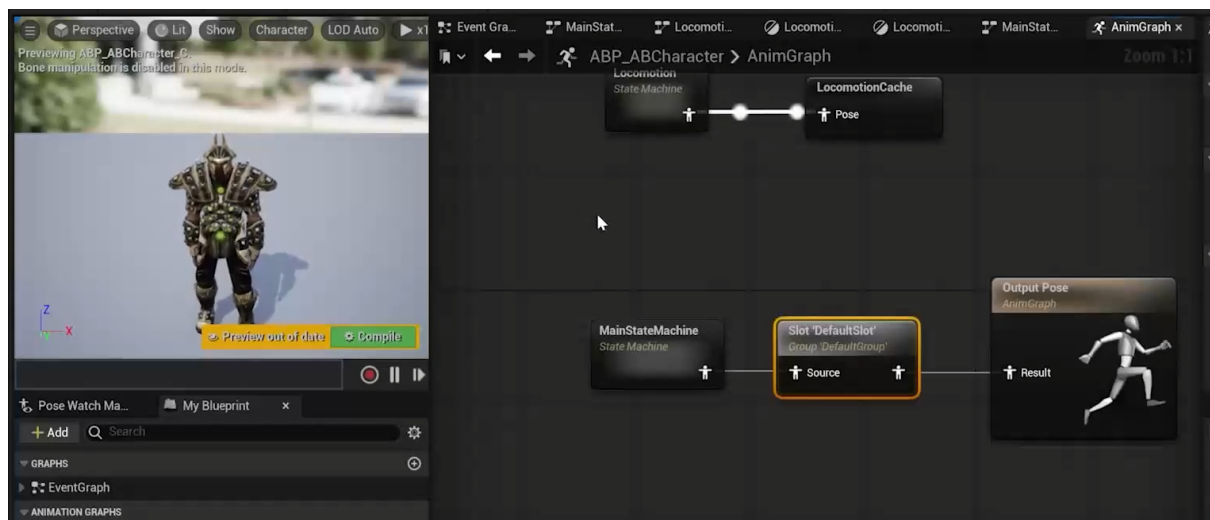


다음과 같이 여러 애니메이션들을 섹션별로 구분해 사용할 수 있다.

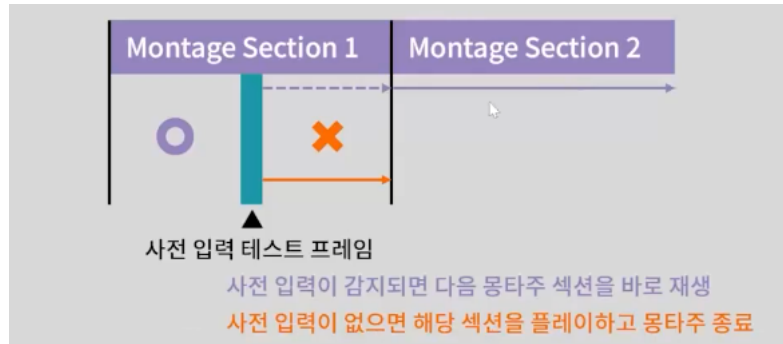
이를 활용해 입력이 들어올때, 특정 섹션을 플레이하는 방식(`AnimInstance->Montage_Play()`)으로 하나의 애셋에서 다양한 모션을 출력할 수 있다.

캐릭터 애니메이션에 사용할 몽타주를 할당해줬다면, 몽타주가 실행 될 슬롯을 할당해줘야한다.

그렇게되면 앞선 애니메이션 위에 몽타주가 덮어씌워져서 출력된다.

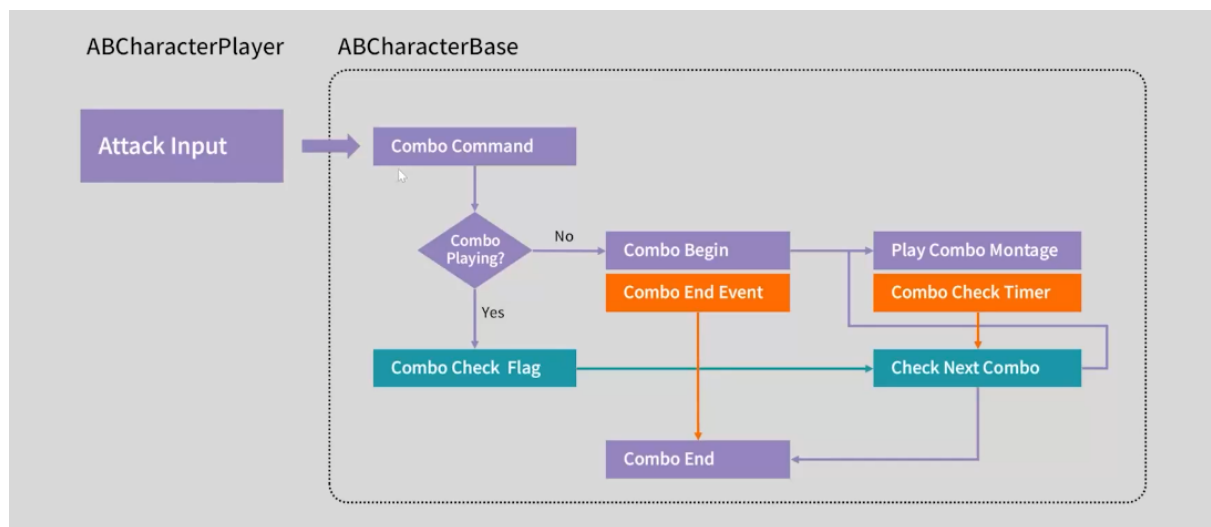


[2] 콤보 공격



앞서 몽타주의 섹션을 나누는 법을 보았다. 이를 더욱 발전시키면 플레이어의 입력 타이밍에 맞춰 새로운 섹션을 플레이해, 매 공격마다 다른 모션을 보여주는 콤보 공격을 구현할 수 있다.

구현 방법은 간단하다. 특정 프레임 구간 전에 입력이 들어온다면 다음 몽타주 섹션을 재생하고 그렇지 않다면 현재 섹션을 플레이 한 뒤 종료하면 된다.



콤보 공격의구체적인 구성도

이런 콤보 섹션, 현재 콤보수, 재생속도에 대한 정보는 앞선 강의에서 배운 **DataSet**을 사용하면 편리하다

```
ABComboActionData.cpp
UCLASS()
class ARENABATTLE_API UABComboActionData : public UPrimaryDataAsset
{
    GENERATED_BODY()

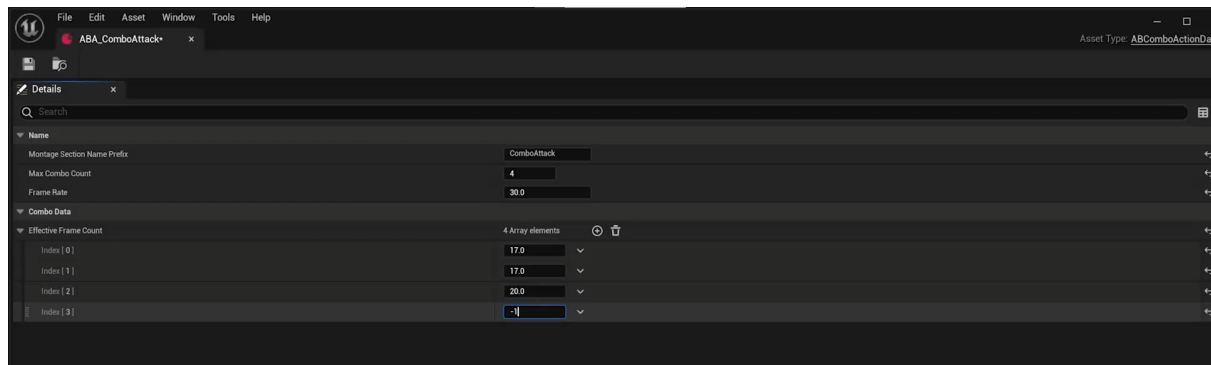
public:
    UABComboActionData();

    UPROPERTY(EditAnywhere, Category = Name)
    FString MontageSectionNamePrefix;

    UPROPERTY(EditAnywhere, Category = Name)
    uint8 MaxComboCount;

    UPROPERTY(EditAnywhere, Category = Name)
    float FrameRate;

    UPROPERTY(EditAnywhere, Category = ComboData)
    TArray<float> EffectiveFrameCount;
};
```



에디터에서 쉽게 만들수 있다.

몽타주의

```

ABCharacterBase.cpp
void AABCharacterBase::ProcessComboCommand()
{
    if (CurrentCombo == 0)
    {
        ComboActionBegin();
        return;
    }

    if (!ComboTimerHandle.IsValid()) //타이머보다 늦게 입력이 들어왔다면 다음 콤보를 불러오지 못하게 HasNextComboCommand를 false로 설정한다.
    {
        HasNextComboCommand = false;
    }
    else //타이머 내에 성공적으로 입력했다면 다음 콤보가 가능하게 bool값을 true로 설정한다.
    {
        HasNextComboCommand = true;
    }
}

void AABCharacterBase::SetComboCheckTimer()
{
    //타이머를 사용해 원하는 타이밍에 특정 함수를 호출할 수 있다. 지금 같은 경우는 유효한 Frame범위 내에 입력을 받는다면 ComboCheck를 불러오는 구조다.
    int32 ComboIndex = CurrentCombo - 1;
    ensure(ComboActionData->EffectiveFrameCount.IsValidIndex(ComboIndex));

    const float AttackSpeedRate = 1.0f;
    float ComboEffectiveTime = (ComboActionData->EffectiveFrameCount[ComboIndex] / ComboActionData->FrameRate) / AttackSpeedRate;
    if (ComboEffectiveTime > 0.0f)
    {
        GetWorld()->GetTimerManager().SetTimer(
            ComboTimerHandle,
            this,
            &AABCharacterBase::ComboCheck,
            ComboEffectiveTime,
            false
        );
    }
}

void AABCharacterBase::ComboCheck()
{
    ComboTimerHandle.Invalidate();
    if (HasNextComboCommand)
    {
        UAnimInstance* AnimInstance = GetMesh()->GetAnimInstance();

        CurrentCombo = FMath::Clamp(CurrentCombo + 1, 1, ComboActionData->MaxComboCount);
        FName NextSection = *FString::Printf(TEXT("%s%d"), *ComboActionData->MontageSectionNamePrefix, CurrentCombo);
        AnimInstance->Montage_JumpToSection(NextSection, ComboActionMontage); //실질적으로 Section별 몽타주를 출력하는 코드
        SetComboCheckTimer(); //이번 콤보는 성공했으니 다음콤보를 위한 타이머도 다시 재생해야된다.
        HasNextComboCommand = false;
    }
}

void AABCharacterBase::ComboActionBegin()
{
    // Combo Status
    CurrentCombo = 1;

    // Movement Setting
    GetCharacterMovement()->SetMovementMode(EMovementMode::MOVE_None); //None으로 지정해주면 캐릭터가 움직이지 않는다!

    // Animation Setting

```

```

const float AttackSpeedRate = 1.0f;
UAnimInstance* AnimInstance = GetMesh()->GetAnimInstance();
AnimInstance->Montage_Play(ComboActionMontage, AttackSpeedRate);

FOnMontageEnded EndDelegate; //AnimInstance에는 몽타주 종료에 Delegate를 사용하며, 양식은 AnimInstance.h에서 확인할 수 있었다.
EndDelegate.BindUObject(this, &AABCharacterBase::ComboActionEnd);
AnimInstance->Montage_SetEndDelegate(EndDelegate, ComboActionMontage);

ComboTimerHandle.Invalidate();
SetComboCheckTimer(); //콤보의 시작과 동시에 타이머를 작동한다.
}

void AABCharacterBase::ComboActionEnd(UAnimMontage* TargetMontage, bool IsProperlyEnded) //AnimMontage.h에 있는 OnMontageEnded라는 델리게이트
{
    ensure(CurrentCombo != 0);
    CurrentCombo = 0;
    GetCharacterMovement()->SetMovementMode(EMovementMode::MOVE_Walking); //콤보가 끝나면 걸을수있게 MovementMode를 되돌려준다.
}

```

0n. 캐릭터 콤보 액션 강의 과제

? Q1. 게임에서 애니메이션 블루프린트의 상태와 구별되는 애니메이션 몽타주를 어떻게 활용하면 좋을지 정리해보시오.



Reference