

언리얼 프로그래밍 Part2-4

제목: 캐릭터 애니메이션 설정

**강의 내용 : 새로운 캐릭터와 애니메이션 시스템 제작

강의 내용

새로운 캐릭터와 애니메이션 시스템 제작



C++ 클래스를 사용해서 해당 캐릭터의 애니메이션을

**강의에서 다루는 게임 프레임워크 요소

강의에서 다루는 게임프레임워크 요소

게임	월드	모드	상태	
기믹	트리거	스폰	물리	
플레이어	입력	카메라	HUD	상태
폰	이동	모션	액션	위젯
데이터	애셋	테이블	설정	저장
인공지능	길찾기	BT		

**강의 목표

강의 목표

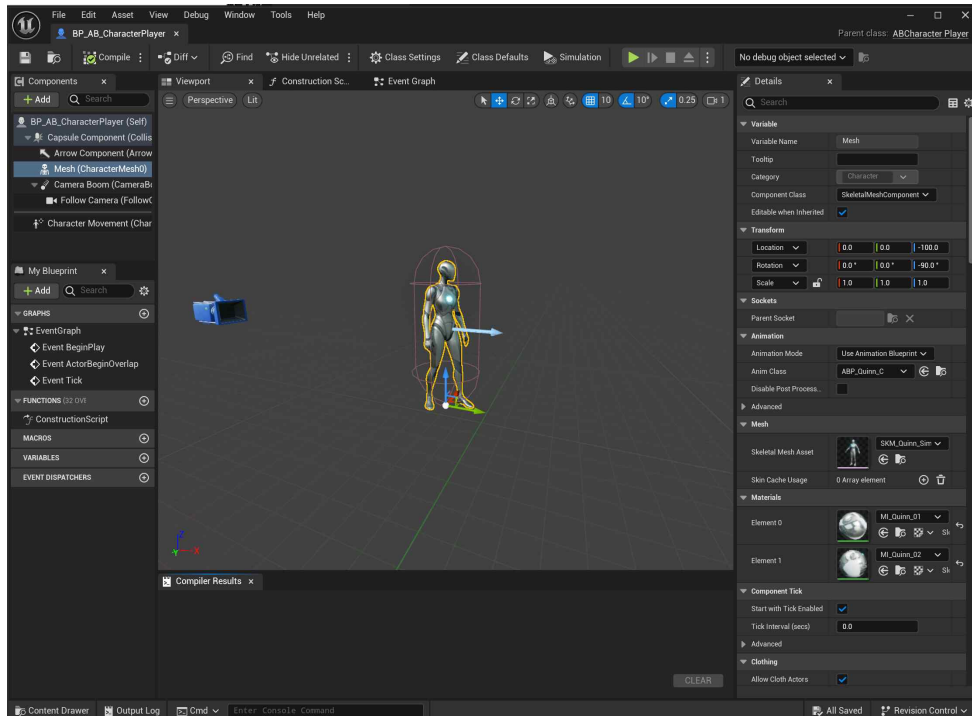
- 캐릭터의 애니메이션 시스템을 생성하는 방법의 학습
- 캐릭터의 애니메이션 시스템을 효과적으로 설계하는 방법의 학습

Inflearn

1138208

이득우의 언리얼 프로그래밍 Part2 - 언리얼 게임 프레임워크의 이해
이득우

****실습**



-애니메이션 모드가 Use Animation Blueprint로 되어있는데 여기에서 Anim class로 등록 되어있는 클래스 정보를 이 캐릭터가 생성될 때 해당 클래스로부터 인스턴스를 생성해서 애니메이션을 담당하는 인스턴스 언리얼 오브젝트를 생성해서 관리를 시키도록 일임을 한다.

-이러한 애니메이션 시스템을 애니메이션 블루프린트라고 한다. ==>전체적인 모션을 담당함

-우리가 할 것은 C++로 AnimInstance클래스를 직접 만들고 이 애니메이션 블루프린트가 우리가 생성한 C++클래스를 상속받아 기본적인 로직을 구현한 후에 AnimGraph를 사용하여 구체적인 모션을 지정하도록 설계하자.

**캐릭터 애니메이션 시스템의 생성

캐릭터 애니메이션 시스템의 생성

- 스�কে탈 메시 컴포넌트의 애니메이션 블루프린트 클래스를 지정한다.
- 캐릭터가 초기화될 때 AnimInstance 클래스의 인스턴스를 생성한다.
- 캐릭터는 GetAnimInstance 함수를 사용해 애니메이션 인스턴스를 얻을 수 있음
- 애니메이션 인스턴스는 GetOwningActor 함수를 사용해 자신을 소유한 액터 정보를 얻을 수 있음



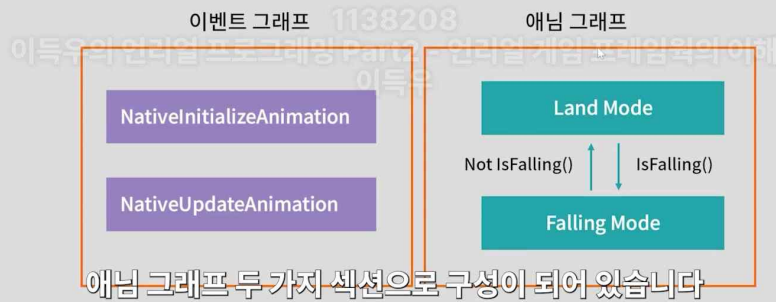
먼저 캐릭터에서 캐릭터 애니메이션 시스템이

- 캐릭터는 GetAnimInstance()를 통해 AnimInstance에 대한 포인터를 가져올 수 있다.
- AnimInstance는 GetOwningActor()를 통해 자기를 소유중인 액터에 대한 정보를 가져올 수 있다.
- 이러한 API를 통해 서로 상호 참조가 가능하다.

**캐릭터 애니메이션 시스템의 설계

캐릭터 애니메이션 시스템의 설계

- 애니메이션 블루프린트는 이벤트 그래프와 애니 그래프의 두 영역으로 구성되어 있음.
- 이벤트 그래프에서는 이벤트로부터 상태를 파악할 수 있는 주요 변수를 저장하는데 사용
- 애니 그래프에서는 저장된 변수로부터 지정된 상태의 애니메이션을 재생
- 애니 그래프의 복잡한 상태는 State Alias로 분리해 효과적으로 설계할 수 있음



-이벤트 그래프에 있는 두가지 이벤트가 발생하면 앞선 슬라이드에 있는 GetOwningActor()로 캐릭터 정보를 가져와 캐릭터 상태를 살펴보고 그 상태를 우리가 지정한 변수에 저장 한다. 이 AnimInstance에 저장된 변수를 애니그래프가 활용한다.

-이 애니 그래프는 저장된 변수로부터 스테이트 머신이라고 하는 우리가 설계한 상태를 중심으로 해서 애니메이션을 자동으로 재생하도록 우리가 설계해주면 된다.

-이 애니 그래프는 굉장히 복잡한 상태를 가지게 되는데 언리얼 엔진에서는 State Alias라는 기능을 사용하여 복잡한 상태를 분리하여 효과적인 설계가 가능하다.

-우리는 앞으로 크게 두 가지 모드를 사용하는데 땅위를 걸어다닐 때 사용하는 Land Mode와 땅을 벗어났을 때 발동되는 Falling Mode에 필요한 애니메이션을 분리해서 구현해 볼 것이다.

**실습

-기본적인 걷기 뛰기의 상태를 로코모션(Locomotion)이라고 한다.

-로코모션의 상태는 엄청 복잡할수도 있는데 로코모션에 대한 구성을 다른 곳에서 진행하고 거기에 결과를 캐시로 저장해서 불러오는 방법이 있다.