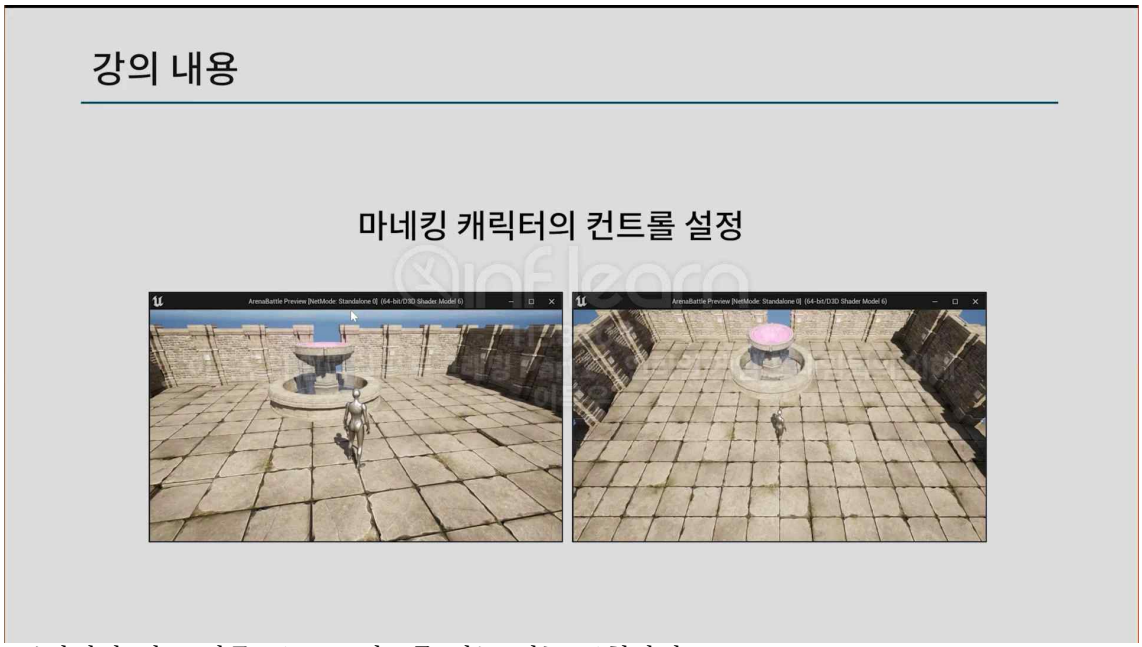
# 언리얼 프로그래밍 Part2-3

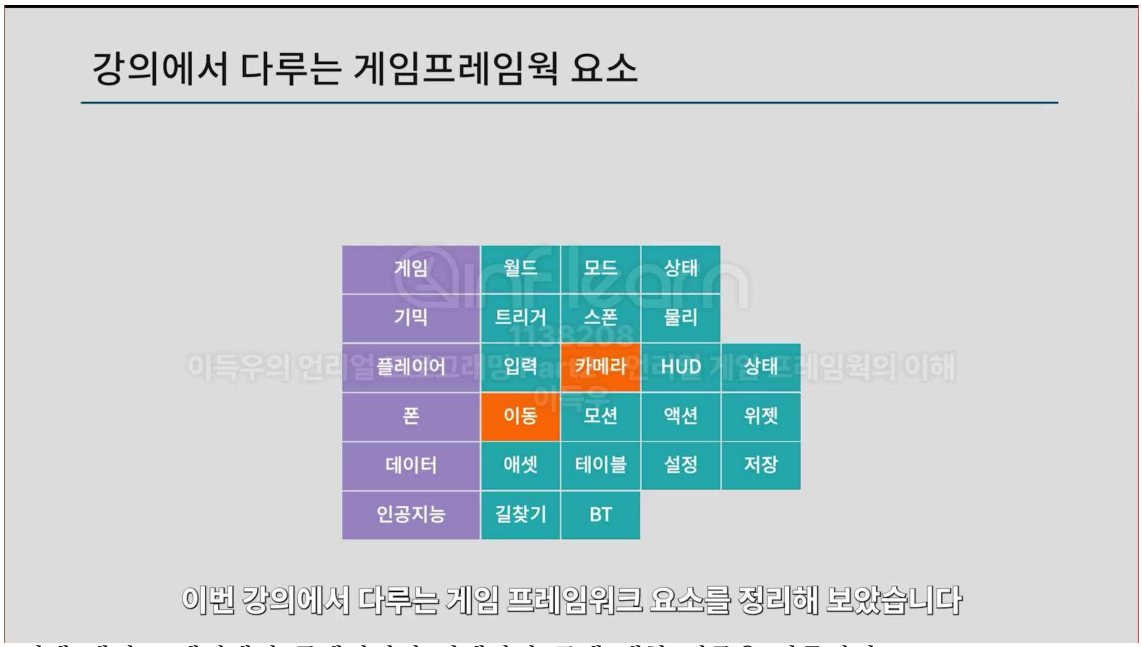
## 제목: 캐릭터 컨트롤 설정

**강의 내용 : 마네킹 캐릭터의 컨트롤 설정



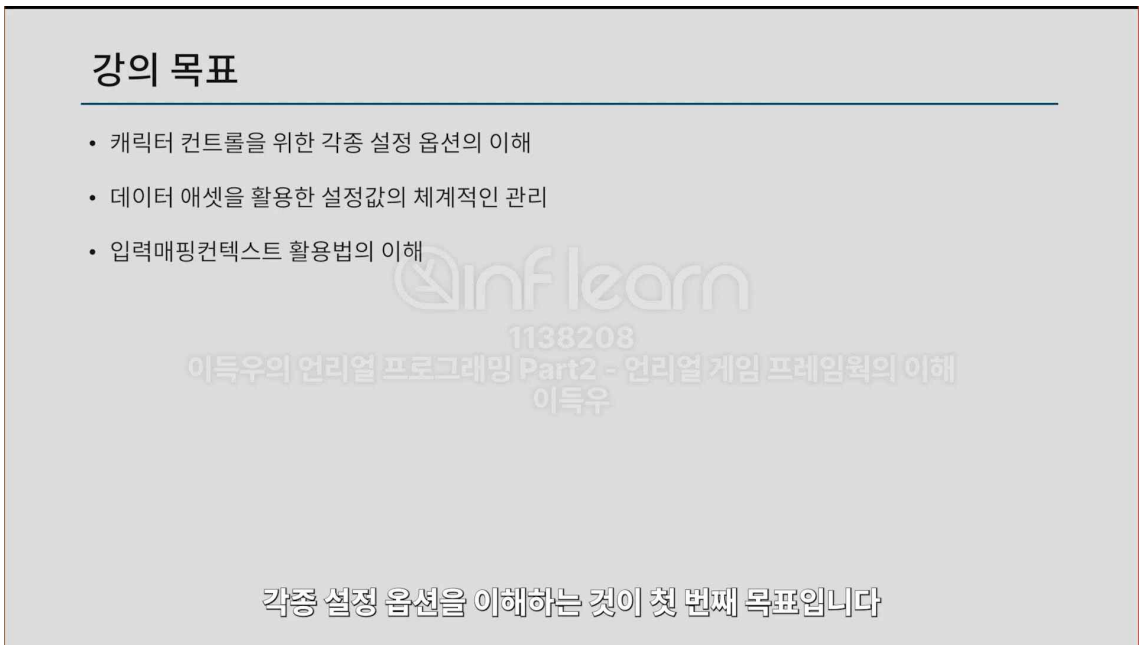

-두가지의 서로 다른 구도로 컨트롤 하는 기능 구현하기

**강의에서 다루는 게임 프레임웍 요소

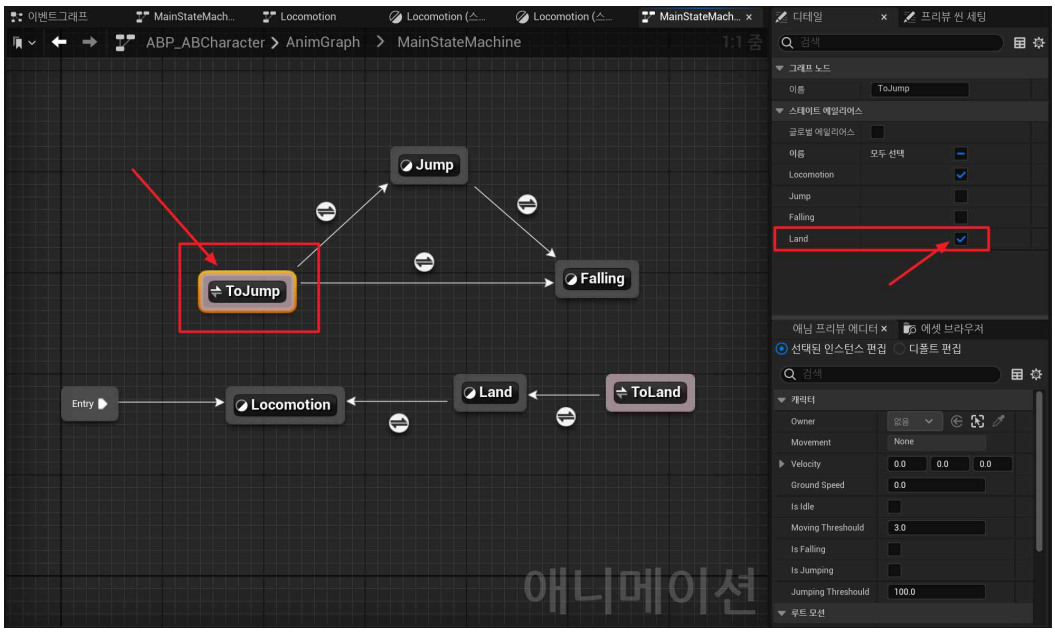



-전체 게임 프레임에서 플레이어의 카메라와 폰에 대한 이동을 다룰것임.

**강의 목표

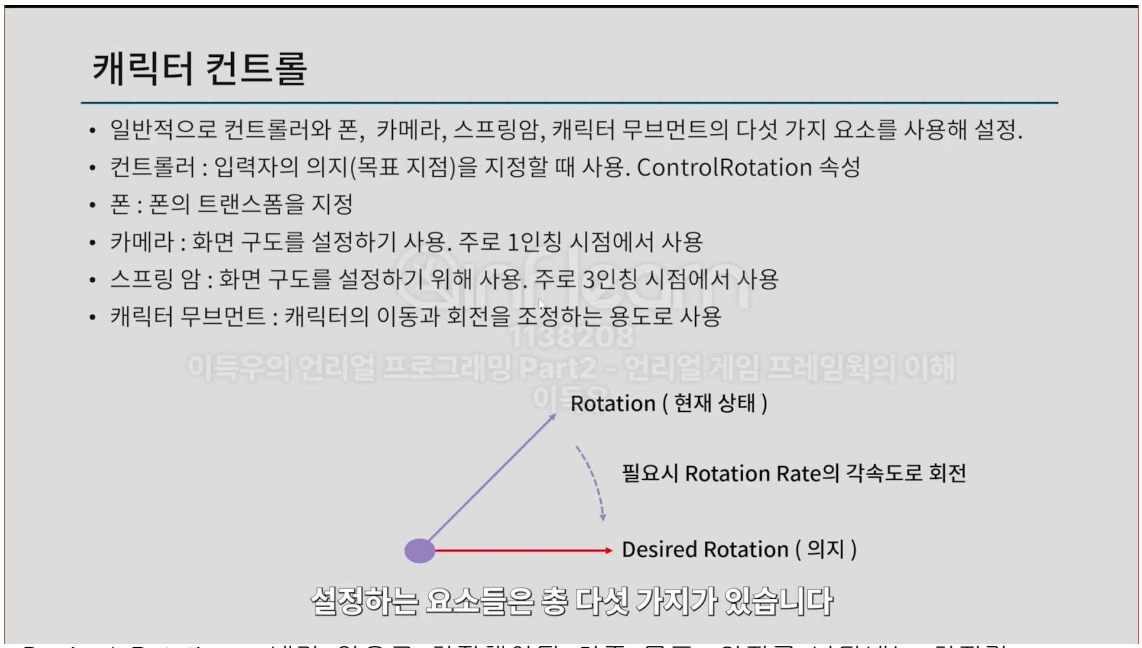



**[수강생 노트]
애님 블루프린트 설정에서 모든 상태의 제작을 마무리하면, 아래 그림과 같이 ToJump 에일리어스에서 Land 상태에서 접근하도록 추가로 설정해주는 것이 좋습니다. Land 상태일 때 스페이스 바를 눌러 점프를 수행하면 점프 모션이 수행되지 않고 Locomotion의 애니메이션이 수행되기 때문입니다.



***캐릭터 컨트롤 요소

**캐릭터 컨트롤




-Desired Rotation : 내가 앞으로 회정해야될 최종 목표, 의지를 나타내는 회전값.

-Rotation : 현재 회전된 상태

-현재 Rotation값을 Desired Rotation값으로 덮어 씌우면 캐릭터는 바로 (의지)값으로 회전하게 되는데 바로 팍 틀어지면 보기가 부자연스러워진다. 따라서 Desired Rotation을 설정하고 필요시에 Rotation Rate로 지정된 각속도로 Rotation에서 DesiredRotation 값으로 서서히 회전하도록 구현하는 것이 부드러운 움직임을 줄 수가 있어서 많이 사용이 된다.

-이러한 것들은 코드로 짤 수가 있는데 언리얼이 제공하는 설정값들을 이요하면 우리는 설정을 통해서 자연스럽게 구현할 수가 있습니다.

**폰의 이동 함수


-Look과 Move함수들은 Control Rotation이라고 하는 컨트롤러의 속성값을 통해 이동과 시점을 조절한다.

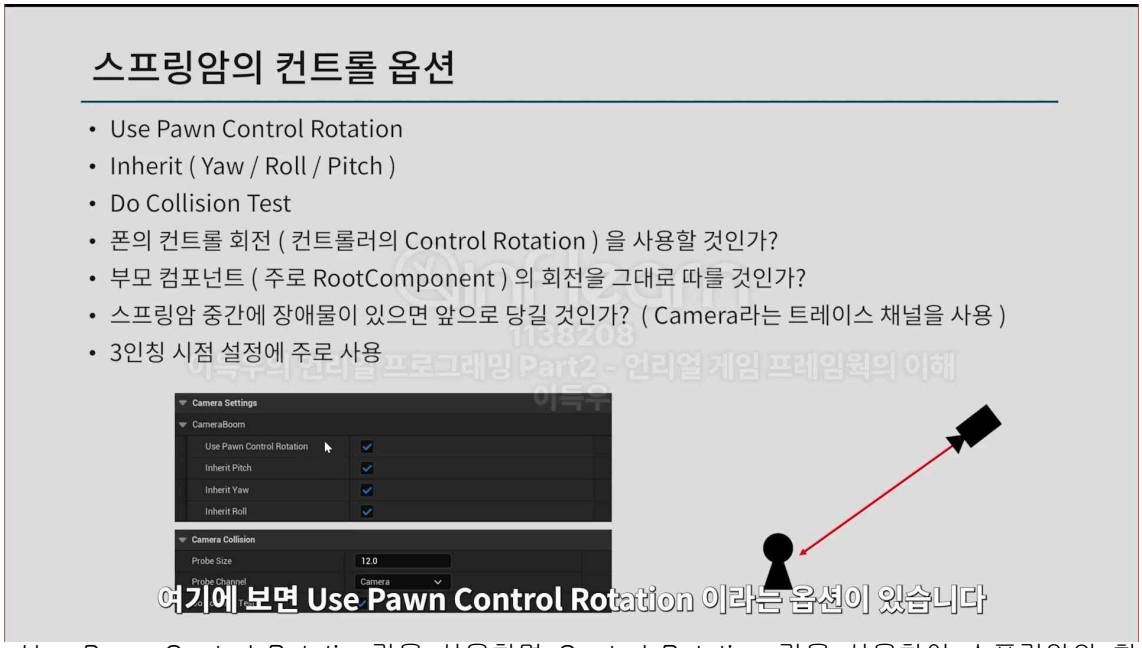
-커맨드를 입력하면 UPROPERTY로 지정된 속성값은 우리가 직접 확인이 가능하다.

**폰의 컨트롤 옵션


-옵션을 키게 되면 폰의 회전이 컨트롤러의 Control Rotation에 바로 동기화가 되므로 감안하여 필요한 경우에 체크하자

-chat gpt의 말에 따르면 FPS처럼 카메라의 시점이 캐릭터에 따라 회전해야 할 때 사용해야 한다고 한다.

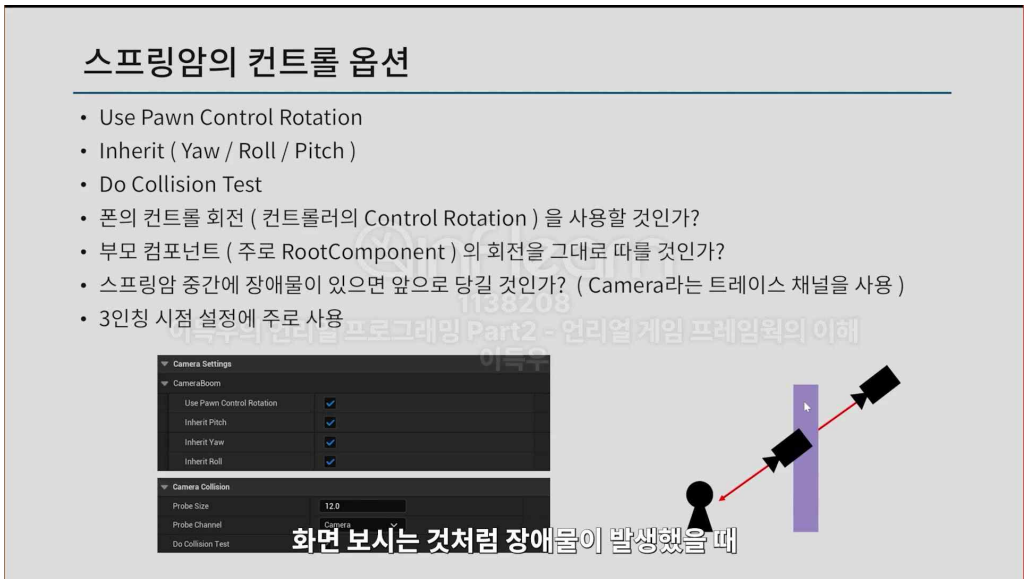
**스프링암의 컨트롤 옵션




-Use Pawn Control Rotation값을 사용하면 Control Rotation 값을 사용하여 스프링암의 회전이 동기화 되도록 설정한다.

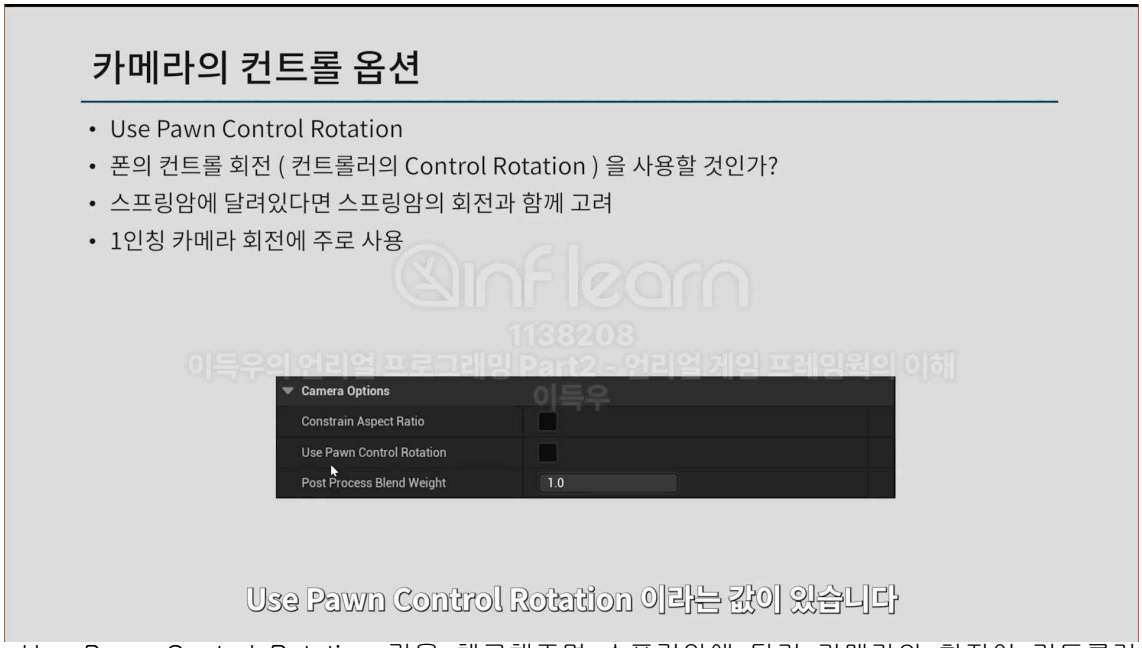
-추가하여 부모 컴포넌트의 회전값인 요,롤,피치 값을 추가로 상속 받아 최종 회전을 구현하게 된다.(우리는 이 옵션이 체크되어 있어 마우스를 움직이면 스프링암도 같이 회전이 됨)

-




Do Collision Test는 카메라와 캐릭터 사이를 지탱해주는 스프링암의 중간에 장애물이 생겼을 경우 장애물 앞으로 카메라를 당겨주는 역할을 수행한다. 이러한 스프링암 컴포넌트는 3인칭 시점 설정에 주로 사용이 된다.

**카메라의 컨트롤 옵션




-Use Pawn Control Rotation 값을 체크해주면 스프링암에 달린 카메라의 회전이 컨트롤러의 Control Rotation 값과 동기화 되면서 회전하게 된다.(1인칭 카메라 회전에 주로 사용한다.)

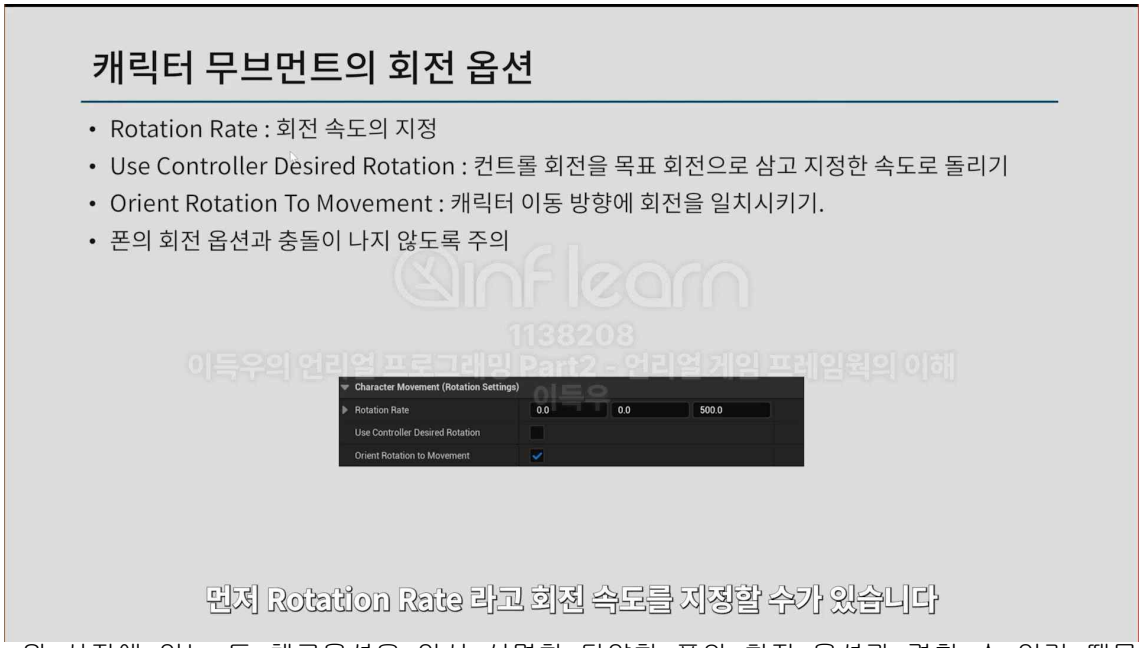
**캐릭터 무브먼트의 이동 옵션


-MovementMode는 다양한 옵션을 제공한다. 땅 위에 있으면 일반적으로 Walking모드이며 땅을 벗어나면 Falling 우리가 만약 수동으로 이동하는 기능을 끄고싶을땐 None모드를 사용한다.

-일반적으로 이동할땐 워킹 모드를 쓰는데 최대값수치는 MaxWalkSpeed속성을 사용하며, Falling모드에서 많이 사용되는게 점프인데 점프를 최초로 도약할 때 사용하는 수치는 JumpZVelocity이다.

-이러한 속성들을 사용하여 걷기와 점프의 크기를 우리가 지정할 수가 있다.

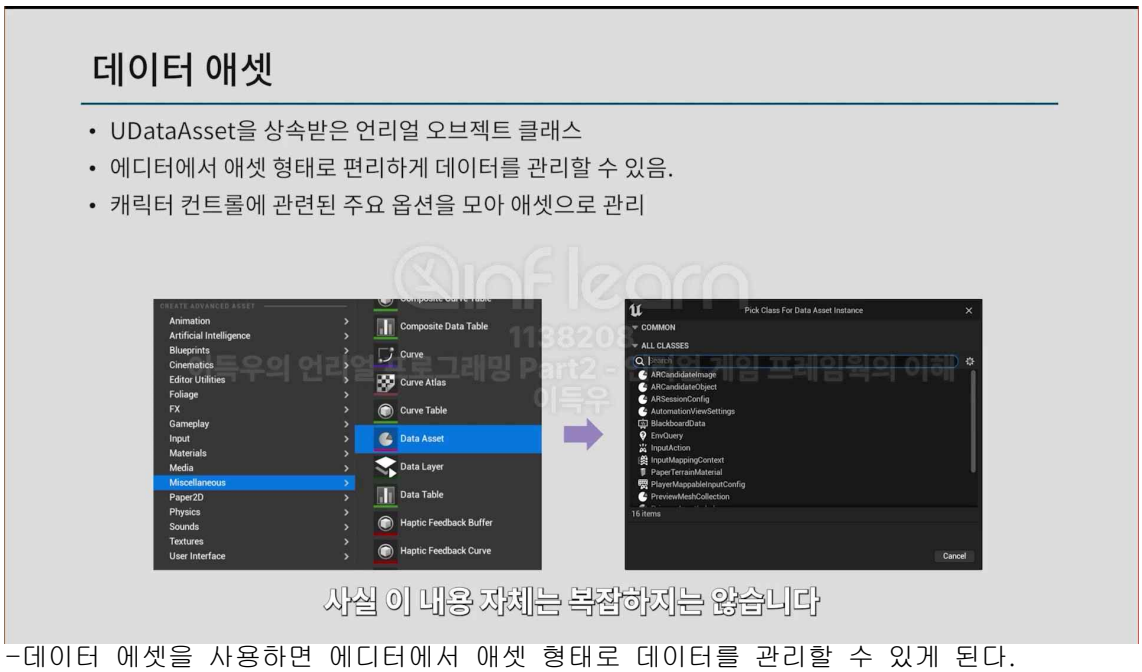
**캐릭터 무브먼트의 회전 옵션




-위 사진에 있는 두 체크옵션은 앞서 설명한 다양한 폰의 회전 옵션과 겹칠 수 있기 때문에 충돌이 나지 않도록 잘 설정해야 한다.

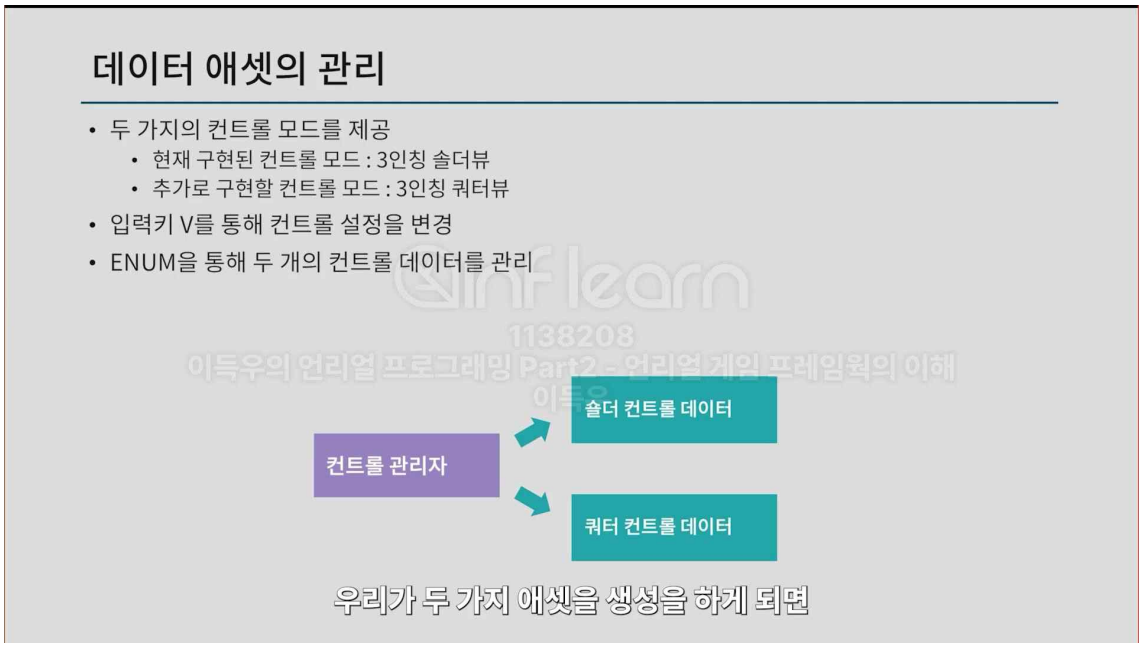
*** 데이터 에셋

**데이터 애셋

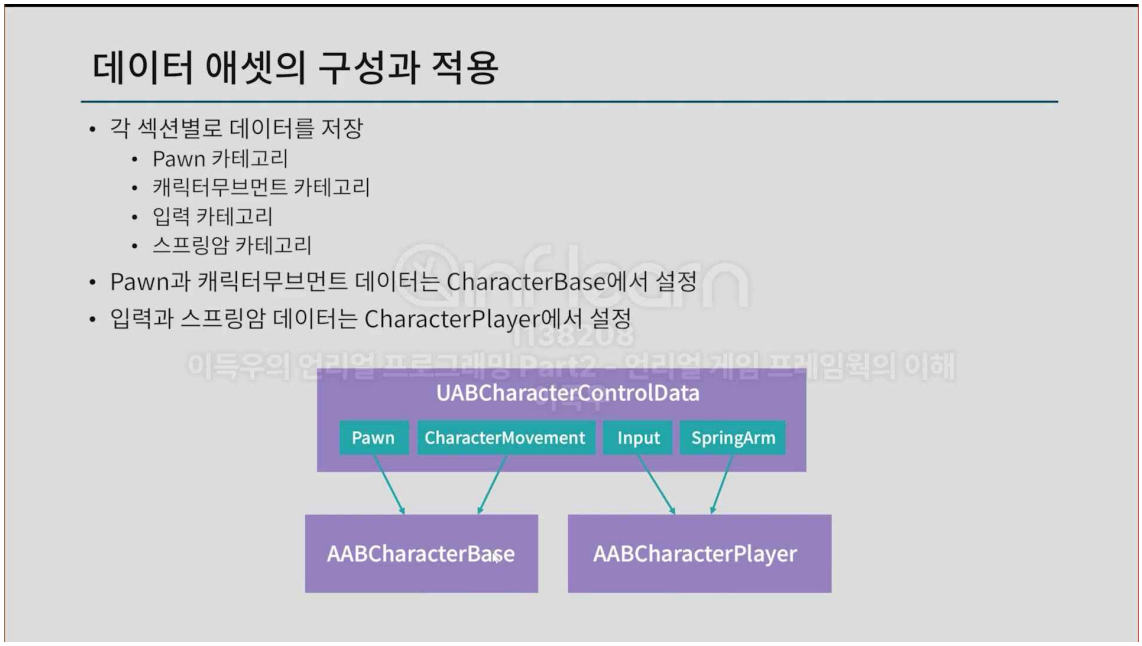



-데이터 에셋을 사용하면 에디터에서 애셋 형태로 데이터를 관리할 수 있게 된다.
-클래스를 만들 때 좀 더 기능이 추가된 PrimaryAssetData를 상속받아 생성하자.

**데이터 애셋의 관리

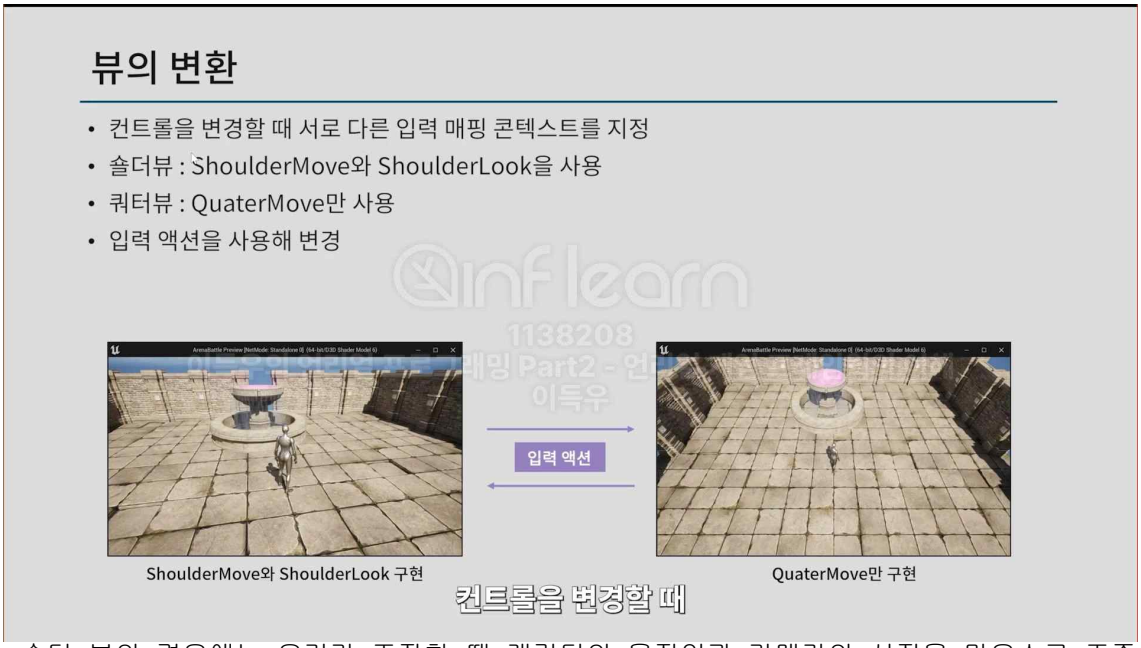


**데이터 애셋의 구성과 적용




-베이스 클래스의 경우에는 폰과 무브먼트를 가지고 있기 때문에 두가지 설정값 지정
-베이스 클래스를 상속받은 캐릭터 플레이어 클래스의 경우에는 입력과 스프링 암에 관련된 값들을 설정하도록 두단계로 나누어 진행한다.

**뷰의 변환



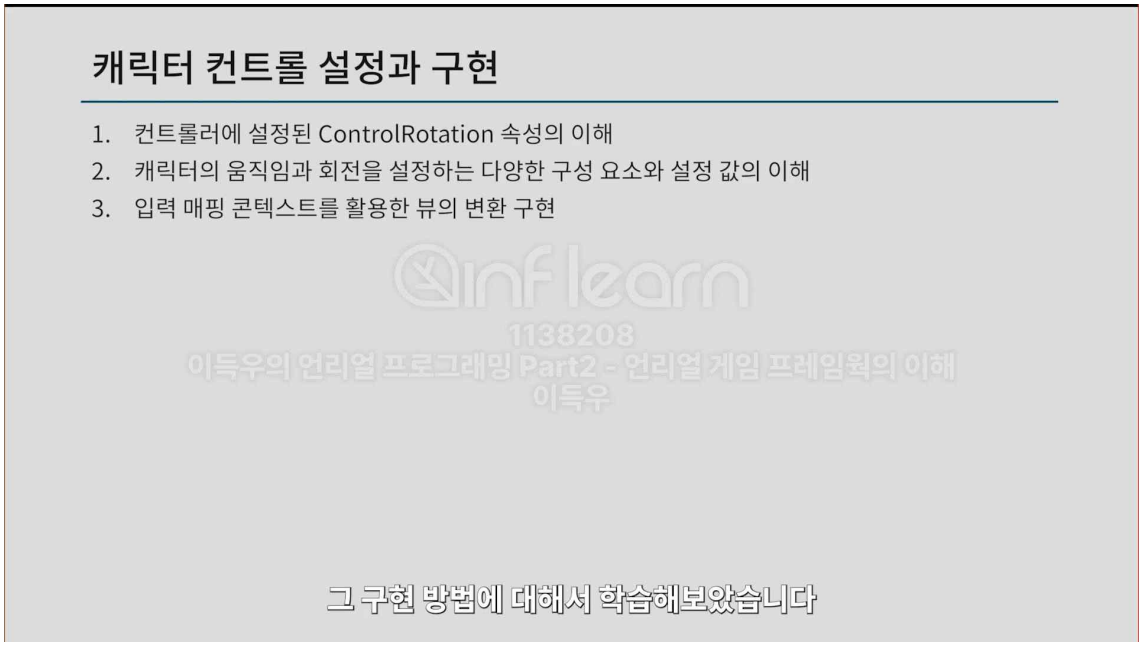

-숄더 뷰의 경우에는 우리가 조작할 때 캐릭터의 움직임과 카메라의 시점을 마우스로 조종할 수 있다.

-쿼터뷰의 경우에는 캐릭터의 움직임만 지정할 수 있고 시점을 마우스로 조작할 필요가 없다.

-그렇기 때문에 기존에 구현했던 Move함수와 Look 함수는 숄더뷰에서 ShoulderMove와 ShoulderLook QuaterView에서는 QuaterMove만 구현하면 된다.

-V라는 입력 액션을 통해 두 개의 뷰가 전환되도록 할것임.

**정리


캐릭터 컨트롤 설정과 구현
1. 컨트롤러에 설정된 ControlRotation 속성의 이해
2. 캐릭터의 움직임과 회전을 설정하는 다양한 구성 요소와 설정 값의 이해
3. 입력 매핑 콘텍스트를 활용한 뷰의 변환 구현
inflearn
1138208
이득우의 언리얼 프로그래밍 Part2 - 언리얼 게임 프레임웍의 이해
이득우
그 구현 방법에 대해서 학습해보았습니다

*** 데이터 에셋