

## CHAPTER 6

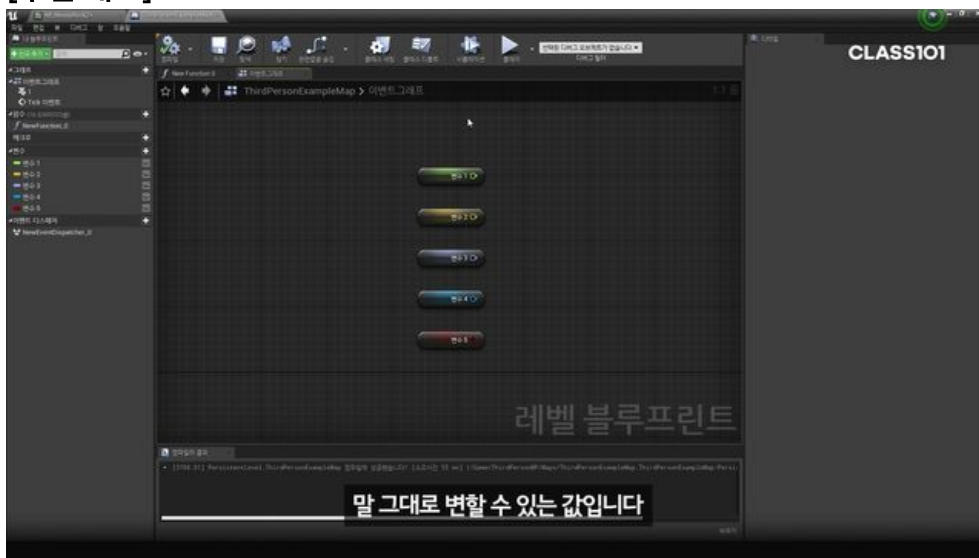
## 블루프린트 활용 I: 가까이 가면 문이 열리고 불이 켜지는 오브젝트

## [수업 목표]

여러분 반갑습니다! 러셀입니다.

이번 시간엔 집 앞에 가면 자동으로 문이 열리고 불이 켜지는 기능을 만들어볼겁니다. 변수, 타임라인, 그리고 Lerp에 대해 배우는 것이 핵심입니다.

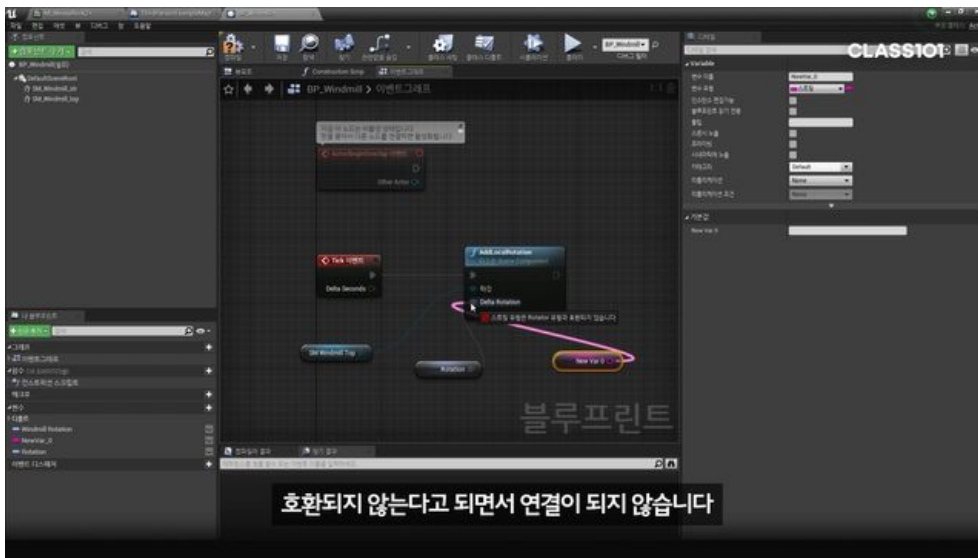
## [수업 개요]



## 0:32 변수에 대해서

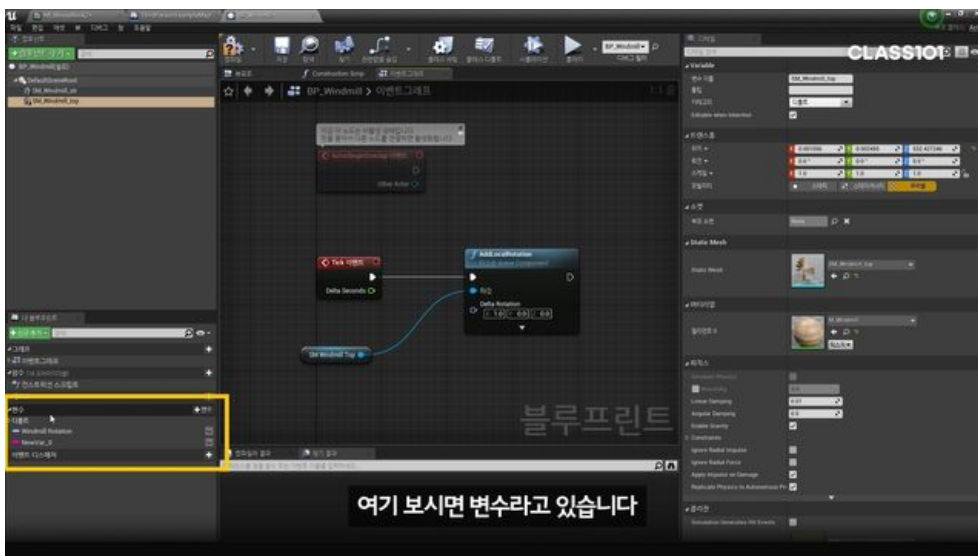
변수는 말 그대로 변할 수 있는 데이터입니다. 머티리얼의 파라미터와 유사하죠. 전 시간의 풍차 회전값에 직접 1을 넣어주었던 값은 플레이 도중에 바뀔 수 없습니다.

바뀔 수 없는 값을 보고 상수(Constant)라고 하고, 플레이 도중이나 다른 블루프린트, 노드에 의해 변할 수 있는 값을 변수(Variable)이라고 합니다.



변수는 다양한 유형이 있으며, 색깔로 구분할 수 있습니다.

당연하게도 노드에서 필요로 하는 변수를 알맞은 유형으로 연결시켜주어야 하며, 변수를 변형, 조합, 분해하여 다른 유형의 변수로 바꿀 수도 있습니다. 활용도가 광범위하죠.



변수는 좌측 아래의 변수 탭에서 만들고, 관리할 수 있습니다.

변수를 누르면 우측의 디테일 패널이 바뀌는데, 그곳에서 변수의 유형 또한 바꿔줄 수 있습니다.

그럼 변수의 유형에 대해 알아보도록 하겠습니다.



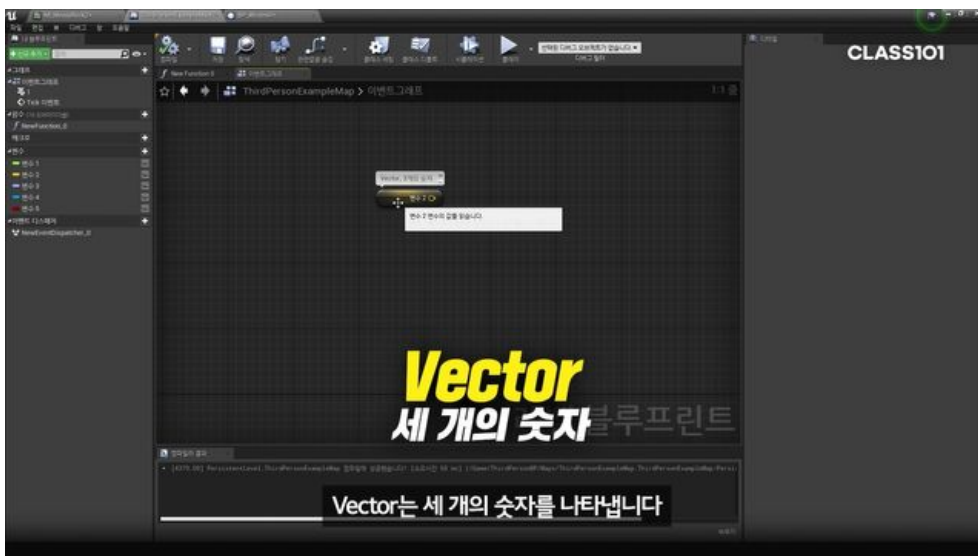
#### 4:53 Float 유형의 변수

Float는 소수점을 사용하는 단일 숫자 유형입니다. 유리수이죠.

가장 많이 쓰게 될 유형 중 하나입니다.

Tick에 Print String이라는 노드를 연결하고, String 부분에 변수를 연결해주면 현재 변수가 어떤 값을 출력하고 있는지 플레이하며 살펴볼 수 있습니다.

이 변수가 저희 생각대로 잘 작동하는지 확인할 수 있죠.

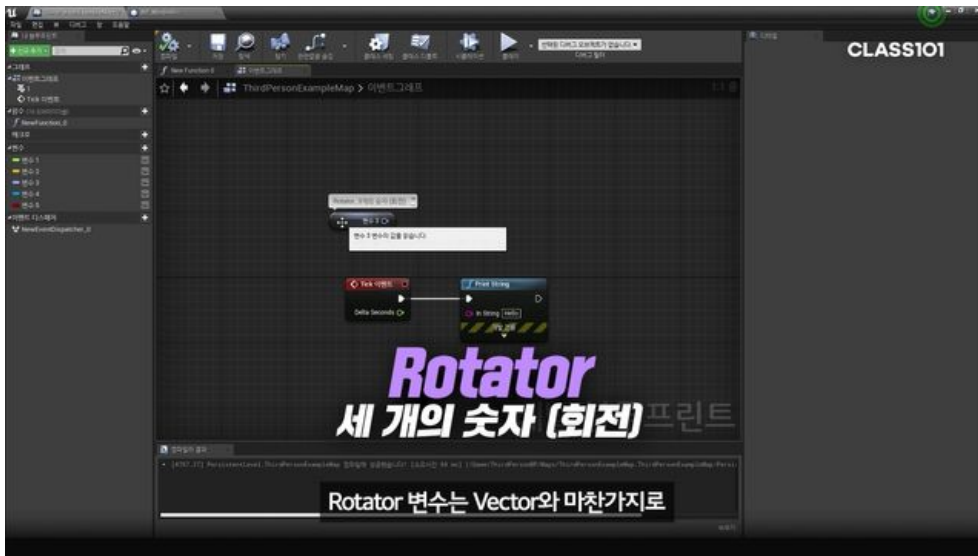


#### 6:00 Vector 유형의 변수

Vector는 세 개의 숫자를 나타냅니다.

주로 X, Y, Z값을 사용하는 위치, 스케일에 사용하며 R, G, B값으로 활용해 색을 정의할 수도 있습니다.

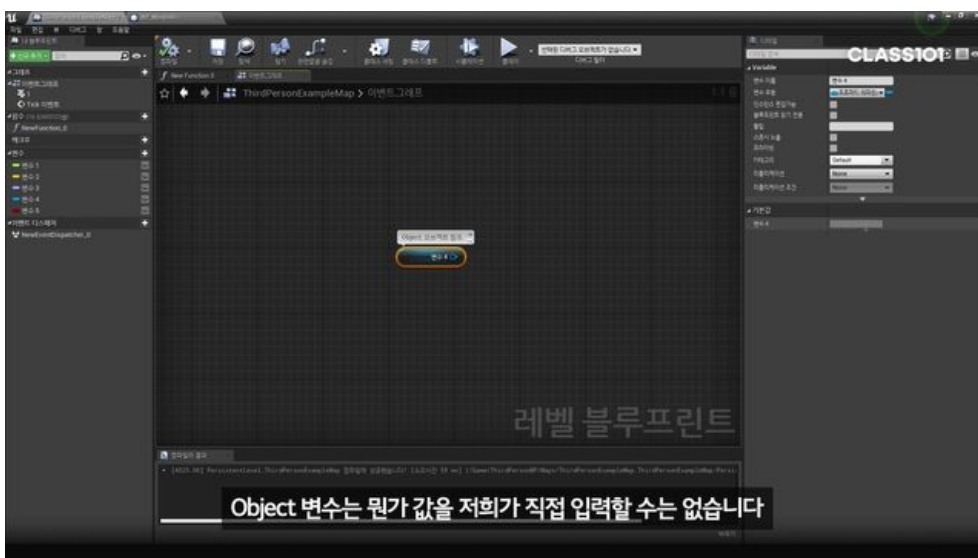
Float 3개를 합쳐 Vector를 만들 수도 있죠.



## 7:21 Rotator 유형의 변수

Rotator는 Vector처럼 3개의 숫자를 사용하지만 오로지 회전에만 활용되는 유형의 변수입니다.

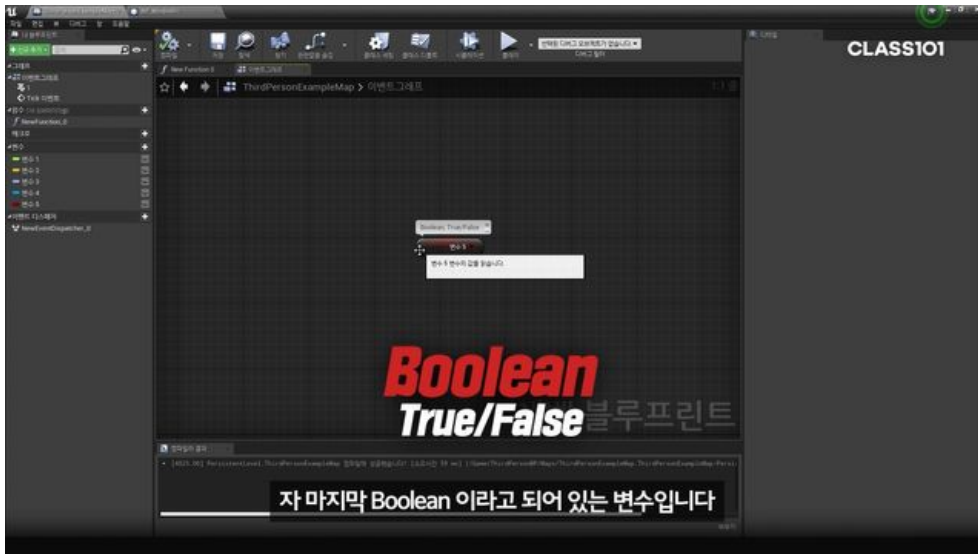
그 이유는 회전에 Pitch, Roll, Yaw라는 단위가 사용되기 때문에 그렇습니다. 그래서 Print String으로 확인해보면 Vector과는 다르게 출력하는 것을 볼 수 있죠.



## 8:35 Object 유형의 변수

오브젝트 변수는 직접 값을 입력하지는 않고, 주로 다른 액터를 불러올 때 사용되는 변수입니다.

저희가 지금까지 봤던, 타깃에 연결되어 있던 파란색 변수가 오브젝트 유형의 변수였던 것입니다.



## 9:20 Boolean 유형의 변수

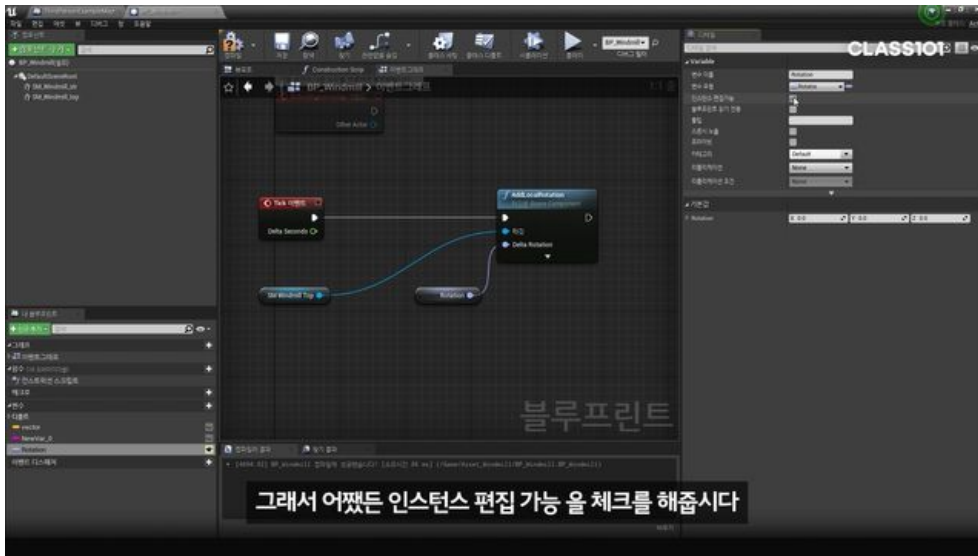
Boolean 변수 유형은 True/False의 값을 갖는 변수입니다.

어떤 상태가 참이냐 거짓이냐를 따져서 다른 이벤트를 실행할 때 유용하게 사용됩니다.

예를 들면 캐릭터의 HP를 점검해서, HP가 50 이상일 때 Boolean 변수에 참을 저장하고, HP가 50 이하일때 False를 저장합니다.

그리고 어떤 이벤트를 실행시킬 때, 이 변수를 불러와서 참인지 거짓인지 물어본 후, 결과에 따라 다른 이벤트를 실행시킬 수 있는 것입니다.

이 내용에 대해선 6-7강에서 깊게 다룹니다.



## 11:18 변수의 인스턴스 편집 옵션

변수의 디테일 패널에는 인스턴스 편집 가능이라는 옵션이 있습니다.

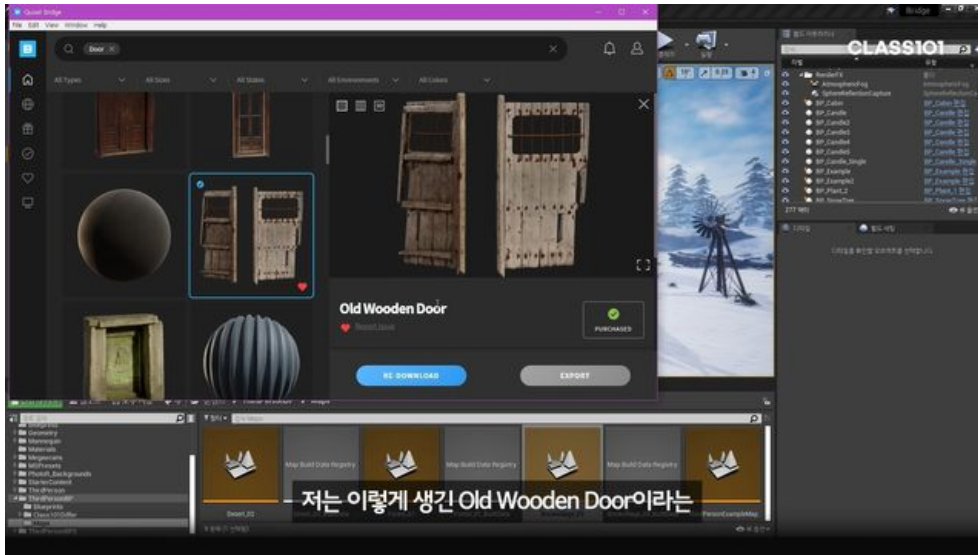
이 옵션을 체크하면, 블루프린트 애셋을 레벨에 배치했을 때 배치된 각각의 인스턴스에서 별개의 변수 값을 적용시킬 수 있습니다.



같은 애셋을 불필요하게 복제할 것 없이, 레벨에 여러 개를 배치한 뒤 다양성을 위해 변수 값만 각각 다르게 설정할 수 있어서 환경 디자인에 매우 용이합니다.

어떤 풍차는 3의 속도로, 다른 풍차는 1의 속도로, 또 어떤 풍차는 멈춰있게도 할 수 있는 것이죠. 매우 유용한 기능이니 클래스 블루프린트 애셋을 만드실 때 꼭 사용해하시기 바랍니다.

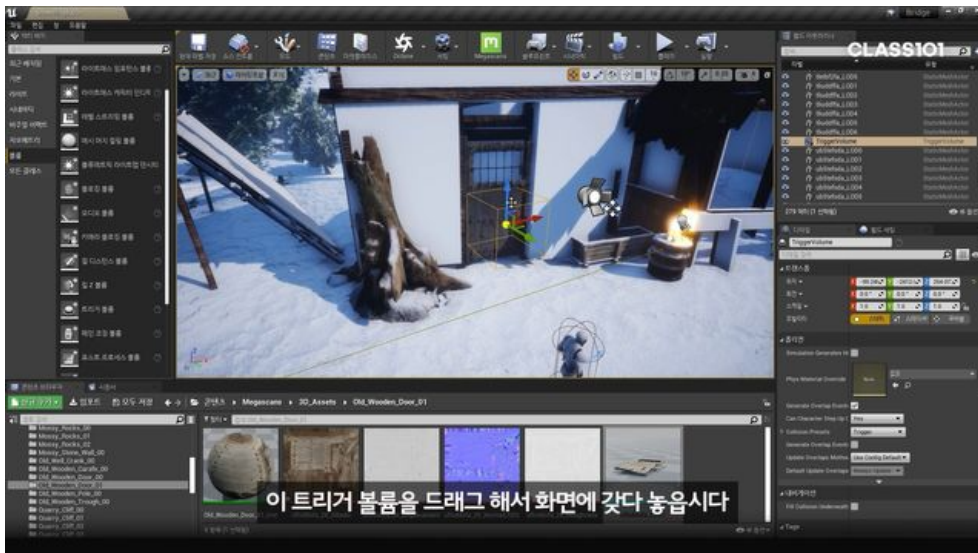




## 14:00 문 열리는 기능 만들기

먼저 퀵셀 브릿지에서 Old Wooden Door라는 애셋을 임포트합니다.

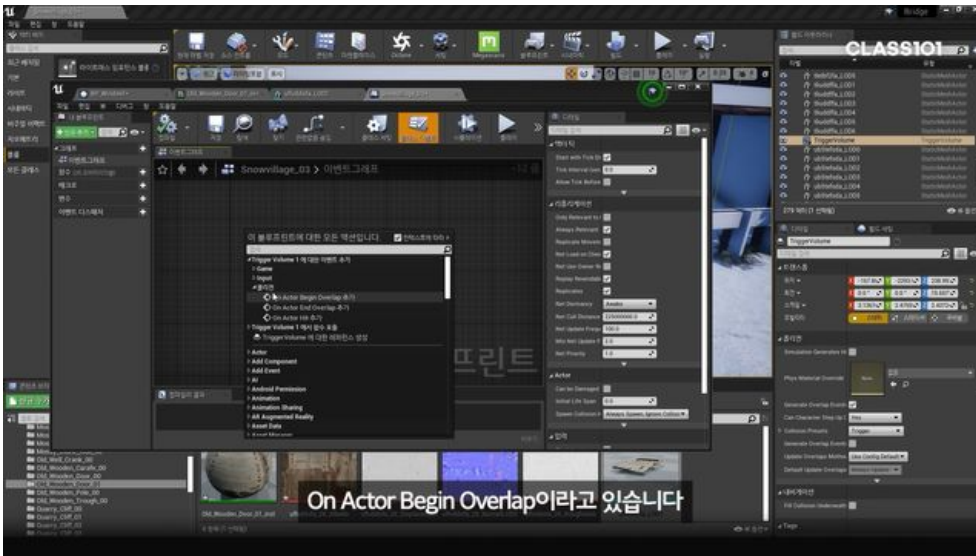
콜리전을 잡아주신 뒤, 집 입구에 스케일을 적당히 조절해 배치해주세요.



액터 배치 패널에서 볼륨 - 트리거 볼륨을 꺼내 환경에 배치합니다.

트리거 볼륨에 들어갔을 때 이벤트를 실행시키는 액터입니다.

트리거 볼륨의 스케일, 위치를 조절해서 특정 구간에 들어갔을 때 이벤트를 실행시킬 수 있도록, 문 앞쪽에 잘 배치해주세요.



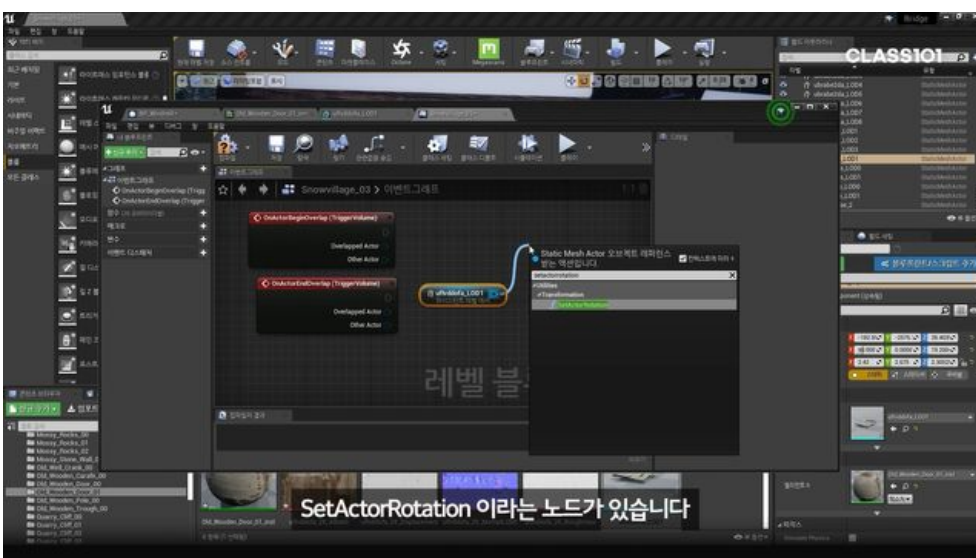
## 16:50 트리거 볼륨 노드를 블루프린트에 배치

월드 아웃라이너에서 트리거 볼륨을 선택한 뒤, 선택된 상태로 레벨 블루프린트를 열고 빈 공간에 우클릭을 합니다.

검색 창의 최상단에 Trigger Volume에 대한 이벤트 추가 - 콜리전 - OnActorBeginOverlap를 선택해 꺼냅니다.

이 노드는 플레이어가 트리거 볼륨에 겹쳤을 때 이벤트를 실행하는 노드입니다.

아래에 있는 OnActorEndOverlap은 캐릭터가 트리거 볼륨 안에 있다가 나올 때 이벤트를 실행하는 노드입니다.

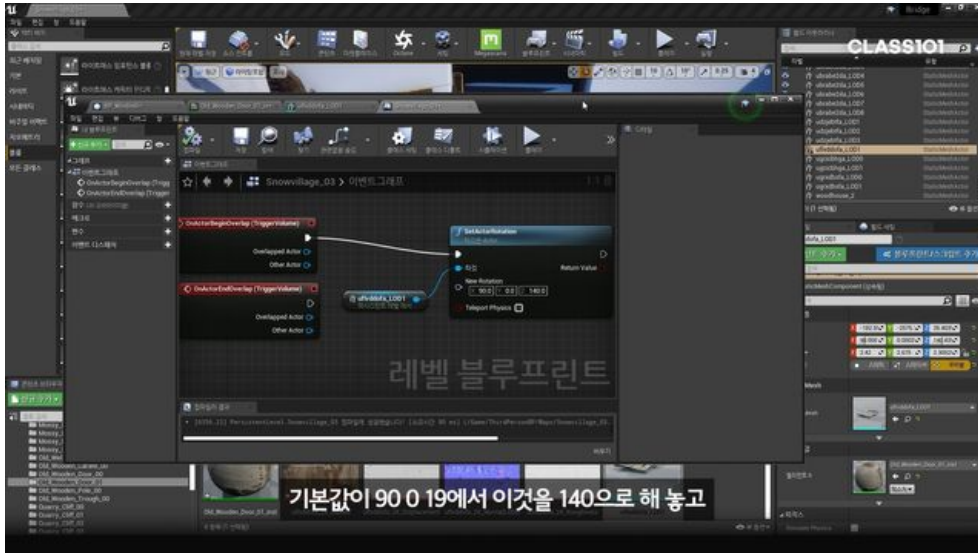




월드 아웃라이너에서 브릿지로부터 임포트한 문 오브젝트를 드래그해서 레벨 블루프린트로 가져옵니다. 그러면 푸른색의 오브젝트 변수로 불러와집니다.

이 문짝 오브젝트 변수의 핀을 드래그해서 검색 창을 띄운 뒤, SetActorRotation이라는 노드를 검색해서 꺼내줍니다.

이 노드는 연결된 액터의 회전값을 Set한다는 뜻입니다. 풍차의 노드는 Add였기에 기존 회전값에 새로운 회전값을 더하지만, **Set**의 경우 원래 값을 무시하고 새로 정해진 값으로 재설정합니다.

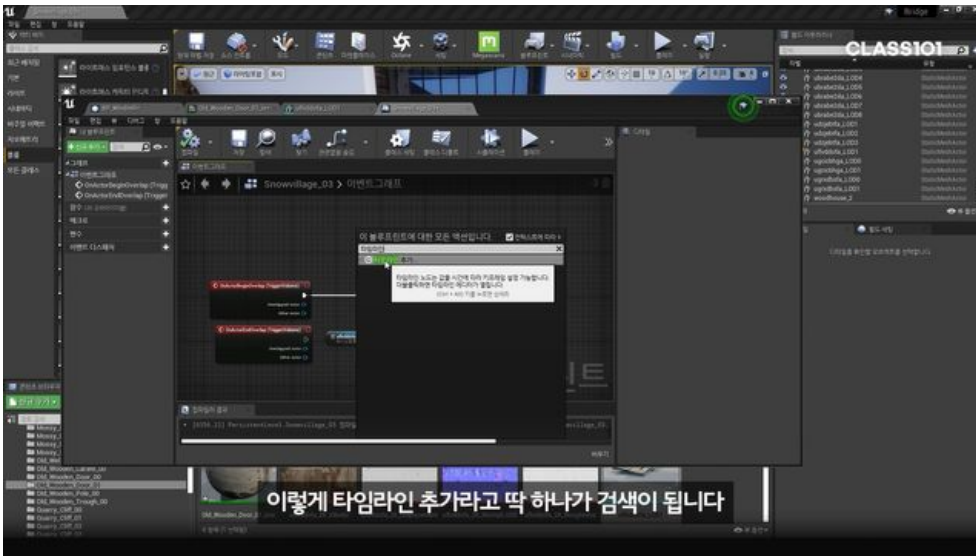


문이 닫혀있을 때의 회전 값은 X: 90 / Y: 0 / Z: 19입니다.

문이 열렸을 때의 회전 값은 X: 90 / Y: 0 / Z: 140입니다.

그러므로 레벨에 배치된 문의 기본 상태를 90 / 0 / 19로 설정해두고, SetActorRotation의 새 회전값에 90 / 0 / 140을 입력합니다.

이렇게 하면 캐릭터가 트리거 볼륨에 닿았을 때, 문짝의 Z값이 19였다가 140으로 변경됩니다.



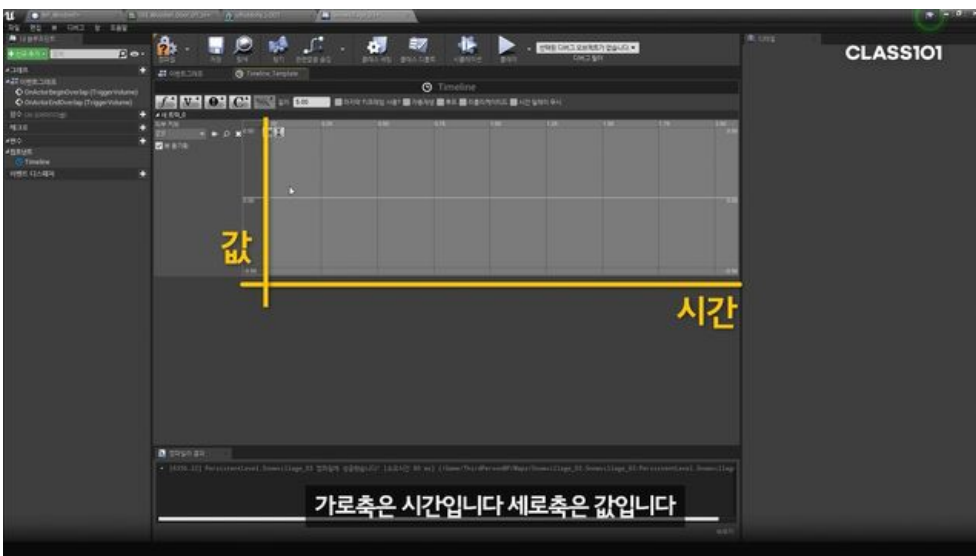
## 21:15 타임라인 노드 소개

문짝이 열린 건 하나 부드럽게 열리지 않고 갑자기 뚝 열려버립니다. 애니메이션이 없어 부자연스럽죠.

그래서 이럴 때 애니메이션을 만들 수 있는 노드가 있는데, 바로 타임라인입니다.

OnActorBeginOverlap를 타임라인 노드의 Play에 연결합니다.

그리고 OnActorEndOverlap를 타임라인 노드의 Reverse에 연결합니다.



타임라인 노드를 눌러 들어가보면 이렇게 생겼습니다.

세로 축은 값, 가로 축은 시간입니다.

Shift를 누르고 클릭하면 점이 생기는데, 이 점은 키프레임으로 점을 하나 더 만들면 두 키프레임이 이어집니다.

그리고 각 키프레임의 시간과 값을 변경해서 원하는 형태의 애니메이션을 구성할 수 있습니다.

또, 키프레임을 모두 선택해서 우클릭을 하면 키프레임이 연결된 그래프의 모양도 변경할 수 있습니다. 이는 애니메이션의 느낌을 바꿀 수 있는 부분입니다.

타임라인에서 전체 길이를 1로 설정하고, 첫 번째 키프레임은 0초 - 값 0으로, 두 번째 키프레임은 1초 - 값 1로 설정합니다. 그리고 두 키프레임을 선택해서 키 보간을 자동으로 설정했습니다.

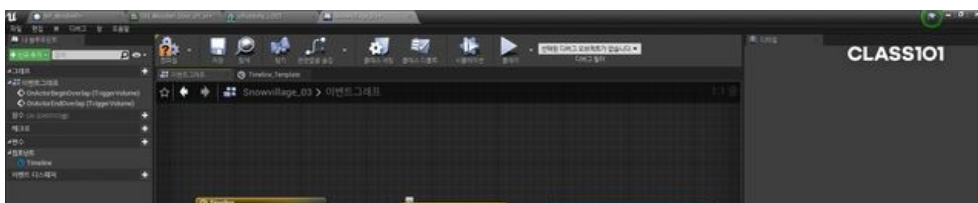


## 25:20 Lerp 노드

타임라인은 Lerp라는 노드와 자주 함께 사용됩니다.

Lerp는 Linear Interpolate의 줄임말로, 이 노드는 0과 1사이의 값을 입력받을 때, 그 값을 우리가 원하는 값의 범위로 증폭/축소시키는 역할을 합니다.

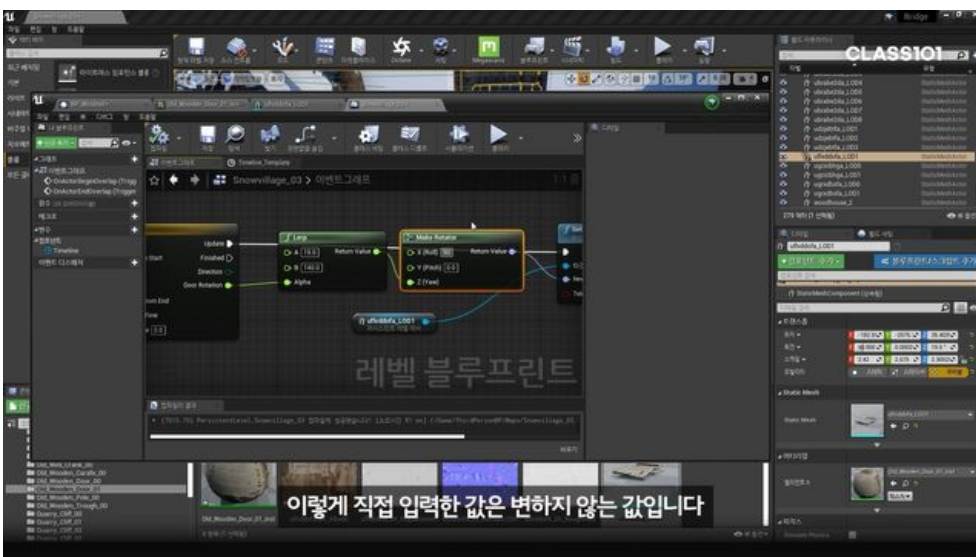
마치 Map Range Clamped와 같지만 이것은 입력값이 0과 1사이로 제한되어 있다는 차이가 있습니다. 그렇기에 타임라인의 값을 최소 0, 최대 1로 세팅한 것 입니다.





타임라인의 Float 출력값을 Lerp의 Alpha에 연결합니다.

그리고 A에는 문이 열리기 전 값인 19, B에는 문이 열린 후 값인 140을 입력합니다.



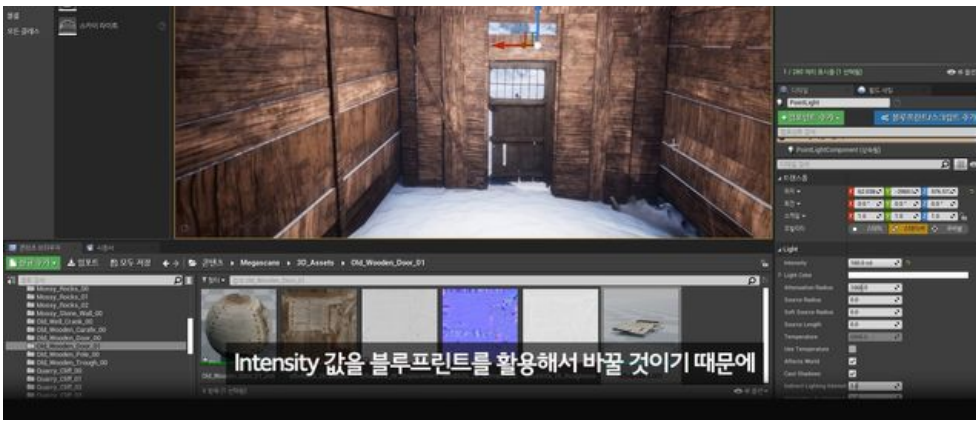
Lerp에서 만들어진 데이터는 실수 값, 즉 단일 숫자이기 때문에 회전 값에 사용하기 위해서 Make Rotator라는 노드를 이용해 Rotator 변수로 합쳐주어야 합니다.

문짝의 회전 값에서, 변하는 것은 Z값 뿐이고 X와 Y는 각각 90과 0으로 고정되어 있으니 Make Rotator에 각각 상수값을 입력해줍니다.

Make Rotator를 SetActorRotation의 Delta Rotation에 연결하면 완료입니다.

컴파일과 저장을 한 뒤 레벨에서 플레이해보면 잘 작동하는 것을 볼 수 있습니다.

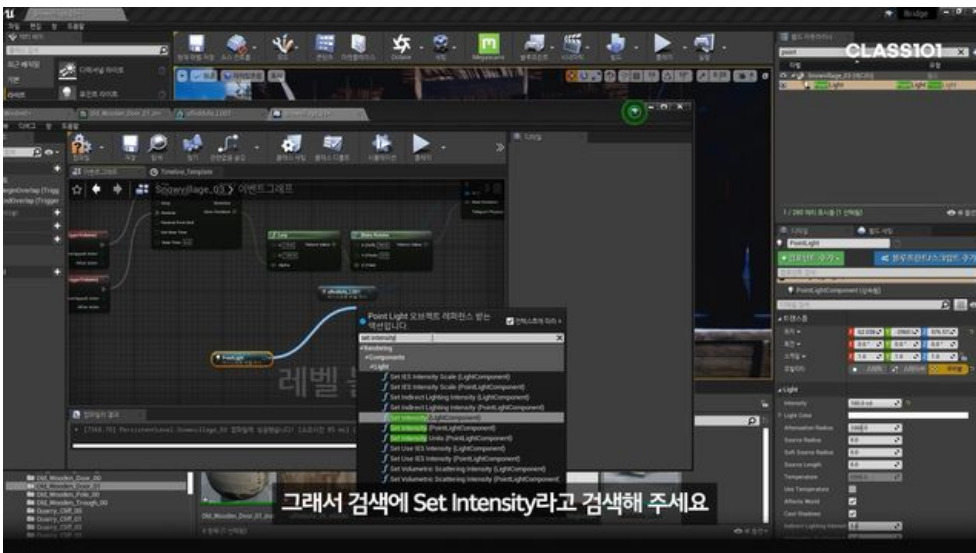




### 33:46 실내에 들어왔을 때 조명이 켜지는 것 만들기

문이 열리는 것과 같은 원리로, 트리거 볼륨, 타임라인 그리고 Lerp를 이용해 포인트 라이트의 Intensity를 세팅해주도록 하겠습니다.

우선 실내에 포인트 라이트를 적당히 배치해주시고, 적당히 밝을 때의 Intensity를 기억해주세요.



배치한 포인트 라이트를 월드 아웃라이너로부터 끌어서 레벨 블루프린트로 가져옵니다. 그리고 라이트 변수에서 드래그해 Set Intensity를 검색해 꺼내옵니다. 이 노드는 새로운 라이트 강도를 정하는 노드입니다.







이 노드를 SetActorRotation 뒤에 연결해서 배치하면 기존 트리거 볼륨에 닿았을 때 해당 포인트 라이트의 Intensity를 0으로 설정합니다.

단, 타임라인 및 Lerp에 연결되지 않았기 때문에 애니메이션 없이 갑자기 0으로 변하게 되죠.



한 개의 타임라인 노드에서 여러 개의 Float 그래프를 만들 수 있습니다. (물론 다른 유형의 변수도 혼용 가능합니다.)

여기서 라이트의 애니메이션에 사용할 그래프를 만들어줍니다. 조명이 깜빡이며 켜지는 것을 만들기 위해 사진처럼 구성했습니다.

구성의 유의점은 최소값 0, 최대값 1만 맞춰주시면 됩니다. 그 이상으로 넘어갈 경우 범위를 벗어난 값에 대해 선 Lerp에서 처리하지 않기 때문에 의도대로 애니메이션이 만들어지지 않을 수 있습니다.





그래프를 만든 후 타임라인에서 새로 생긴 Float 출력값을 새로운 Lerp 노드에 연결한 후, 문짝과 같은 방식으로 A에는 불이 꺼졌을때의 값 0을 입력하고, B에는 불이 켜졌을때의 값인 160을 입력합니다. 그리고 그것을 Set Intensity의 New Intensity에 연결하면 완료입니다.



조명이 켜지는 이벤트 또한 처음 설치한 트리거 볼륨에 기반합니다. 하지만 문이 닫히는 범위와 조명이 켜져 있을 때의 범위를 다르게 해야합니다.

왜냐하면 처음 설치한 트리거 볼륨은 입구 부근에 걸쳐있기 때문에, 캐릭터가 실내에 완전히 들어왔을 때 트리거 볼륨을 벗어나서 타임라인의 Reverse에 의해 조명이 꺼지기 때문입니다.

이럴 경우, 트리거 볼륨을 추가로 설치해서 범위를 다르게 설정해주면 되겠죠. 새로 만든 트리거 볼륨은 조명용 트리거 볼륨으로, 캐릭터가 실내에 있을 때와 그렇지 않을 때를 구분하는 용도로 배치합니다. 곧, 실내를 가득 채우는 범위를 구성하면 됩니다.

같은 방식으로 새로 만든 트리거볼륨2를 선택한 상태에서 레벨 블루프린트에서 OnActorBeginOverlap과 OnActorEndOverlap를 꺼내준 다음, 타임라인은 기존의 것을 복사하여 새로운 이벤트에 연결해줍니다.

이후 과정 또한 같이 진행해주시면 됩니다. 즉, 이 과정을 통해 문 열림과 조명이 켜지는 것은 완전히 별개의 이벤트로 동작하게 되는 것입니다. 단지 트리거 볼륨들이 비슷한 위치에 있기에 함께 동작하는 기능처럼 느껴지는 것이죠.

### [다음 수업 예고]

다음 수업에는 이번 시간에 배운 타임라인과 Lerp에 더욱 익숙해지기 위해, 훨씬 많은 값을 일괄적으로 조정하는 애니메이션을 만들 것입니다. 특정 영역에 들어가면 SkySpherBlueprint와 Exponential Height Fog 값을 바꾸어 낮에서 밤으로 바꾸는 것을 만들어봅시다.

고맙습니다! 러셀이었습니다 :)