

## CHAPTER 6

## 블루프린트로 UI 버튼 만들기

\*만약 독립형 게임 실행 모드에서, 폴리지로 배치된 나무의 텍스처가 로딩되지 않는 이슈가 있을 경우 아래 내용을 참조해주세요.

- 나무 애셋의 마스터 머티리얼로 이동해서, 머티리얼의 디테일 패널에서 **Used with Instanced Static Meshes** 를 체크해주시면 해결됩니다.

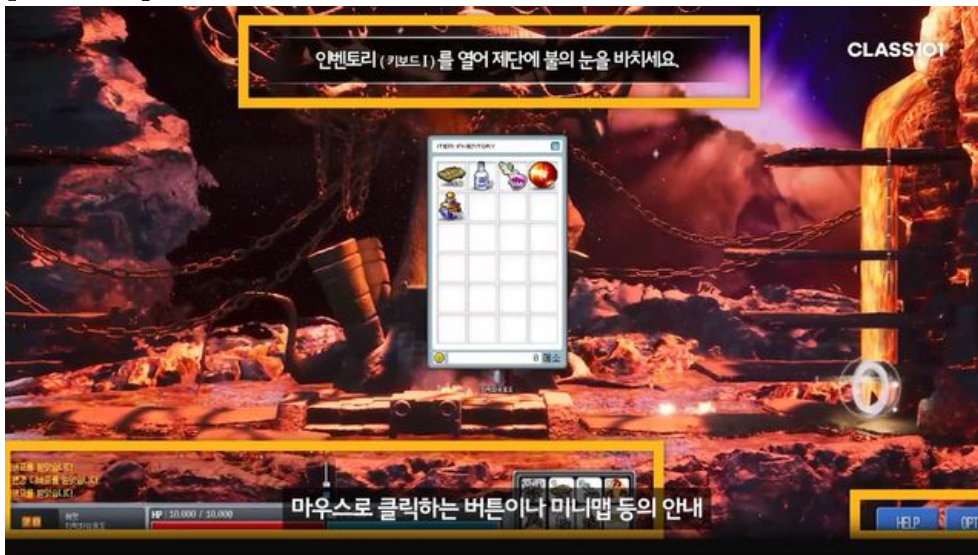
\*추가 팁: 버튼의 크기를 이미지 해상도와 같은 비율로 늘리거나 줄이면 버튼의 크기를 유동적으로 조정할 수 있습니다. 버튼의 크기 X, Y에 우선 이미지 해상도를 입력하고,  $353 \times 0.8$ ,  $136 \times 0.8$  같은 식으로 비율만 맞춰 조절한다면 어색하지 않게 조정할 수 있습니다.

## [수업 목표]

여러분 반갑습니다! 러셀입니다.

이번 시간에는 3D가 아닌 2D로, 버튼을 통해 이벤트를 실행시킬 수 있는 UI 제작에 대해 알아보겠습니다.

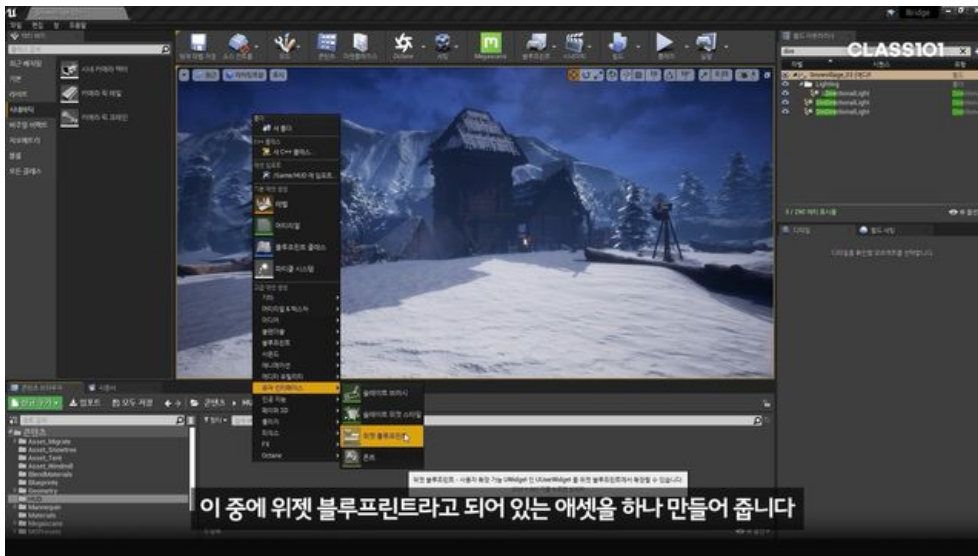
## [수업 개요]



UI는 게임 플레이에서 다양한 보조 역할을 하는 요소로, 상태 표시, 안내 메시지, 버튼 등 다양한 기능을 합니다. 일반적인 게임엔 없어도 안 될 요소이죠.

언리얼 엔진에서 UI는 주로 UMG 디자이너라는 곳에서 만들어집니다.

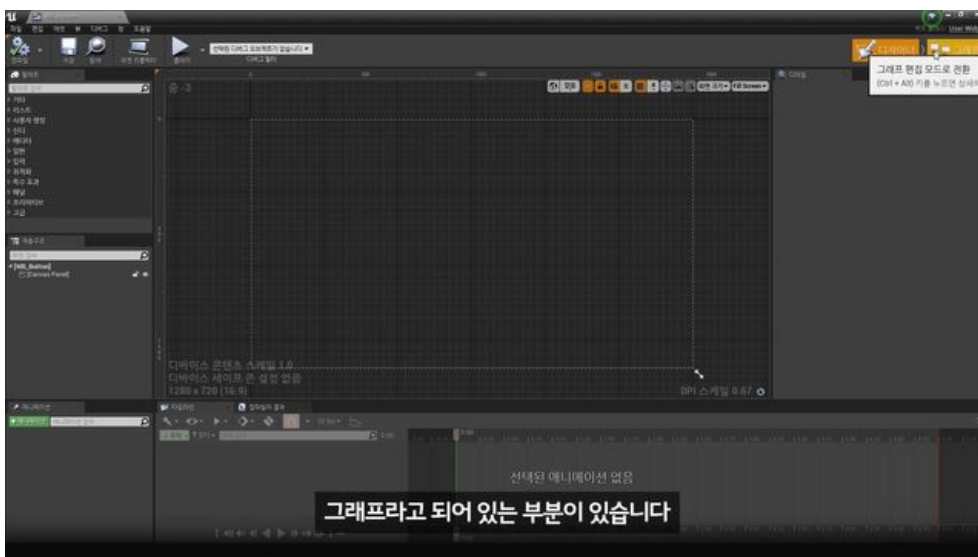
이것은 위젯 블루프린트라는 새로운 유형의 블루프린트를 통해 접할 수 있습니다.



## 1:00 위젯 블루프린트 애셋 생성

위젯 블루프린트는 클래스 블루프린트처럼 애셋의 형태로 존재합니다.

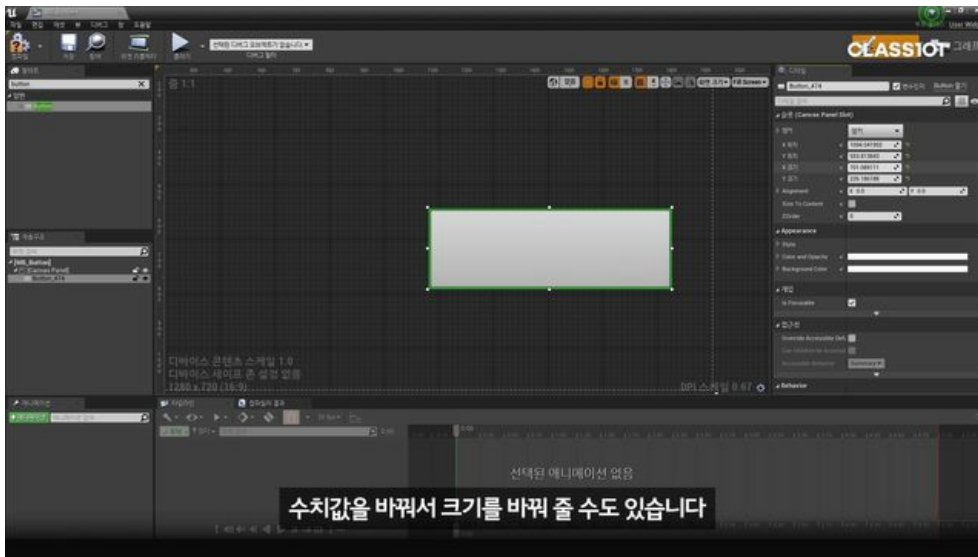
콘텐츠 브라우저에서 HUD라는 이름의 폴더를 하나 만들고, 우클릭 - 유저 인터페이스 - 위젯 블루프린트 애셋을 생성해주세요.



위젯 블루프린트는 기존 블루프린트와 다르게 생겼습니다.

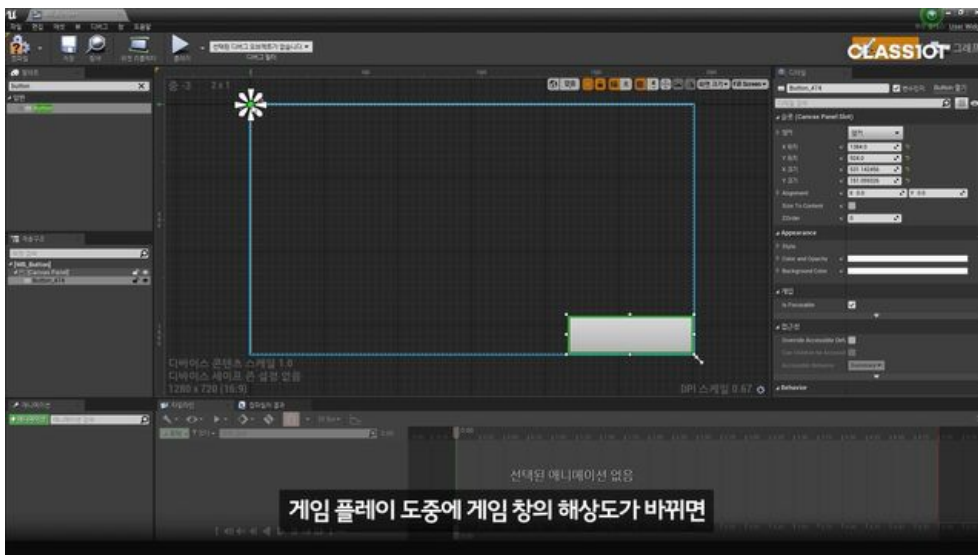
상단의 디자이너라고 되어 있는 부분이 현재 모습으로, 여기에서 텍스트, 버튼, 사진 등을 추가할 수 있고 그것이 화면에 그대로 나타납니다.

그래프라고 되어 있는 부분은 다른 블루프린트의 이벤트그래프와 동일한 기능을 하는 곳으로, 그곳에선 노드를 추가하여 위젯에 기능을 추가할 수 있습니다.



좌측의 팔레트 부분에서 다양한 유형의 기능을 꺼내 배치할 수 있습니다.

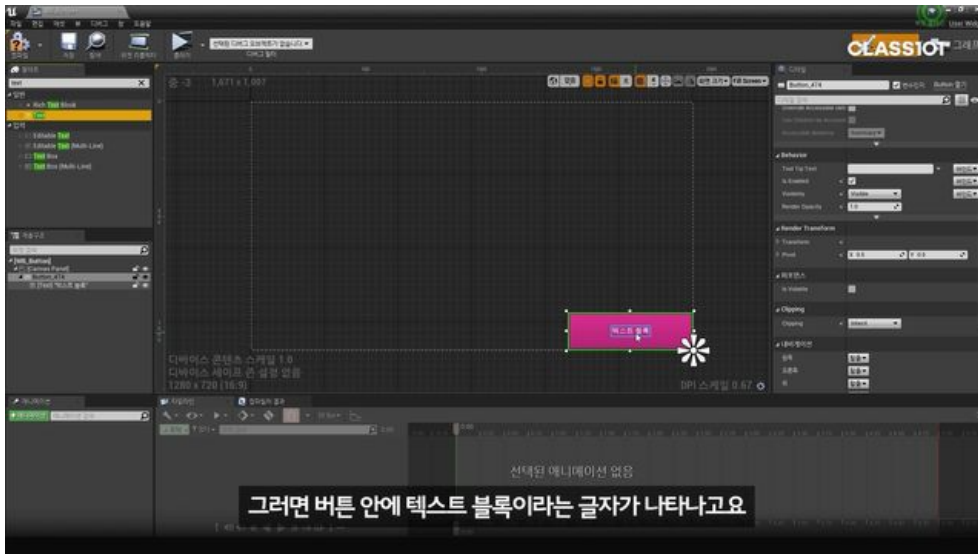
여기에 Button을 꺼내 배치해서 간단히 버튼을 만들 수 있고 크기와 위치를 쉽게 조절할 수 있습니다. 색깔 등도 쉽게 변경할 수 있습니다.



배치한 버튼을 살펴보면 핫님 모양의 아이콘을 볼 수 있습니다.

이것은 앵커라는 것으로, 현재 화면이 16:9비율을 기준으로 제작되었기 때문에 해상도가 바뀔 때 해당 버튼이 어디를 기준으로 정렬될 것인지를 정하는 요소입니다.

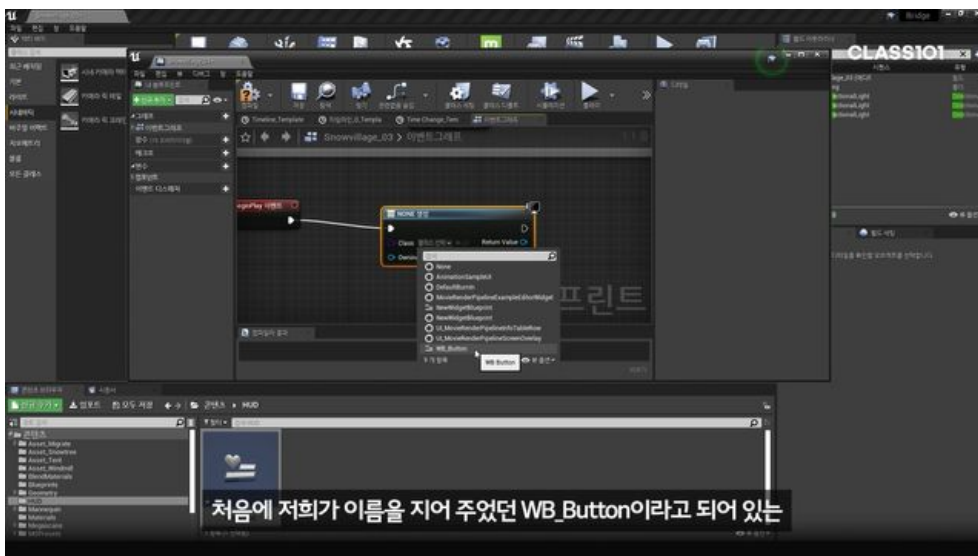
사진처럼 버튼이 우측 하단 구석에 배치됐다면 앵커 또한 우측 하단 구석에 배치해주는 것이 바람직하겠지요. 앵커는 직접 옮겨 수정할 수도 있지만 버튼을 눌러 디테일 패널에서도 빠르게 바꿀 수 있습니다.



그러면 버튼 안에 텍스트 블록이라는 글자가 나타나고요

팔레트에서 텍스트를 검색해 버튼 아래로 종속시키면 버튼에 텍스트가 나타납니다.

텍스트의 디테일 패널에서 내용을 작성할 수 있고 색, 선 등을 바꿀 수 있습니다. 폰트 또한 바꿔줄 수 있지만 언리얼에선 콘텐츠 브라우저에 폰트 애셋을 임포트해야 사용자 설정 폰트로 교체할 수 있다는 점 참고해주세요.



처음에 저희가 이름을 지어 주었던 WB\_Button이라고 되어 있는

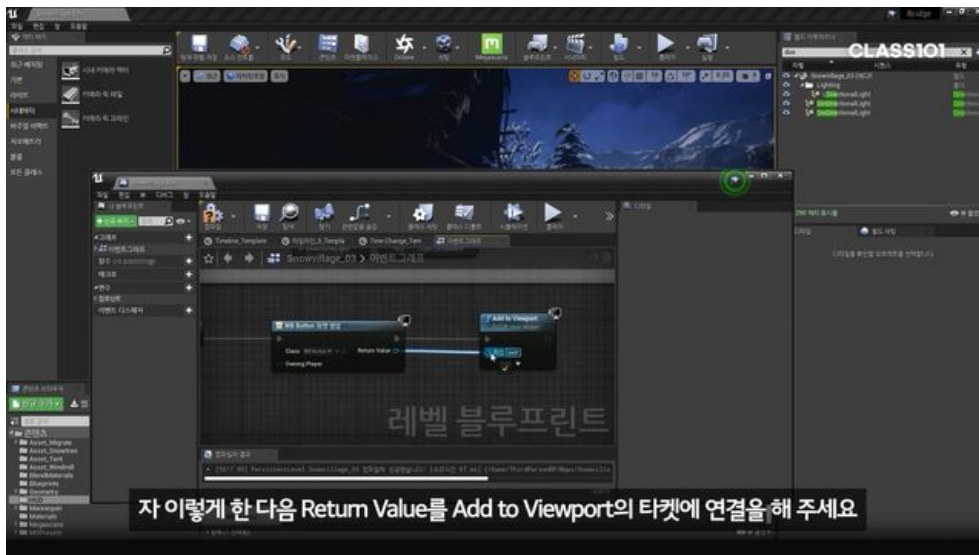
7:05 위젯 블루프린트를 레벨에 추가하기



위젯 블루프린트에서 작업을 마쳤다면 이것을 레벨에 추가하여 화면에 나타나게 해야 합니다.

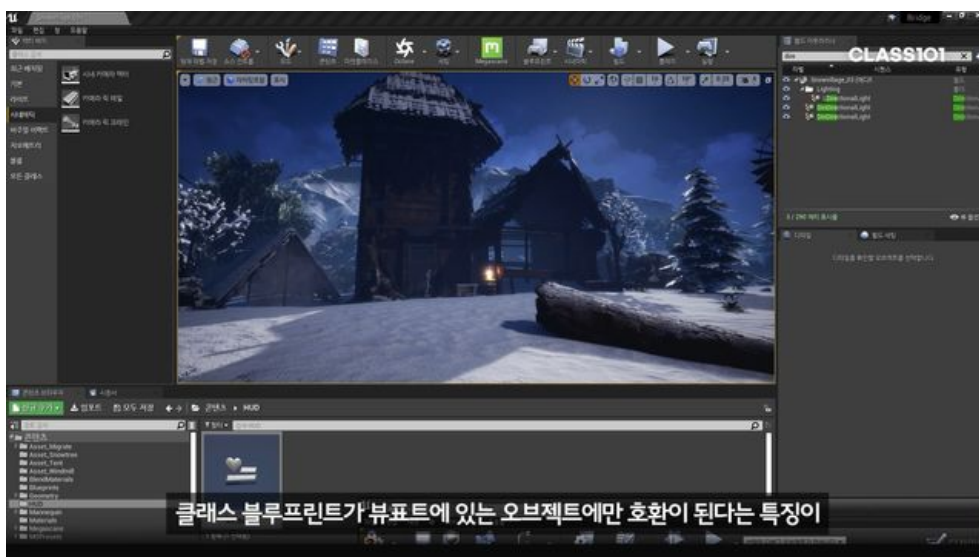
레벨 블루프린트를 열어 이벤트 **BeginPlay** 노드를 검색해 꺼내줍니다. BeginPlay는 레벨이 시작됐을 때 발생하는 이벤트입니다. 따라서 무조건 실행되는 이벤트라고 생각하시면 됩니다.

BeginPlay에 이어 **위젯 생성**이라는 노드를 검색합니다. 그리고 애셋 선택을 눌러 방금 만든 WB\_Button 위젯 애셋을 선택해줍니다.



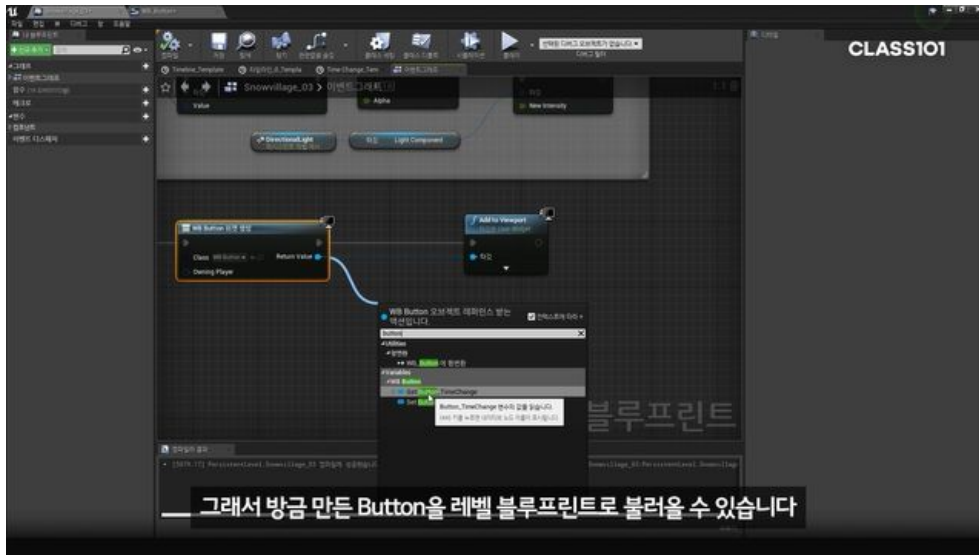
다음으로 **Add to Viewport** 를 연결한 후 위젯 생성의 Return Value를 타겟에 연결해줍니다. Add to Viewport는 화면에 해당 위젯을 표시하는 역할을 합니다.

그리고 플레이해보면 조금 전에 만든 버튼이 화면에 잘 나타납니다.



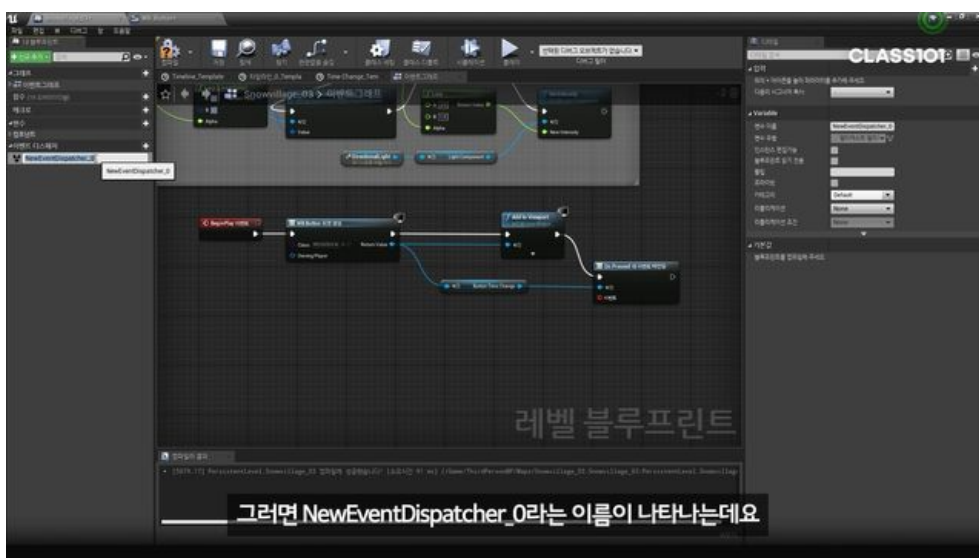
하지만 위젯 블루프린트도 클래스 블루프린트와 마찬가지로 레벨에 있는 특정 오브젝트를 불러오려면 추가적인 작업을 해주어야 합니다.

그래서 저희는 버튼 입력 이벤트를 레벨 블루프린트에서 호출해 하늘 변경을 버튼을 눌러 실행시켜보겠습니다.



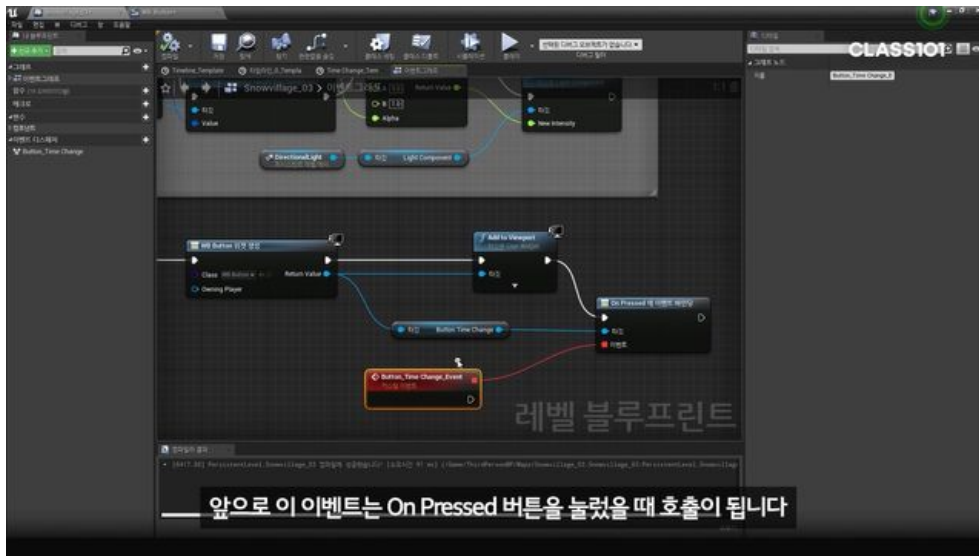
조금 전에 만든 위젯 생성 노드의 Return Value를 끌어 Get Button으로 불러옵니다.

불러와진 버튼 변수를 드래그해 On Pressed에 이벤트 바인딩을 누릅니다. 그러면 특정 이벤트에 버튼을 누르는 액션을 할당할 수 있습니다.



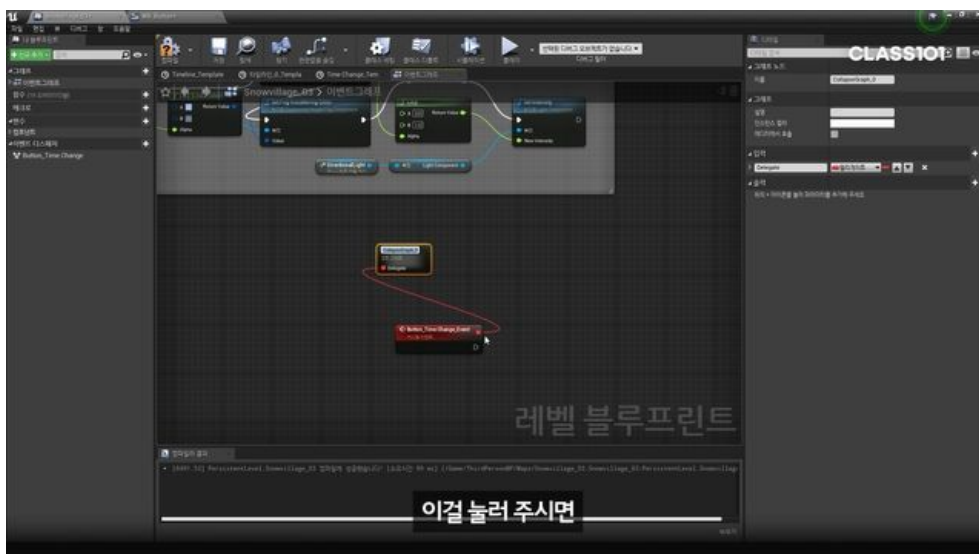
레벨 블루프린트의 좌측에 이벤트 디스패처라는 부분에서 커스텀 이벤트를 생성할 수 있습니다.

이렇게 만든 커스텀 이벤트를 이름을 바꿔주고 드래그해서 이벤트그래프로 가져옵니다.

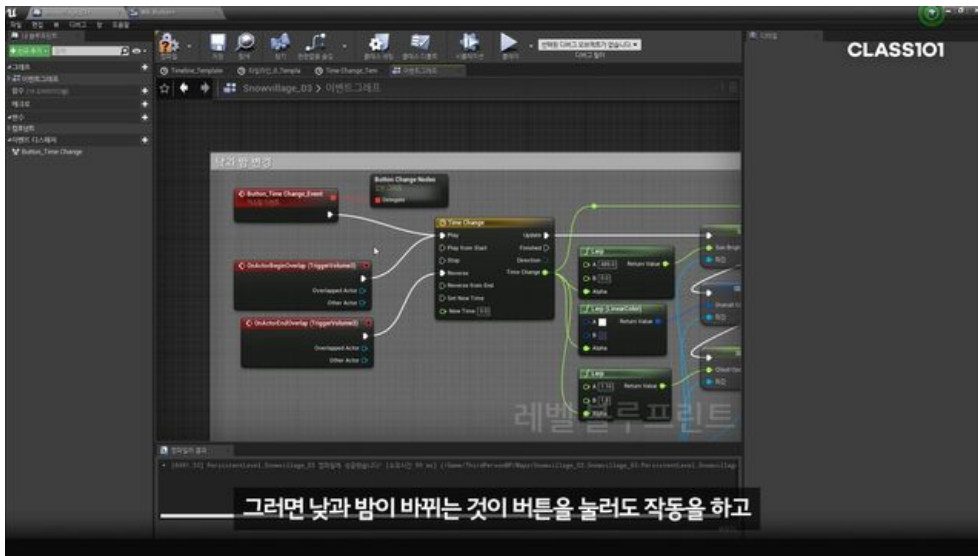


커스텀 이벤트 노드의 윗쪽 붉은 부분을 드래그해 On Pressed에 이벤트 바인딩 노드의 이벤트 부분이랑 연결합니다.

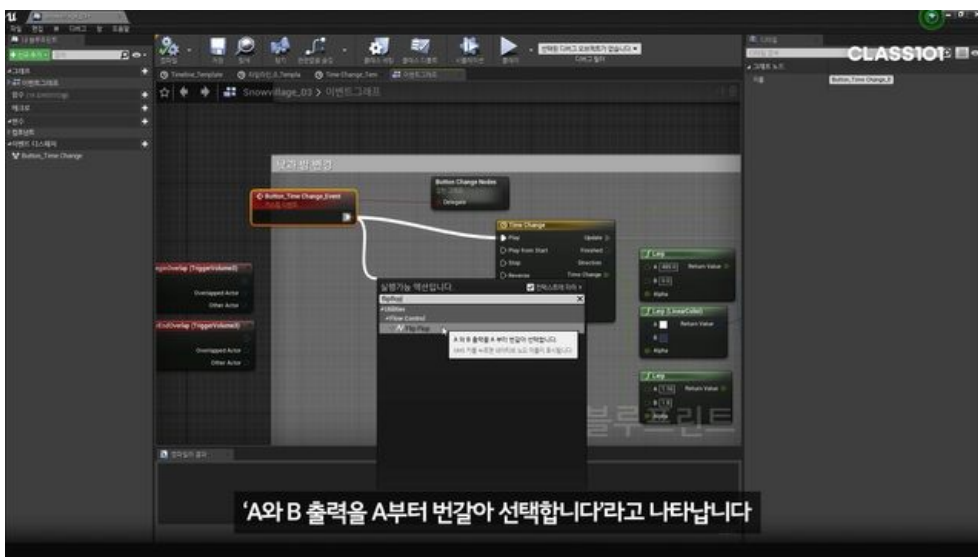
이제 커스텀 이벤트는 버튼을 누를 때 이벤트를 실행합니다.



이어진 다른 노드들은 이제 수정하지 않을 것이기에 편의를 위해 접어줍니다. 노드들을 모두 선택하고 우클릭해서 접을 수 있습니다. 접힌 노드는 기능을 하는 것에 변화를 주지 않으니 안심하셔도 됩니다.



이렇게 만들어진 커스텀 이벤트를 기존의 타임라인에 연결해주면 완성입니다.



버튼을 눌러보니 낮에서 밤으로 잘 변하지만 한번 더 눌렀을 때 다시 낮으로 변하게 하려면 어떻게 해야 할까요?

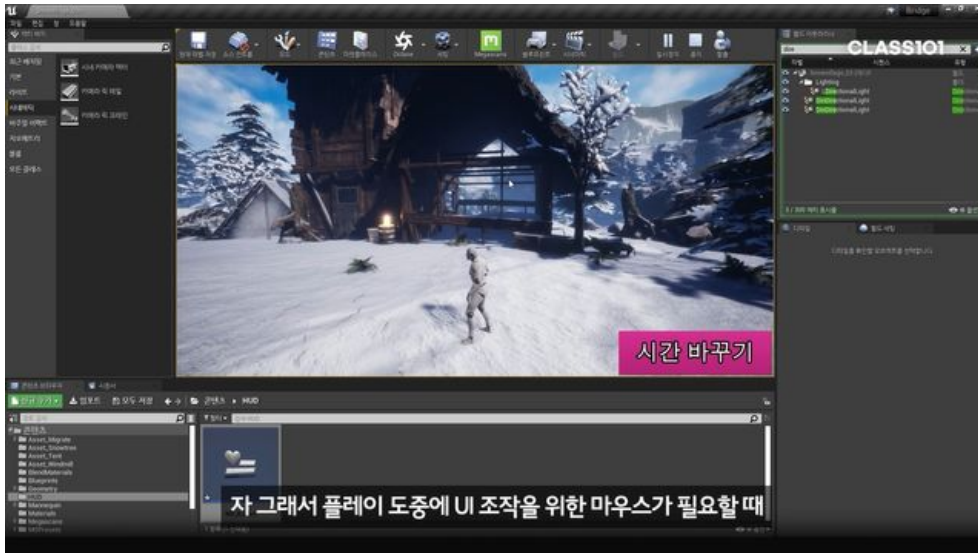
그럴 땐 Flip Flop이라는 노드가 유용합니다.

Flip Flop은 이벤트가 여러 번 실행되었을 때 각각 다른 이벤트를 번갈아 실행시키는 노드입니다.

그래서 A에는 타임라인의 Play에, B에는 Reverse에 연결해줍니다.

이렇게 하면 버튼을 여러번 눌렀을 때 시간대가 의도대로 잘 변하는 것을 볼 수 있습니다.

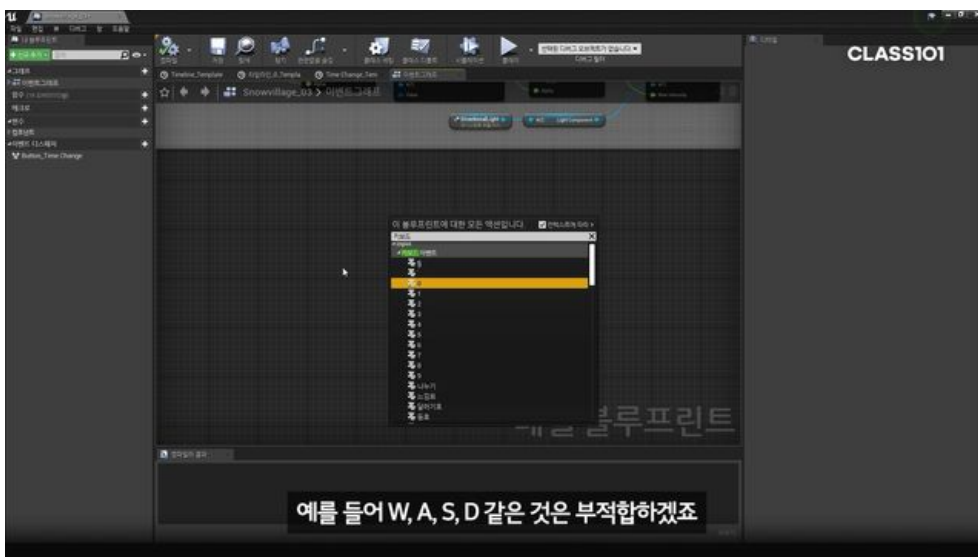




## 18:40 플레이 도중 UI 조작을 위한 마우스 꺼내는 기능 만들기

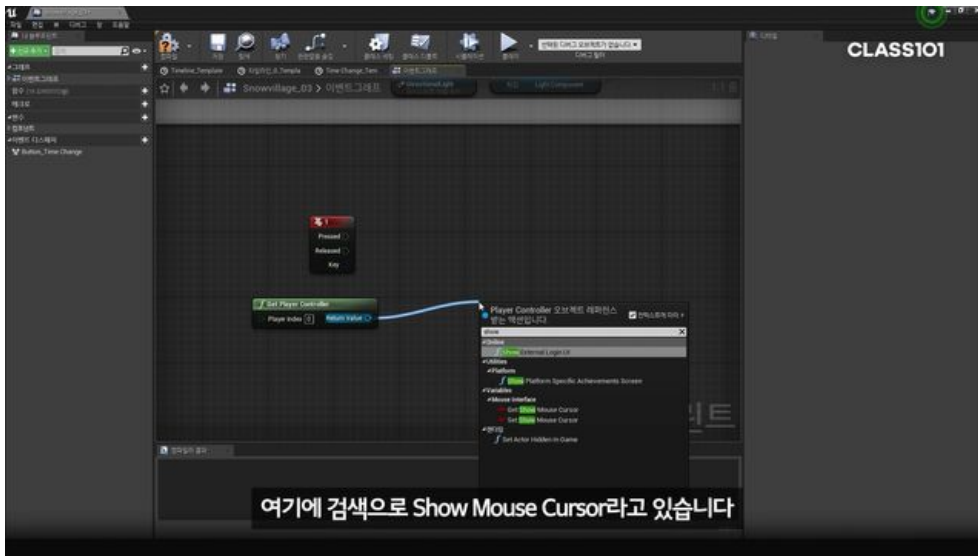
3인칭 플레이에선 마우스로 카메라를 조작하기 때문에 UI를 클릭하기 어렵습니다. 기본적으로 Shift+F1을 눌러 꺼낼 수 있지만 다소 불편하죠.

이 기능마저도 블루프린트로 직접 만들어줄 수 있습니다.



이벤트그래프에서 노드를 검색할 때 키보드 ~ 를 붙히고 검색하면 키보드에서 입력되는 대부분의 키로부터 이벤트를 실행할 수 있습니다.

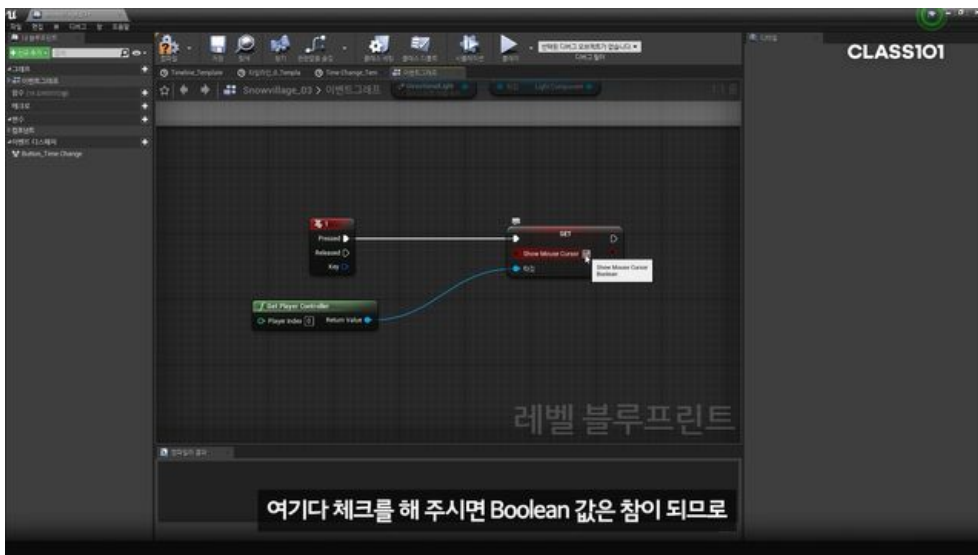
마우스 on/off 기능을 만들 것인데, 저는 키보드 1번에서 시작하는 이벤트를 가져오겠습니다. 이 때 다른 조작키와 겹치지 않게 유의해주세요.



노드 중 **Get Player Controller** 라는 노드가 있습니다. 이 노드는 플레이어의 컨트롤러, 즉 마우스와 키보드 관련 이벤트를 실행할 때 사용하는 노드입니다.

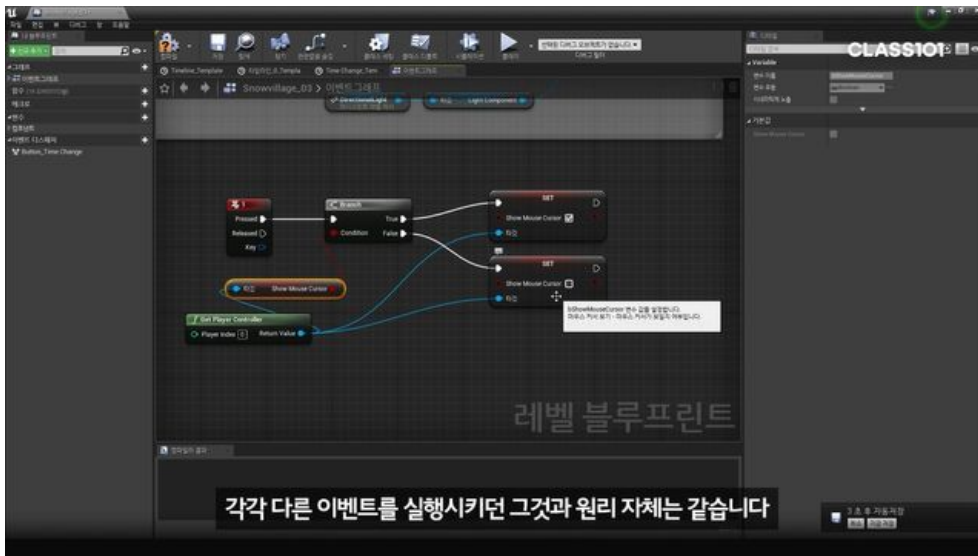
이 노드를 꺼낸 후 Return Value에서 이어서 Show Mouse Cursor라고 검색한 후 **Set Show Mouse Cursor** 를 꺼냅니다.

Show Mouse Cursor는 Boolean 유형의 변수입니다.



이렇게 연결해주면 1번을 눌렀을 때 마우스 커서가 보이게 됩니다.

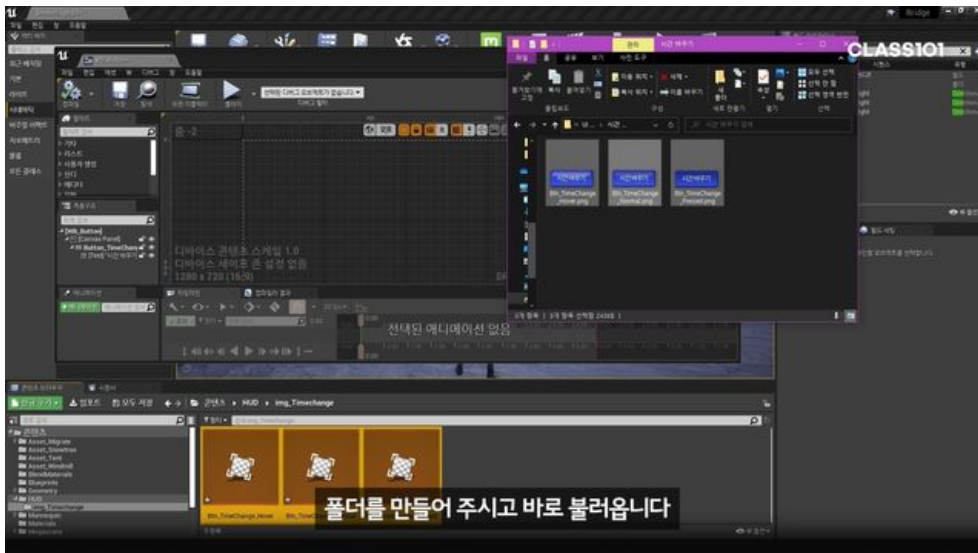
하지만 역시 다시 마우스를 끄고 3인칭 조작에 집중하기 위한 옵션이 없기 때문에 만들어주도록 합시다.



Flip Flop을 사용해서 해도 되지만 이번엔 다른 방식으로 해보겠습니다.

Get Show Mouse Cursor로 Boolean 변수의 현재 상태를 점검한 다음, 현재 마우스 커서가 보이고 있다면 보이지 않게, 반대로 보이지 않는다면 보이게 설정하는 원리입니다.

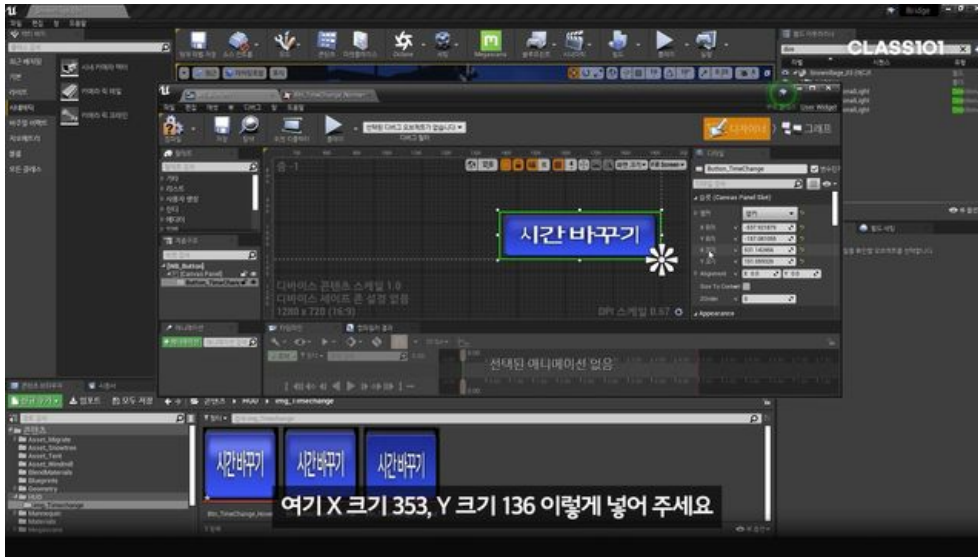
Branch는 현재 Boolean 변수의 True/False를 바탕으로 각각 다른 이벤트를 실행하는 아주 유용한 노드입니다. 이것을 활용해 위 사진처럼 구성해주시면 됩니다.



## 27:00 버튼에 사진을 넣어 디자인하기

지금 만든 버튼의 디자인이 너무 초라하기 때문에 버튼의 모양에 맞게 외부 프로그램에서 버튼을 사진 형태로 디자인하여 언리얼로 불러오게 됩니다.

여기에선 제가 미리 디자인해둔 사진 애셋들을 활용하도록 하겠습니다.



버튼에는 세 종류가 있습니다. 바로 Normal, Hovered, Pressed입니다.

Normal은 일반 상태에서 보이는 형태를 나타냅니다.

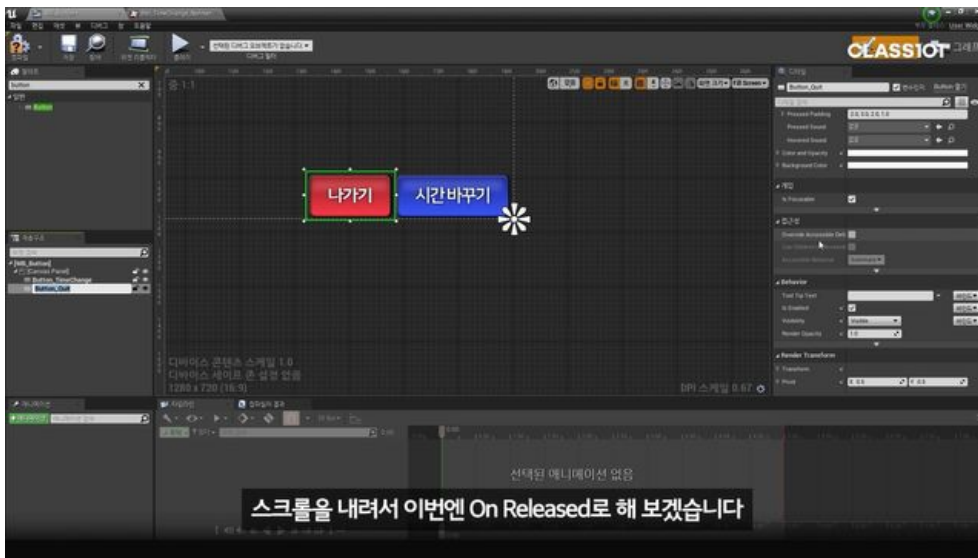
Hovered는 마우스 커서가 올라가있을 때 보이는 형태를 나타냅니다.

Pressed는 마우스 커서로 눌렀을 때 보이는 형태를 나타냅니다.

버튼을 눌러 디테일 패널에 이미지를 넣는 공간에 넣어줄 수 있습니다.

버튼의 크기를 이미지의 해상도에 맞게 세팅해주세요.

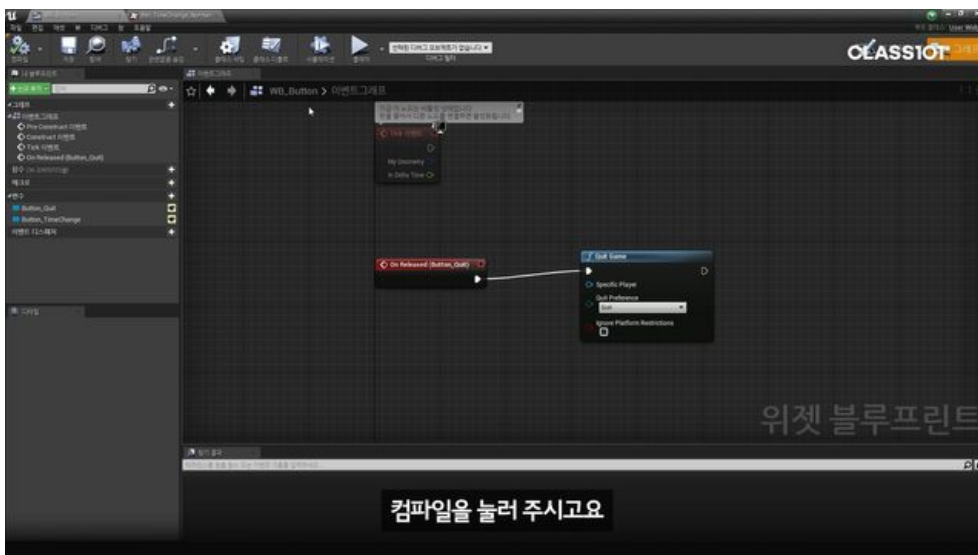
그리고 각 상태 (Normal, Hovered, Pressed)에 맞게 이미지를 넣어주시면 됩니다.



다음으로는 나가기 버튼을 만들어볼겁니다.

게임 종료는 위젯 블루프린트에서 실행할 수 있기 때문에 굳이 레벨 블루프린트에 이벤트 디스패처를 활용하지 않아도 됩니다.

버튼을 같은 방식으로 디자인해주시고 디테일 패널 하단의 On Released를 추가해 이벤트그래프에 버튼의 On Released 이벤트를 불러옵니다.



그리고 아주 간단하게 Quit Game 노드를 연결해주면 완료입니다.

## [다음 수업 예고]

이렇게 언리얼 엔진에서의 UI 제작에 대해 알아보았습니다.

다음 시간에는 시퀀서라는 기능을 이용해 환경을 촬영하는 것에 대해 알아보시다! 감사합니다.



