19_ 사운드

<제목 차례>

인천대학교 컴퓨터공학부 박종승

1. 개요

이 장에서는 사운드에 대해서 다룬다.

사운드는 게임에 있어서 필수적인 요소이다. 사운드는 게임의 현실감을 높이고 몰입감을 높이고 플레이어를 즐겁게 한다.

<참고> 사운드에 대한 전체적인 문서는 다음의 링크를 참조하자.
https://docs.unrealengine.com/working-with-audio-in-unreal-engine/

언리얼의 오디오 시스템에 대해서 알아보자. 엔진에서는 16비트 또는 24비트 PCM 포멧의 .wav 파일을 임포트할 수 있다. 최대 채널 수는 최대 8 채널까지 지원하며 샘플 레이트는 어느것이든 상관없다.

<참고> 엔진이 지원하는 임포트 포맷은 .wav 파일 뿐만 아니라 편의를 위해서 .ogg, .aiff, .flac의 포멧도 지원한다. 이들 포멧은 임포트시에 내부적으로 16 비트 PCM 파일로 변환되어 임포트된다. 이들 포멧은 Xiph.Org 재단의 오픈소스 코덱인 .ogg (Ogg-Vorbis), 애플의 .aiff (AIFF; Audio Interchange File Format), Xiph.Org 재단의 무손실 압축 포멧인 .flac (FLAC; Free Lossless Audio Codec)이다.

에디터에서 사운드 파일을 임포트하면 **사운드 웨이브**(Sound Wave) 애셋이 생성된다. **사운드 웨이브** 에셋은 레벨이 바로 배치할 수도 있고 **사운드 큐 에디터**에서 **사운드 큐**를 만드는데 사용될 수도 있다. 엔진에서는 **사운드 큐**(Sound Cue)의 형태로 합성 사운드를 만들 수 있다. 엔진에서는 **사운드 큐**의 편집을 위한 **사운드 큐 에디터**를 제공한다. 에디터에서는 사운드 노드를 통해서 사운드들을 합성하고 수정해서 최종 출력을 만들도록 한다.

이제부터 사운드와 관련된 다양한 애셋 타입에 대해서 알아보자.
먼저, **사운드 큐**(Sound Cue)에 대해서 알아보자. 사운드 큐는 오디오 재생 방법을 수정하거나 오디오 효과를 결합하거나 오디오 내용을 수정해서 만든 합성 사운드이다. 사운드 큐에서는 사운드 노드를 사용하여 합성 사운드를 만들고 이를 최종 출력으로 내보낸다. 사운드 큐 에디터에서는 사운드 노드들을 사용하여 사운드 큐를 만들 수 있도록 편집 기능을 제공한다.
다음, **사운드 감쇠**(Sound Attenuation) 애셋에 대해서 알아보자.
사운드 감쇠 애셋은 사운드의 감쇠 속성의 정의를 독립 애셋으로 저장할 수 있도록 하였다. 감쇠 속성은 사운드 볼륨이 거리에 따라 어떻게 들리는지를 결정하는 속성이다. 매번 감쇠 속성을 명시할 필요 없이 저장되어있는 사운드 감쇠 애셋을 사용하여 편리하게 지정할 수 있다.
그 외에도 **리버브 이펙트**(Reverb Effects), **사운드 클래스**(Sound Class), **사운드 믹스**(Sound Mix), **다이얼로그 보이스**(Dialogue Voice), **다이얼로그 웨이브**(Dialogue Wave) 등의 애셋이 있다.

<참고> 사운드 감쇠에 대한 자세한 내용은 다음의 문서를 참조하자.
https://docs.unrealengine.com/sound-attenuation-in-unreal-engine/
다양한 사운드 애셋 유형에 대해서는 다음의 문서를 참조하자.
https://docs.unrealengine.com/4.27/WorkingWithAudio/Overview/

2. 사운드 웨이브와 사운드 큐와 앰비언트 사운드 액터

이 절에서는 **사운드 웨이브**와 **사운드 큐**와 **앰비언트 사운드** 액터에 대해서 학습한다.
사운드 웨이브는 웨이프 포맷의 사운드 데이터를 가지는 에셋이다. 사운드 웨이브 애셋은 외부의 웨이브 파일을 콘텐츠 브라우저로 임포트하여 만들어진다.
사운드 큐는 사운드 큐 에디터을 통해서 만드는 애셋이다. 사운드 큐 에디터에서는 여러 사운드 웨이브 에셋을 사용하고 여러 노드들의 조합으로 최종 사운드 출력을 만들어내도록 하는 그래프를 만들 수 있다.
사운드 웨이브와 **사운드 큐**는 **오디오 컴포넌트** 내에 탑재된다. **오디오 컴포넌트**는 다른 컴포넌트처럼 일반적인 액터 내에 포함될 수 있다. 또한, 엔진은 **오디오 컴포넌트**가 단독으로 레벨에 배치될 수 있도록 **앰비언트 사운드** 액터를 제공한다.
이러한 액터와 컴포넌트의 관계를 더 자세하게 알아보자. 레벨에는 액터 유형만 배치할 수 있다. 그리고 액터는 컴포넌트들의 컨테이너의 역할을 한다. 이와 관련하여 우리가 이미 자주 다루었던 스태틱 메시의 경우를 설명한 후에, 이와 유사하게 앞으로 다룰 사운드의 경우를 설명한다.
레벨을 꾸밀 때에 우리는 **콘텐츠 브라우저**에서 스태틱 메시 파일을 드래그해서 레벨에 배치하였다. 우리가 A라는 스태틱 메시 파일을 레벨에 배치한다고 하자. 레벨에는 새로운 **StaticMeshActor** 유형의 액터 인스턴스가 추가된다. 추가된 액터의 내부에는 하나의 **StaticMeshComponent**가 포함되어 있다. 이 컴포넌트는 내부의 **StaticMesh** 속성값으로 우리가 드래그해서 배치한 A 스태택 메시 파일을 가지고 있다. 즉, 레벨에 배치할 수 있는 객체는 액터 유형이지만 실제로 모든 기능은 컴포넌트에서 수행하고 있음을 의미하고 있다.
이제 사운드의 경우를 생각해보자. 사운드의 실제 내용이 담긴 애셋 파일은 **사운드 웨이브** 애셋 파일이나 또는 **사운드 큐** 애셋 파일이다. 우리가 A라는 사운드 웨이브 파일 또는 사운드 큐 파일을 레벨에 배치한다고 하자. 레벨에는 새로운 **앰비언트 사운드**(**AmbientSound**) 유형의 액터 인스턴스가 추가된다. 추가된 액터의 내부에는 하나의 **오디오 컴포넌트**(**AudioComponent**)가 포함되어 있다. 이 컴포넌트는 내부의 **Sound** 속성값으로 우리가 드래그해서 배치한 A 애셋 파일을 가지고 있다.
오디오 컴포넌트(**AudioComponent**)를 다른 액터에 추가할 수도 있다. 예를 들어 장작불 액터를 만든다면, 액터에 장작불의 타는 모습을 표현하는 컴포넌트를 추가하고 장작불의 타는 소리를 표현하는 오디오 컴포넌트를 추가하면 된다.

이제부터 예제를 통해서 학습해보자

---

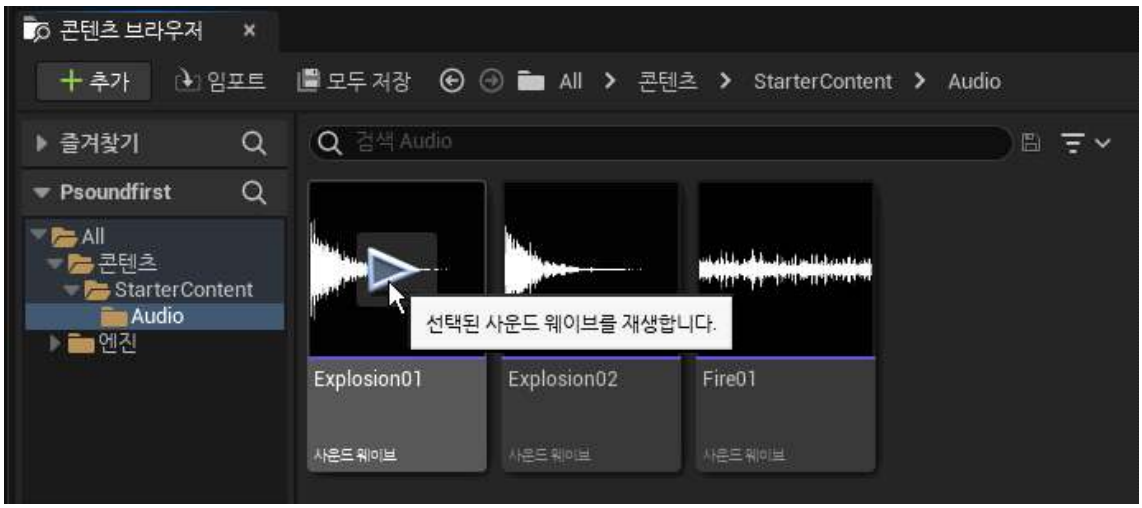
1. 새 프로젝트 **Psoundfirst**를 생성하자. 먼저, 언리얼 엔진을 실행하고 **언리얼 프로젝트 브라우저**에서 왼쪽의 **게임** 탭을 클릭하자. 오른쪽의 템플릿 목록에서 **기본** 템플릿을 선택하자. **프로젝트 이름**은 Psoundfirst로 입력하고 **생성** 버튼을 클릭하자. 프로젝트가 생성되고 언리얼 에디터 창이 뜰 것이다. 창이 뜨면, 메뉴바에서 **파일** » **새 레벨**을 선택하고 **Basic**을 선택하여 기본 템플릿 레벨을 생성하자. 그리고, 레벨 에디터 툴바의 저장 버튼을 클릭하여 현재의 레벨을 MyMap으로 저장하자.
그리고, **프로젝트 세팅** 창에서 **맵&모드** 탭에서의 **에디터 시작 맵** 속성값을 MyMap으로 수정하자.

2. 이번 프로젝트에서 필요한 외부 애셋을 준비하자.
필요한 애셋은 **시작용 콘텐츠**에 포함되어 있다. **시작용 콘텐츠**가 포함되어 있는 다른 프로젝트를 찾아보거나 또는 **시작용 콘텐츠**가 포함되도록 새 임시 프로젝트를 만들자. 임시 프로젝트에서 **Content**

폴더 아래에서 다음의 파일들을 찾아보자. 다음의 파일들을 우리의 프로젝트로 복사하여 가져오자. 폴더 구조가 동일하게 되도록 하자. 가져온 후에 임시 프로젝트는 종료하고 삭제하면 된다.

```
StarterContent/Audio/Explosion01.uasset
StarterContent/Audio/Explosion02.uasset
StarterContent/Audio/Fire01.uasset
```

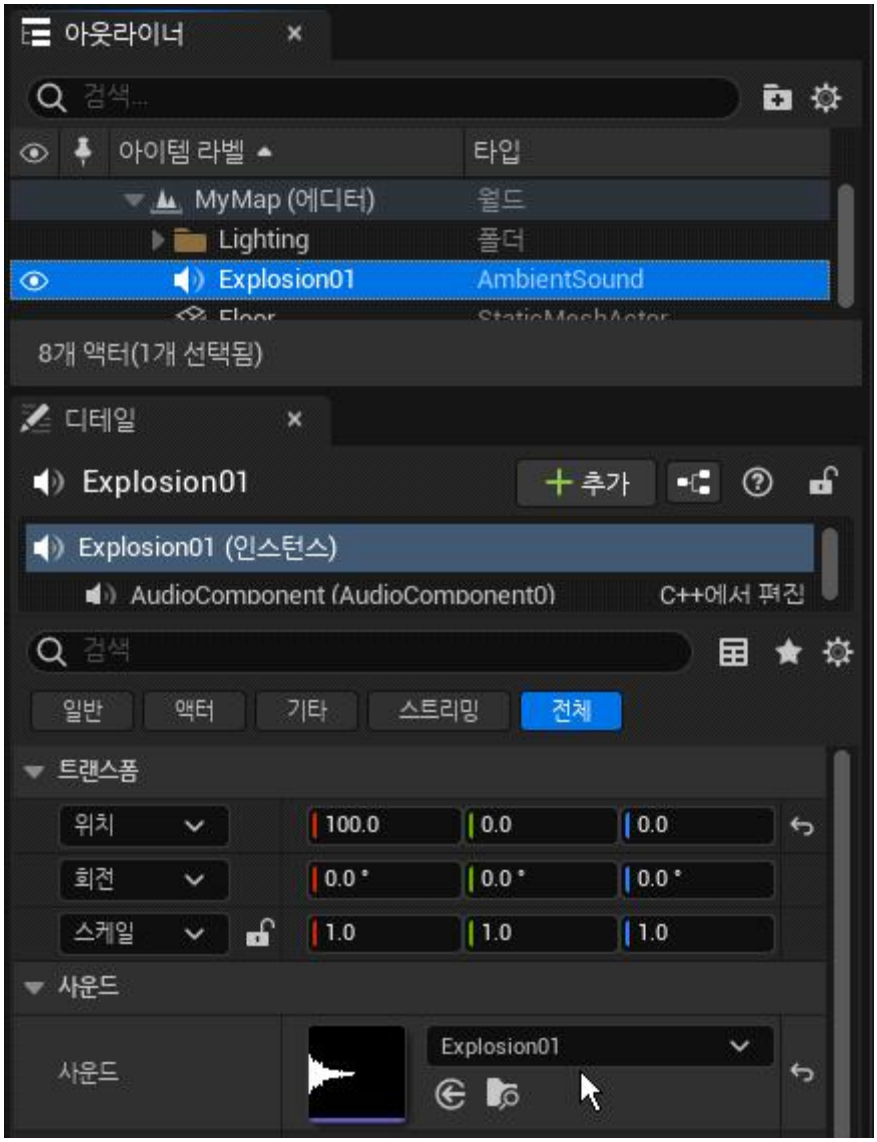
3. 콘텐츠 브라우저에서 **콘텐츠** » **StarterContent** » **Audio** 아래로 이동하자. 여러 사운드 웨이브 에셋이 보일 것이다.
사운드 웨이브 애셋은 사운드 웨이브 데이터 파일에 대한 애셋이다. 마우스를 애셋의 중간에 가져가면 플레이 아이콘이 보인다. 클릭하면 재생해볼 수 있다.



사운드 웨이브 애셋을 생성하는 방법을 알아보자.
사운드 웨이브 애셋을 생성하려면 외부의 .wav 파일로부터 임포트하면 된다. 사운드 파일을 임포트하는 방법은 **콘텐츠 브라우저**에서 **임포트** 버튼을 클릭하고 사운드 파일을 선택하면 된다. 또는 윈도우 파일 탐색기에서 사운드 파일을 **콘텐츠 브라우저**로 드래그해도 된다.

4. 첫 번째 테스트를 해보자.
콘텐츠 브라우저에서 **사운드 웨이브** 애셋인 **Explosion01**을 드래그해서 레벨에 배치해보자. 드래그할 때에 애셋 아이콘 정중간의 재생 아이콘이 보일 때에 드래그하는 것이 아니라 약간 비켜서 재생 아이콘이 보이지 않을 때에 드래그하자. 배치한 후에 플레이해보자. 플레이 시작과 더불어 바로 폭발 소리가 재생될 것이다.

5. 배치된 **Explosion01**이 선택된 상태에서 **아웃라이너** 탭을 살펴보자. 액터 유형이 **AmbientSound**인 인스턴스 **Explosion01**이 배치되어 있을 것이다. 이 인스턴스는 내부에 컴포넌트로 **AudioComponent**를 가지는 것을 알 수 있다.
디테일 탭의 속성을 보면 **사운드(Sound)** 속성값이 **Explosion01**으로 되어 있다. 이는 바로 우리가 드래그해서 배치했던 **사운드 웨이브** 애셋인 **Explosion01**에 해당한다. 이 속성값을 다른 것으로 바꾸면 바꾼 **사운드 웨이브** 애셋의 사운드가 재생될 것이다.

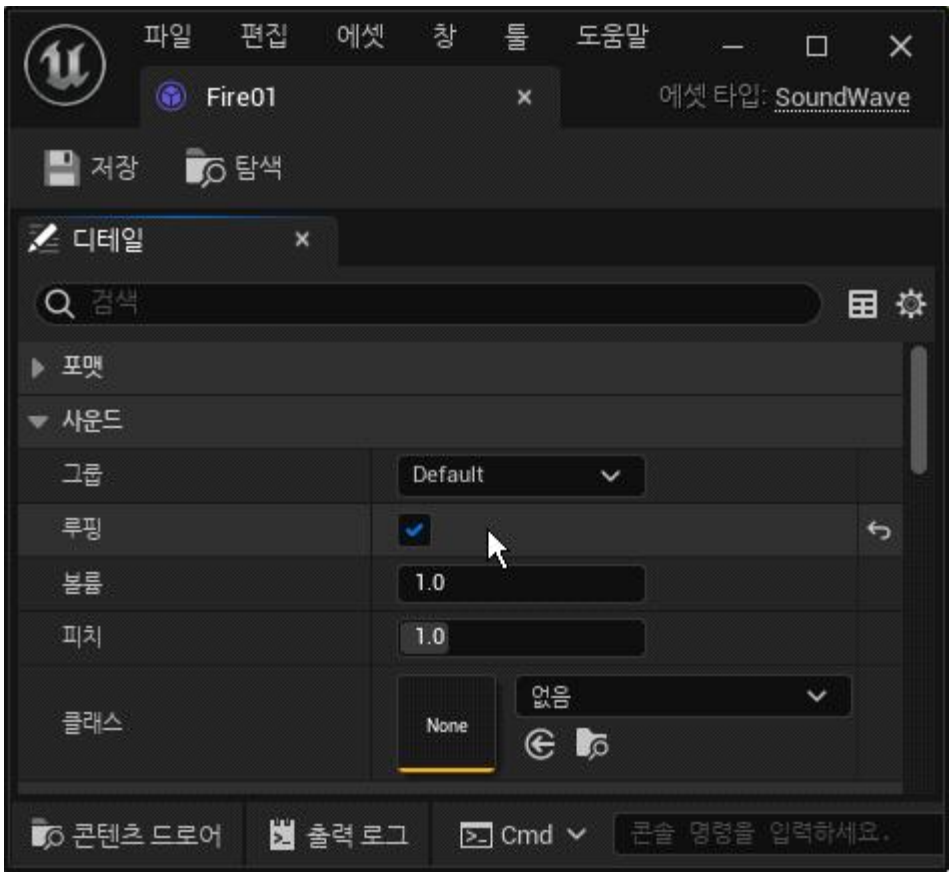


테스트를 해본 후에는, 레벨에 추가했던 **Explosion01** 인스턴스를 제거해서 다시 원래 상태로 되돌려 두자. **Explosion01** 인스턴스가 **아웃라이너**에서 사라질 때까지 **Ctrl+Z**를 클릭해서 되돌려도 된다.

6. 두 번째 테스트를 해보자.
콘텐츠 브라우저에서 **사운드 웨이브** 애셋인 **Fire01**을 드래그해서 레벨에 배치해보자. 배치한 후에 플레이해보자. 불타는 소리가 재생될 것이다. 재생은 멈추지 않고 계속될 것이다. 반복 재생되기 때문이다.

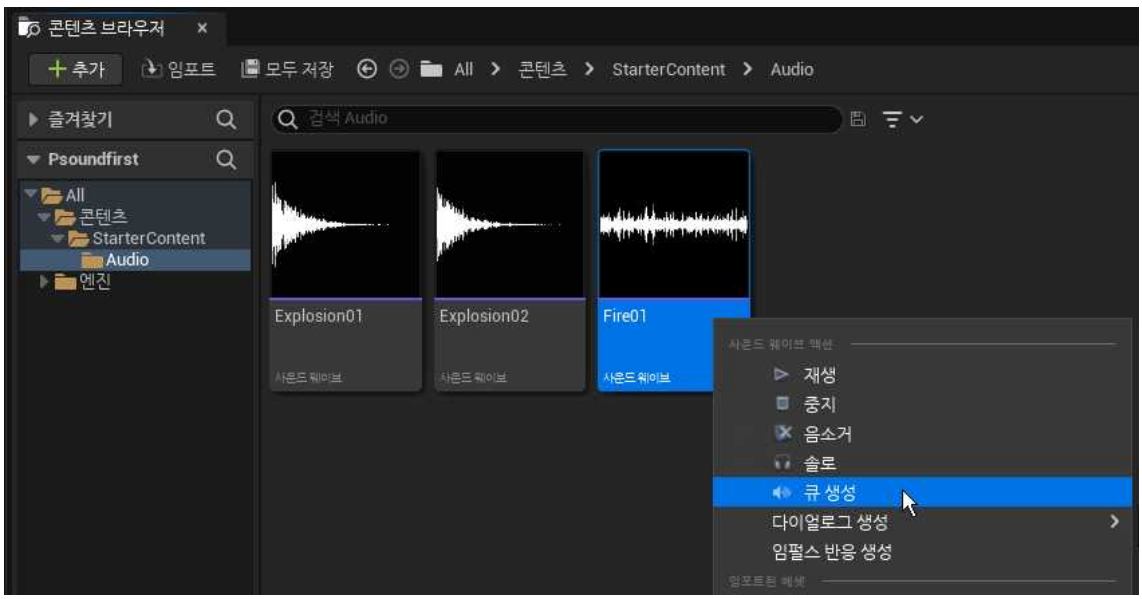
7. 콘텐츠 브라우저에서 **Fire01** 사운드 웨이브 애셋으로 마우스를 가져가서 정중간을 약간 비켜서 재생 아이콘이 보이지 않을 때에 더블클릭해보자. 애셋의 속성을 보여주는 디테일 탭이 단독 창으로 뜰 것이다.
사운드 영역에 **루핑(Looping)** 속성의 체크박스가 체크되어 있을 것이다. 이 속성값에 따라서 반복 재생 여부가 정해진다.
체크박스를 해제하고 레벨을 플레이해보자. 사운드가 한 번만 재생될 것이다.
테스트를 해본 후에는, 체그박스에 원래대로 체크해두자. 또한 레벨에 추가했던 **Fire01** 인스턴스를

제거해서 다시 원래 상태로 되돌려두자.



8. 이제 처음으로 사운드 큐를 만들어보자.

콘텐츠 브라우저에서 **사운드 웨이브** 애셋인 **Fire01**에서 우클릭하자. 드롭다운 메뉴에서 **큐 생성**을 클릭하자. 선택된 **사운드 웨이브** 애셋을 사용하여 **사운드 큐** 에셋을 생성한다. 생성된 **사운드 큐**의 이름을 **Fire01_Cue**로 그대로 두자.



9. 사운드 큐 **Fire01_Cue**를 더블클릭하자. 아래와 같은 모습의 **사운드 큐 에디터**가 열릴 것이다.

먼저, **사운드 큐 에디터**에 대해서 알아보자.

사운드 큐 에디터의 레이아웃은 단순하다. 가장 위에는 메뉴바와 툴바가 있다. 중간에는 오디오 노드 그래프를 작성할 수 있는 격자판 모양의 오디오 노드 그래프 탭 있다. 왼쪽에는 디폴트 속성값을 보여주는 디테일 탭이 있다. 오른쪽에는 노드를 나열해주는 팔레트 탭이 있다.

사운드 큐 에디터의 주요 기능은 오디오 노드 그래프를 작성하는 것이다. 가장 오른쪽에 **출력** 노드가 있다. 노드 그래프를 작성하고 그 결과의 최종 사운드를 이 출력 노드에 연결해주면 된다.

출력 노드를 클릭하면 왼쪽의 디테일 탭에서 여러 속성을 지정할 수 있다. 이 **출력** 노드의 디테일 탭에서 속성값을 지정해두면 이 **사운드 큐**가 레벨에 배치될 때에 디폴트 속성값으로 사용된다. 한편, 만약 레벨 에디터에서 레벨에 배치된 인스턴스에 대한 디테일 탭에서의 속성값을 수정한다면 그 수정된 내용이 여기의 **출력** 노드에서의 속성값에 우선하여 사용된다.

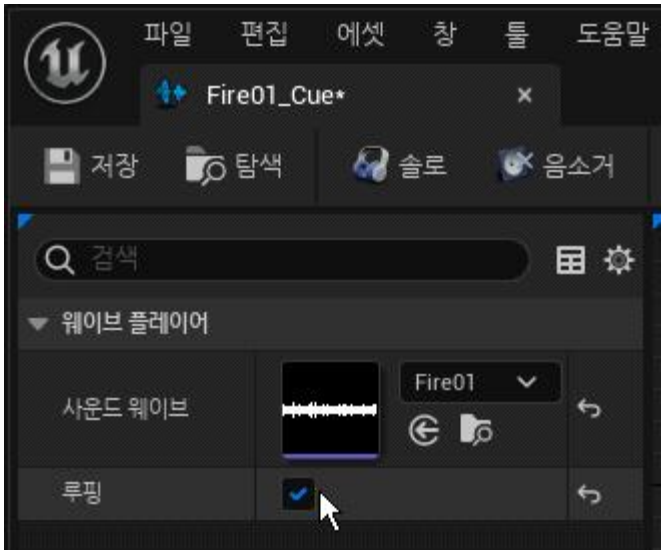
지금까지, **사운드 큐 에디터**에 대해서 알아보았다.

10. 예제를 계속하자. 우리는 **Fire01 사운드 웨이브** 애셋으로부터 생성하였으므로 **Fire01**의 **웨이브 플레이어(Wave Player)** 노드가 디폴트로 배치되었다. 그리고 최종 **출력** 노드에 연결시켜주었다. **사운드 큐**를 바로 사용할 수 있도록 노드 그래프를 만들어준 것이다.

툴바의 **큐 재생** 버튼을 클릭해서 재생해보자. 한 번만 재생될 것이다. **Wave Player** 노드는 자신이 재생할 **사운드 웨이브** 에셋의 **Looping** 속성에 체크된 경우에도 이를 무시하고 별도의 **Looping** 속성을 사용한다.

<참고> 사운드 큐 에디터에 대해서는 다음의 문서를 참조하자.
https://docs.unrealengine.com/sound-cue-editor-in-unreal-engine/

11. 오디오 노드 그래프 탭에서 **웨이브 플레이어(Wave Player)** 노드를 클릭하자. 왼쪽 디테일 탭에 속성이 표시될 것이다. 이 중에서 **루핑(Looping)**에 체크하자. 툴바에서 **저장** 버튼을 눌러 저장하고 **큐 재생** 버튼을 눌러 재생해보자. 계속 반복해서 재생될 것이다.



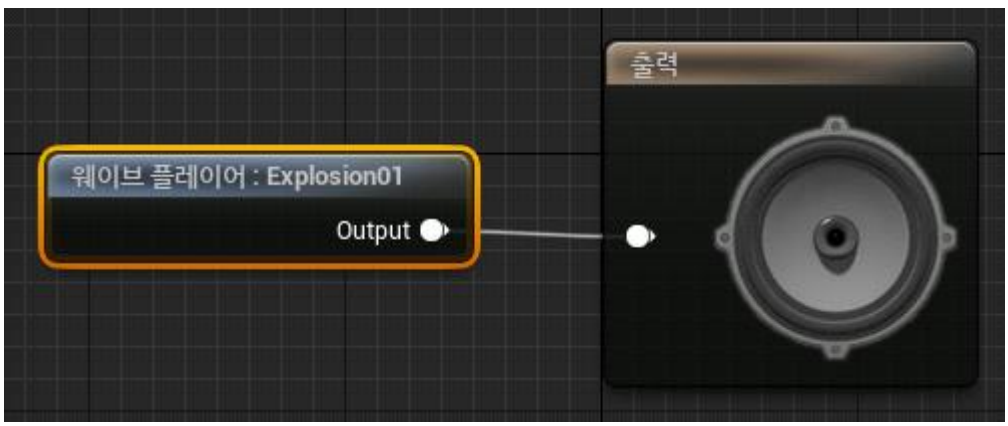
이제, **콘텐츠 브라우저**에서 **Fire01_Cue**를 드래그하여 레벨에 배치하자. 그리고 플레이해보자. 계속 반복해서 재생될 것이다.
테스트를 해본 후에는, 레벨에 추가했던 **Fire01_Cue** 인스턴스를 제거해서 다시 원래 상태로 되돌려두자.

12. 이제 또다른 사운드 큐를 만들어보자. 이번에는 다른 방식으로 만들어보자.
콘텐츠 브라우저에서 툴바의 **+추가**를 클릭하고 **사운드** » **사운드 큐**를 선택하자. 생성된 사운드 큐의 이름을 **Explosion_Cue**로 수정하자.

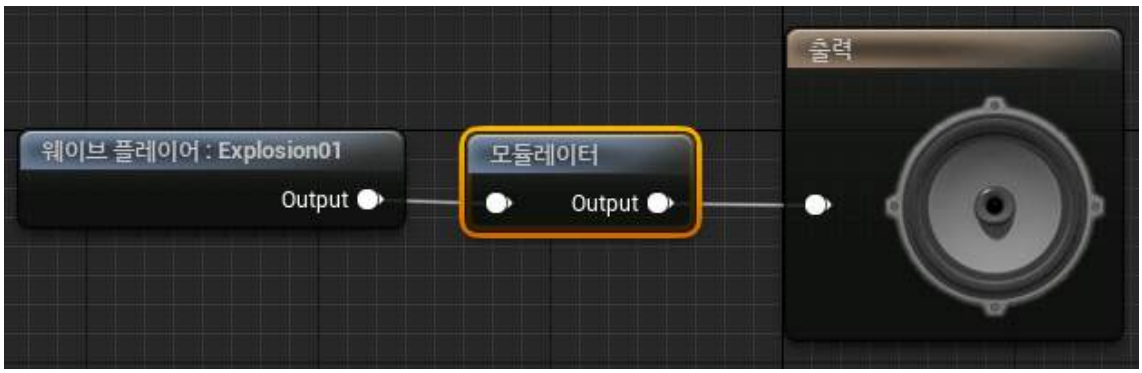
13. **Explosion_Cue**를 더블클릭하여 **사운드 큐 에디터**를 열자. 오디오 노드 그래프 탭에는 **출력** 노드만 있을 것이다.

먼저, **콘텐츠 브라우저**에서 **Explosion01** 사운드 웨이브를 드래그해서 오디오 노드 그래프 탭에 배치하자. **Explosion01**의 **웨이브 플레이어** 노드가 배치될 것이다.

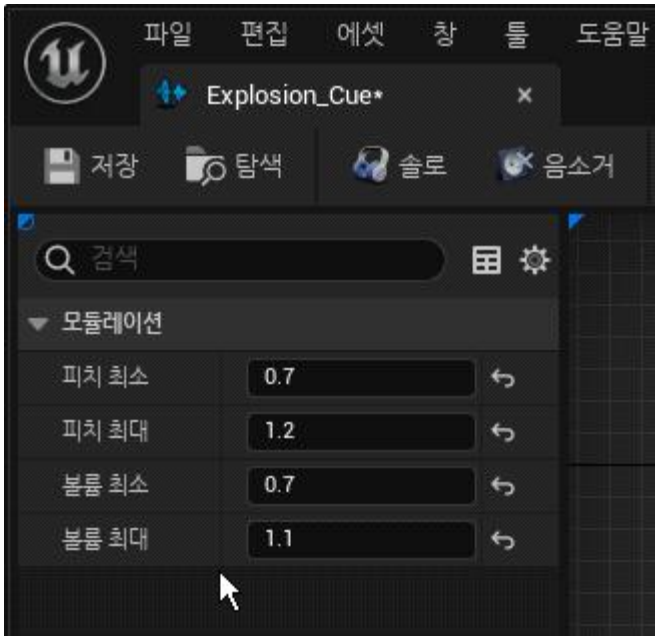
그다음, **웨이브 플레이어** 노드의 출력핀을 **출력** 노드의 입력핀에 연결하자. 그리고, 툴바의 **큐 재생** 버튼을 클릭하여 잘 재생되는지 확인해보자.



14. 게임 플레이 시에 동일한 사운드를 재생할 때에도 재생시마다 조금씩 다르게 재생한다면 더욱 자연스러워질 것이다.
이제, **모듈레이터(Modulator)** 노드를 사용해보자. 이 노드는 사운드가 시작될 때에 피치와 볼륨을 랜덤하게 수정할 수 있는 기능을 제공한다.
먼저, 오디오 노드 그래프 탭에서 **웨이브 플레이어: Explosion01** 노드의 출력핀 위에서 **Alt+좌클릭**하여 기존의 연결을 끊자.
그다음, 오른쪽의 팔레트에서 **모듈레이터(Modulator)** 노드를 드래그해서 배치하자. 우클릭하고 액션 선택 창에서 선택해도 된다. 이 노드는 입력 핀 중에서 랜덤하게 결정해서 출력해주는 노드이다.
그다음, 아래와 같이 연결해보자.



15. **모듈레이터** 노드의 디테일 탭에 **피치(Pitch)**의 **최소(Min)**,**최대(Max)**가 0.95, 1.05일 것이다. 또한 **볼륨(Volume)**의 **최소(Min)**,**최대(Max)**가 0.95, 1.05일 것이다. 피치와 볼륨이 이 범위 내에서 랜덤하게 지정되는 것을 의미한다.
피치는 소리의 높낮이를 의미한다. 피치 값은 [0.4, 2]의 범위에서 지정해주면 된다. 테스트를 위해서 볼륨을 1,1로 지정한 후에 피치값을 [0.4, 2]에서 바꾸어가며 재생해보자. 소리의 높낮이가 바뀌는 것을 확인할 수 있다.
볼륨은 소리의 강도이다. 0 이상의 값을 지정하면 된다.
우리는 피치의 범위를 [0.7, 1.2]로 볼륨의 범위를 [0.7, 1.1]로 지정하자.



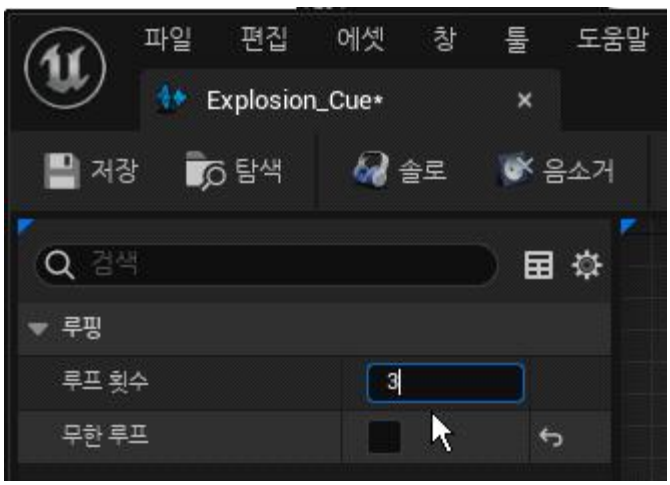
16. 게임 플레이 시에 폭발이 일어날 때에 하나의 폭발음만 사용하기보다는 약간 다른 폭발음을 랜덤하게 선택해서 재생해준다면 더 자연스러울 것이다.
먼저, **콘텐츠 브라우저**에서 **Explosion02** 사운드 웨이브를 드래그해서 오디오 노드 그래프 탭에 드롭하자. **웨이브 플레이어: Explosion02** 노드가 배치될 것이다.
그다음, 팔레트에서 **랜덤(Random)**을 드래그해서 **랜덤(Random)** 노드를 배치하자. 이 노드는 입력 핀 중에서 랜덤하게 결정해서 출력해주는 노드이다.
그다음, 두 **웨이브 플레이어** 노드의 출력을 **랜덤** 노드에 연결하고, **랜덤** 노드의 출력을 **모듈레이터** 노드로 연결하자. 이제 아래 그림처럼 될 것이다.

툴바의 **큐 재생** 버튼을 여러 번 누르면서 테스트해보자.
사운드 큐를 재생하는 것을 반복하면 두 사운드가 한 번씩 번갈아 가며 재생되는 것을 알 수 있다.

17. **랜덤** 노드를 선택하고 왼쪽의 디테일 탭을 살펴보자.
대체 없이 랜덤화(Randomize Without Replacement) 속성을 찾아보자. 체크박스에 체크되어 있을 것이다. 여기서 **대체**(Replacement)라는 용어의 의미는 이미 선택된 것을 다시 선택 가능 그룹에 넣고 다음번 선택을 하겠다는 것이다. 따라서 **대체 없이**(Without Replacement)는 이미 선택된 것은 다음번 선택에서 배제하겠다라는 의미이다. 따라서 각각이 한 번씩 번갈아 가며 재생된다.
우리는 이번에는 체크박스를 해제하자. 그리고 다시 테스트해보자. 이번에는 번갈아가며 재생하는 것이 아니라 매번 임의로 선택하여 재생할 것이다.

18. 이제 반복 재생하는 노드를 추가해보자.
먼저, 팔레트에서 **루핑(Looping)**을 드래그해서 **루핑(Looping)** 노드를 추가하자.
그다음, **루핑** 노드를 **모듈레이터** 노드와 최종 **출력** 노드 사이에 배치하자. 다음과 같이 보일 것이다.

19. 이제, **루핑** 노드를 선택하고 디테일 탭을 살펴보자.
디폴트로 **무한 루프(Loop Indefinitely)**에 체크가 되어 있다. 체크되어 있으며 무한히 계속 반복한다의 의미이다. 이 체크를 해제하자. 그리고 바로 위의 속성인 **루프 횟수(Loop Count)**를 1에서 3으로 수정하자. 이제 세 번만 반복할 것이다.



이제, 테스트해보자.
콘텐츠 브라우저에서 **Explosion_Cue**를 드래그해서 레벨에 배치하자. 플레이해보자. 세 번 반복된 후에 멈출 것이다.

이 절에서는 사운드 웨이브와 사운드 큐와 앰비언트 사운드 액터에 대해서 학습하였다.

3. 입체 사운드와 사운드 감쇠 설정

이 절에서 입체 사운드(3D 사운드)에 대해서 알아보자. 입체 사운드란 감쇠와 패닝의 기능을 가진 사운드를 의미한다. 감쇠(attenuation)는 음원에서 가까울수록 사운드 볼륨이 높아지고 멀어질수록 낮아지는 것을 의미한다. 패닝(panning)은 음원이 플레이어의 왼쪽인지 오른쪽인지에 따라서 왼쪽 스피커와 오른쪽 스피커의 볼륨이 달라지는 것을 의미한다.
엔진은 입체 사운드 기능을 지원한다. 입체 사운드 기능을 사용하려면 반드시 사운드 감쇠 설정을 해 주어야 한다. 이 절에서는 사운드 감쇠(attenuation, 어테뉴에이션) 설정에 대해서 학습한다. 감쇠 설정에는 감쇠에 대한 설정과 더불어 패닝에 대한 설정도 함께 한다.

이제부터 예제를 통해서 학습해보자

---

1. 새 프로젝트 Pattenuation를 생성하자. 먼저, 언리얼 엔진을 실행하고 **언리얼 프로젝트 브라우저**에서 왼쪽의 **게임** 탭을 클릭하자. 오른쪽의 템플릿 목록에서 **기본** 템플릿을 선택하자. **프로젝트 이름**은 Pattenuation로 입력하고 **생성** 버튼을 클릭하자. 프로젝트가 생성되고 언리얼 에디터 창이 뜰 것이다.
창이 뜨면, 메뉴바에서 **파일** » **새 레벨**을 선택하고 **Basic**을 선택하여 기본 템플릿 레벨을 생성하자. 그리고, 레벨 에디터 툴바의 저장 버튼을 클릭하여 현재의 레벨을 MyMap으로 저장하자.
그리고, **프로젝트 세팅** 창에서 **맵&모드** 탭에서의 **에디터 시작 맵** 속성값을 MyMap으로 수정하자.
그리고, 툴바의 **액터 배치** 아이콘을 클릭하고 **기본** 아래의 **플레이어 스타트** 액터를 드래그하여 레벨에 배치하자. 위치를 (0,0,92)로 지정하자.
그리고, **아웃라이너**에서 **Floor**를 선택하고, **디테일** 탭에서 **스케일**을 (8,8,8)에서 (1,1,1)로 수정하자.

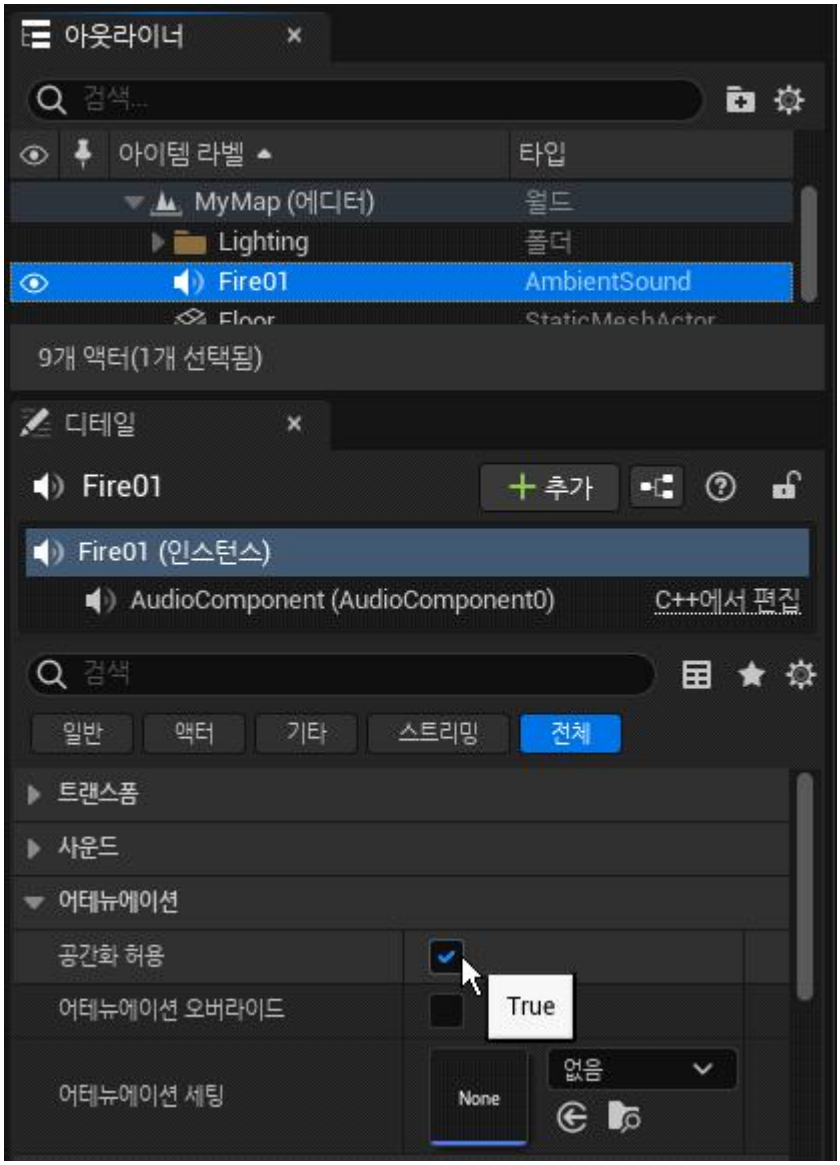
2. 이번 프로젝트에서 필요한 외부 애셋을 준비하자.
필요한 애셋은 **시작용 콘텐츠**에 포함되어 있다. **시작용 콘텐츠**가 포함되어 있는 다른 프로젝트를 찾아보거나 또는 **시작용 콘텐츠**가 포함되도록 새 임시 프로젝트를 만들자. 임시 프로젝트에서 **Content** 폴더 아래에서 다음의 파일들을 찾아보자. 다음의 파일들을 우리의 프로젝트로 복사하여 가져오자. 폴더 구조가 동일하게 되도록 하자. 가져온 후에 임시 프로젝트는 종료하고 삭제하면 된다.

```
StarterContent/Audio/Fire01.uasset
```

3. 첫 번째 테스트를 해보자.
콘텐츠 브라우저에서 **콘텐츠** » **StarterContent** » **Audio**에 있는 **사운드 웨이브** 애셋인 **Fire01**을 드래그해서 레벨에 배치해보자. 이 애셋은 단일 채널 모노(Mono) 사운드 웨이브 애셋이다. 배치한 후에 플레이해 보자. 불타는 소리가 반복 재생될 것이다. 사운드의 배치 위치와 무관하게 사운드가 모든 위치에서 동일하게 들리도록 플레이된다.

4. 지금부터는 사운드 액터가 배치된 위치에 따라서 다르게 들리도록 해보자.
먼저, 배치된 **Fire01** 사운드 액터의 위치를 (100,0,0)로 하자.
그다음, 디테일 탭의 **어테뉴에이션** 영역으로 가자. **공간화 허용**(**Allow Spatialization**) 속성이 있을 것이다. 그리고 디폴트로 체크되어 있을 것이다. 레벨에 배치된 액터의 모든 **AudioComponent** 속성에는 이 **공간화 허용**(**Allow Spatialization**) 속성이 있다. 이 속성이 **true**이면 이 사운드 에셋에 감쇠 설정을 적용할 수 있도록 허용하겠다는 의미이다. 만약 **false**라면 이 사운드 에셋은 입체감 없이 어디에서나

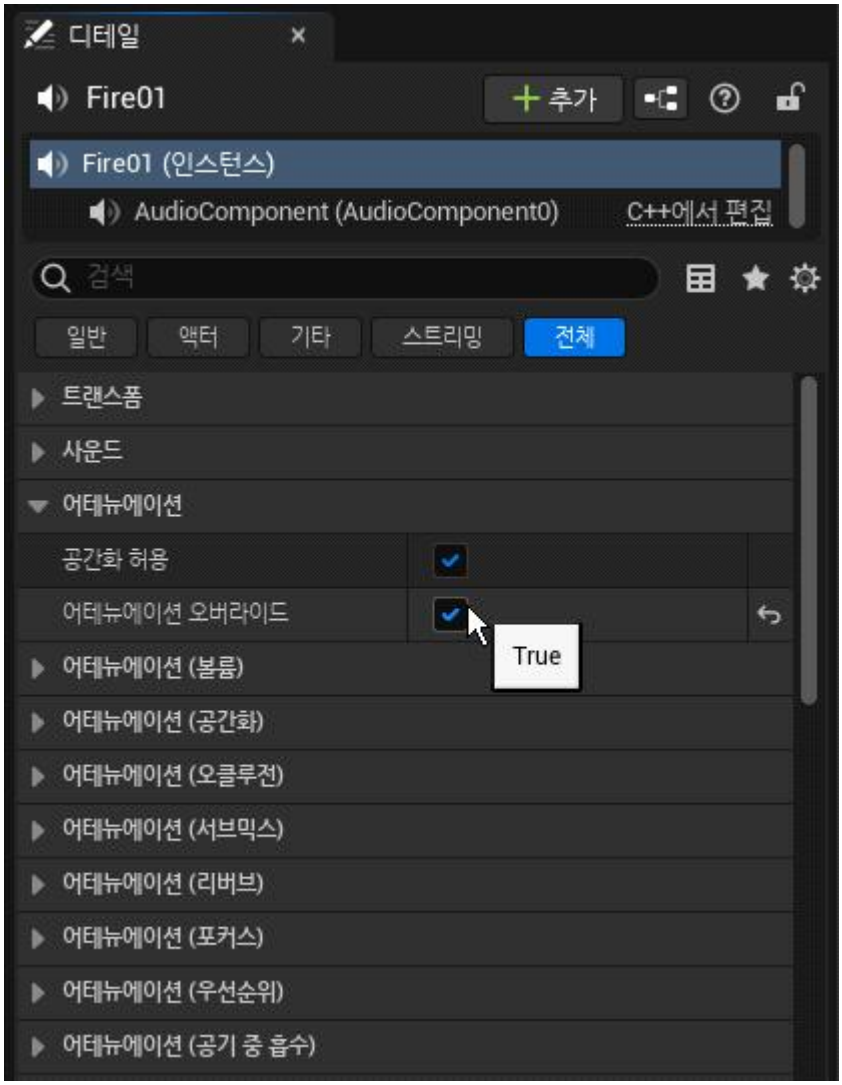
동일하게 들리도록 재생되는 단순한 재생 방식으로만 활용된다. 우리는 특별한 경우가 아니면 디폴트인 **true**로 그대로 두자.



5. 그다음, **공간화 허용** 속성 바로 아래에 **어테뉴에이션 오버라이드**(**Override Attenuation**) 속성이 있다. 이 속성은 디폴트로 체크되어 있지 않아서 **false** 값을 가진다. **어테뉴에이션 오버라이드** 속성이 **false**인 경우에는 바로 아래의 **어테뉴에이션 세팅**(Attenuation Settings) 속성값에 지정된 **사운드 감쇠** 애셋의 설정값을 사용한다. **사운드 감쇠** 애셋은 별도로 제작해야 한다. **어테뉴에이션 세팅** 속성값은 디폴트로 **없음**으로 되어 있다. 따라서 입체 사운드로 동작하지 않을 것이다.
한편, **사운드 웨이브** 에셋에도 이 **어테뉴에이션 세팅** 속성이 있다. 따라서 **사운드 감쇠** 애셋이 별도로 준비되어 있다면 사운드 웨이브 에셋에 미리 지정해두어도 좋다.

6. 우리는 아직 **사운드 감쇠** 애셋은 별도로 제작하지 않았으므로 **어테뉴에이션 오버라이드** 속성을 **true**로 체크하자. **true**가 되도록 체크하면 현재의 디테일 탭에서 상세한 감쇠 설정을 직접하겠다는 의미이다. 체크하면 바로 그 아래에 매우 많은 감쇠 속성이 추가로 나타난다.

모든 감쇠 속성들이 **어테뉴에이션(볼륨)**, **어테뉴에이션(공간화)**, **어테뉴에이션(오클루전)**, **어테뉴에이션(서브믹스)**, **어테뉴에이션(리버브)**, **어테뉴에이션(포커스)**, **어테뉴에이션(우선순위)**, **어테뉴에이션(공기 중 흡수)**의 8개의 영역으로 구분하여 나열된다.



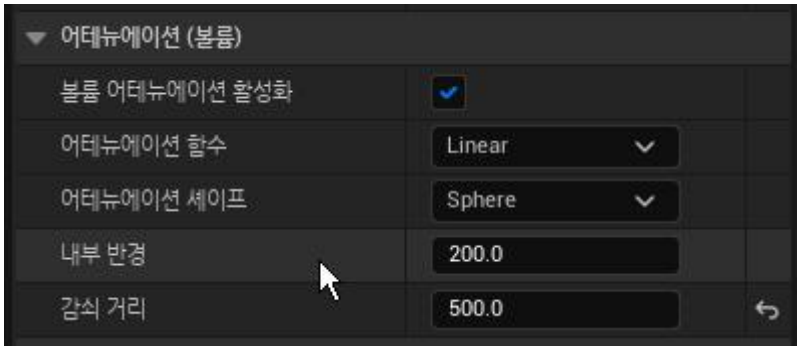
일단 상세한 속성들은 그대로 두자.

7. 플레이해보자. 사운드가 재생될 것이다. 뒤로 멀리 가보자. 소리가 점점 약해지다가 안들리게 될 것이다. 다시 원래 위치에서 이번에는 왼쪽 방향으로 바닥 메시를 벗어날 만큼 멀리 가보자. 소리가 오른쪽 귀에서만 잘 들릴 것이다. 그리고 그 위치의 제자리에서 180도 돌아보자. 이번에는 왼쪽 귀에서만 잘 들릴 것이다. 입체 음향으로 재생되고 있음을 확인할 수 있다. 좌우 소리의 구분을 위해서는 스피커보다는 이어폰을 사용하는 것이 좋다.

8. 이제 감쇠 속성을 수정해보자.

어테뉴에이션(볼륨) 영역에 **내부 반경**(Radius)과 **감쇠 거리**(Fall off distance)가 있다. 이 속성은 배치된 사운드 액터와 플레이어와의 거리에 따라서 소리를 페이드 인/아웃 하는 기능을 제어한다. 플레이어와의 거리가 내부 반경 이내인 경우에는 감쇠되지 않은 풀 볼륨으로 재생된다. 거리가 내부 반경을 넘어서면 그때부터 볼륨이 약해지기 시작한다. 감쇠 거리에 도달하거나 넘어서게 되면 볼륨이 0으로

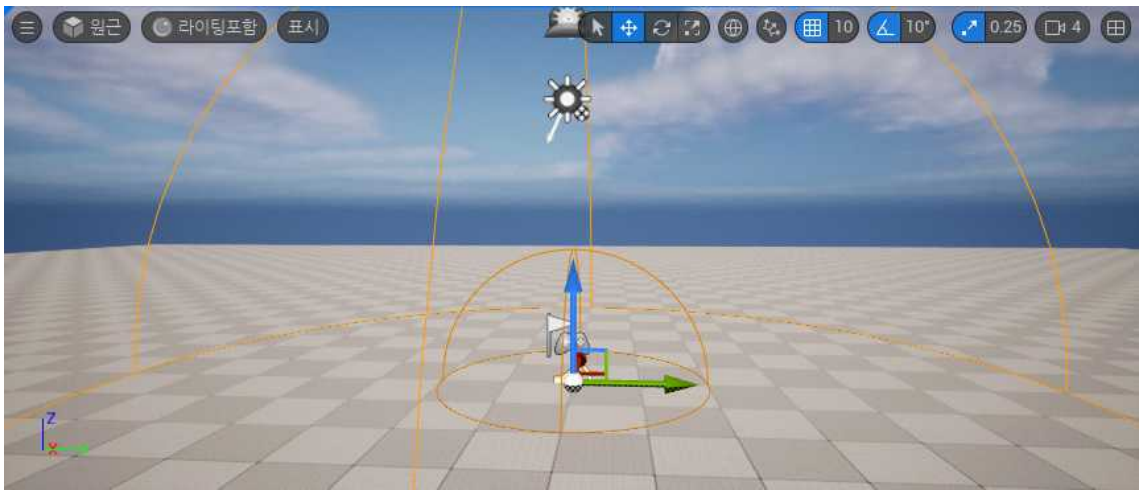
된다.
디폴트로 내부 반경과 감쇠 거리가 400, 3600으로 되어 있다. 우리는 이것을 200, 500으로 수정하자.



내부 반경에서 감쇠 거리 사이에서는 사운드 볼륨은 거리에 따라 감쇠가 일어난다. 거리와 볼륨의 감쇠 관계는 **어테뉴에이션 함수** 속성값에 의해서 결정된다. 디폴트로 **Linear**로 되어 있다.
또한 감쇠 모양이 구체가 아닌 경우도 지원한다. 다른 모양으로 바꾸려면 **어테뉴에이션 셰이프** 속성을 수정하면 된다. **어테뉴에이션 셰이프**가 바뀌면 그에 따라 새로운 속성이 추가로 나타난다.

<참고> 감쇠와 관련된 속성의 자세한 내용은 아래의 문서를 참조하자.
https://docs.unrealengine.com/sound-attenuation-in-unreal-engine/

9. 속성값 **어테뉴에이션 오버라이드(Override Attenuation)**을 **true**로 수정하면 레벨의 뷰포트에서 **내부 반경**과 **감쇠 거리**를 구체 모양으로 보여준다. 안쪽의 작은 구체가 내부 반경에 해당하고 외부의 큰 구체가 감쇠 거리에 해당한다.



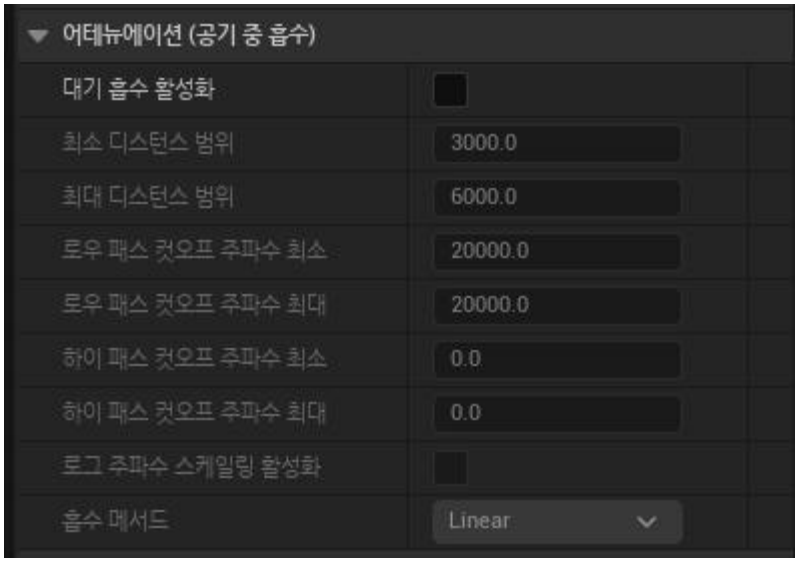
10. 플레이해보자. 플레이어를 움직여보자.
현재 **Fire01** 사운드 액터는 (100,0,0)에 있을 것이다. 그리고 플레이어의 시작 위치는 (0,0,92)일 것이다. 바닥 메시는 1000x1000의 크기이다. 따라서 바닥 메시를 벗어날 정도로 멀리 가면 소리가 들리지 않게 될 것이다.

11. 이제 그다음의 속성 영역인 **어테뉴에이션(공간화)**를 살펴보자. 이 영역에서는 좌우 스피커의 볼륨의 차이를 발생시키는 패닝 기능에 대한 설정을 한다.
속성값 **어테뉴에이션 오버라이드(Override Attenuation)**이 **true**이면 **감쇠(공간화)** 속성 영역이 나타나며 그 첫 번째 속성인 **공간화 활성화** 속성을 디폴트로 **true**로 지정되어 있다. 이 속성이 **false**인 경우에는

패닝이 작동하지 않는다. 즉 거리에 따른 감쇠는 되지만 방향에 따른 좌우 스피커의 차이는 생기지 않는다. 따라서 음원으로부터의 거리감만 파악할 수 있고 음원 위치에 대한 방향감은 파악할 수 없다. 따라서 방향감이 있는 스테레오 사운드를 원한다면 **true**로 체크해야 한다. 우리는 **공간화 활성화** 속성을 항상 **true**로 체크되어 있도록 하자.

다음으로, **공간화 메서드** 속성에서는 패닝 방법을 선택한다. 속성값 **Panning**은 엔진의 표준 패닝 방법이다. 다른 속성값인 **Binaural**을 선택하고 패닝이 구현된 플러그인을 사용할 수도 있다.
다음으로, **공간화 되지 않은 반경** 속성이 있다. 이 속성값에 지정된 거리보다 음원에 가까이 있는 경우에는 패닝 기능을 작동하지 않게 한다. 라디오로부터 멀리 있을 때에는 패닝 기능으로 라디오의 위치를 짐작해서 찾아갈 수 있도록 하는 것이 좋다. 그러나 라디오에 충분히 가까이에 있으면서 청취하고자 하는 경우에는 라디오 소리를 제대로 잘 들을 수 있도록 패닝 기능을 꺼주는 것이 좋다. 디폴트로는 0으로 설정되어 있다. 0인 경우에는 이 기능이 작동하지 않고 항상 패닝이 적용된다.
다음으로, **3D 스테레오 스프레드**는 패닝의 적용에 있어서 가정하는 가상의 왼쪽 스피커와 오른쪽 스피커의 거리를 지정한다.

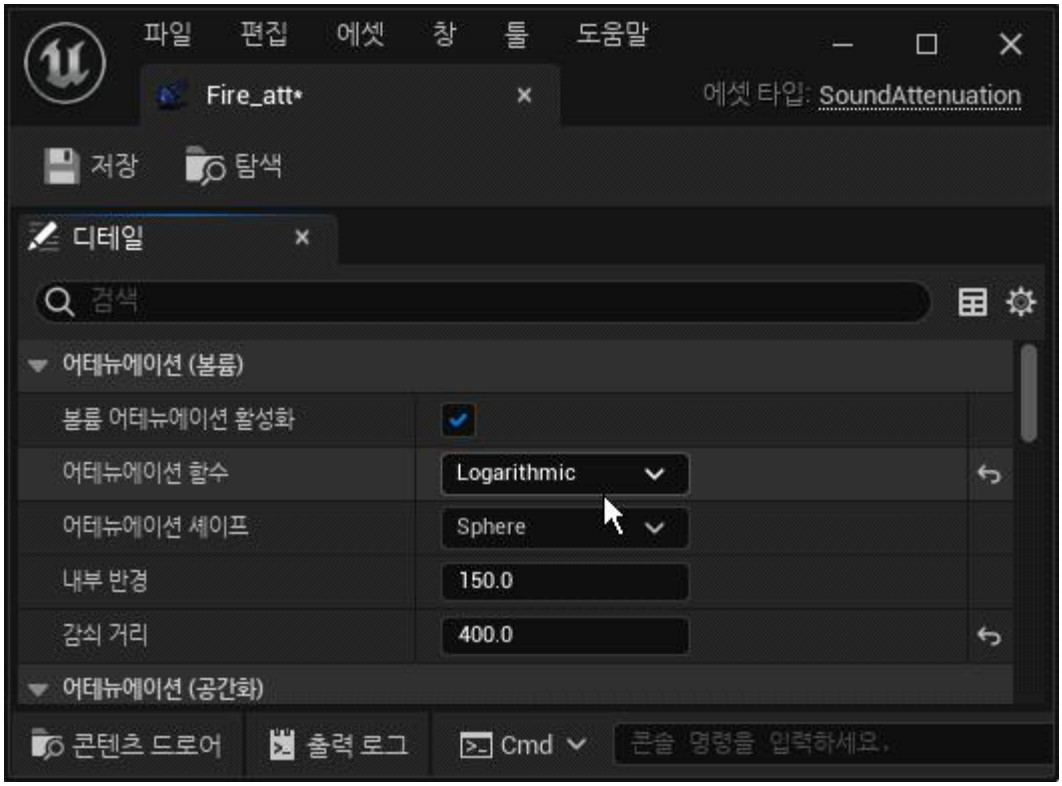
12. 그 외의 속성 영역에 대해서도 살펴보자.
어테뉴에이션((공기 중 흡수) 영역의 속성은 소리가 대기에 흡수되는 현상을 표현할 수 있도록 한다. 소리가 멀리 전달될 때에 소리가 대기에 흡수되어 볼륨이 줄어드는데 모든 주파수의 소리가 동일하게 줄어드는 것이 아니라 높은 음역의 소리가 낮은 음역의 소리보다 더 많이 흡수되어 줄어든다. 따라서 낮은 소리가 더 멀리 전달되는 현상이 나타난다. 낮은 음역의 소리만 통과시키도록 하는 로우 패스 필터를 구현하고 있다.
대기 흡수 활성화 속성값이 **true**로 지정되어 있다면 **최소 디스턴스 범위, 최대 디스턴스 범위** 속성값으로 명시된 거리 구간에서 로우 패스 필터를 적용한다. 이 거리 구간은 **감쇠 거리**(Fall off distance)보다 약간 이전에서 시작해서 더 먼 거리에서 끝나도록 설정하면 자연스럽게 된다. 또한, 적용할 로우 패스 주파수의 범위는 추가적인 속성값으로 명시한다.



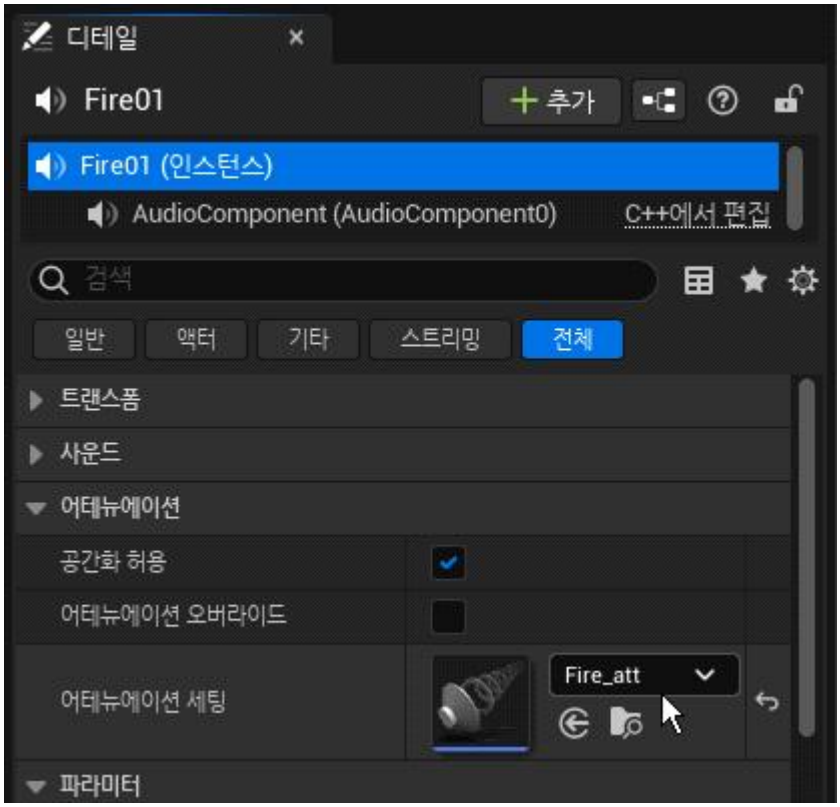
지금까지 **어테뉴에이션 오버라이드(Override Attenuation)**를 **true**로 하고 다양한 **어테뉴에이션** 속성값을 직접 설정하는 방법에 대해서 알아보았다.

13. 레벨에 배치된 모든 사운드 액터마다 감쇠 설정을 일일이 하는 것은 불편할 것이다. 공통적으로 자주 사용되는 감쇠 설정을 템플릿으로 만들어두고 이를 재사용하면 편리할 것이다. 이제부터 감쇠 설정을 애셋으로 만들고 이를 사용해보자.
먼저, **콘텐츠 브라우저**에서 **+추가**를 클릭하고 **사운드** » **사운드 어테뉴에이션**을 선택하자. 생성된 **사운드 어테뉴에이션** 애셋의 이름을 **Fire_att**로 수정하자.

14. 이제 **Fire_att**를 더블클릭하여 **사운드 어테뉴에이션** 애셋의 에디터를 열자. 애셋의 디테일 탭만 보여주는 창이 뜰 것이다.
먼저, **어테뉴에이션(볼륨)** 영역에서 **어테뉴에이션 함수**를 **Linear**에서 실제의 감쇠와 비슷한 **Logarithmic**으로 수정하자.
그다음, **내부 반경**과 **감쇠 거리**를 400, 3600에서 150,400으로 수정하자.
나머지 속성은 생략하고 이제 저장하자.



15. 이제 **아웃라이너**에서 배치된 **Fire01**을 선택하자. 인스턴스의 **디테일** 탭에서 **어테뉴에이션** 영역의 **어테뉴에이션 오버라이드(Override Attenuation)**의 체크를 해제하자. 디테일 탭에서 감쇠 속성 지정과 관련된 많은 영역들이 사라질 것이다.
그다음, 바로 아래의 **어테뉴에이션 세팅**에서 **없음**을 클릭하고 우리가 만든 **Fire_att**를 선택하여 지정하자.



뷰포트에서 범위를 나타내는 구체가 바뀌는 것을 확인할 수 있다.

16. 이 절에서는 **사운드 웨이브** 애셋만 다루었고 **사운드 큐** 애셋을 사용하지는 않았다.
그러나 **사운드 큐** 애셋에 대해서도 위에서와 동일한 방식으로 감쇠 속성을 지정하면 된다.

이 절에서는 입체 사운드와 사운드 감쇠 설정에 대해서 학습하였다.

4. 블루프린트에서의 사운드 큐 활용

이 절에서 블루프린트에서의 사운드 큐 활용에 대해서 학습한다.

지금까지는 사운드 웨이브 애셋이나 사운드 큐 에셋을 앰비언트 사운드 액터 형태로 레벨에 배치하는 것에 대해서 알아보았다. 지금부터는 플레이 시에 동적으로 사운드 애셋을 다루는 방법에 대해서 알아보자.

이제부터 예제를 통해서 학습해보자

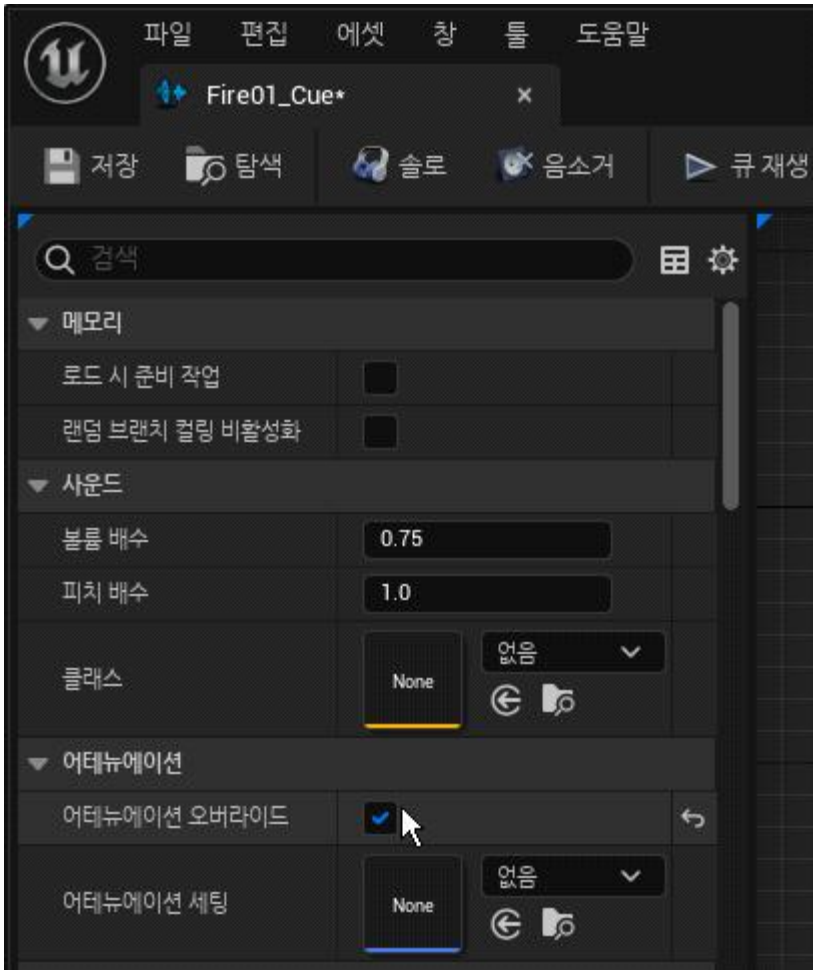
---

1. 새 프로젝트 Psoundbp를 생성하자. 먼저, 언리얼 엔진을 실행하고 **언리얼 프로젝트 브라우저**에서 왼쪽의 **게임** 탭을 클릭하자. 오른쪽의 템플릿 목록에서 **기본** 템플릿을 선택하자. **프로젝트 이름**은 Psoundbp로 입력하고 **생성** 버튼을 클릭하자. 프로젝트가 생성되고 언리얼 에디터 창이 뜰 것이다. 창이 뜨면, 메뉴바에서 **파일** » **새 레벨**을 선택하고 **Basic**을 선택하여 기본 템플릿 레벨을 생성하자. 그리고, 레벨 에디터 툴바의 저장 버튼을 클릭하여 현재의 레벨을 MyMap으로 저장하자.
그리고, **프로젝트 세팅** 창에서 **맵&모드** 탭에서의 **에디터 시작 맵** 속성값을 MyMap으로 수정하자.
그리고, 툴바의 **액터 배치** 아이콘을 클릭하고 **기본** 아래의 **플레이어 스타트** 액터를 드래그하여 레벨에 배치하자. 위치를 (0,0,92)로 지정하자.
그리고, **아웃라이너**에서 **Floor**를 선택하고, **디테일** 탭에서 **스케일**을 (8,8,8)에서 (1,1,1)로 수정하자.

2. 이번 프로젝트에서 필요한 외부 애셋을 준비하자.
필요한 애셋은 **시작용 콘텐츠**에 포함되어 있다. **시작용 콘텐츠**가 포함되어 있는 다른 프로젝트를 찾아보거나 또는 **시작용 콘텐츠**가 포함되도록 새 임시 프로젝트를 만들자. 임시 프로젝트에서 **Content** 폴더 아래에서 다음의 파일들을 찾아보자. 다음의 파일들을 우리의 프로젝트로 복사하여 가져오자. 폴더 구조가 동일하게 되도록 하자. 가져온 후에 임시 프로젝트는 종료하고 삭제하면 된다.

```
StarterContent/Audio/Explosion01.uasset
StarterContent/Audio/Fire01.uasset
```

3. **콘텐츠 브라우저**에서 **StarterContent** » **Audio** 아래의 **Fire01** 사운드 웨이브 애셋을 찾자. 애셋 위에서 우클릭하고 **큐 생성**을 선택하여 사운드 큐 애셋을 생성하자. 이름은 **Fire01_Cue**로 그대로 두자.
그다음, **Fire01_Cue**를 더블클릭하여 사운드 큐 에디터를 열자. 왼쪽 속성 탭에서 **어테뉴에이션** 영역에 있는 **어테뉴에이션 오버라이드(Override Attenuation)** 속성의 체크박스에 체그하여 **true**로 지정하자. 다른 감쇠 속성값들은 디폴트로 그대로 두자.



4. **콘텐츠 브라우저**에서 **Explosion01** 사운드 웨이브 애셋을 찾고, 동일한 방법으로 사운드 큐 애셋 **Explosion01_Cue**를 생성하자. 그리고 **Explosion01_Cue**를 더블클릭하여 사운드 큐 에디터를 열고, 동일한 방법으로 **어테뉴에이션 오버라이드**(**Override Attenuation**) 속성을 **true**로 지정하자.

5. 이제부터, 레벨 에디터에서 사운드를 플레이해보자.
레벨 에디터의 툴바에서 **블루프린트** » **레벨 블루프린트 열기**를 실행하여 레벨 블루프린트를 열자. 이벤트 그래프의 격자판에서 우클릭하고 액션선택 창에서 '키 s'를 검색하여 **S 키 입력** 이벤트 노드를 배치하자. 노드의 **Pressed** 실행핀을 당기고 **Play Sound at Location** 노드를 검색하여 배치하자.
그다음, 노드의 **Sound** 입력핀을 클릭하면 선택 가능한 모든 사운드 웨이브 애셋과 사운드 큐 애셋이 나열된다. 목록에서 **Fire01_Cue**를 찾아 배치하자. **Location** 입력핀에는 (300,-300,0)을 지정하자.

6. 플레이해보자. S 키를 눌러보자. 사운드가 재생될 것이다. 입체 사운드이므로 소리가 대각선 왼쪽 앞에서 들릴 것이다.

7. 사운드 재생 함수로 입체 사운드를 재생하는 **Play Sound at Location** 노드가 있고 비입체 사운드를 재생하는 **Play Sound 2D** 노드가 있다. 이들 노드의 입력핀을 살펴보자. 노드 하단의 펼침 아이콘을 클릭하면 많은 입력핀이 나타난다.

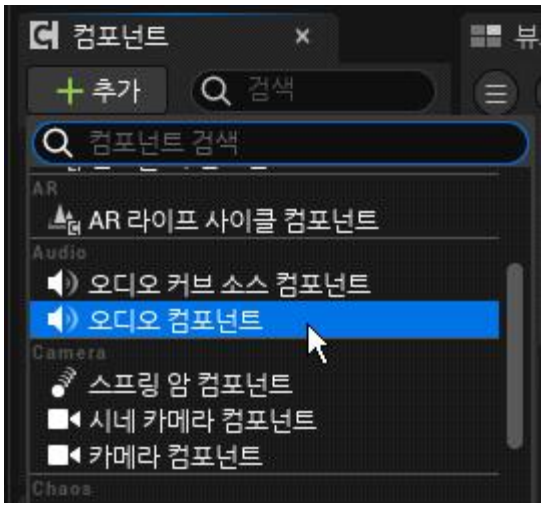
공통적인 입력핀들을 살펴보자. 먼저, 재생할 사운드 애셋 선택을 위한 **Sound** 입력핀이 있다. 그다음, 볼륨 조절을 위한 **Volume Multiplier**가 있고 피치 조절을 위한 **Pitch Multiplier**가 있다. 그다음, 사운드에서 재생을 시작할 위치를 지정할 수 있는 **Start Time** 입력핀이 있다. 그다음, **사운드 동시실행** 애셋을 지정할 수 있는 **Concurrency Settings** 입력핀이 있다. 그리고 오너 액터를 지정하는 **Owing Actor** 입력핀이 있다. 이것은 오너별로 동시 재생의 제한을 받게 한다.

<참고> 사운드 동시실행 (Sound Concurrency) 애셋은 사운드의 동시 재생 문제와 관련된 설정값을 모아둔 애셋이다. 설정에는 동시에 재생 가능한 사운드의 최대 개수를 명시하는 속성이 대표적이다. 또한 최대 개수에 도달했을 때 어떻게 처리할지(예전 사운드를 종료하고 새 사운드를 재생할지 또는 새 사운드를 거절할지)를 명시하는 속성도 있다.

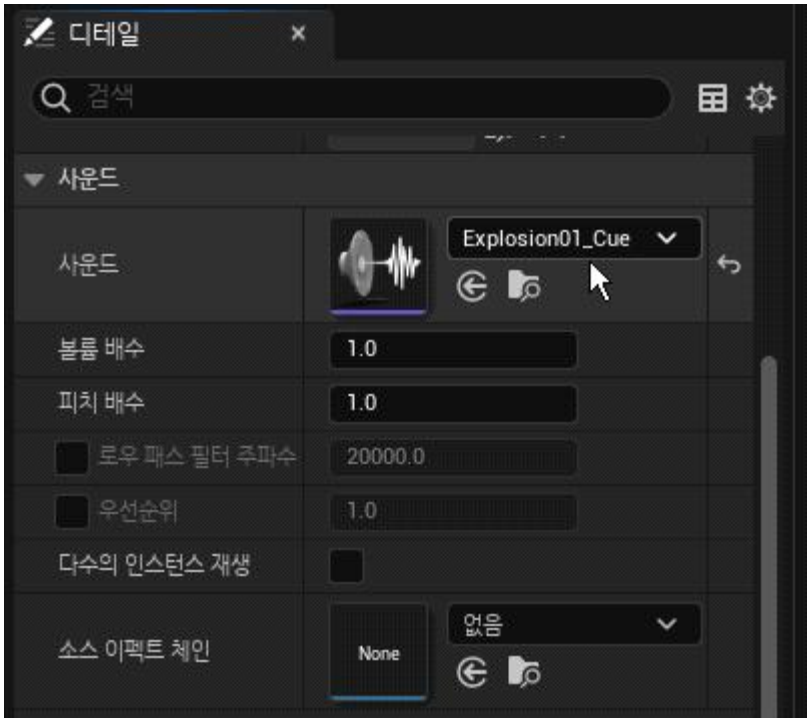
입체 사운드에만 있는 입력핀으로 **Location**, **Rotation**가 있다. 이 속성은 사운드 액터의 위치와 회전을 명시한다. 그리고, **Attenuation Setttings** 속성은 사운드 감쇠 애셋을 지정할 수 있게 한다.
비입체 사운드에만 있는 입력핀은 **Is UISound**가 있다. 사운드가 UI에서 사용되는 사운드이면 **true**를 지정하면 된다. UI 사운드인 경우에는 게임이 일시정지 되어도 계속 플레이되는 등의 약간 다른 특징이 있다.

8. 이제부터는, 액터 내에 오디오 컴포넌트를 만들어서 추가해보자.
이렇게 하면 사운드 재생이 필요한 액터를 배치할 때에 별도로 사운드를 배치할 필요가 없어지므로 편리하다.
우선, 테스트를 위해서 이전 예제에서 만들었던 스핀되는 큐브인 **BP_MyCube**를 동일한 방법으로 만들어보자.
먼저, **콘텐츠 브라우저**에서 **콘텐츠** 폴더를 선택하고 툴바에서 **+추가**를 클릭하고, 드롭다운 메뉴에서 **블루프린트 클래스**를 선택하고, **부모 클래스 선택** 창에서 **액터(Actor)**를 부모 클래스로 선택하자. 이름은 **BP_MyCube**으로 수정하자.
그다음, **BP_MyCube**을 더블클릭하여 **블루프린트 에디터**를 열자. **컴포넌트** 탭에서 **+추가**를 클릭하고 **큐브**를 선택하자. 큐브 스태틱 메시 컴포넌트가 추가될 것이다. 디폴트로 **Cube**라는 이름으로 그대로 두자.

9. 계속해서 **BP_MyCube** 블루프린트 에디터에서 작업하자. 오디오 컴포넌트를 추가하자. **컴포넌트** 탭에서 **+추가**를 클릭하고 **오디오 컴포넌트**를 선택하면 오디오 컴포넌트가 추가된다. 이름은 **Audio**로 두자.



10. 계속해서 **BP_MyCube** 블루프린트 에디터에서 작업하자. 컴포넌트 탭에서 **Audio**를 선택한 상태에서 오른쪽 디테일 탭으로 가자.
사운드 영역에 있는 **사운드(Sound**)속성값에 **없음**을 클릭하고 **Explosion01_Cue**를 선택하여 지정하자.
이제 오디오 컴포넌트가 추가되었다.

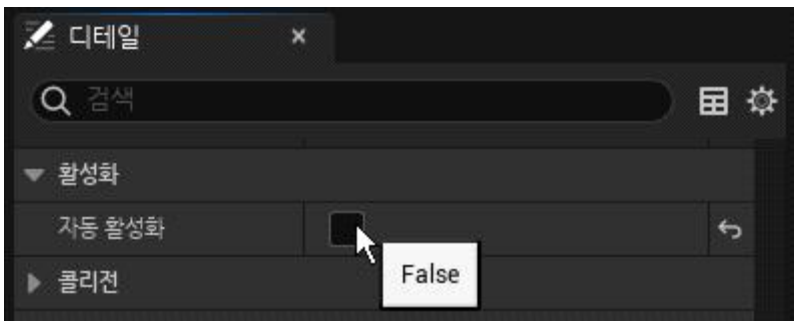


11. 이제, 레벨 에디터로 가자. **BP_MyCube**를 레벨에 배치하자. 위치는 (300,300,100)에 배치하자. 플레이해보자. 바로 폭발음이 재생될 것이다. 입체 사운드이므로 소리가 대각선 오른쪽 앞에서 들릴 것이다.

12. 레벨을 플레이하면 액터의 소리가 바로 재생되는 이유는 액터의 **자동 활성화**(**Auto Activate**) 속성이 **true**로 되어 있기 때문이다. 우리는 플레이 시에 바로 재생되도록 하지 말고 특정한 키가 눌러질 때에 재생되도록 수정해보자.

BP_MyCube 블루프린트 에디터로 가자.

먼저, 컴포넌트 탭에서 **Audio** 컴포넌트를 선택하고 디테일 탭에서 **활성화** 영역 아래에 있는 **자동 활성화**(**Auto Activate**) 속성을 찾아보자. 디폴트로 체크되어있을 것이다. 이것을 클릭하여 체크해제하자.

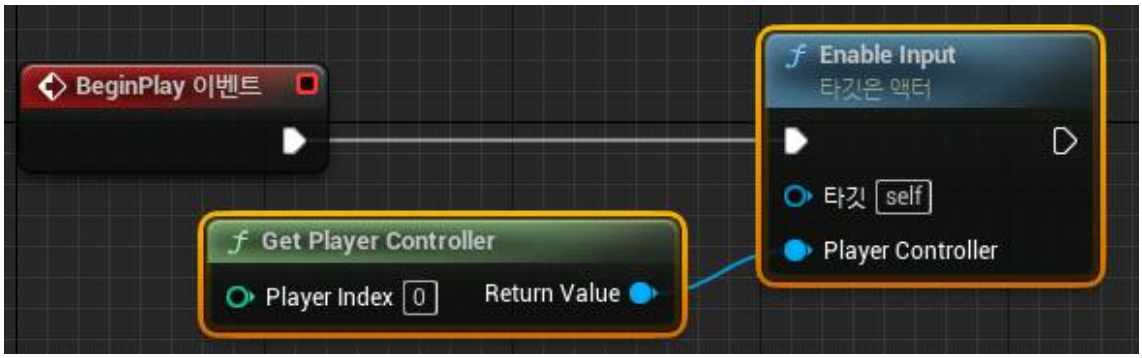


<참고> **자동 활성화**(**Auto Activate**) 옵션을 변경하는 또다른 방법에 대해서 알아보자.

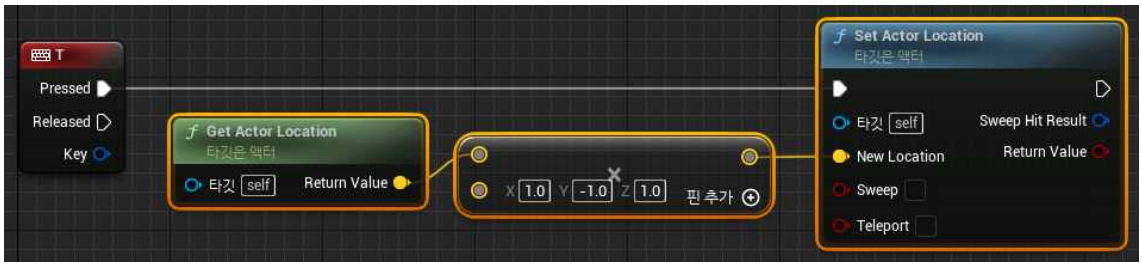
레벨 에디터의 **아웃라이너**에서 큐브 액터를 선택하고 디테일 탭에서 **Audio** 컴포넌트를 클릭하자. 디테일 탭의 **활성화** 영역 아래의 **자동 활성화**(**Auto Activate**) 속성이 보일 것이다. 이 속성값을 해제해도 자동 활성화가 해제된다. 그러나 이렇게 하면 해당 인스턴스에 대해서만 해제되는 것이고, 액터의 기본 디폴트 값에는 변동이 없다. 기본적인 큐브 액터의 속성으로 지정하려면 블루프린트 내에서 수정해주어야 한다. 블루프린트 클래스가

수정되면 배치된 모든 인스턴스의 옵션값이 블루프린트 클래스의 디폴트 값으로 리셋된다.

13. 계속해서 **BP_MyCube** 블루프린트 에디터에서 작업하자.
이제 특정 키가 눌러지면 액터의 사운드가 플레이되도록 해보자.
먼저, 액터에서 키 입력을 받을 수 있도록 조치하자.
액터에서는 기본적으로 입력 이벤트가 전달되지 않는다. 입력 이벤트가 액터에 전달되도록 하려면 액터에서 입력을 활성화해야 한다.
계속해서 **BP_MyCube** 블루프린트 에디터에서 작업하자. 이벤트 그래프 탭에서 **BeingPlay** 이벤트 노드를 당기고 **EnableInput** 노드를 배치하자. 그리고, 배치된 **EnableInput** 노드의 **PlayerController** 입력핀을 왼쪽으로 드래그하고 **GetPlayerController** 노드를 배치하자. 그래프가 아래와 같이 완성될 것이다. 컴파일하고 저장하자. 이제부터 플레이어 컨트롤러로 오는 입력 이벤트가 이 액터에게도 오게 된다.



14. 계속해서 **BP_MyCube** 블루프린트 에디터에서 작업하자.
이제 키를 누르면 큐브의 위치가 바뀌면서 오디오 재생을 활성화하도록 하자.
이벤트 그래프의 격자판에서 우클릭하고 액션선택 창에서 '키 t'를 검색하여 **T 키 입력** 이벤트 노드를 배치하자.
그다음, 노드의 **Pressed** 실행핀을 드래그하고 **SetActorLocation** 노드를 배치하자. 그리고, **GetActorLocation** 노드를 배치하고 그 결과에 (1,-1,1)을 곱해서 **SetActorLocation** 노드의 **New Location** 입력으로 연결하자.



15. **T 키 입력** 이벤트 노드의 오른쪽의 모든 노드를 선택하자. 우클릭하고 **함수로 접기**를 선택하자. 생성되는 함수의 이름을 **ChangeLocation**으로 지정하자.

이제 플레이하면 **T** 키를 누를 때마다 큐브가 왼쪽과 오른쪽으로 번갈아 이동할 것이다.

16. 그다음, **ChangeLocation** 함수 호출 노드의 출력 실행핀을 당기고 **활성화(Activate) (Audio)** 노드를 배치하자.

플레이해보자. **T** 키를 눌러보자. 왼쪽과 오른쪽으로 번갈아 가며 이동할 때에 해당 위치에서 사운드가 재생될 것이다.

17. 만약 **T** 키를 너무 빨리 누르게 되면 새로운 위치에서 재생되지 않는다. 왜냐하면 이전의 재생이 종료되어야 다시 활성화될 수 있기 때문이다. 따라서 수시로 재생해야 하는 경우에는 **활성화(Activate)** 함수 호출이 적당하지 않다.
이를 해결하는 방법을 알아보자.
먼저, 액션선택 창에서 '**키 u**'를 검색하여 **U 키 입력** 이벤트 노드를 배치하자.
그다음, 이벤트 노드의 실행핀을 당겨 **ChangeLocation** 함수 호출 노드를 배치하자.
그다음, **ChangeLocation** 노드의 출력핀을 당기고 **중지(Stop) (Audio)** 노드를 배치하자.
그다음, **중지(Stop) (Audio)** 노드의 출력핀을 당기고 **Play** 노드를 배치하자.
이런 경우에는 다음과 같이 **중지(Stop) (Audio)** 노드 후에 **재생(Play)(Audio)**노드를 배치하자.

이제, 중지한 후에 재생하므로 즉시 재생된다.
플레이해보자. **U** 키를 빨리 눌러보자. 잘 동작할 것이다.

18. 오디오 컴포넌트의 사운드 애셋도 바꿀 수 있다.
키를 누르고 있는 동안에 다른 사운드 애셋으로 잠시 바꾸도록 해보자.
먼저, 키보드 **V** 키 입력 이벤트 노드를 배치하자. 그리고, 노드의 **Pressed** 실행핀을 드래그하고 **SetSound (Audio)** 노드를 배치하자. 그리고, **New Sound** 입력핀에 **Fire01_Cue**를 지정하자.
그다음, 키보드 **W** 키 입력 이벤트 노드를 배치하자. 그리고, 또다른 **SetSound (Audio)** 노드를 배치하자. 그리고, **New Sound** 입력핀에 **Explosion01_Cue**를 지정하자.

플레이해보자. **V** 키와 **W** 키로 사운드 애셋을 바꾸면서 **U** 키를 눌러보자.

이 절에서는 블루프린트에서의 사운드 큐 활용에 대해서 학습하였다.

□