A cinematic still from the video game The Last of Us. A female character, Ellie, is seen from behind, looking out over a desolate, rocky landscape. In the distance, the ruins of a city are visible, including a large, partially collapsed structure. The sky is filled with dramatic, dark clouds, and the horizon is lit with a warm, orange glow from a setting or rising sun.

게임 엔진

LEC 15 블루프린트 통신



한국공학대학교
TECH UNIVERSITY OF KOREA

이대현 교수

Blueprint Communication

- 여러 액터 들 사이에 작업을 시키거나, 정보를 전달하고 공유하는 것.
 - 다른 액터로 하여금 어떤 일을 하게 함 - 예) 방안의 불을 켜.
 - 다른 액터에게 정보를 전달함 - 예) 데미지 전달
 - 다른 액터의 상태, 속성, 변수값, 결과 등을 파악함.
 - 다른 액터들(Listener)에게 공통적인 내용을 전달함(Broadcasting).

블루프린트 통신(Blueprint Communication)

- 통신을 위해서는 보내는 Blueprint 한 개와, 최소한 한 개 이상의 받는 Blueprint들이 존재
- 모든 통신은 1-Way
 - 양방향으로 하려면, 각각 따로 설정이 되어야 함.
- 통신에는 반드시 Reference(참조 이름 - 특정 객체를 지정할 수 있는 이름)가 필요함.
 - Sender, Receiver 둘 중 하나는 상대를 알아야 함 - 예) 이름, 주소

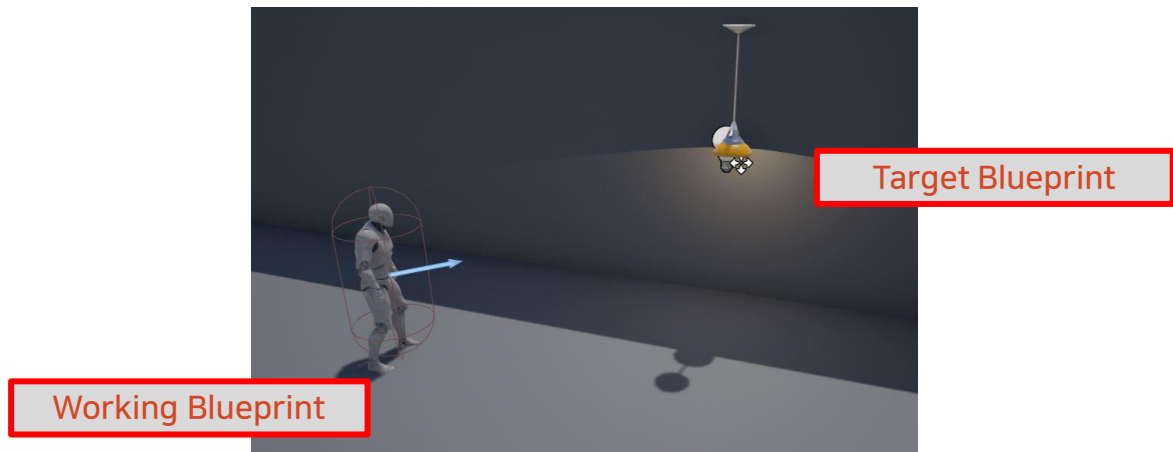


블루프린트 통신 유형

- 상대 액터의 참조를 획득하는 방법과 메시지의 전달 방법에 따라 분류
- 직접 통신(Direct Communication)
- 블루프린트 인터페이스(Blueprint Interface)
- 이벤트 디스패처(Event Dispatcher)

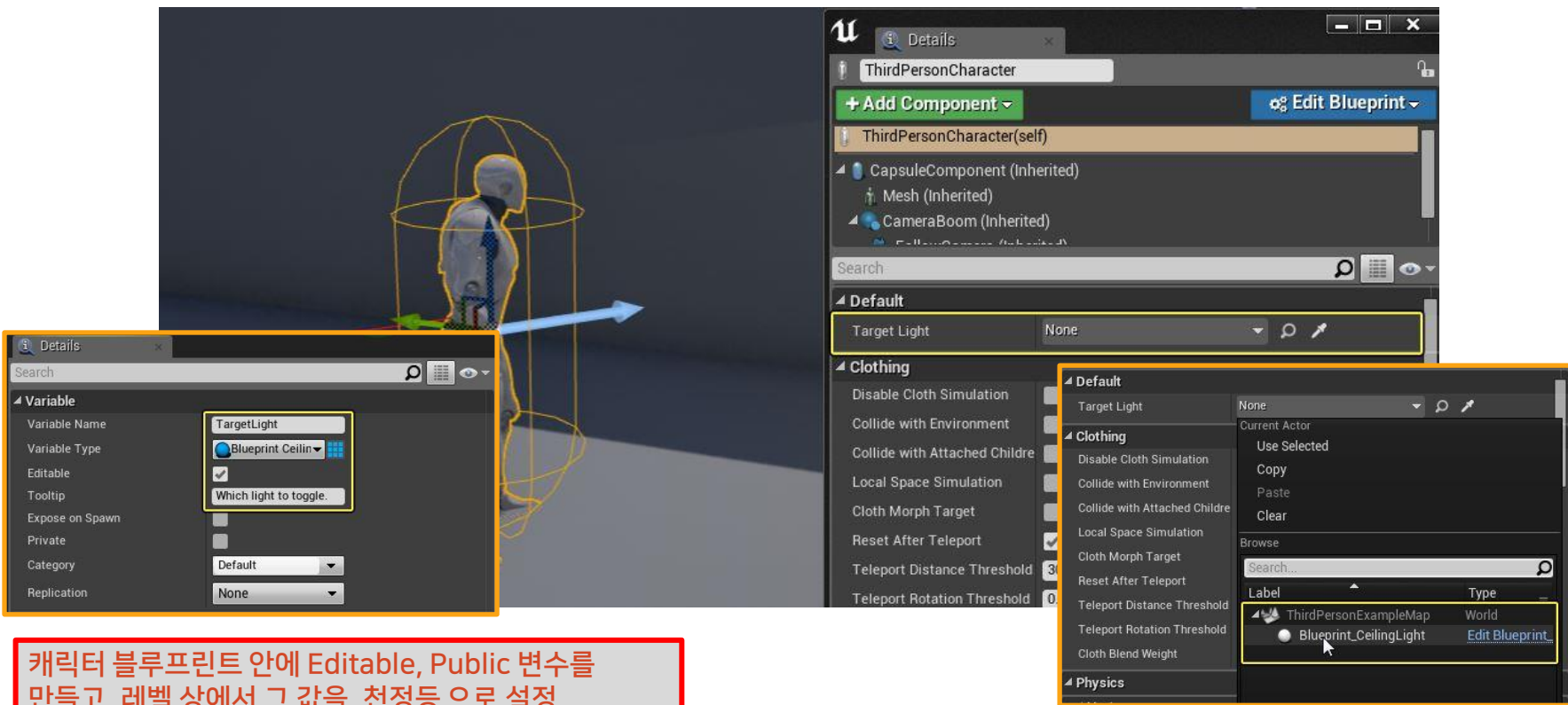
직접 통신

- 메시지를 주고받으려 하는 액터가 두 개 있을 때
 - Working Blueprint → Target Blueprint
 - One-to-One 통신
- Target Blueprint 를 쉽게 찾을 수 있을 때 유리
 - 액터가 맵 상에 이미 존재할 때, 또는 함수를 통해 쉽게 reference를 알 수 있을 때.



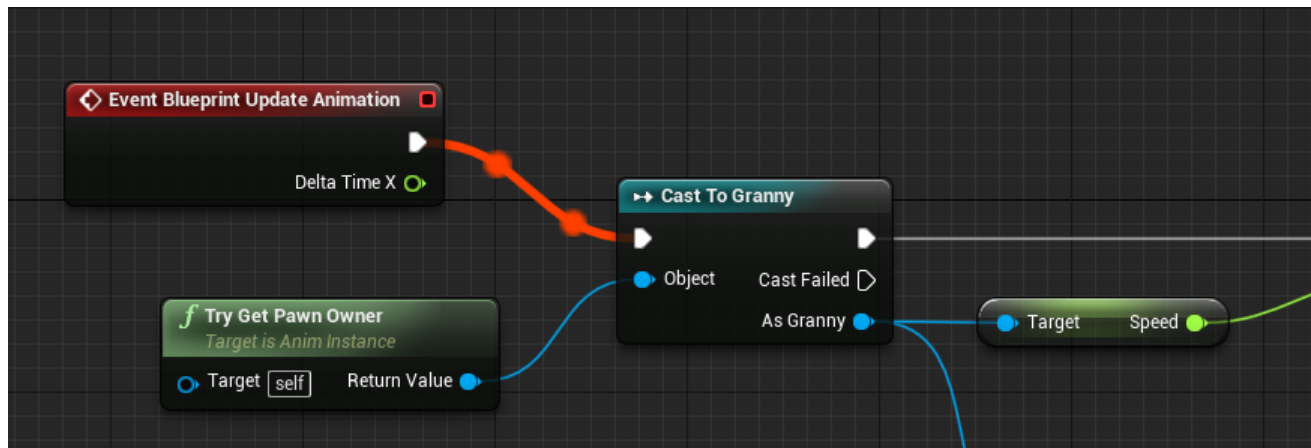
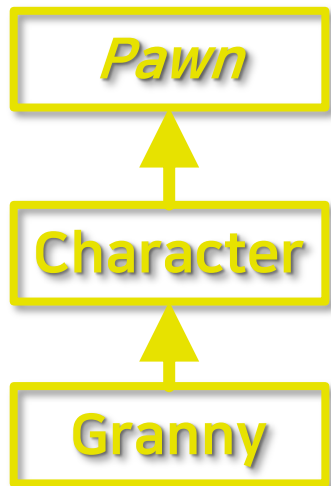
직접 통신 방법

- 가장 간단한 방법은 공개적으로 노출된 오브젝트 타입 변수를 이용하는 것



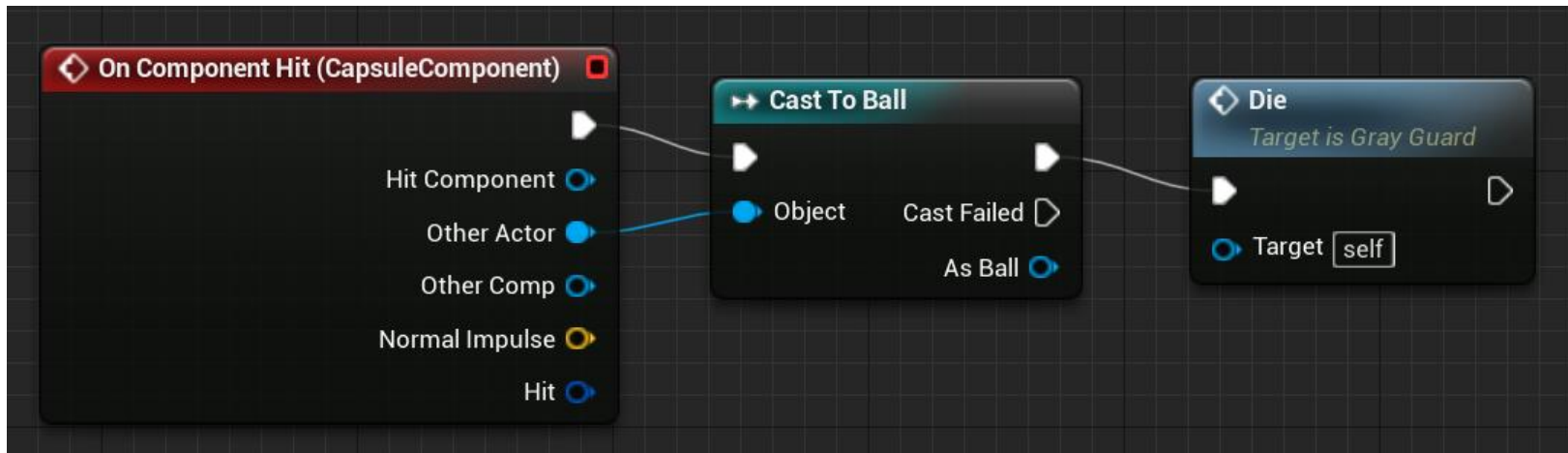
블루프린트 형변환(casting)

- 객체의 타입(클래스)을 특정 타입(클래스)으로 변경하는 것.
- 상속을 통해서 만들어진 자식 블루프린트 안에서 만들어진 내용을 액세스할 때 활용.
- 직접 통신을 할 때, 경우에 따라서, casting 이 필요.



블루프린트 형변환(casting)

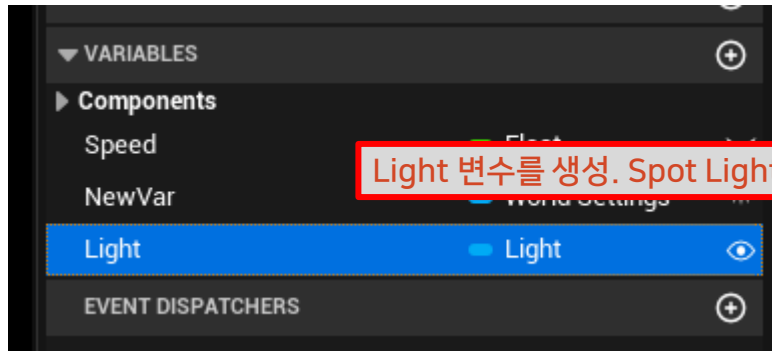
- 형변화를 하려는 액터가 형변환 대상 액터와 같은 검사할 때도 사용할 수 있음.
- 느리기 때문에, 틱에서 호출하는 것은 피해야 함. 미리 다른 변수에 저장해 놓는 것이 유리함.



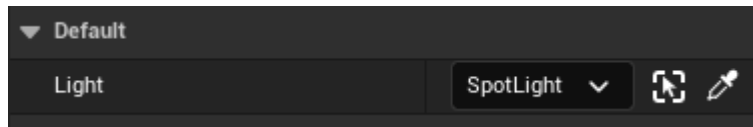


실습 방안의 불켜기

캐릭터 내 변수 생성 및 연결

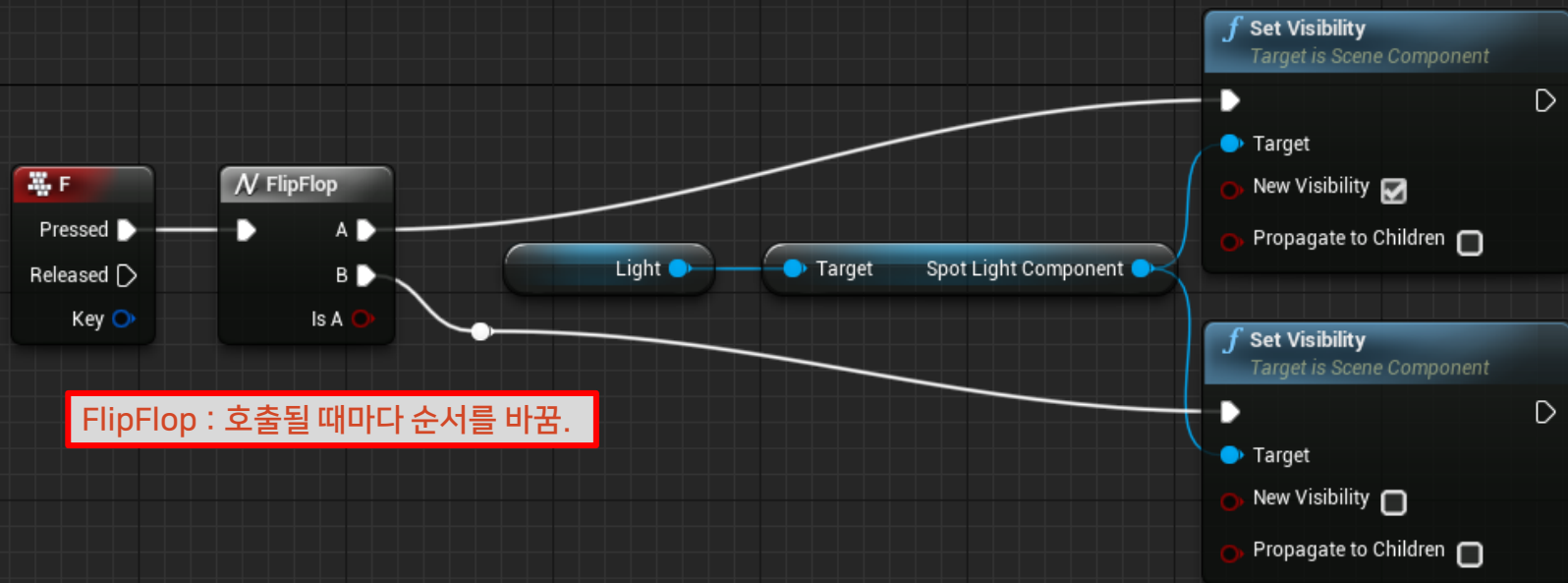


Light 변수를 생성. Spot Light type으로 설정. Instance Editable 설정.

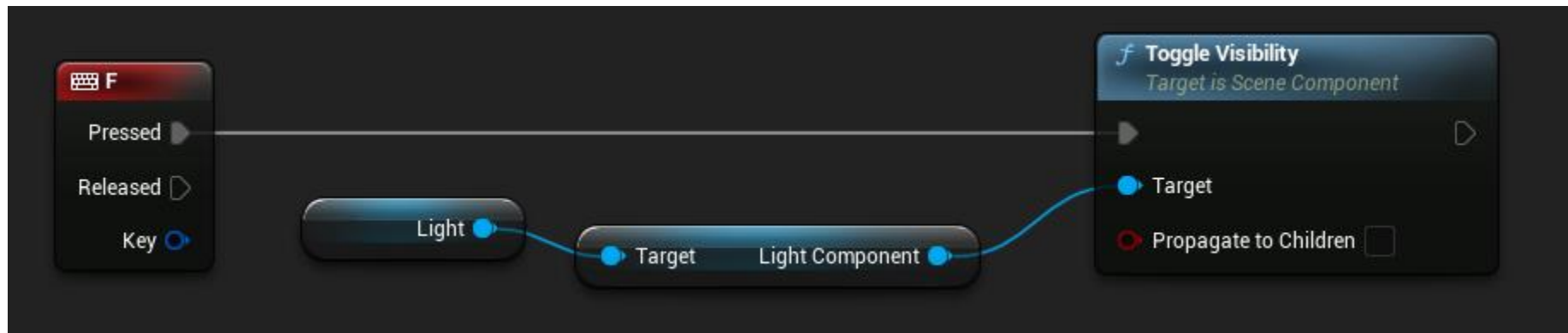


레벨 에디터 상에서, 캐릭터의 Light 변수를, 레벨 상에 배치된 SpotLight로 지정

CollectingGranny > Event Graph

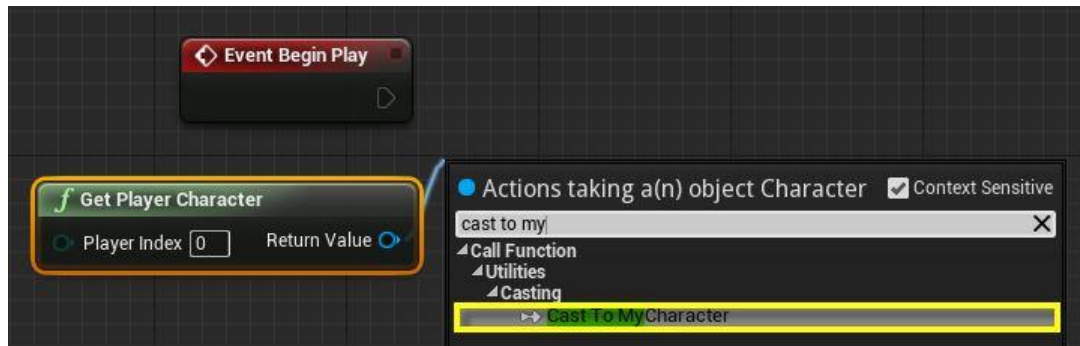


FlipFlop : 호출될 때마다 순서를 바꿈.



Blueprint Casting

- 상속을 통해서 만들어진 자식 블루프린트 안에서 만들어진 내용을 액세스할 때 활용.
- Direct Communication을 위해 경우에 따라서, casting 이 필요.



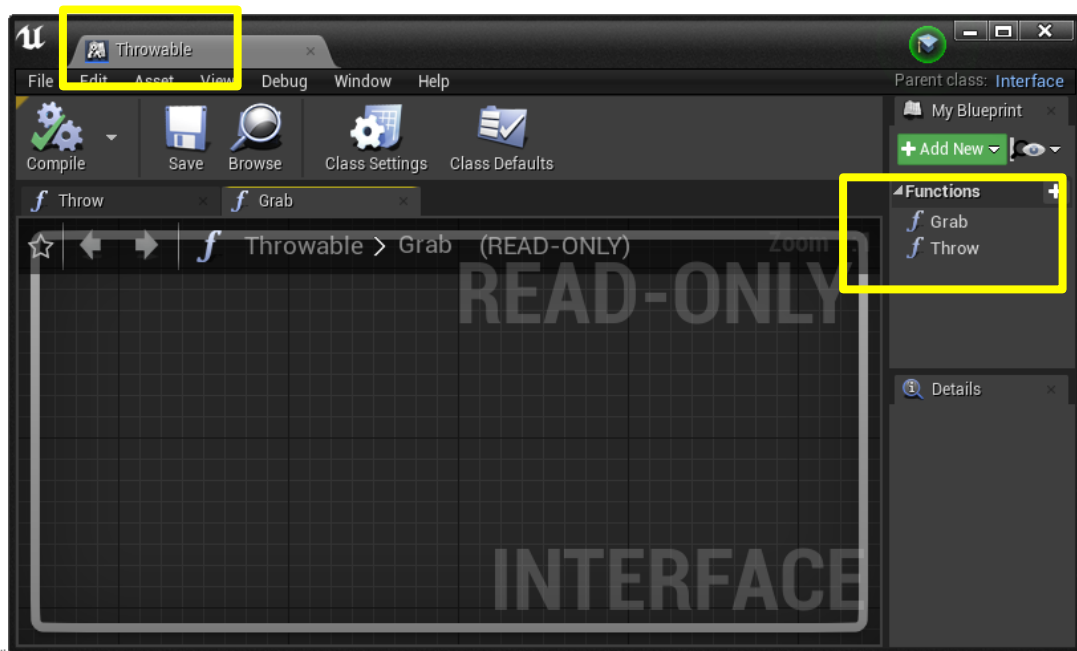
인터페이스의 역할

- 객체들에게 어떤 기능을 공통적으로 부여할 때 사용.
- 대상 객체와의 블루프린트 통신이 가능하게 해줌.



블루프린트 인터페이스

- 이름만 있고 구현은 없는 함수 하나 이상의 집합.
- 다른 블루프린트에 추가할 수 있음.
- 인터페이스가 추가된 블루프린트는 그 안에 함수를 갖게 됨.

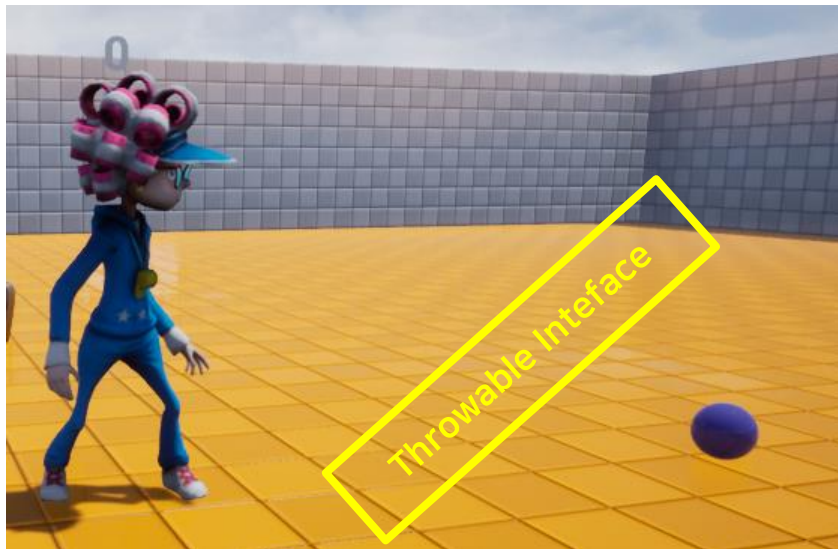


인터페이스의 역할



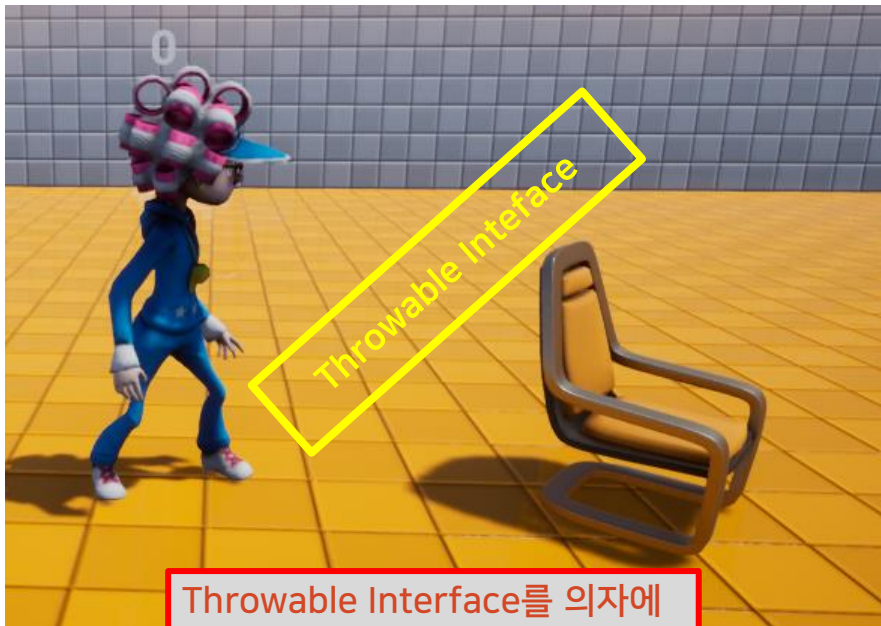
현재 공을 잡거나 던질 수 없음.

인터페이스의 역할



Throwable Interface를 Ball에
추가하고, Throw 및 Grab 함수를
구현해주면, 공을 잡고 던질 수
있게 됨.

인터페이스의 역할



Throwable Interface를 의자에
추가하고, Throw 및 Grab 함수를
구현해주면, 의자를 잡고 던질 수
있게 됨.

블루프린터 인터페이스

■ 일종의 계약

- “B, 너가 이 인터페이스를 구현하며(또는 함수로 구현), 내(A)가 호출하면 거기에 반응함.”
- B에게 인터페이스 구현이 없으면, 무시됨.

A



B



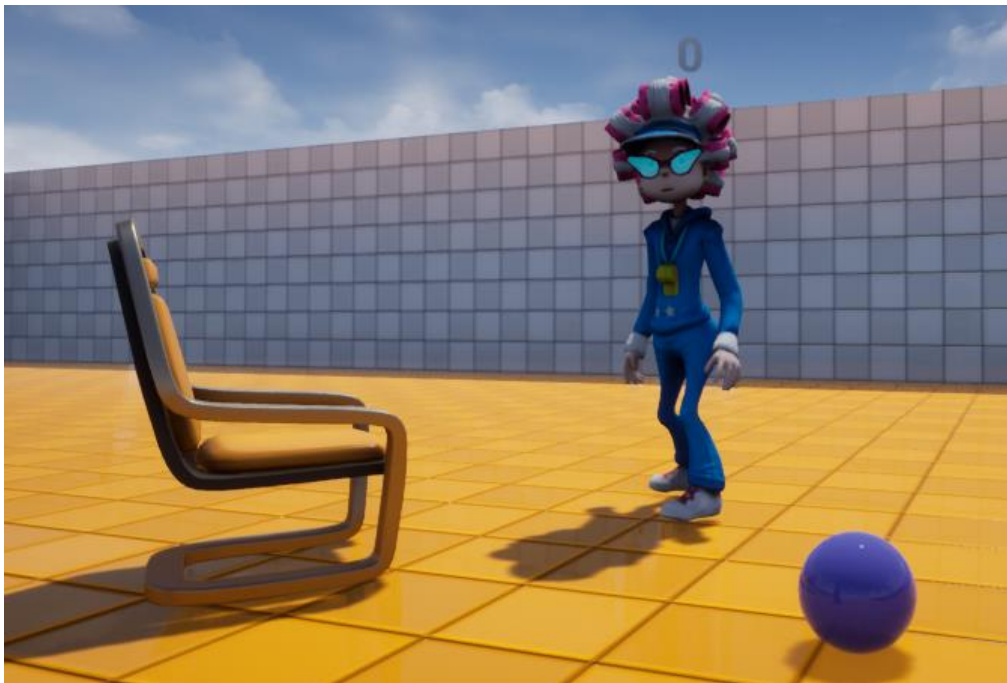


실습

공과 코인을 잡고 던지기

실습 목표

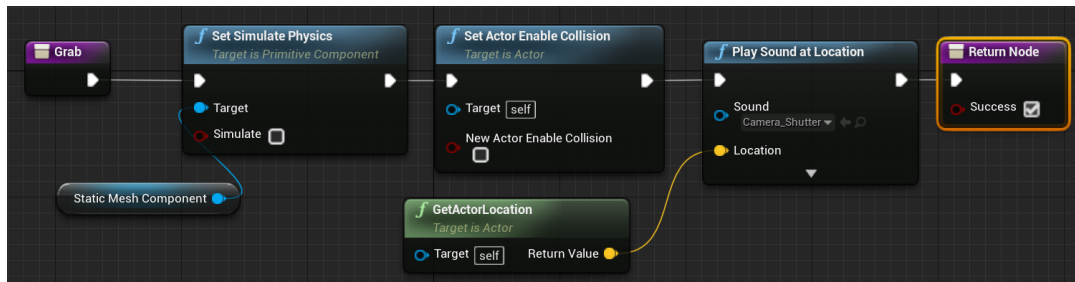
- 블루프린트 인터페이스 구현 절차 이해
- 공과 의자에 인터페이스 기능 구현 추가



인터페이스 기능 구현: 함수 vs 이벤트

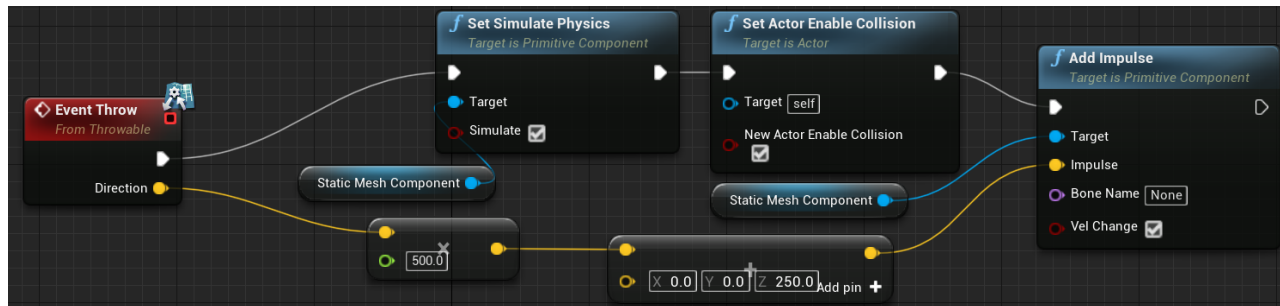
■ 함수로 구현

- 인터페이스 수행 결과로 출력값이 필요할 때



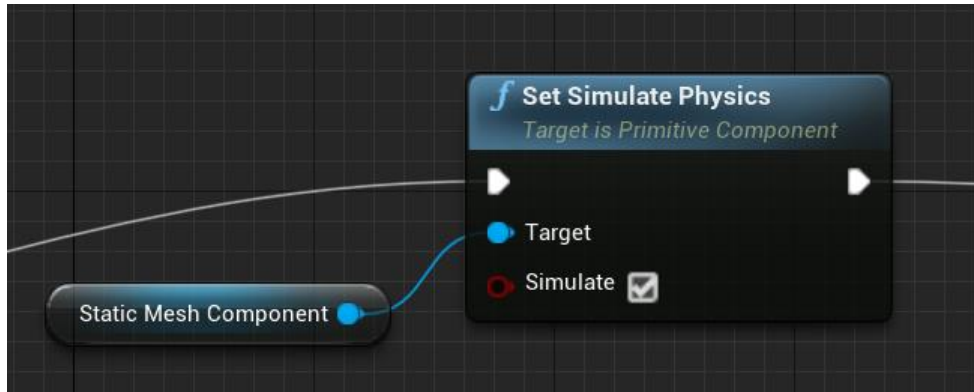
■ 이벤트로 구현

- 인터페이스 수행에 따른 결과값이 필요하지 않을 때



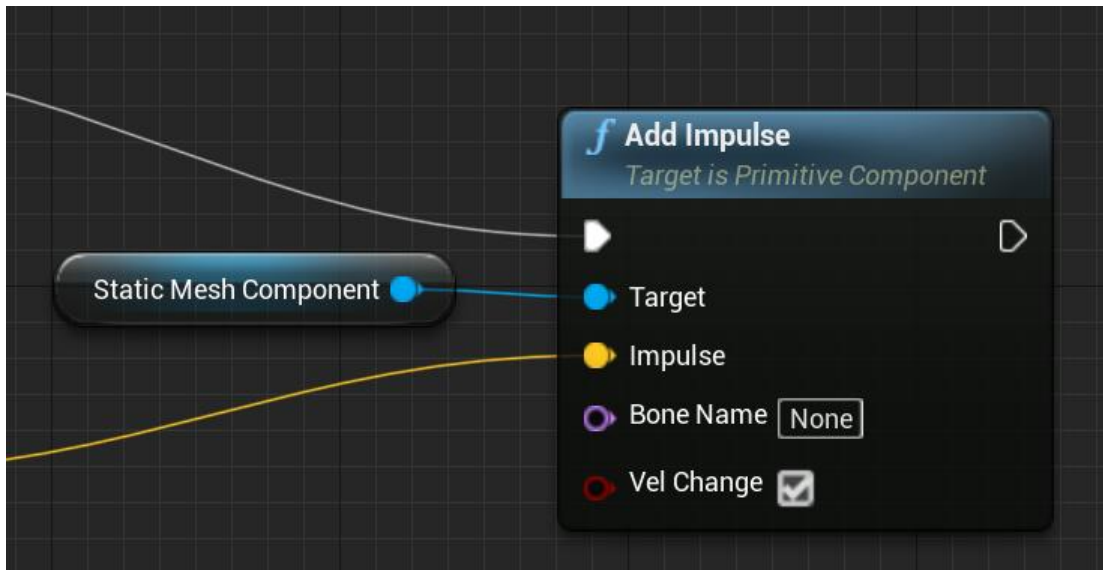
Set Simulate Physics 노드 함수

- 컴포넌트의 물리 시뮬레이션을 활성화.



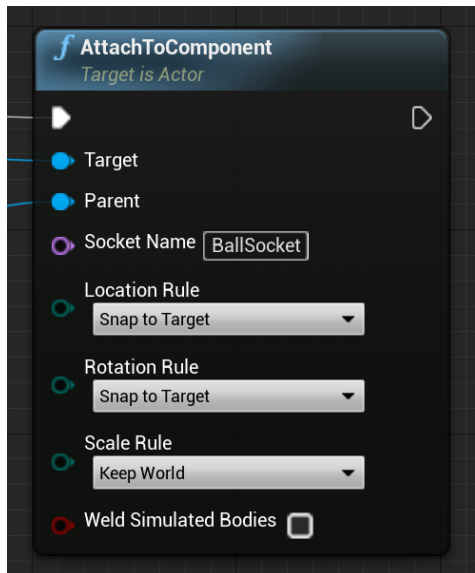
Add Impulse 노드 함수

- 물체에 충격량을 가해서, 물체를 움직이게 하는 함수



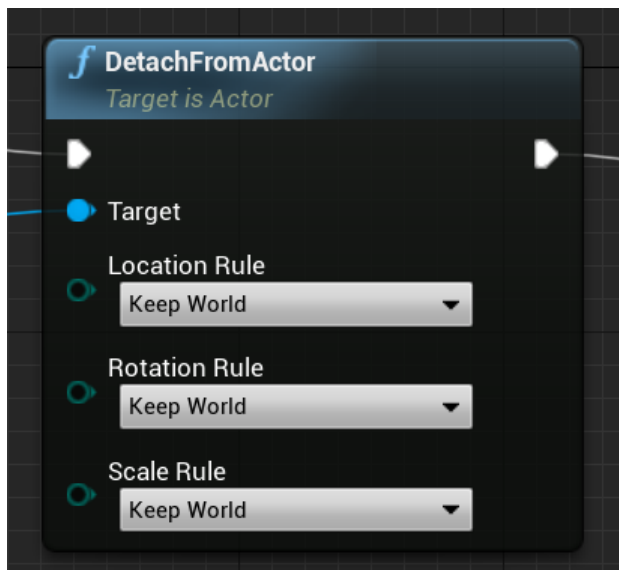
AttachToComponent 노드 함수

- 액터를 어떤 컴포넌트의 특정 소켓 위치에 부착시키는 함수.
 - Keep Relative – 액터의 기존 상대 좌표 유지.
 - Keep World – 기존 월드 좌표를 유지
 - Snap to Target – 소켓에 그대로 부착

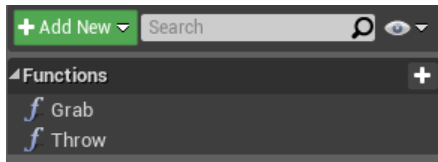
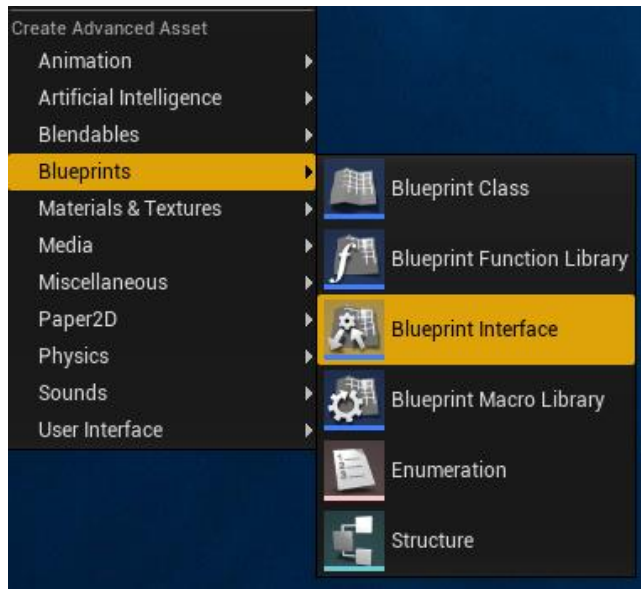


Detach From Actor 노드 함수

- 액터를 떼어내는 함수.
 - Keep World - 원래 위치의 월드 좌표에 그대로 배치
 - Keep Relative - 기존 상대 좌표를 유지



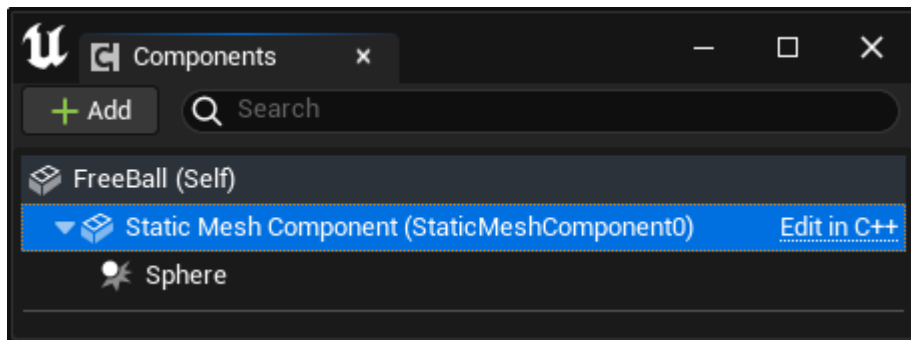
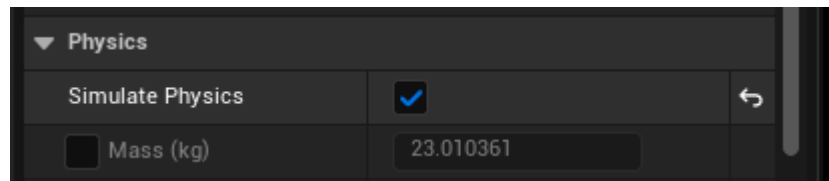
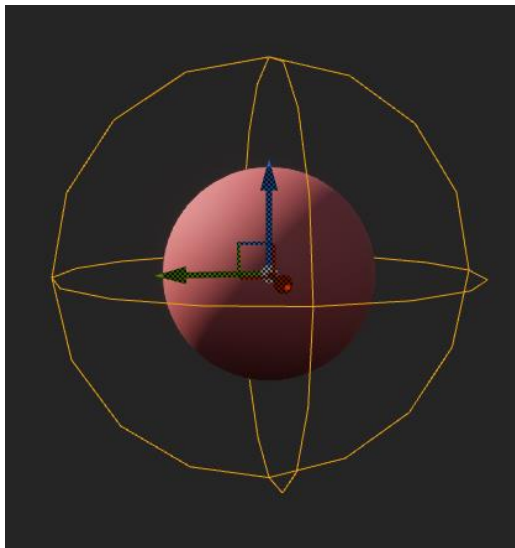
Blueprint Interface 만들기 - Throwable

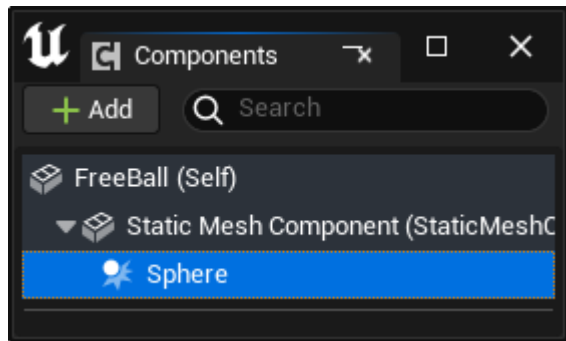


Interface는 입력과 출력 만을 정의하기 때문에, 그 내부는 편집할 수 없음.
따라서 READ-ONLY로 표시됨.

Ball

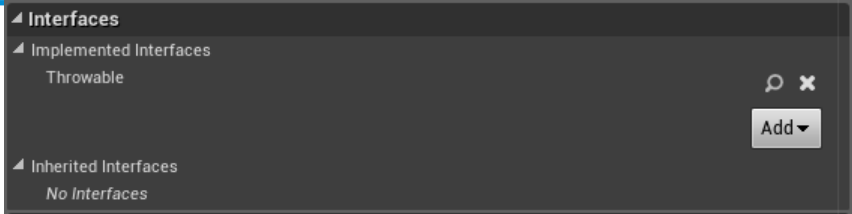
- 물리시뮬레이션과 충돌을 분리해서 처리





Generate Overlap Events	<input checked="" type="checkbox"/>
Can Character Step Up On	Yes ▼
▶ Collision Presets	OverlapAllDynamic ▼

인터페이스 설정



Class 설정에서, Interface 를 Add 해줌.

이 부분은, 마치 C++에서 어떤 클래스가 있을 때, 이 클래스를 어떤 Abstract Class 를 부모로 만들어주는 것처럼 생각할 수 있음.

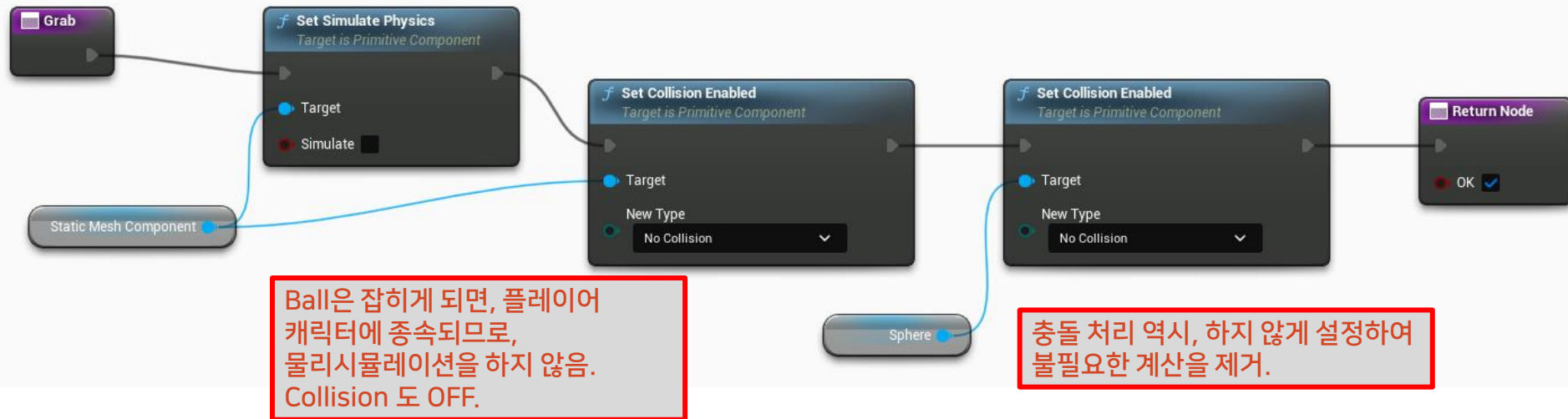


Interface 항목에 Grab 함수가 만들어짐 - 이 내용을 채워야 함(구현).
그런데, Throw는 어디로????

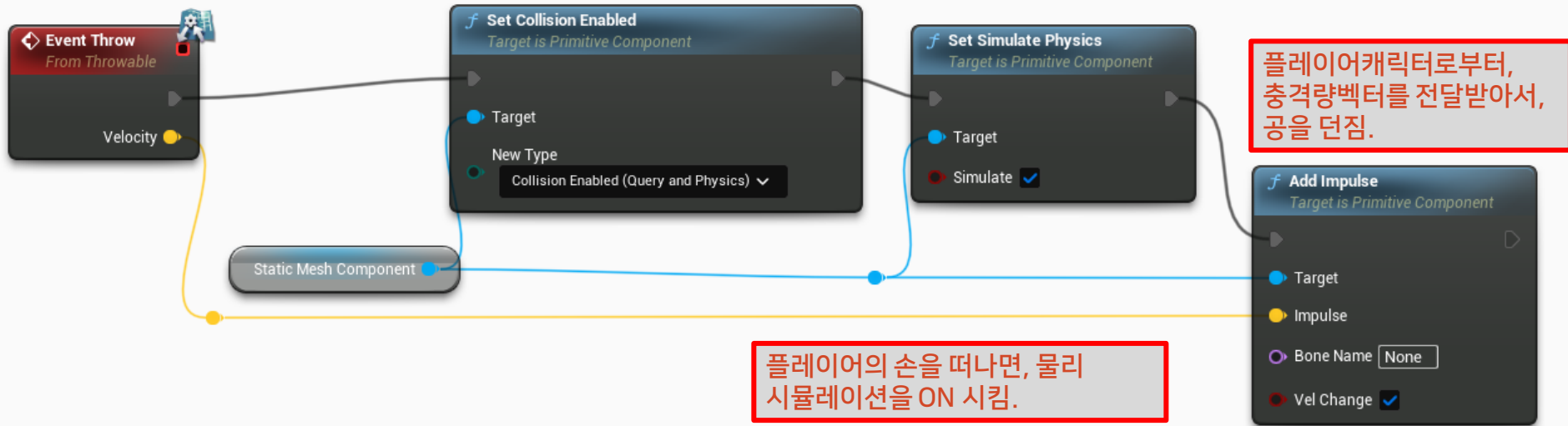
언리얼엔진은 인터페이스의 구현에 대해서,

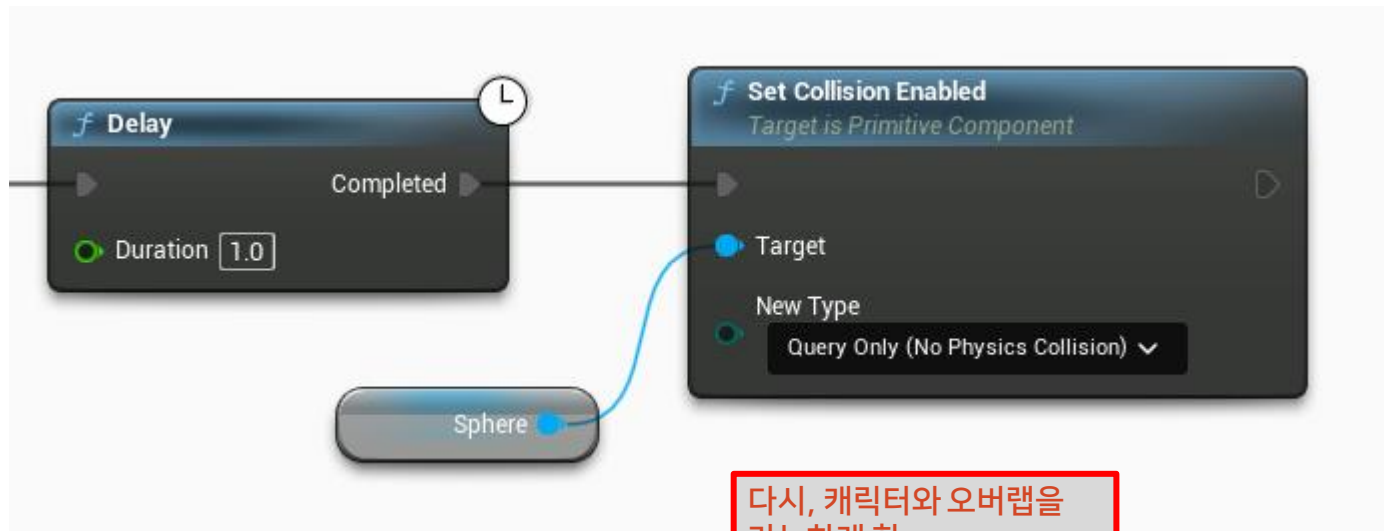
출력이 있는 경우 : 함수(function)로 구현
출력이 없는 경우 : 이벤트(event)로 구현

Grab Interface 구현 - 함수로 구현

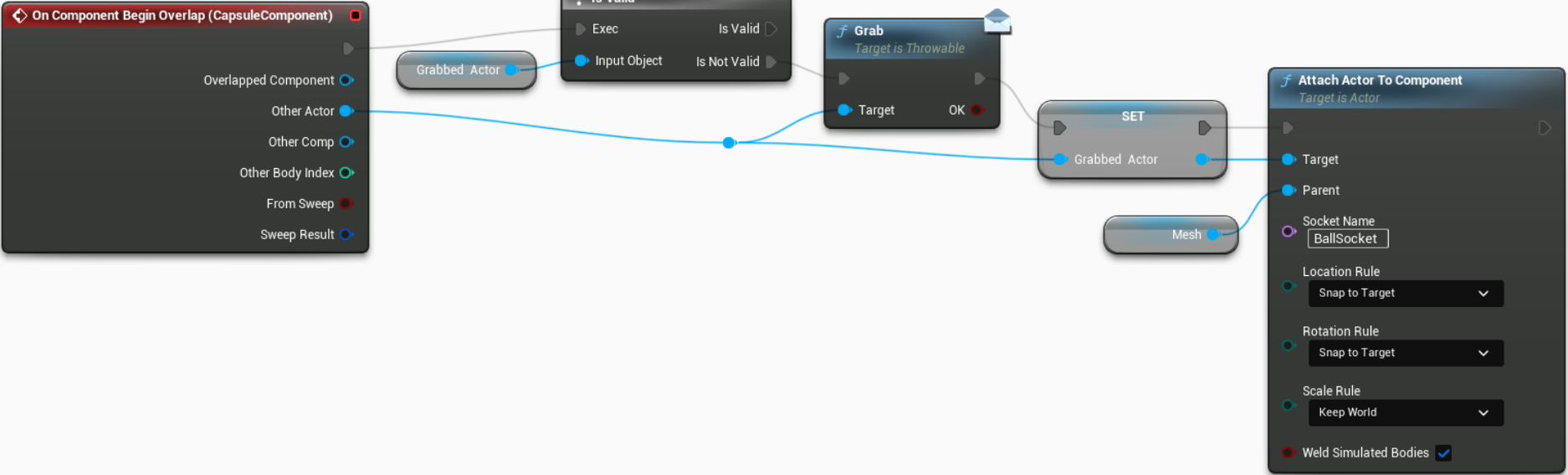


Thow Interface 구현 - Event로 구현





Granny의 캡슐과 Overlap 된 액터들에게,
Grab Interface 함수를 호출함.
Grab 함수없는 액터들은 무시됨.



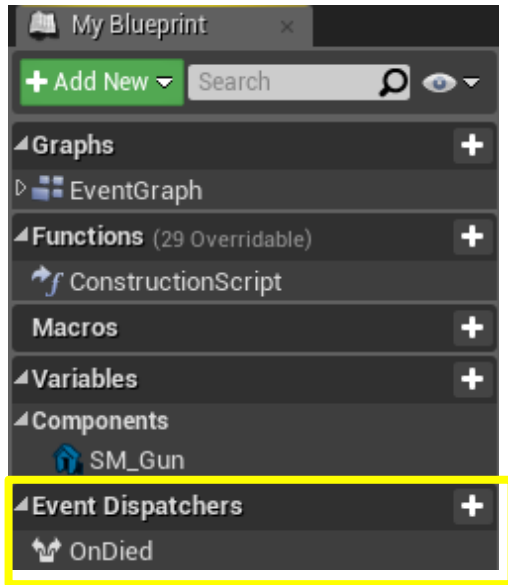
캐릭터의 GrabSocket 위치에 아이템을 부착,
크기는 원래 크기를 유지



Event Dispatcher

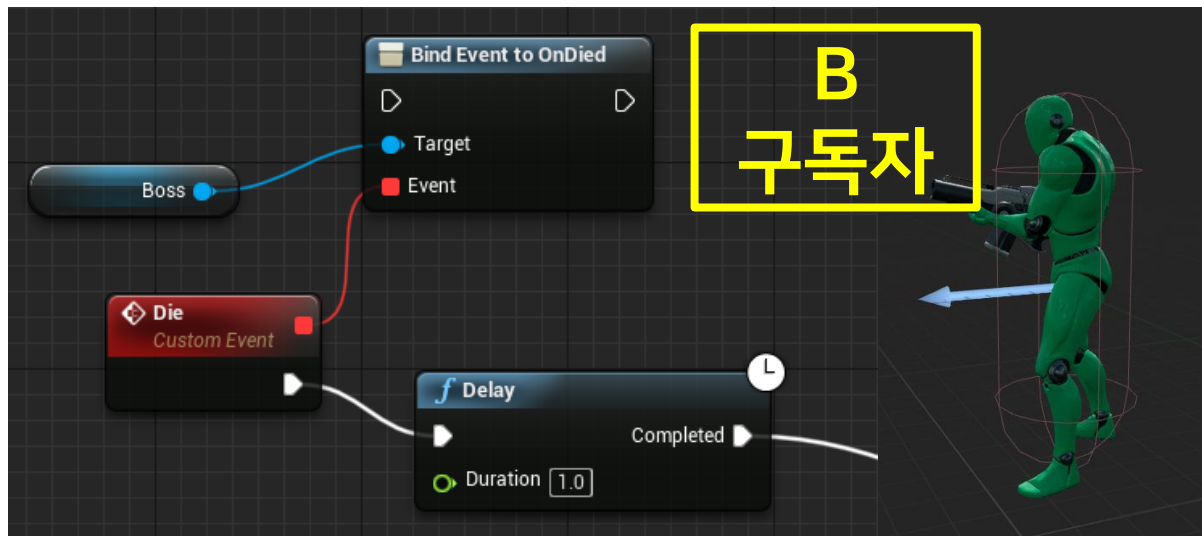
- 이벤트의 발생 여부를 대기 중인 다른 블루 프린트에게 알리기에 적합.
- 이벤트 발송에 따른 처리 여부를 실시간으로 변경할 수 있음.
- Event Dispatcher 가 필요한 경우.
 - 캐릭터 BP에서 Level Blueprint와 통신을 하고 싶어함.
 - 레벨 업함으로써, 레벨 안에 있는 기존의 잠긴 영역을 오픈.
 - 캐릭터가 레벨 전체의 객체들에게 동시다발적으로 어떤 액션을 지시함.
 - 스폰된 액터에서 상태 변화가 일어나면, 이벤트가 발동.
 - 보스를 스폰하고, 보스가 죽으면 월드에 보상이 스폰되는 이벤트가 발동.
 - 레벨에 아이템을 스폰하고, 픽업시 아이템과 캐릭터에게 알림.

이벤트 디스패처의 생성



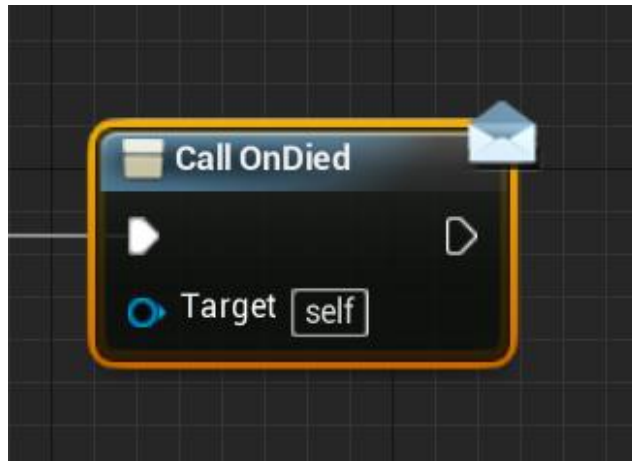
이벤트를 발송하고자 하는 블루프린트에서
Event Dispatcher를 생성

이벤트 연결(Binding) – Bind Event to ~

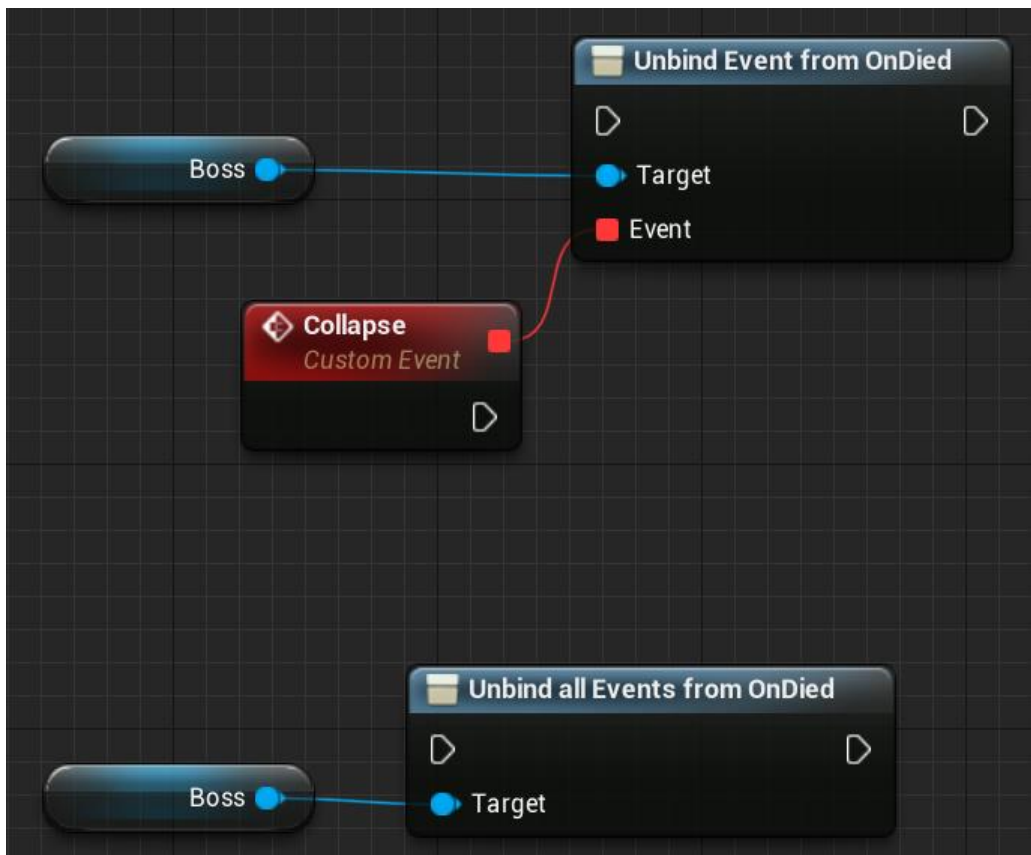


B에서 A의 레퍼런스를 확보한 후, A의 이벤트 발송 "OnDied"과 B의 이벤트 함수 "Die" 를 연결시킴.

이벤트 발송의 실행



이벤트 연결 해제 (Unbind)

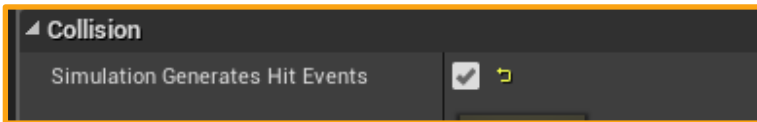
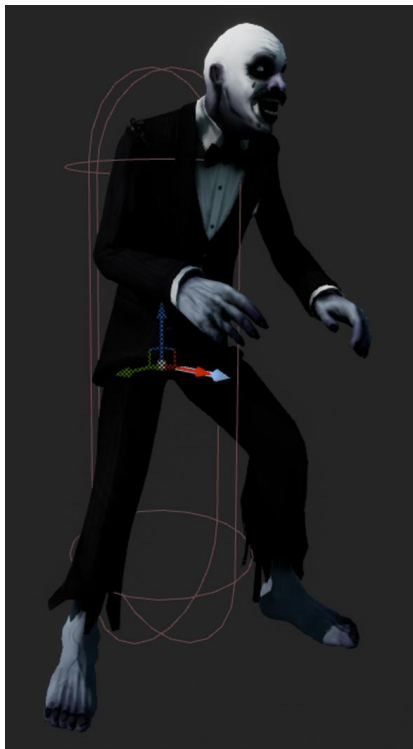




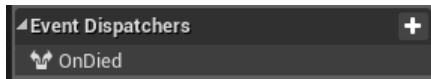
실습

보스 죽음에 따른 좀비 자동 소멸

BossZombie

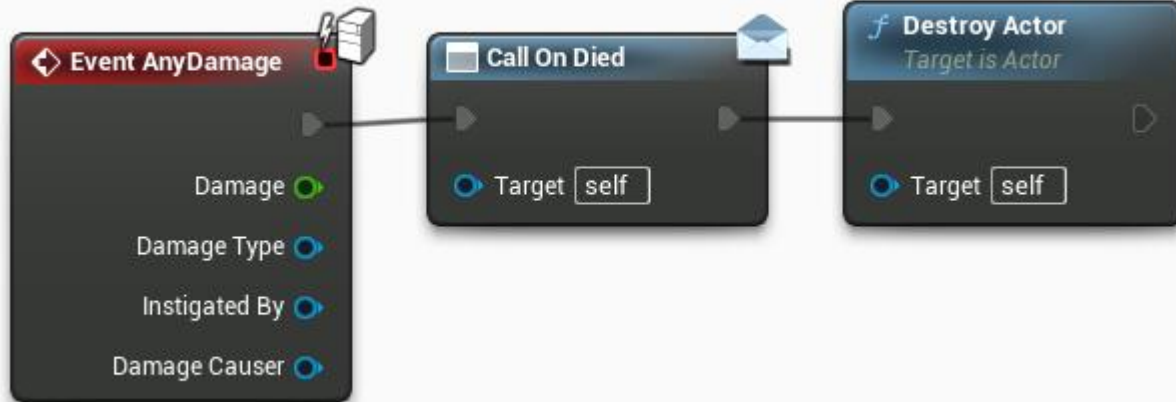


Boss가 죽을 경우, OnDied 라는 이벤트를 날리도록 구현.



OnDied 라는 이벤트 Dispatch 를 생성

OnDied 라는 이벤트 Dispatch 를 보냄.



zombie의 Die 구현

Guard가 BeginPlay로 초기시작을 할때

Boss Type의 액터를 찾아서,

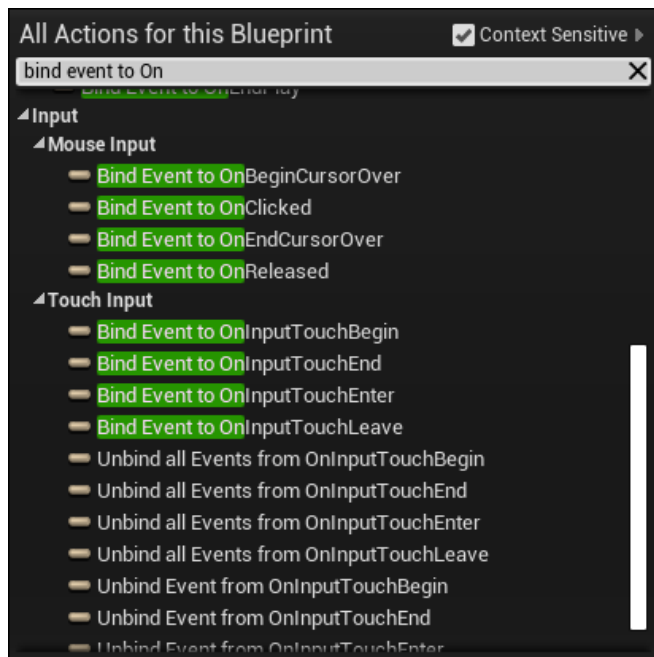
zombie와, BossZombie 모두 Level에 미리 배치된 상태라고 하면, Direct Communication 에서 했듯이, 레벨 에디터 상에서 zombie에게 BossZombie 를 직접적으로 지정할 수도 있음.
하지만, zombie가 동적으로 Spawn되는 상황이면?
→ 실시간으로 레벨에 존재하는 BossZombie를 찾아서 미리 보스의 reference를 확보해야 함.

On Died 라는 메시지에 Die 이벤트를 바인드



내부 이벤트 발송

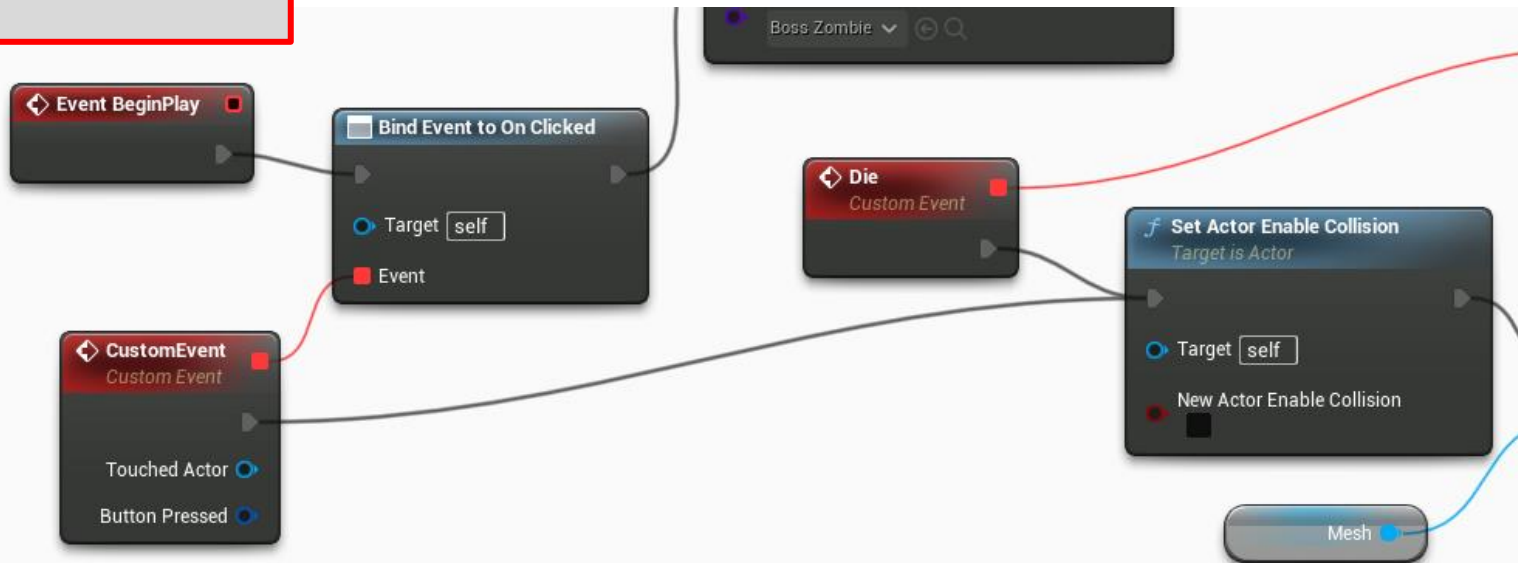
- 언리얼 엔진은 액터들이 다양한 이벤트 발송들을 내부적으로 이미 갖고 있음.
- 새로운 이벤트 발송을 만들기 전에 미리 기존 이벤트 발송들을 확인해야 함.
 - 예) 어떤 액터를 클릭했을 때, 이에 따른 처리를 하려면?



PlayerController



zombie



요약

■ 블루프린트 통신

- 독립적인 객체들이 서로 상호작용하기 위한 방법

■ 통신 유형

- 직접 통신(Direct Communication) A → B
 - 상대방 객체에 대한 직접적인 레퍼런스를 확보한 후, 통신
- 블루프린트 인터페이스(Blueprint Interface)
 - 인터페이스가 필요한 객체가 있으면, 그 객체에 인터페이스를 추가하고, 그 인터페이스 함수를 구현.
- 이벤트 디스패처(Event Dispatcher)
 - 여러 객체들에게 이벤트를 방송(Broadcasting)함.
 - 1 대 다 통신