

언리얼 스터디

2주차

0. 할거

1. 효율적인 충돌 처리 기술 - 채널, 콜리전
2. 상호작용 만들기 - 인터페이스, 채널 탐색

0-1. 참고 사이트

<http://api.unrealengine.com/KOR/Engine/Physics/Tracing/HowTo/MultiLineTraceByChannel/>

<http://api.unrealengine.com/KOR/Engine/Physics/Collision/Overview/#%EC%BD%9C%EB%A6%AC%EC%A0%84%EB%B0%8F%EC%8B%9C%EB%AE%AC%EB%A0%88%EC%9D%B4%EC%85%98%EC%8B%9C%ED%9E%88%ED%8A%B8%EC%9D%B4%EB%B2%A4%ED%8A%B8%EC%83%9D%EC%84%B1>

<https://www.unrealengine.com/ko/blog/collision-filtering?lang=ko>

0-1. 참고사이트

<http://api.unrealengine.com/KOR/Engine/Blueprints/UserGuide/Types/Interface/index.html>

https://www.youtube.com/watch?v=JhzEPfE6vbU&index=2&list=PLkHDai4yit5WG_eizDuOnDINvv4_5tCSYh

1. 오브젝트 채널

1. 각각의 오브젝트는 채널(성격)을 지님.
2. 채널끼리 상호작용 기능을 미리 정해줌
3. 상호작용은 3가지 **block, overlap, ignore**
4. 각각의 상호작용 기능은 이렇.
5. **overlap**은 겹쳐짐 이벤트만.
6. **block**은 충돌 처리 + 이벤트.
7. **ignore**은 무시.

		Object A		
		Ignore	Overlap	Block
Object B	Ignore	Ignore	Ignore	Ignore
	Overlap	Ignore	Overlap	Overlap
	Block	Ignore	Overlap	Block

1. 트레이스 채널

1. 이게 개쩌는거임.
2. 오브젝트랑 비슷한데 좀 다름.
3. **block**을 해야 트레이스가 됨.
4. 말그대로 뭔가를 찾을때 사용.

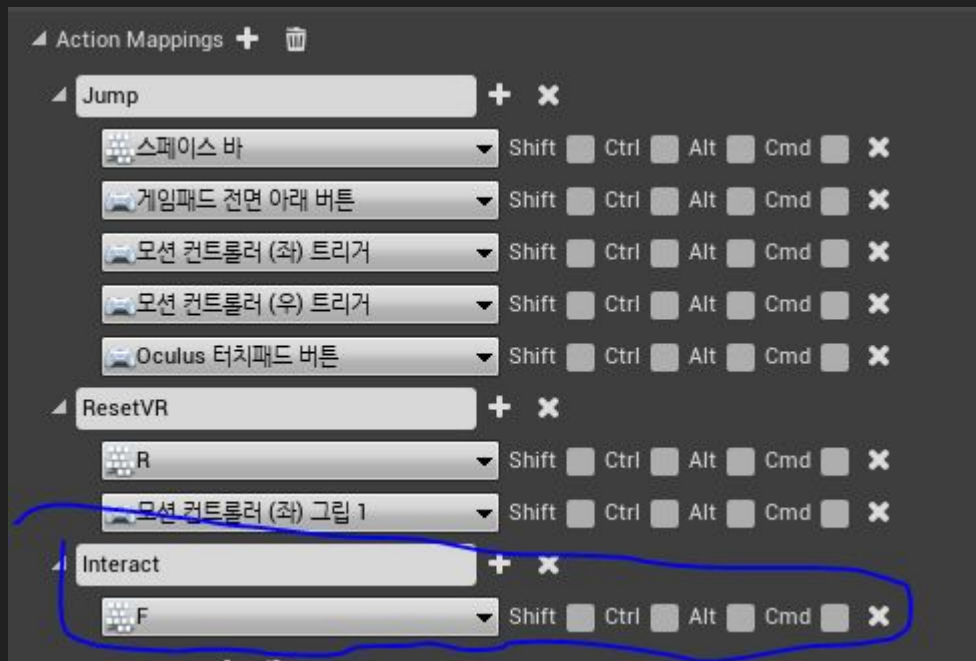
1. 프리셋

1. 미리 오브젝트가 어떻게 상호작용할지 만들어두는거
2. 오브젝트 채널 - 오브젝트 채널 이렇게 관계를 설정하는게 아님.
3. 액터 - 오브젝트 채널 관계를 설정.
4. 미리 만들어두고 편하게 관리하는게 프리셋.

1. 모르겠으니 해보자

1. 지난번거 키자.

1. 입력 추가



1. 오브젝트 채널 만들기

엔진 - 콜리전

콜리전 프로파일 - 콜리전 세팅을 세팅하고 편집합니다.

 이 세팅은 DefaultEngine.ini 에 저장되었으며, 현재 쓰기가능 입니다.

Object Channels

오브젝트 & 트레이스 채널을 포함해서 커스텀 채널을 18 개

이름

Interactable

Trace Channels

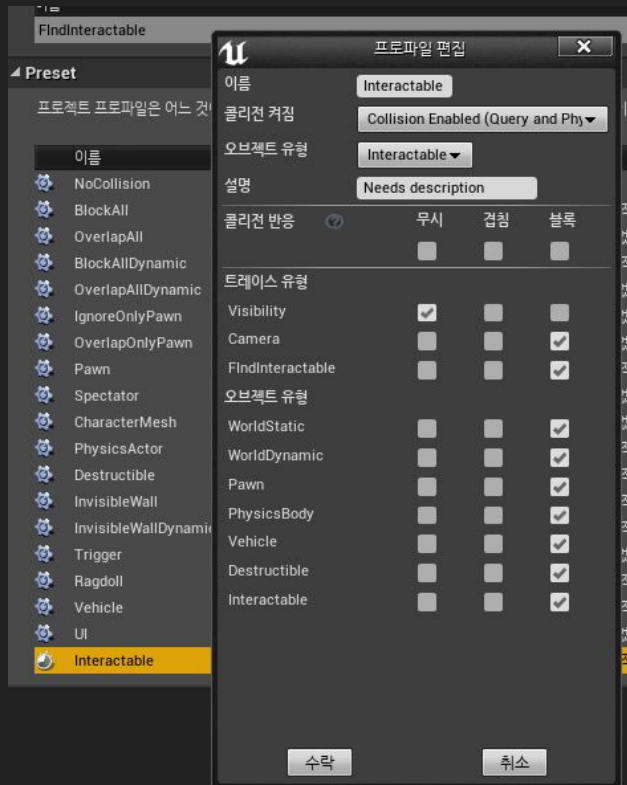
오브젝트 & 트레이스 채널을 포함해서 커스텀 채널을 18 개



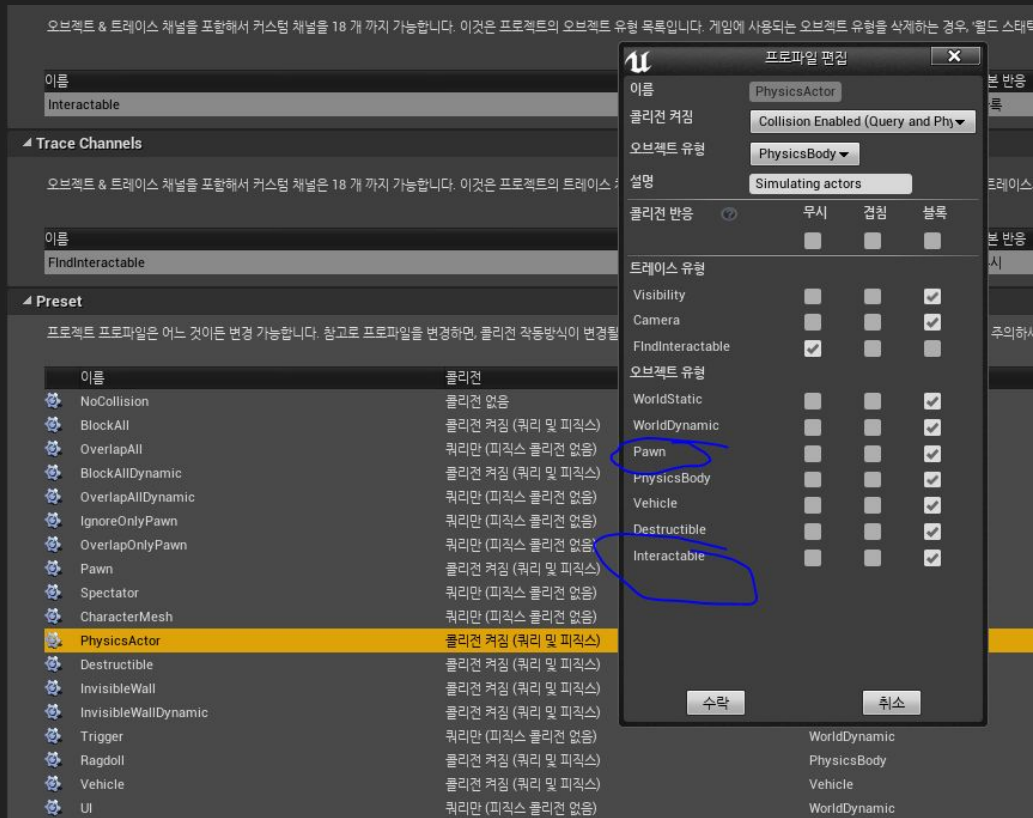
1. 만든다



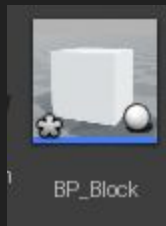
1. 프리셋을 만든다



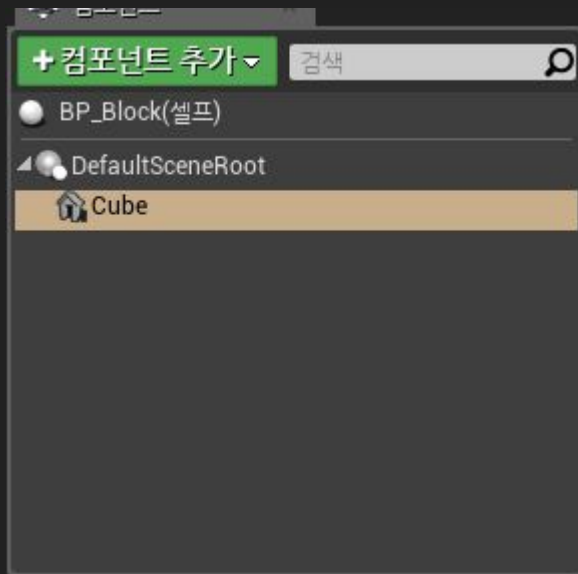
1. 다른 프리셋들의 설정을 pawn 과 일치시킴



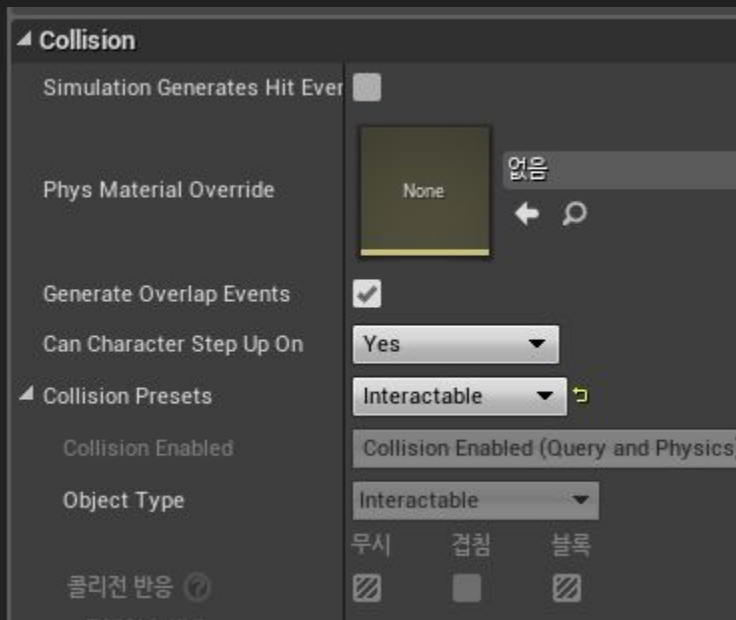
1-1. BP_Block 라는 액터 를 만듬.



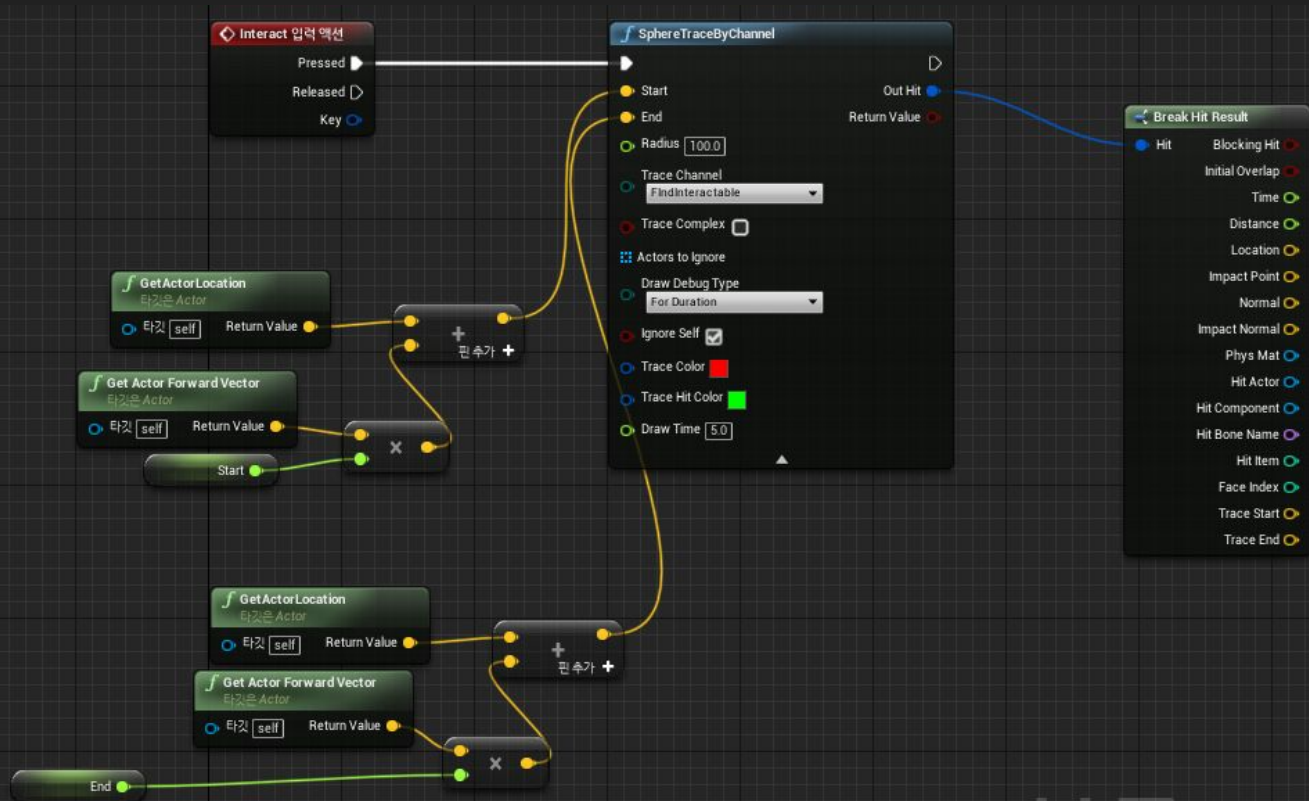
1-1. cube 라는 컴포넌트 추가



1-1. 콜리전 프리셋 변경



1-2. player 에 만들어줌



1-2. 좀 편한 팁

The image displays a workflow in Unreal Engine 4. On the left, a sequence of nodes is shown: an **Interact 입력 액션** (Interact Input Action) with 'Pressed' and 'Released' events, followed by **Get Actor Location** and **Get Actor Forward Vector** nodes, both targeting 'self'. A context menu is open over the workflow, listing various actions such as '삭제' (Delete), '잘라내기' (Cut), '복사하기' (Copy), '복제' (Duplicate), '노드 새로고침' (Refresh Node), '링크 끊기' (Break Link), '정의로 이동' (Move to Definition), '레퍼런스 찾기' (Find Reference), '이름 변경' (Rename), '체계' (Hierarchy), '노드 접기' (Collapse Node), '함수로 접기' (Collapse to Function), '매크로로 접기' (Collapse to Macro), '정렬' (Sort), '문서' (Documentation), '문서 보기' (View Documentation), '코멘트 그룹' (Comment Group), and '선택에서 코멘트 생성' (Generate Comment from Selection). The '함수로 접기' (Collapse to Function) option is circled in blue. A yellow line connects this menu option to a **Find Forward** node on the right, which is also circled in orange. The **Find Forward** node has a 'Return Value' output. Below the workflow, a **Search Details** (검색 디테일) panel is visible, showing settings for a '그래프' (Graph) element, including '카테고리' (Category) set to '기본' (Basic), '키워드' (Keywords), '간단 노드 제목' (Simple Node Title), '접근 지정자' (Access Specifier) set to '퍼블릭' (Public), and '플어' (Language) set to 'C++'. The '출력' (Output) section shows 'ReturnValue' as a 'Vector' type. At the bottom right, a small diagram shows a **Find Forward** node connected to an **End** node, with the connection line highlighted in orange.

1-2. start = 50; end = 100;

다테일

검색 다테일

Variable

변수 이름 Start

변수 유형 플롯

인스턴스 편집가능

블루프린트 읽기 전용

툴팁

스폰시 노출

프라이빗

시네마틱에 노출

카테고리 Default

슬라이더 범위

값 범위

리플리케이션 None

리플리케이션 조건 None

기본값

Start 50.0

다테일

검색 다테일

Variable

변수 이름 End

변수 유형 플롯

인스턴스 편집가능

블루프린트 읽기 전용

툴팁

스폰시 노출

프라이빗

시네마틱에 노출

카테고리 Default

슬라이더 범위

값 범위

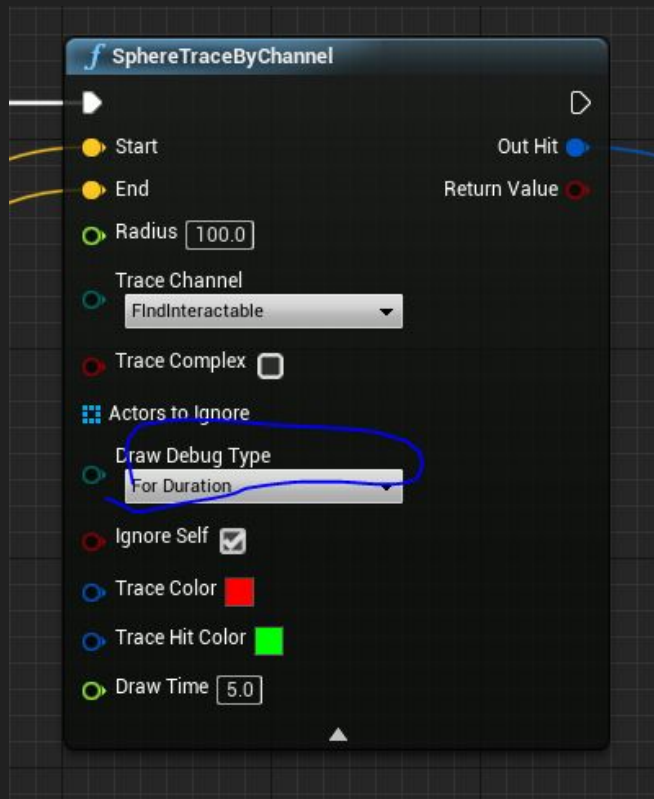
리플리케이션 None

리플리케이션 조건 None

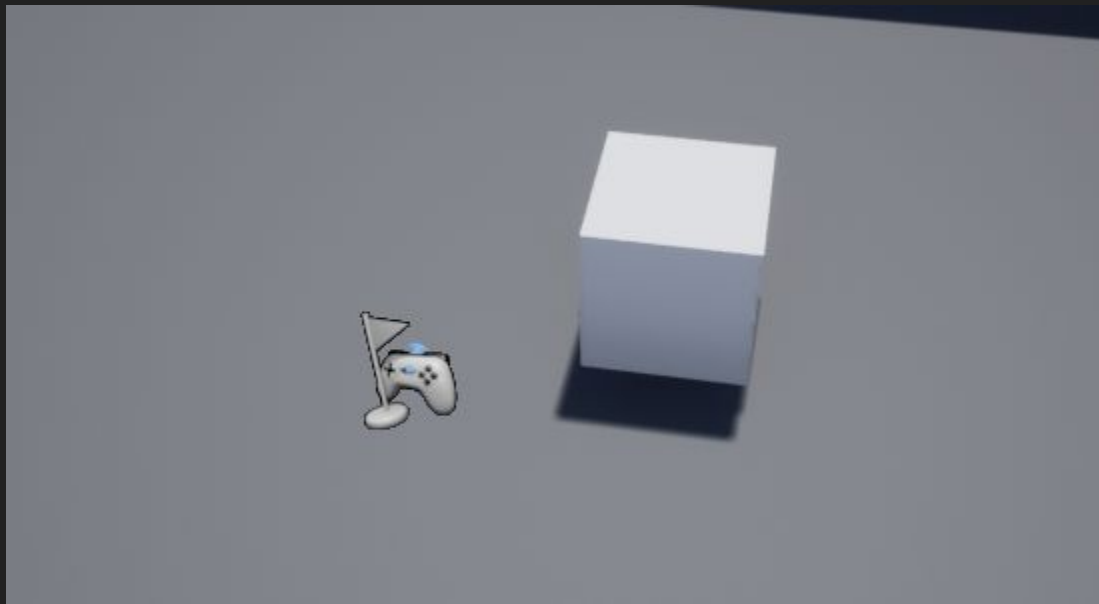
기본값

End 100.0

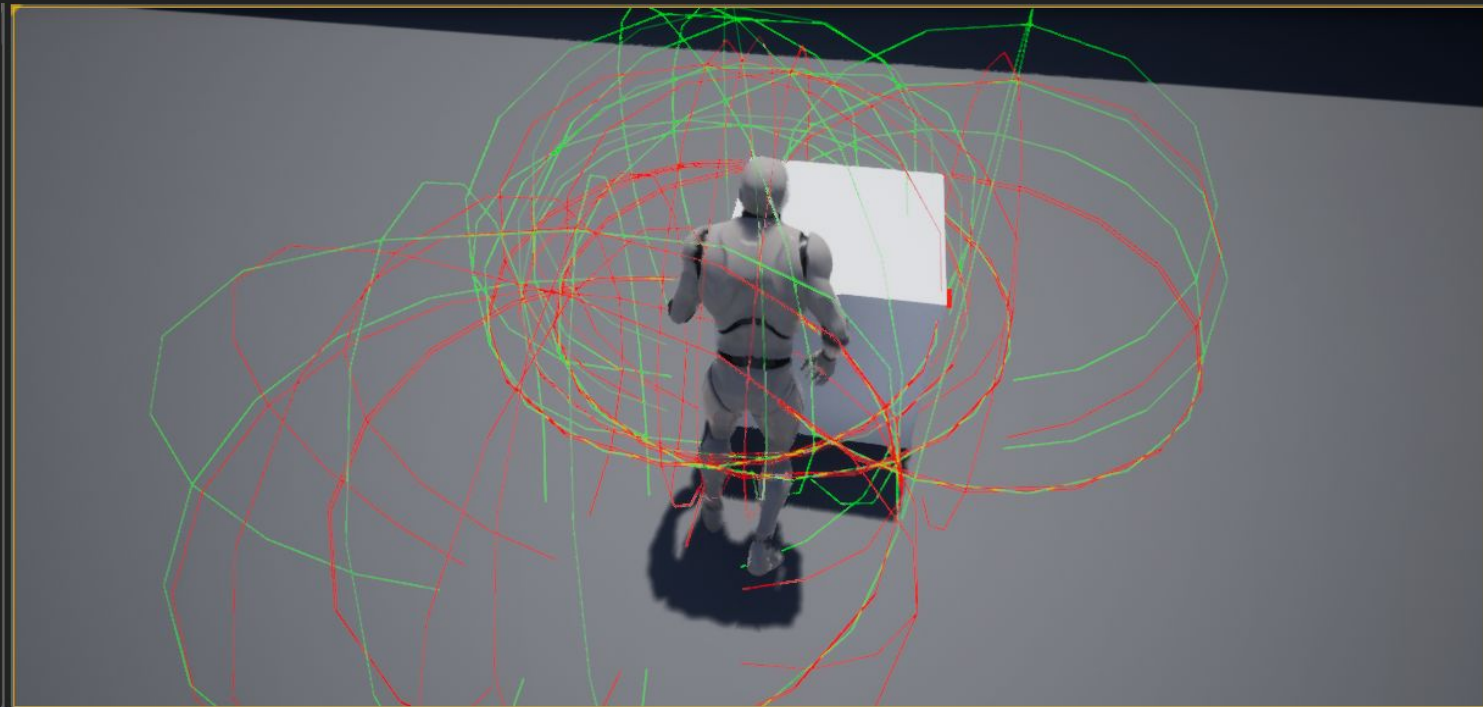
1.2 디버그를 켜준다



1-2. BP_Block 을 화면 위로 올림.



1-2. 박스 옆에서 f 를 열심히 눌러보자.



1-3. 복습

1. 채널은 이렇게 쓰는거임.
2. 유니티 따위와 비교할수 없는 효율적인 충돌 처리.
3. 매우. 효율적.
4. 이해 잘 안갈수 있는데. 하다보면 이해감.

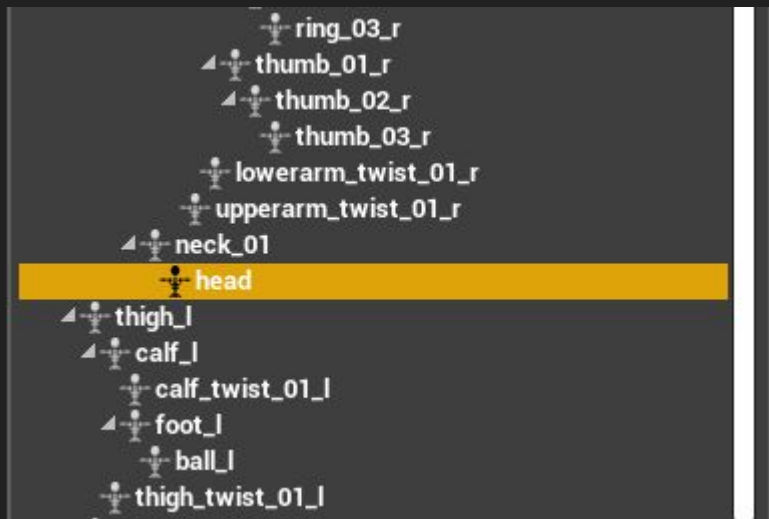
2. 이제 상호작용 만들자.

1. f 누르면 머리에 붙여보자.

2-1.



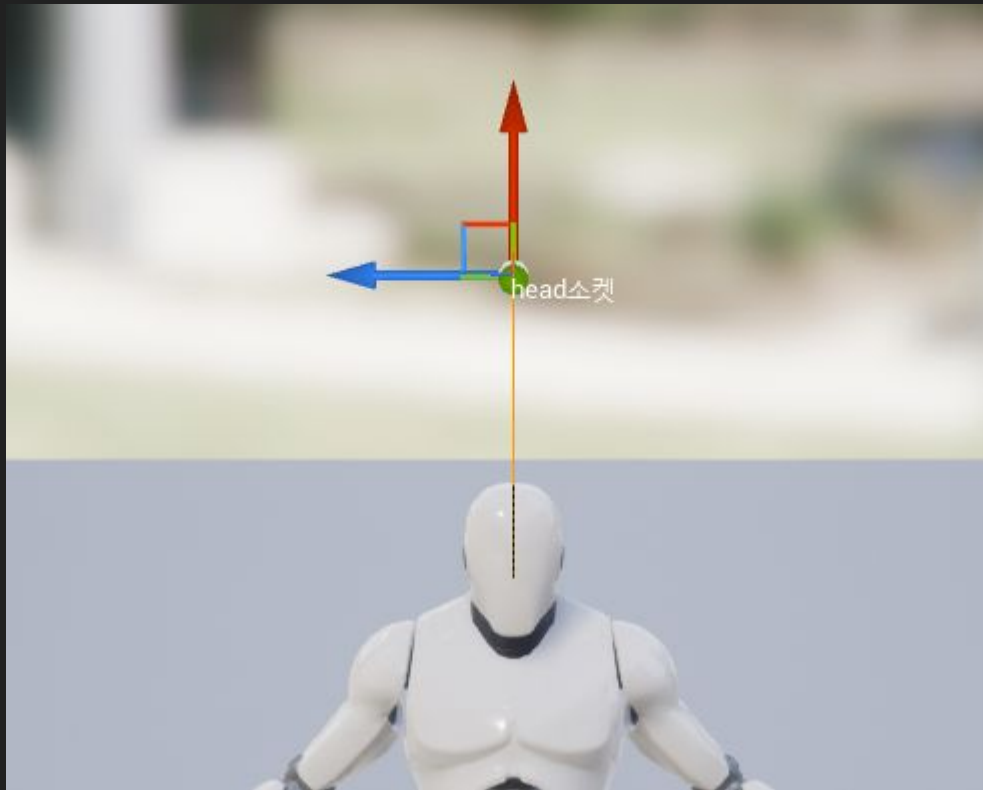
이걸 연다.



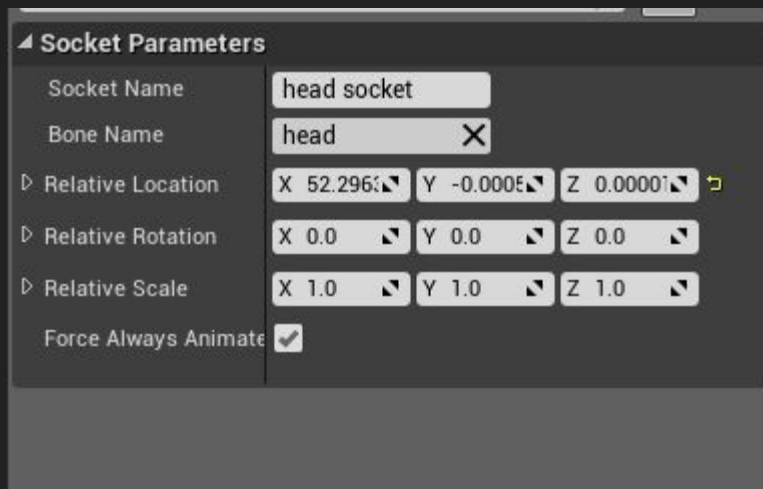
2-1. 소켓(볼을 위치) 추가



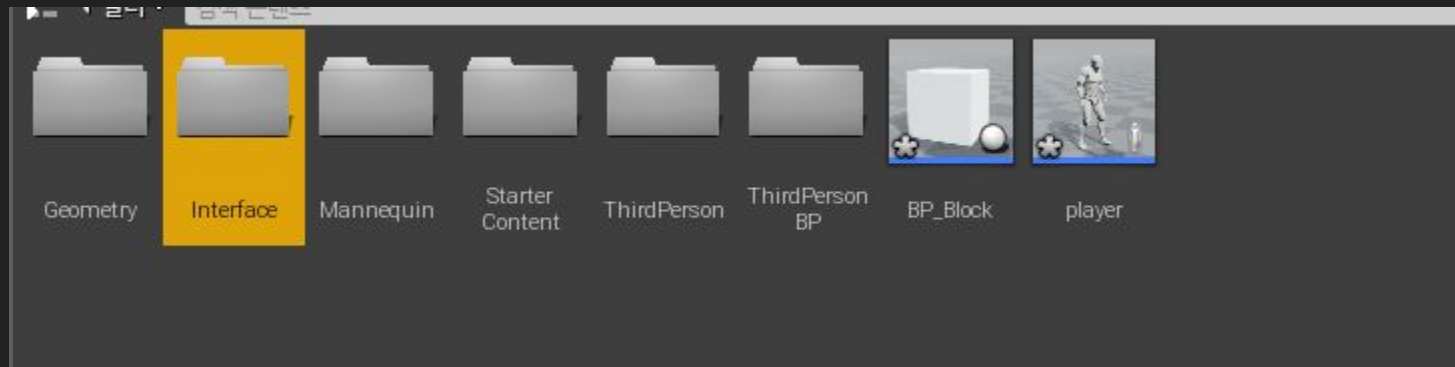
2-1. 소켓 위치 조절 (※ 높이 80 이상 추천)



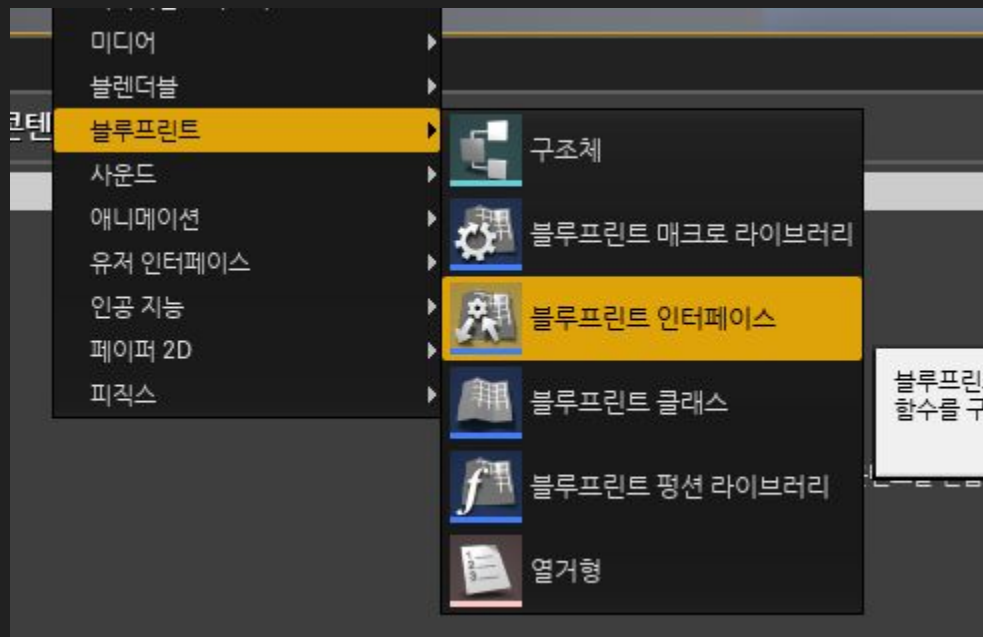
2-1. 소켓 이름 바꾸기



2-2. 인터페이스 폴더 만들기



2-2. 블루프린트 인터페이스 만들기



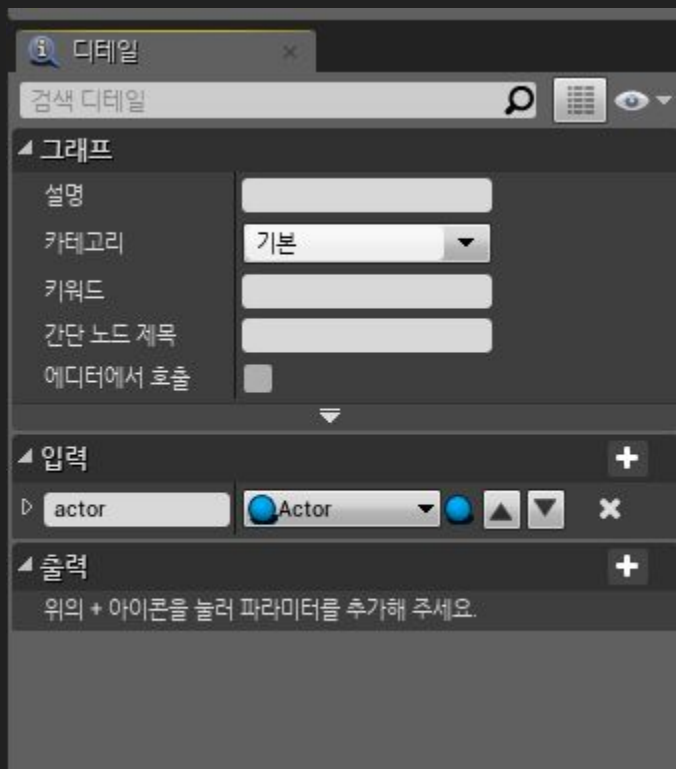
2-2. 총 2개



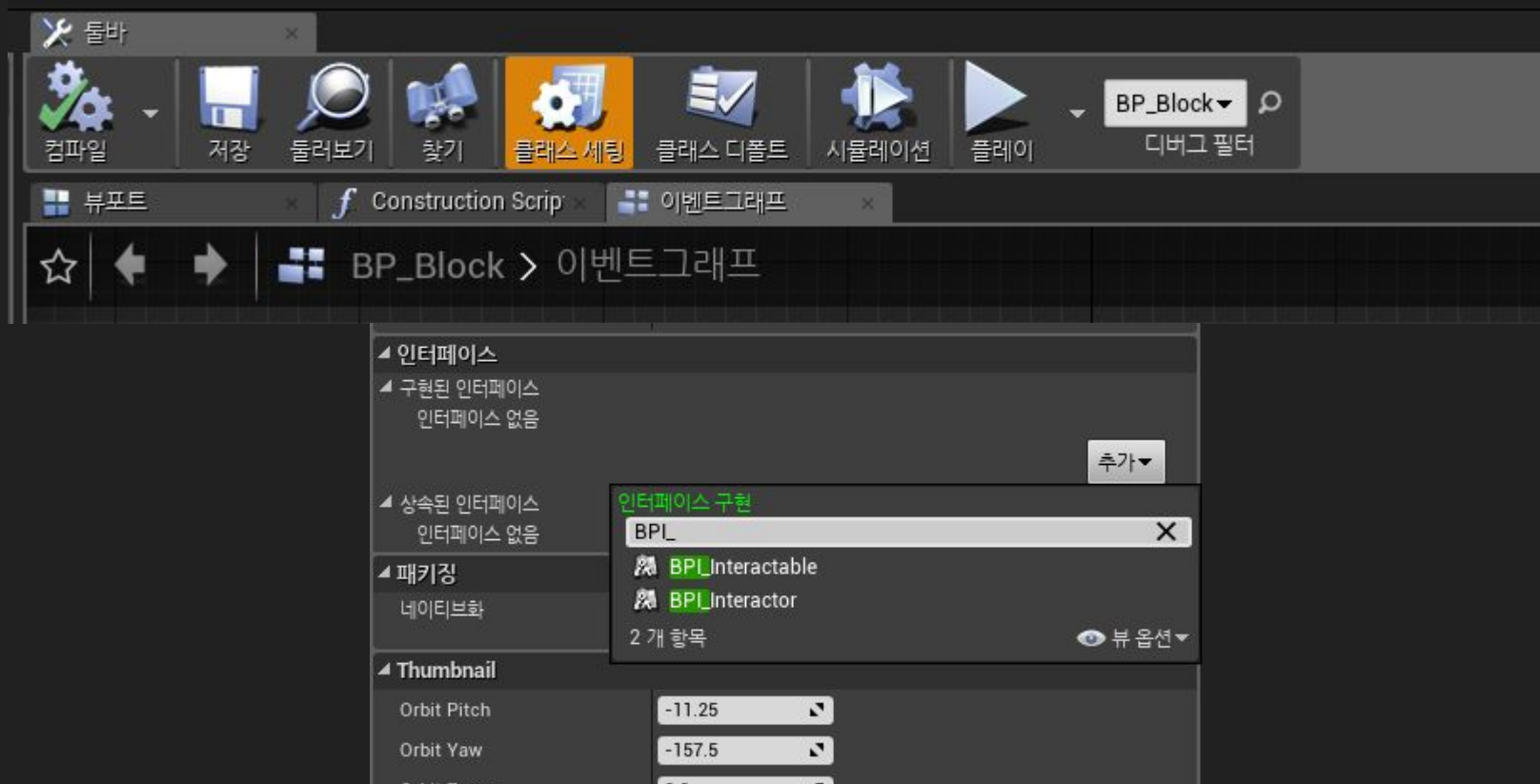
2-2. 둘다 이런 함수를 만들어준다.



2-2. BPI_Interactor 에 함수에는 입력 추가.



2-3. 클래스에 인터페이스 상속



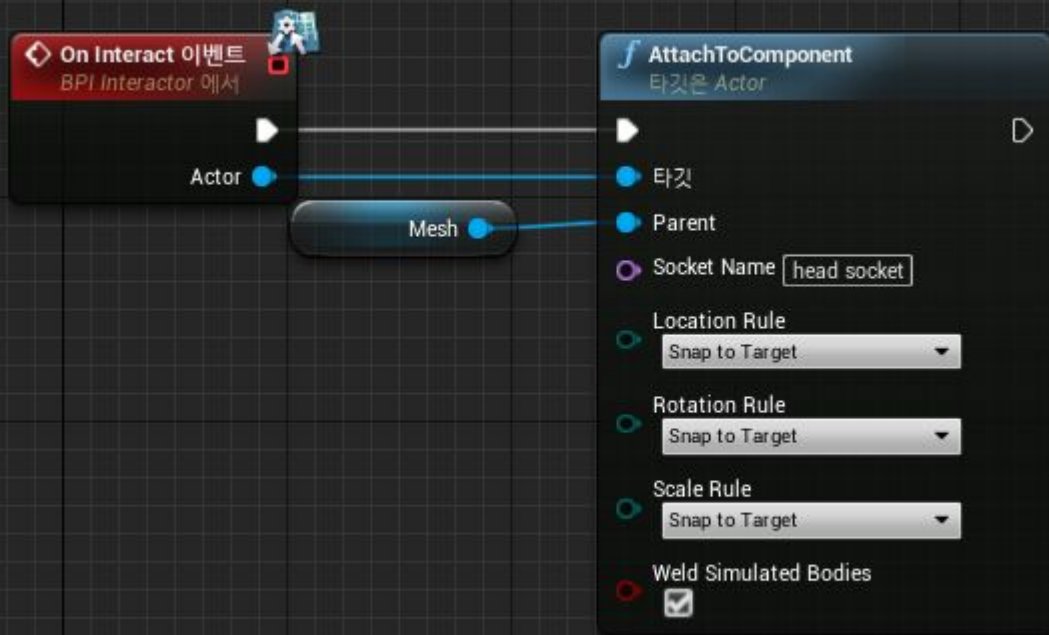
2-3.

1. player 에는 BPI_interactor
2. BP_Block 에는 BPI_interactable
3. 을 각각 상속해준다.

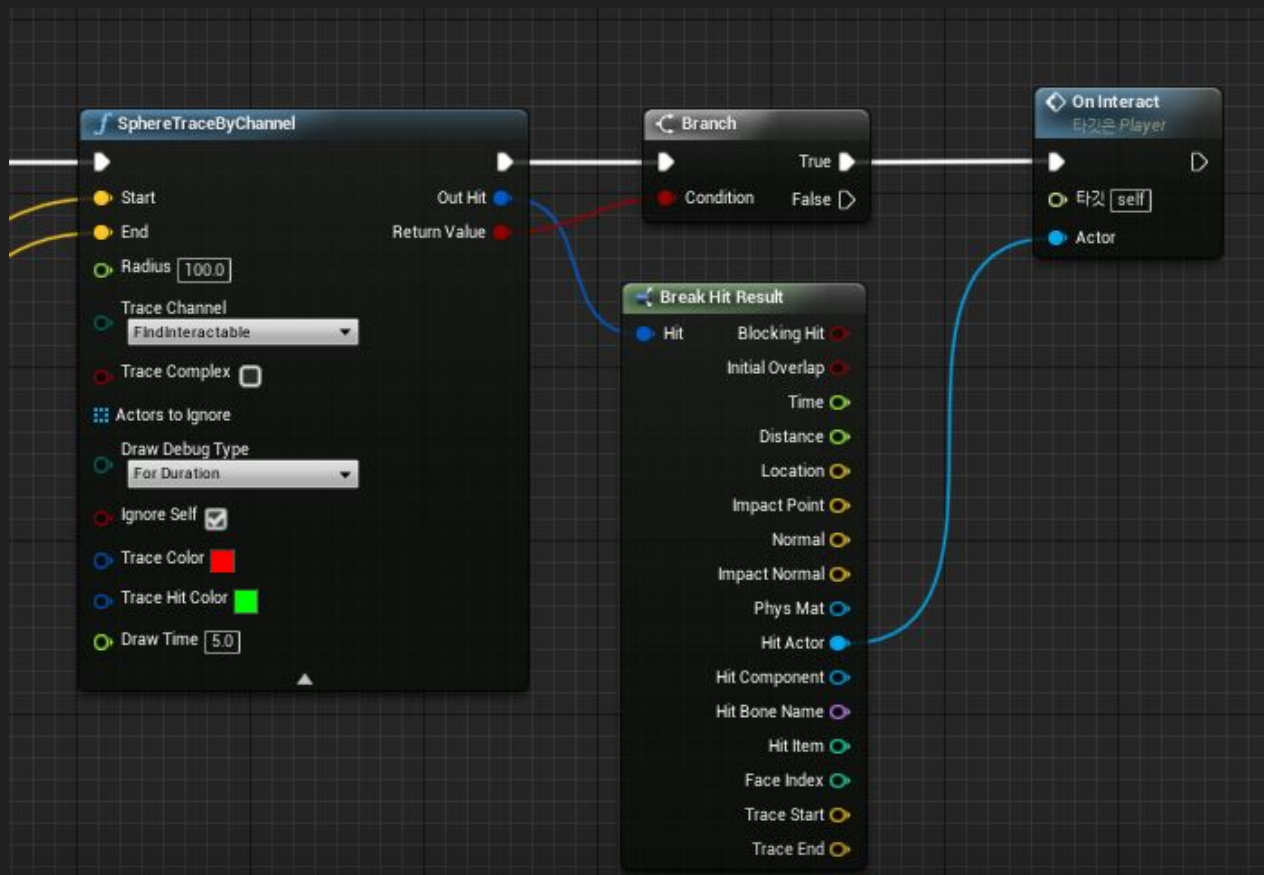
2-3. BP_Block 의 함수 구현.

1. 일단 지금은 할거 없음.

2-3. player이 한스 구현



2-3. player의 함수 추가 구현



2-3. 붙은 모습

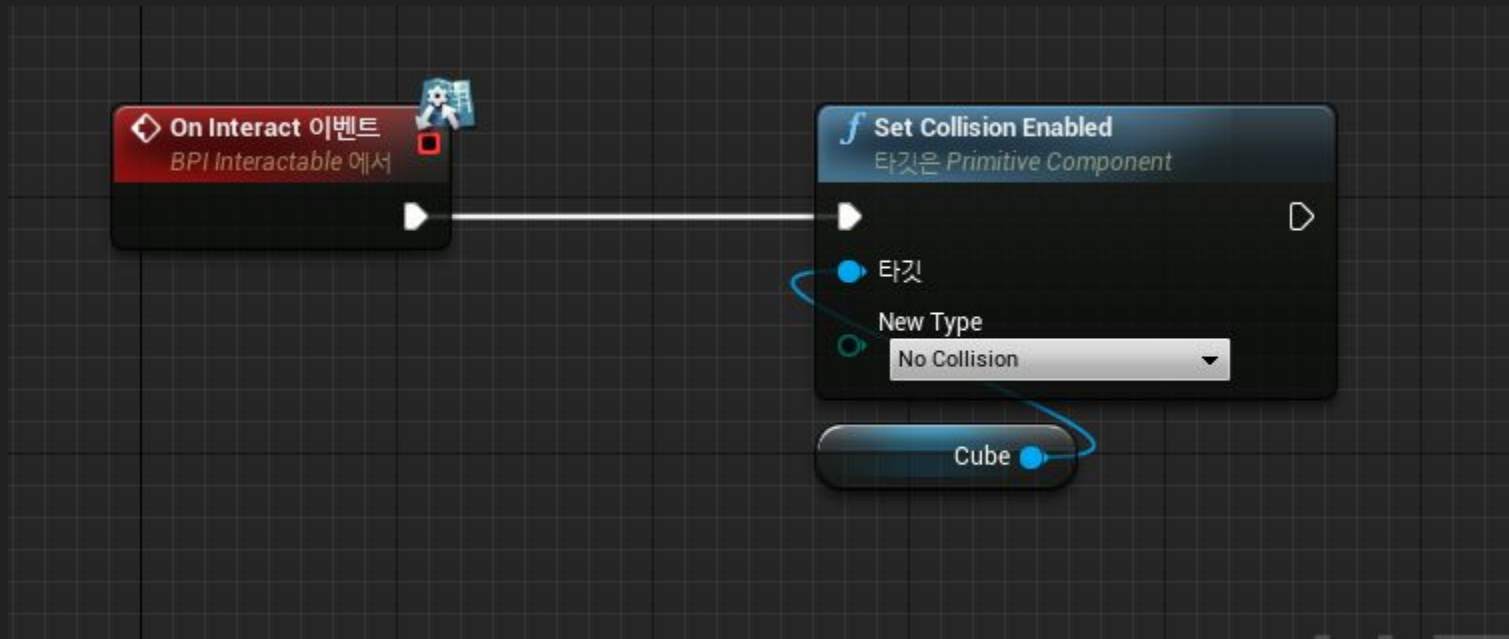


2-3. 만약 움직이지 않을시.

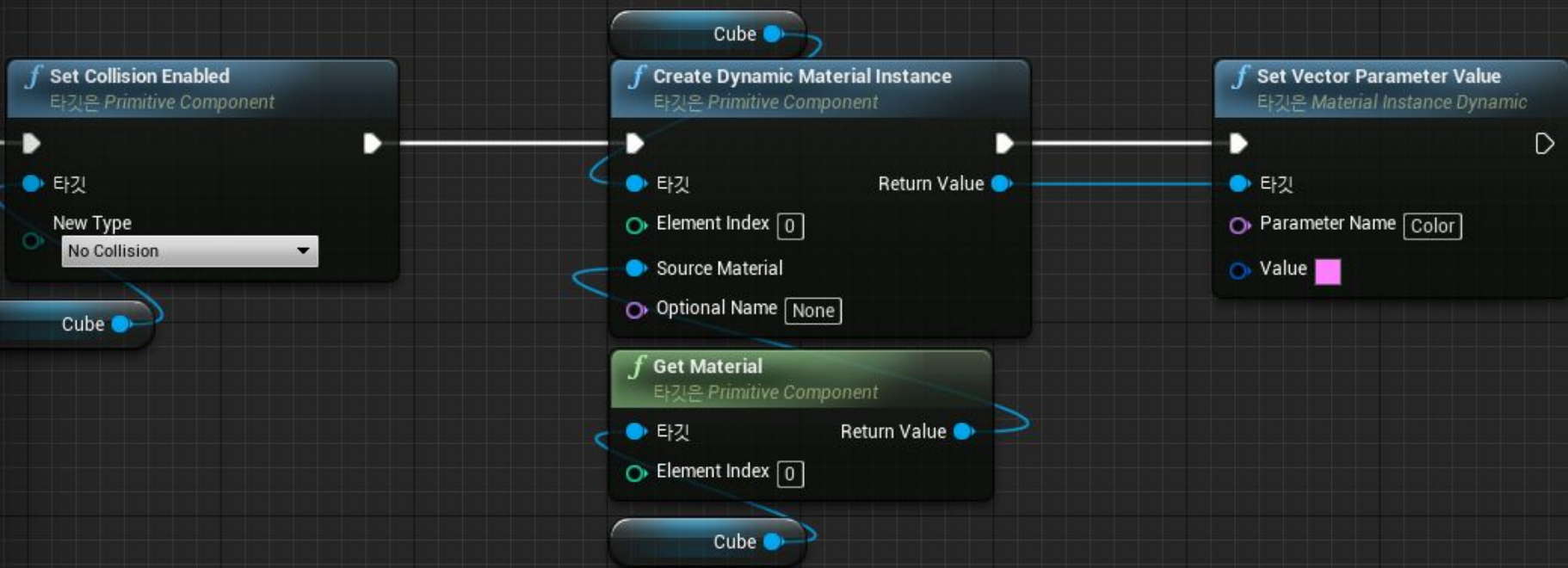
1. 위에 사각형이랑 충돌해서 그런건데.
2. 일단은 소켓 위치를 80 이상으로 바꾸자

2-4. 붙은뒤 충돌 안되게 하고 색 바꿔보자.

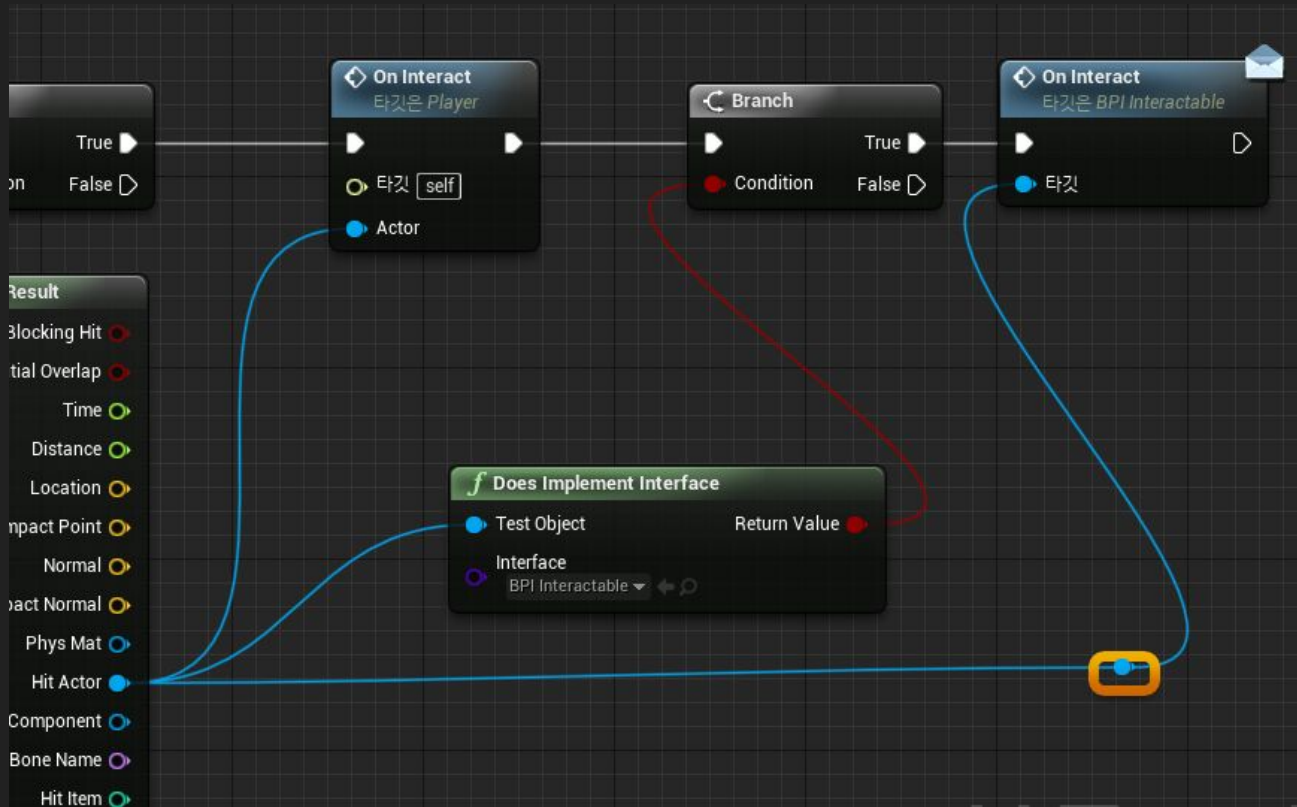
2-4. 일단 충돌 없애는 이벤트 BP_block

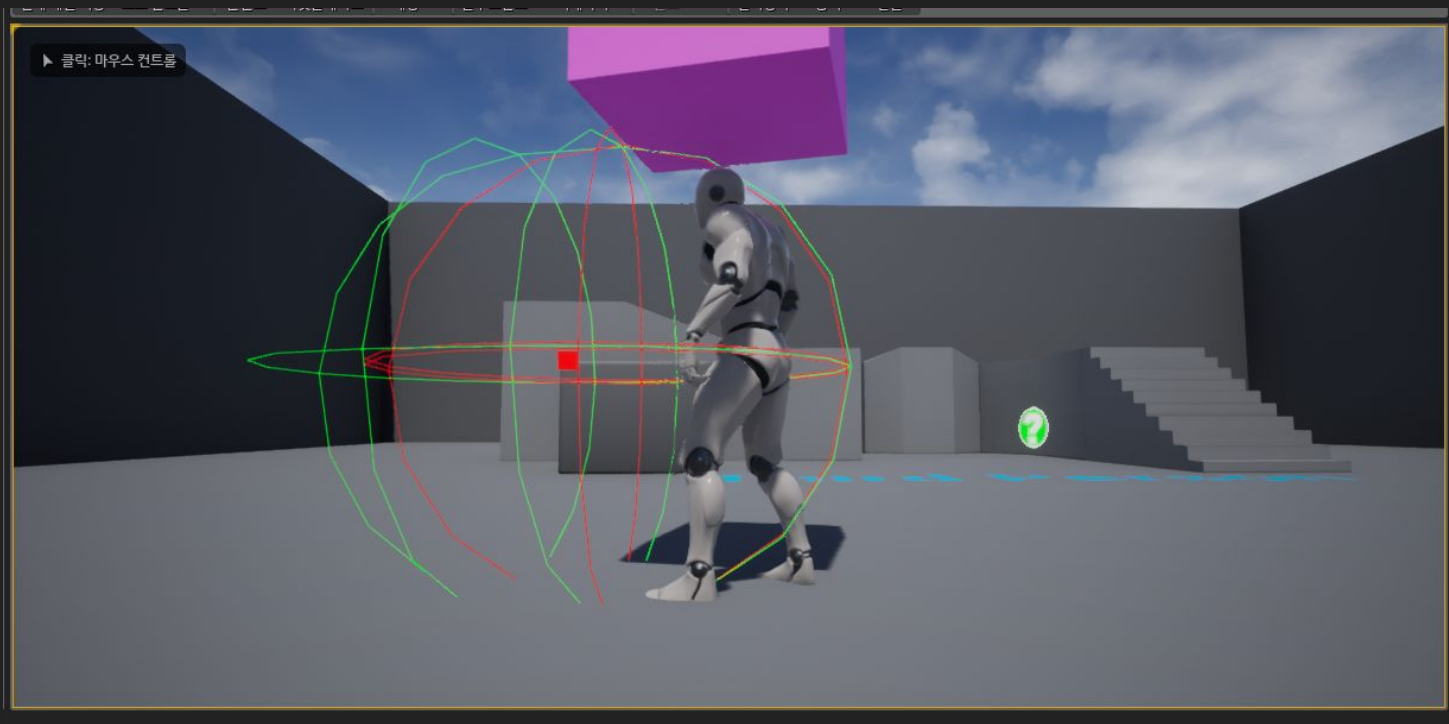


2-4. BP_Block 색 이상한거로 바꿔보자. 좀 복잡.



2-4. player 뒤에 추가





종강 안내

다음주가 마지막일거같음.