게임 엔진

LEC 24 UMG

한국공학대학교 TECH UNIVERSITY OF KOREA 이대현 교수

UMG(Unreal Motion Graphic)

- 언리얼 엔진의 기본 UI 시스템
- HUD, 메뉴, 그래픽 인터페이스 제작에 사용됨.
- 위젯 – UI 구성 요소
  - 버튼, 체크박스, 슬라이더, 프로그레스 바
- 위젯 블루프린트
  - 위젯을 이용하여 UI를 구성하는 핵심 도구

UI 개발 절차

- #1. UI 필요 속성 결정

Score
BulletCount

총탄 개수

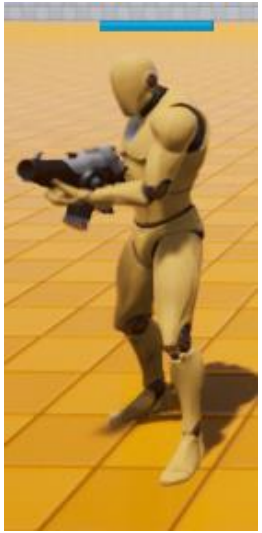
HP

HP

UI 개발 절차

- #2. 레이아웃 제작 - 표현 방식, 위치 등을 결정
-

텍스트

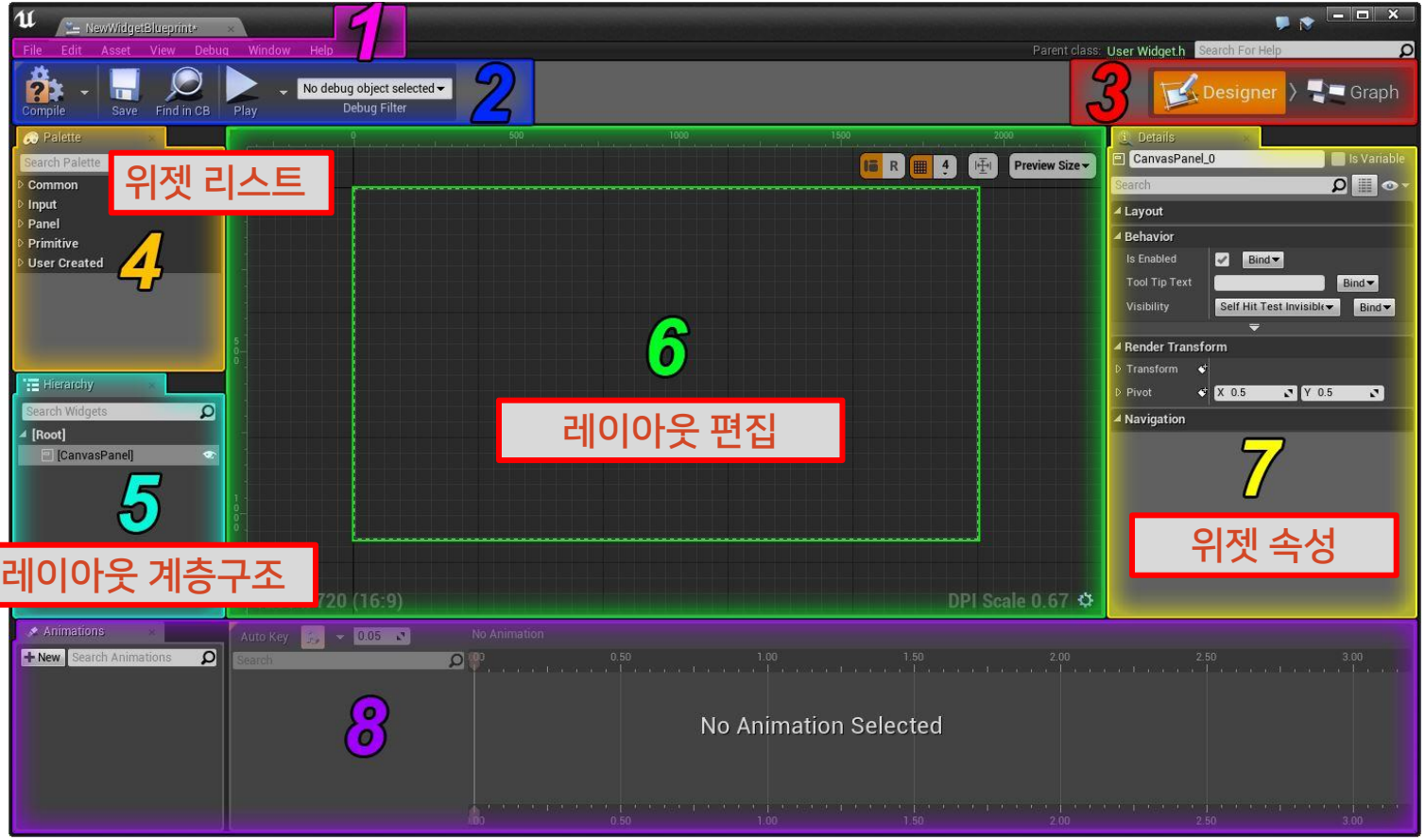


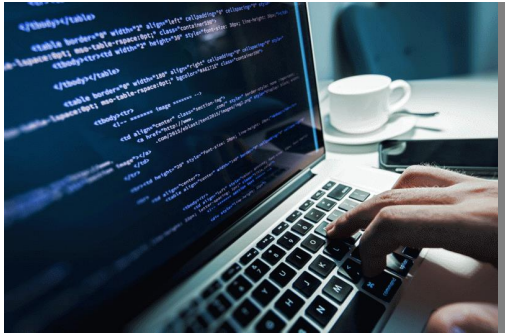
프로그래스 바

UI 개발 절차

- #3. 속성과 위젯의 연결 – 실시간 업데이트 방식 선택, 스크립트 작업
-

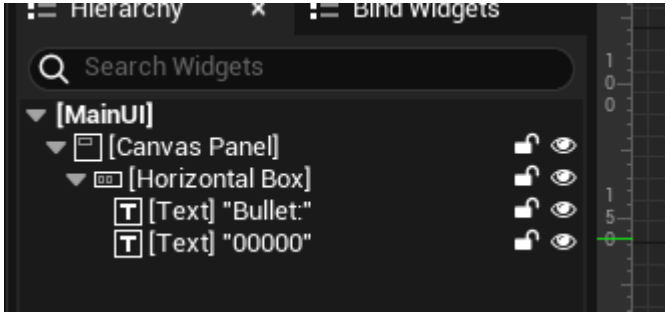
위젯 블루프린트 에디터 – 디자이너 창

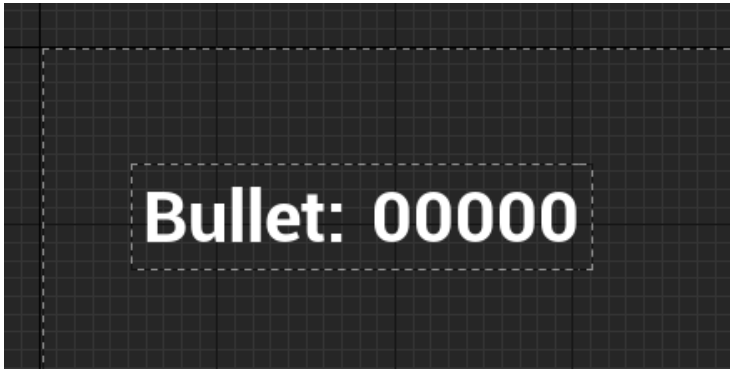


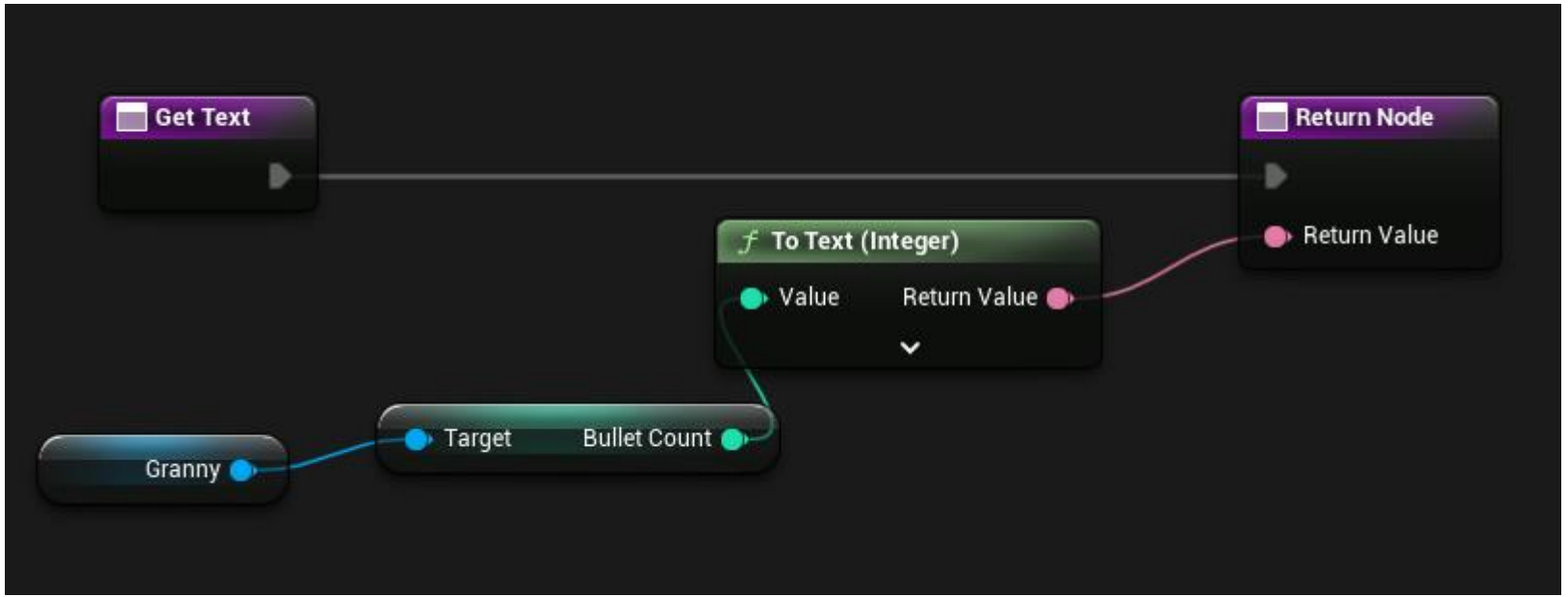


실습

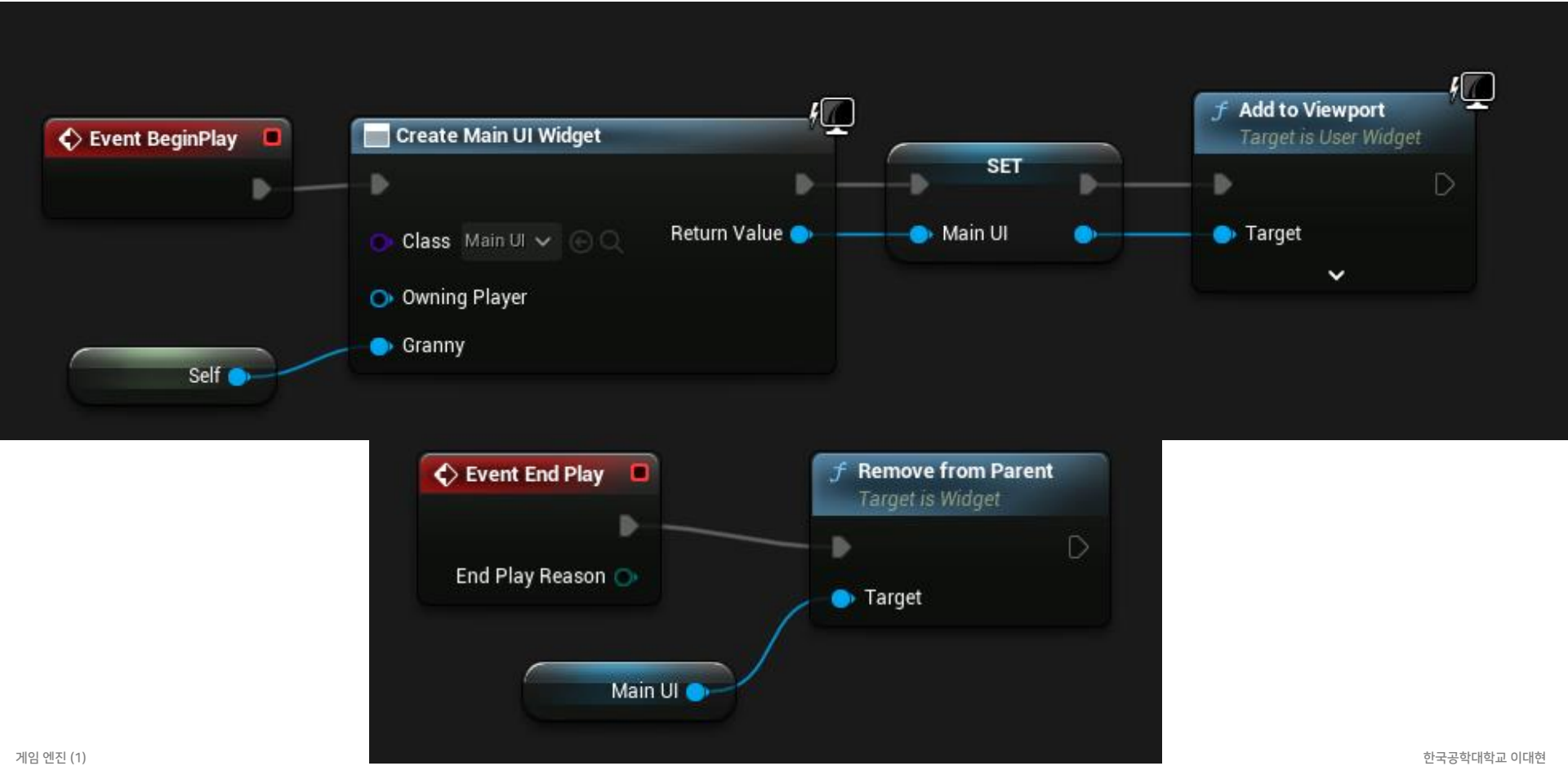
기본 UMG


Hierarchy
Bind Widgets
Search Widgets
[MainUI]
[Canvas Panel]
[Horizontal Box]
[Text] "Bullet:"
[Text] "00000"

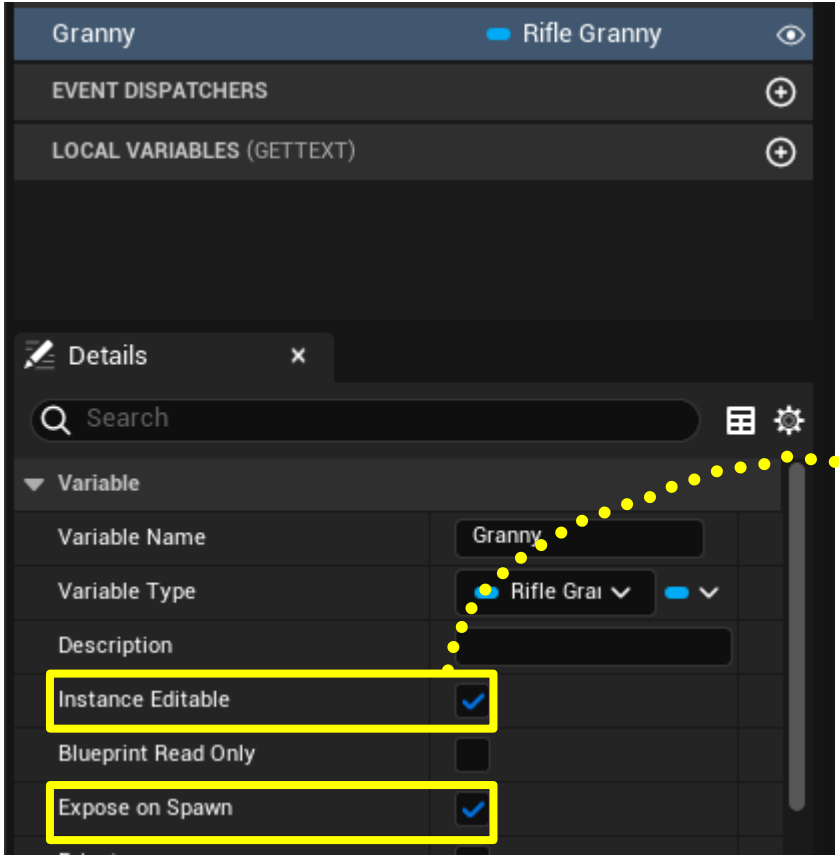

Bullet: 00000

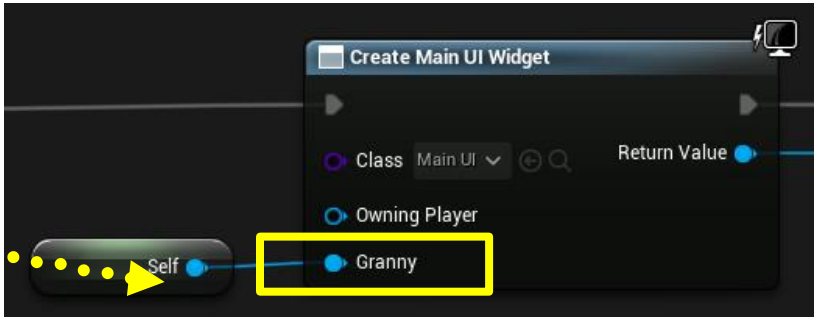

Get Text
Return Node
Return Value
To Text (Integer)
Value
Return Value
Target
Bullet Count
Granny

UI 위젯 생성, 부착 및 제거



위젯 내 액터 레퍼런스의 노출

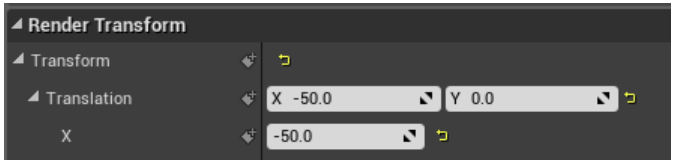


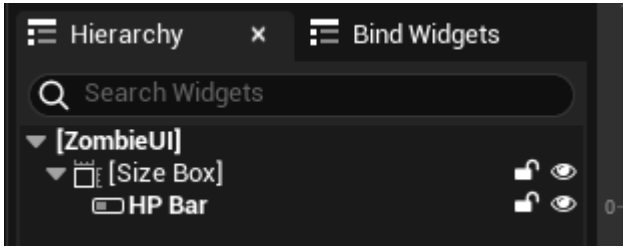


위젯에서 필요로 하는 액터의 레퍼런스를 전달하는 방법

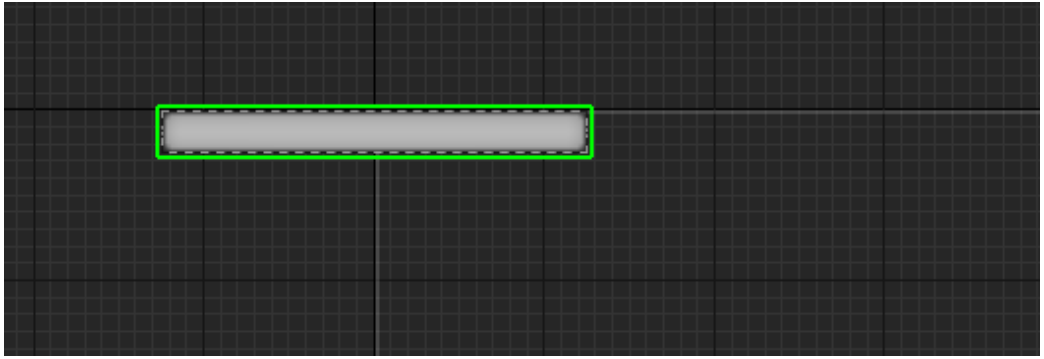
좀비 HP Bar


Child Layout
Width Override
100.0
Height Override
10.0

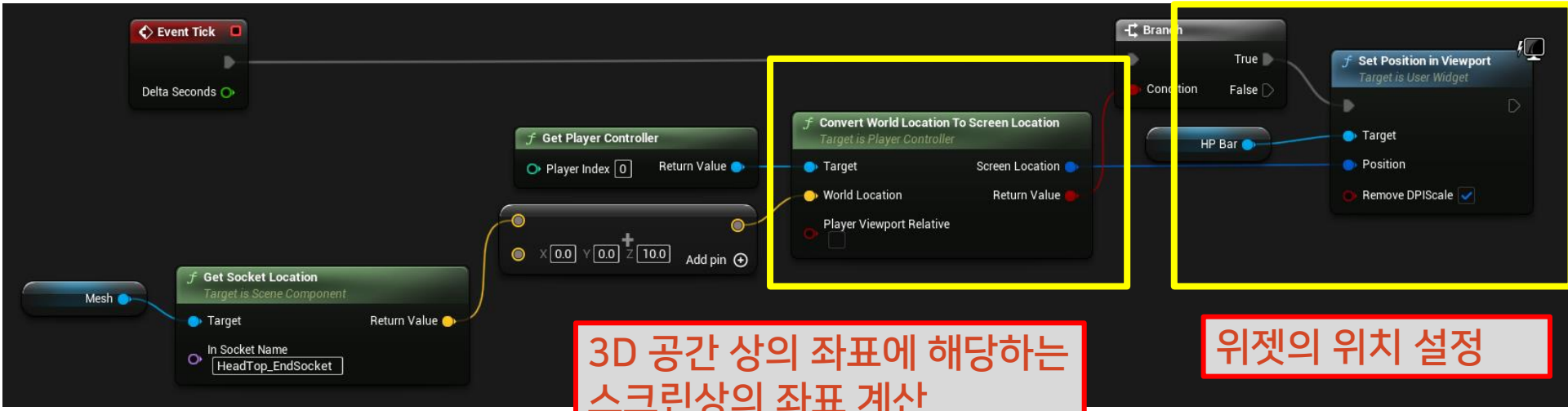

Render Transform
Transform
Translation
X -50.0
Y 0.0
X
-50.0


Hierarchy
Bind Widgets
Search Widgets
[ZombieUI]
[Size Box]
HP Bar


1400
1500
1600
1700
1800
None
R
4
Screen Size
Desired on Screen



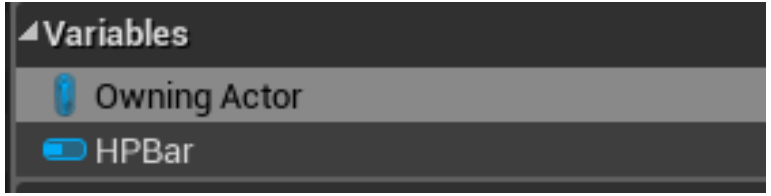
위젯의 위치 계산 및 재설정

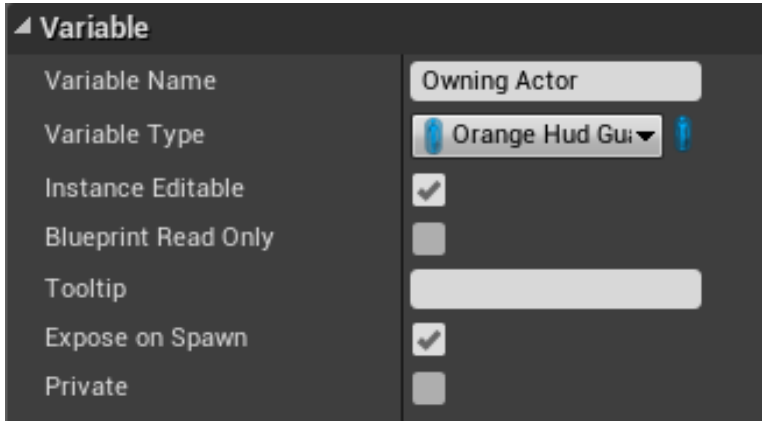


3D 공간 상의 좌표에 해당하는 스크린상의 좌표 계산

위젯의 위치 설정

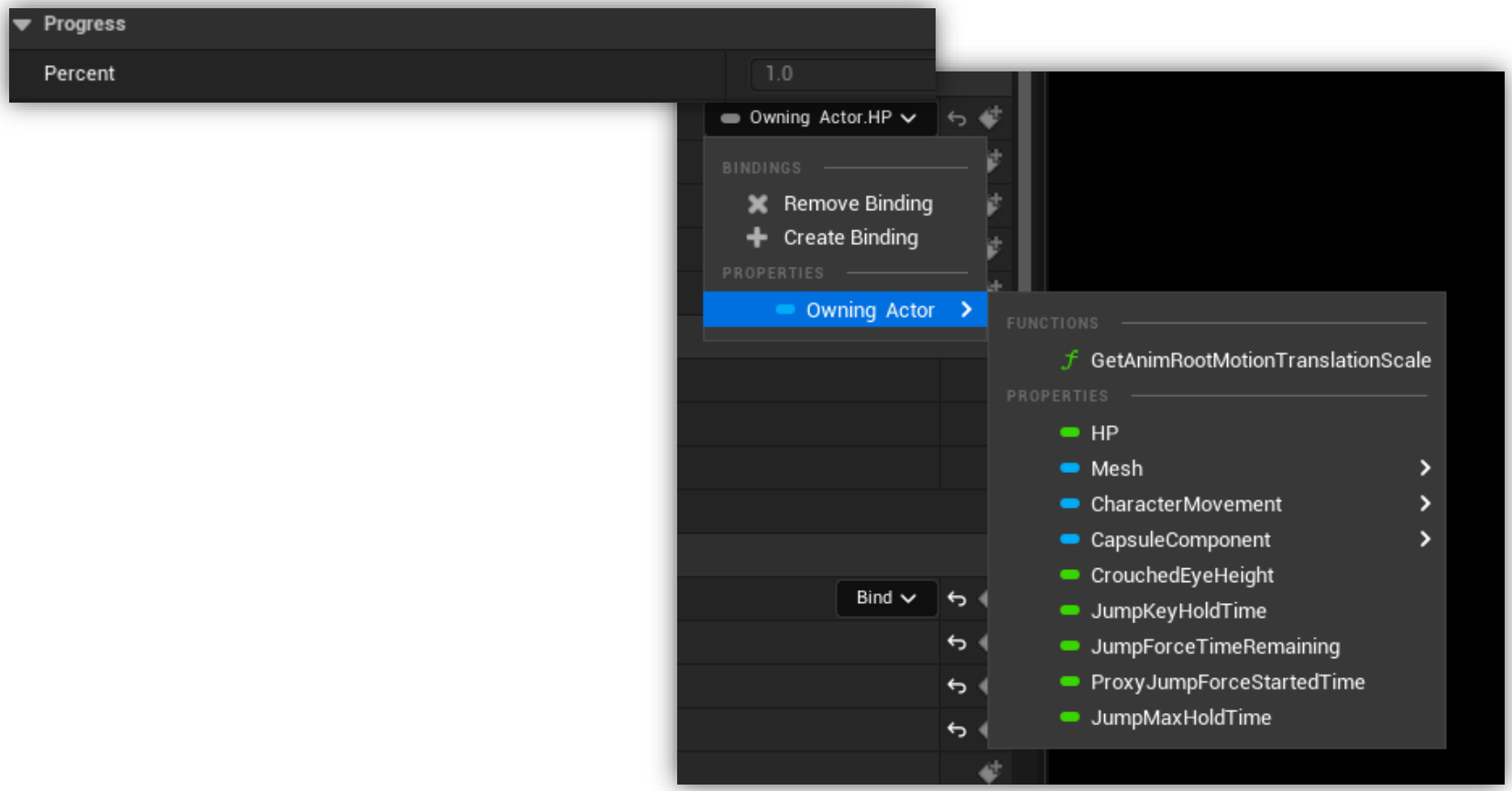
OrangeGuardHud 위젯

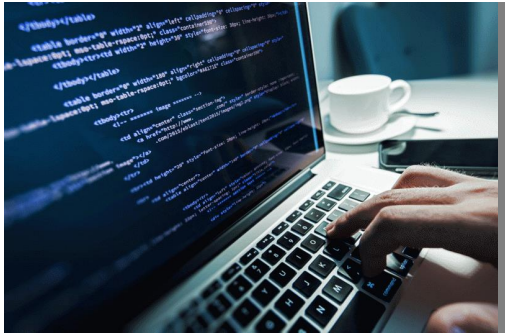




HUD와 연결된 액터의
reference를 외부에서 제공

Progress Bar 바인딩

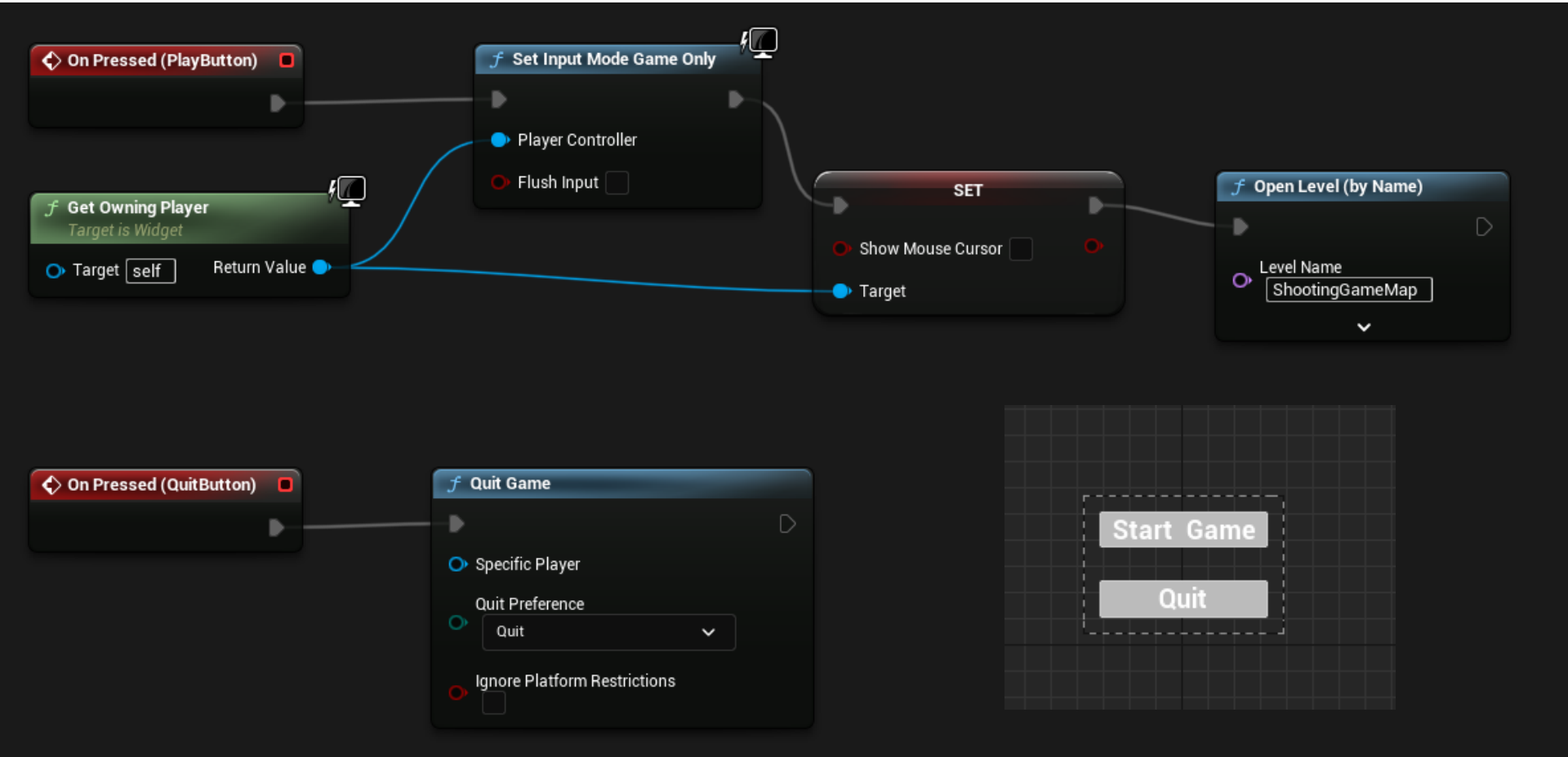




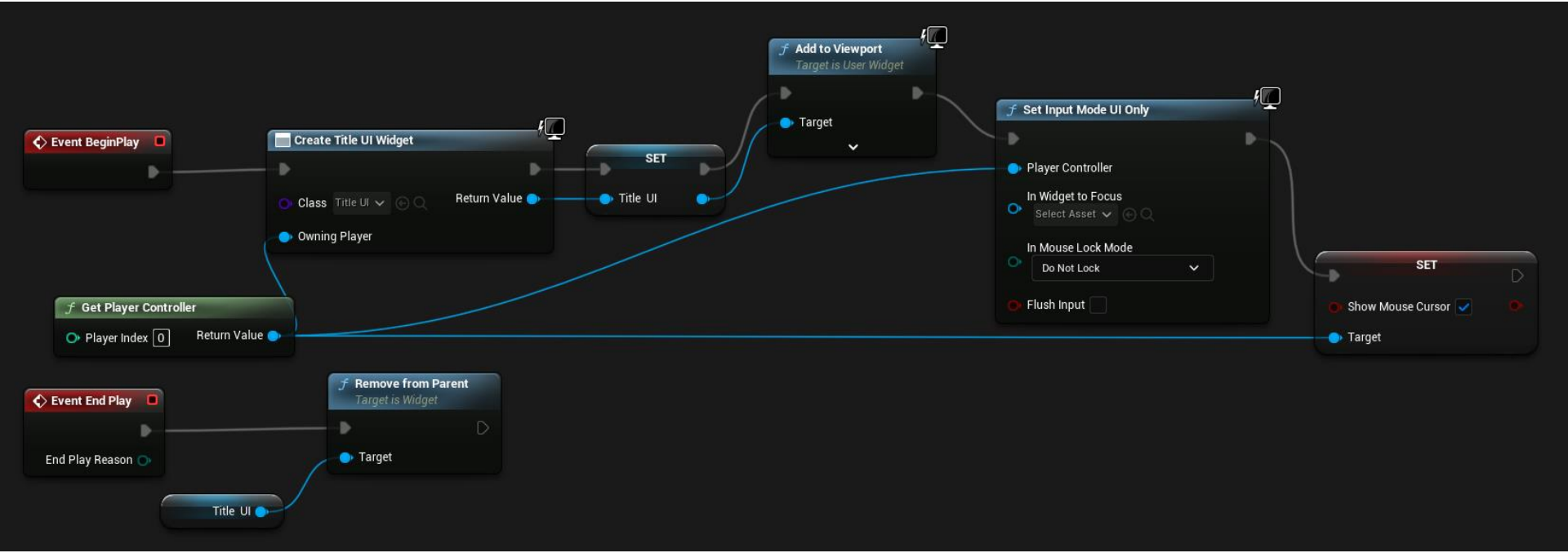
실습

타이틀 UI

Title UI

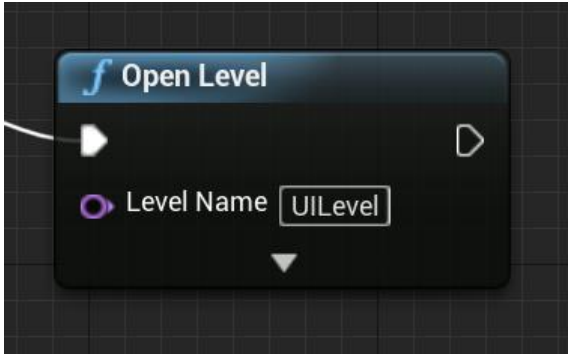

On Pressed (PlayButton)
Set Input Mode Game Only
Player Controller
Flush Input
Get Owning Player
Target is Widget
Target
self
Return Value
SET
Show Mouse Cursor
Target
Open Level (by Name)
Level Name
ShootingGameMap
On Pressed (QuitButton)
Quit Game
Specific Player
Quit Preference
Quit
Ignore Platform Restrictions
Start Game
Quit

Title Level 블루프린트



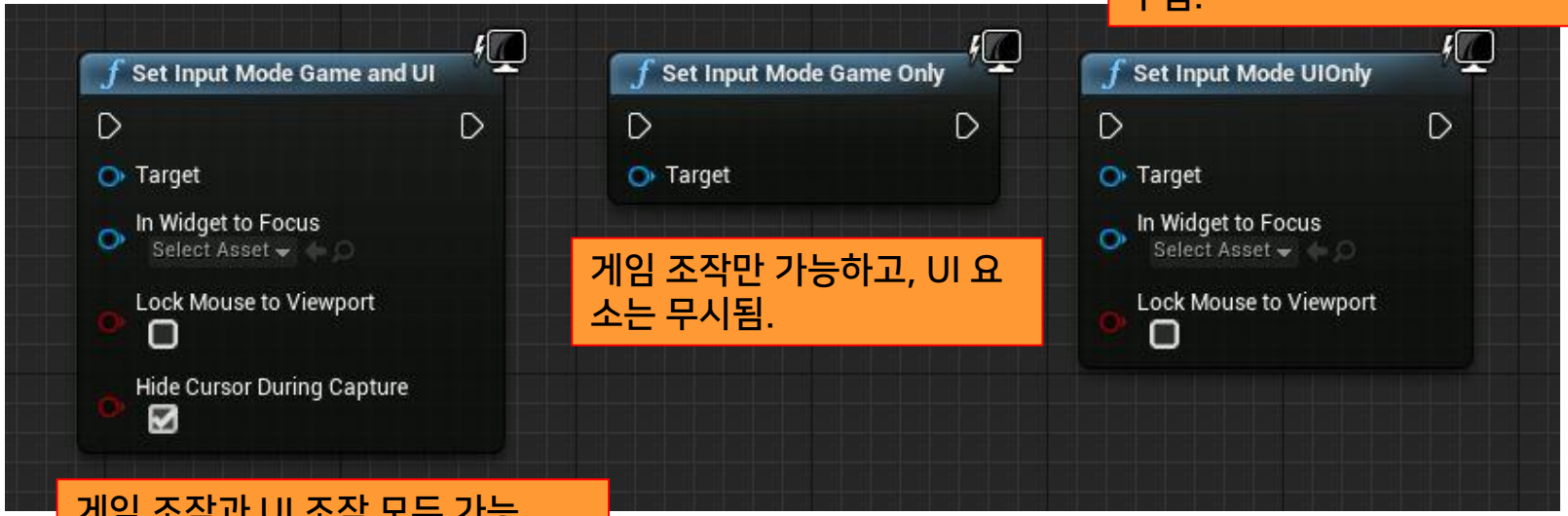
Open Level 노드

- 현재 레벨을 닫은 후, 새 레벨을 열어 실행함.



입력 모드 (Input Mode)

- 게임 컨트롤 및 UI 컨트롤의 사용 여부를 제어함.



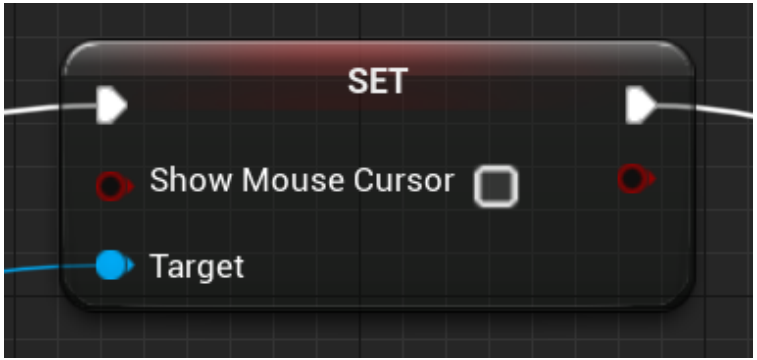
UI 조작만 활성화됨. 신중한 사용이 요구됨.

게임 조작만 가능하고, UI 요소는 무시됨.

게임 조작과 UI 조작 모두 가능

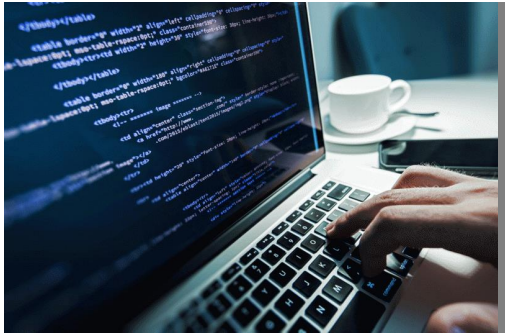
Set Show Mouse Cursor 노드

- 마우스 포인터 표시 여부를 결정



Quit Game 노드

- 게임 실행 중단

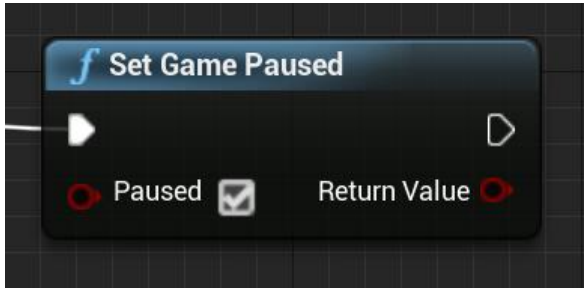


실습

일시정지 UI 와 과녁 UI

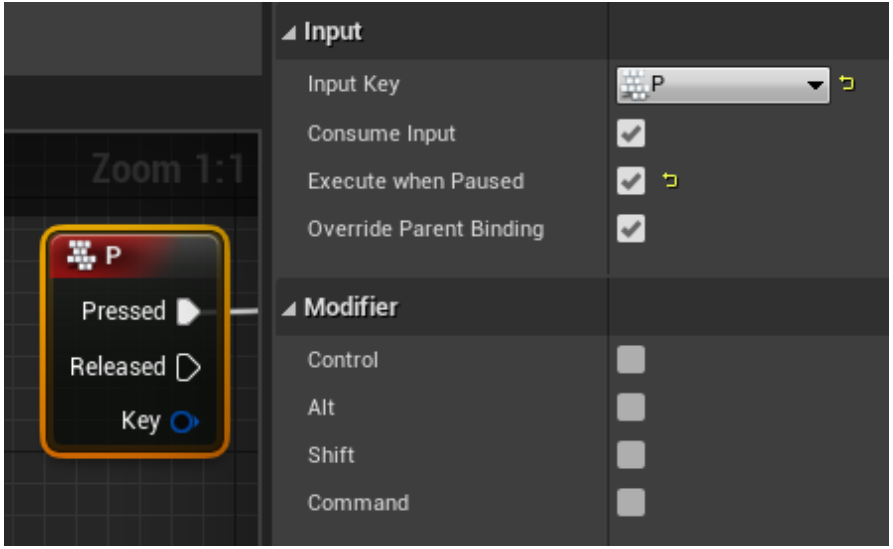
Set Game Paused 노드

- 게임의 일시 정지
- 모든 입력이 막히므로 주의해야 함.

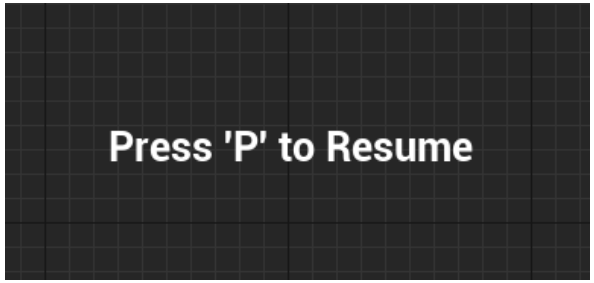


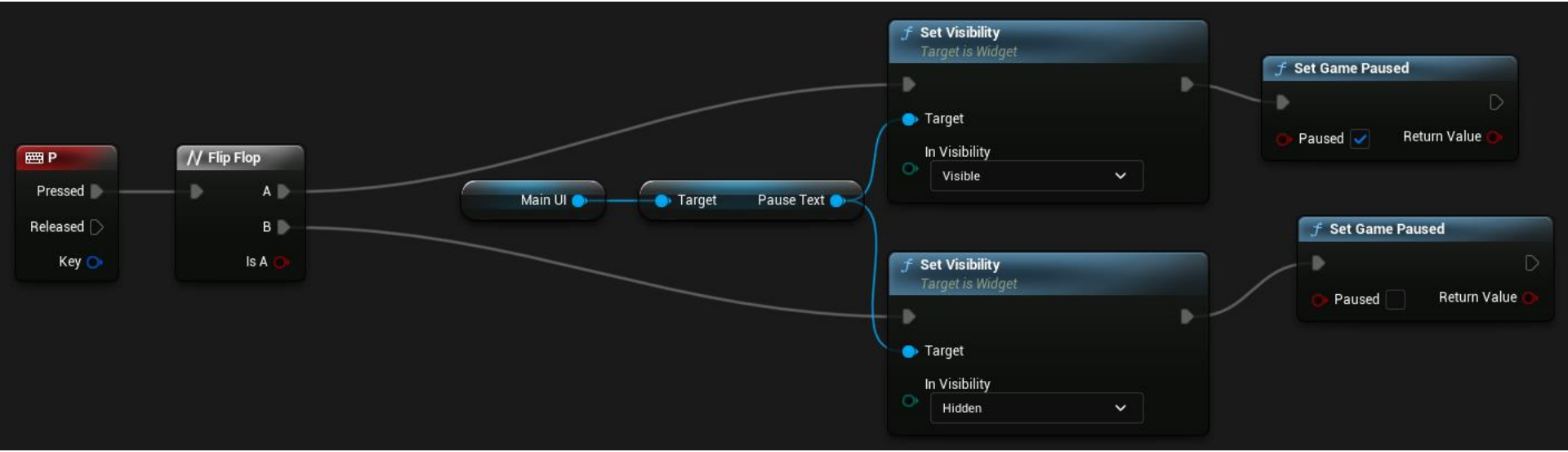
키 이벤트 속성 조정

- 일시정지 상태에서도 입력을 받을 수 있도록 설정이 필요함.

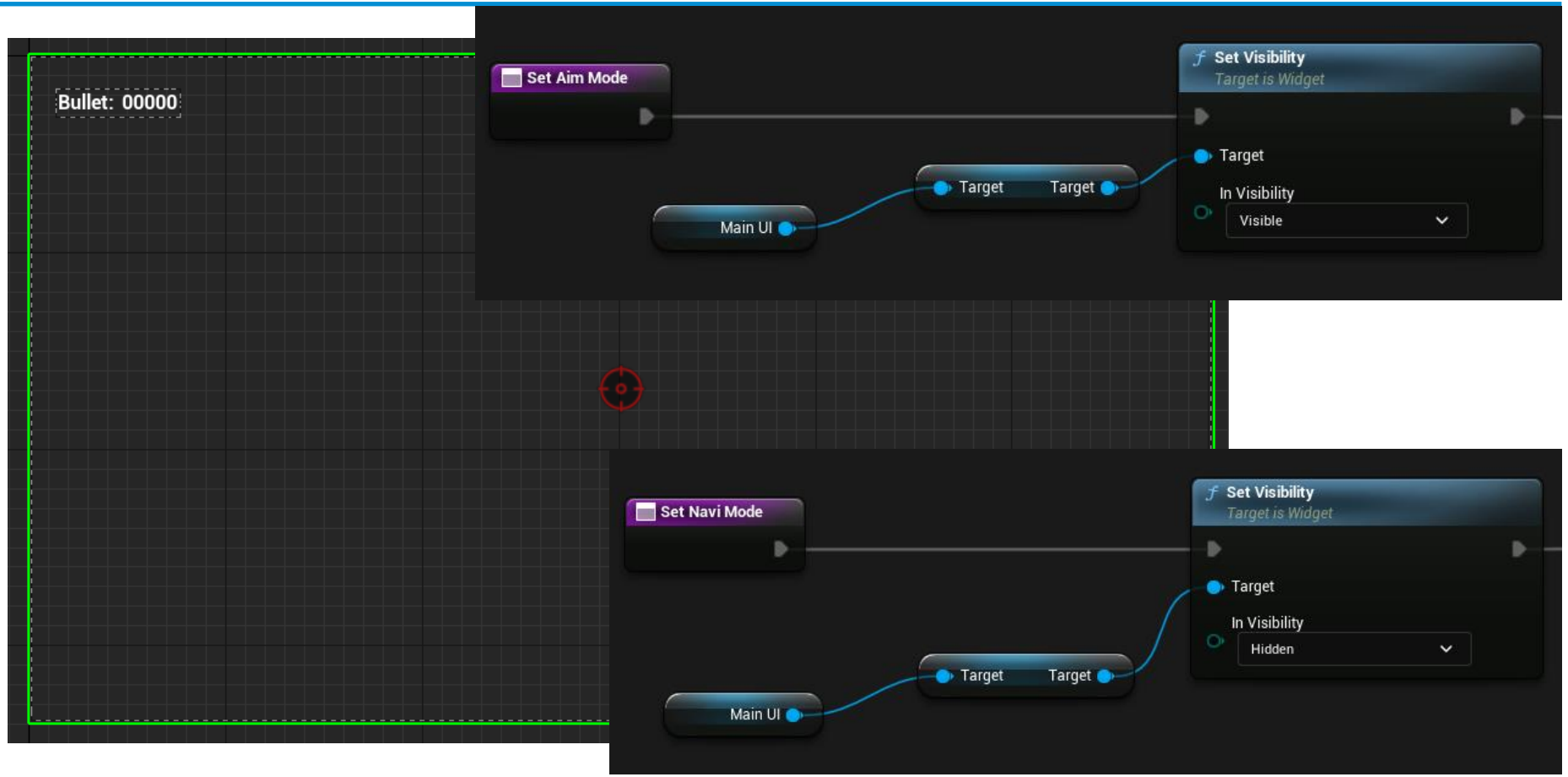


일시 정지 및 재개


Press 'P' to Resume


P
Pressed
Released
Key
Flip Flop
A
B
Is A
Main UI
Target
Pause Text
Set Visibility
Target is Widget
Target
In Visibility
Visible
Set Game Paused
Paused
Return Value
Set Visibility
Target is Widget
Target
In Visibility
Hidden
Set Game Paused
Paused
Return Value

과녁 UI

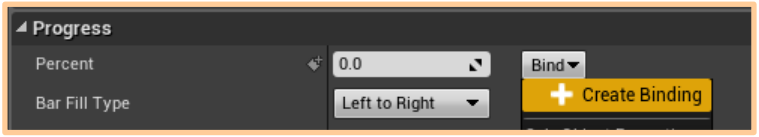

Bullet: 00000
Set Aim Mode
Set Visibility
Target is Widget
Target
In Visibility
Visible
Target
Target
Main UI
Set Navi Mode
Set Visibility
Target is Widget
Target
In Visibility
Hidden
Target
Target
Main UI

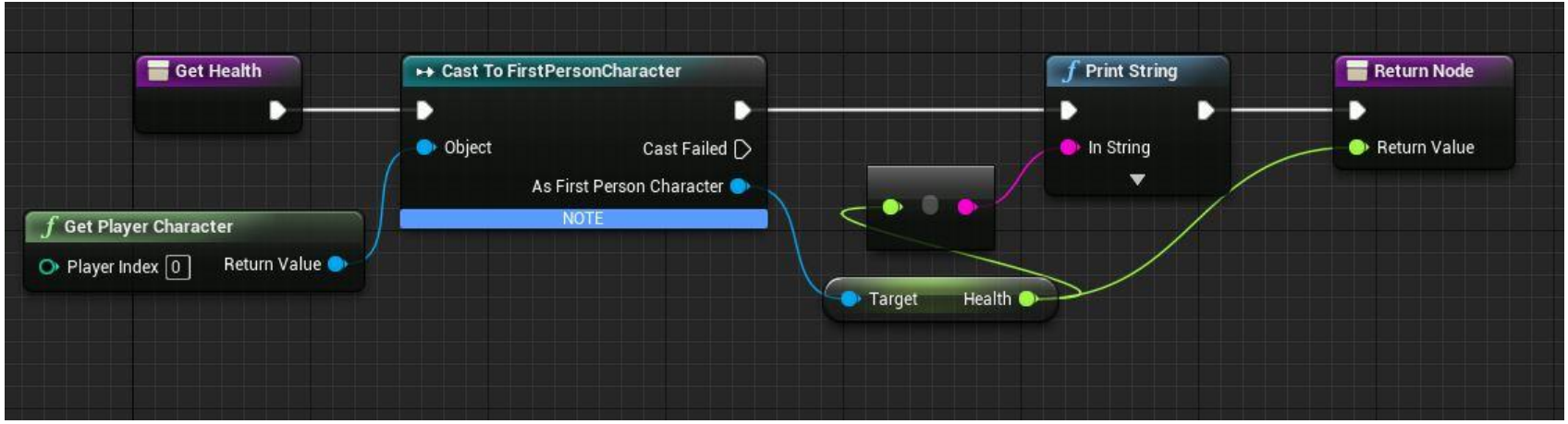
UI 바인딩(Binding)

- 속성값은 변화에 따른 UI 위젯의 업데이트 방식
- 함수 바인딩(Function Binding)
- 속성 바인딩(Property Binding)
- 이벤트 기반 바인딩(Event-Driven Binding)

함수 바인딩

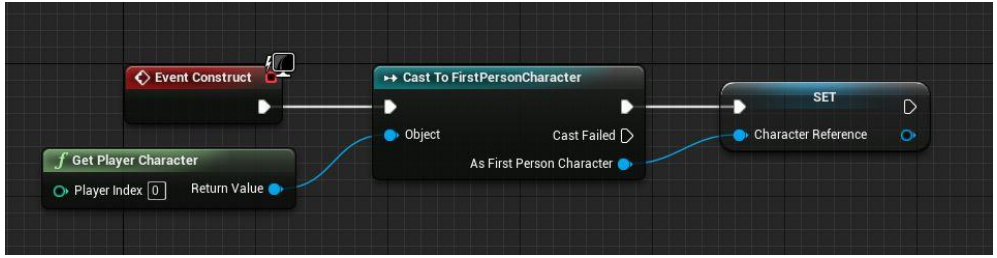
- 바인딩 전용 함수를 생성하고, 여기서 UI 요소의 값을 매프레임마다 업데이트함.
- 간편하지만, 불필요하게 업데이트를 자주 함으로써, 성능 저하의 원인이 됨.

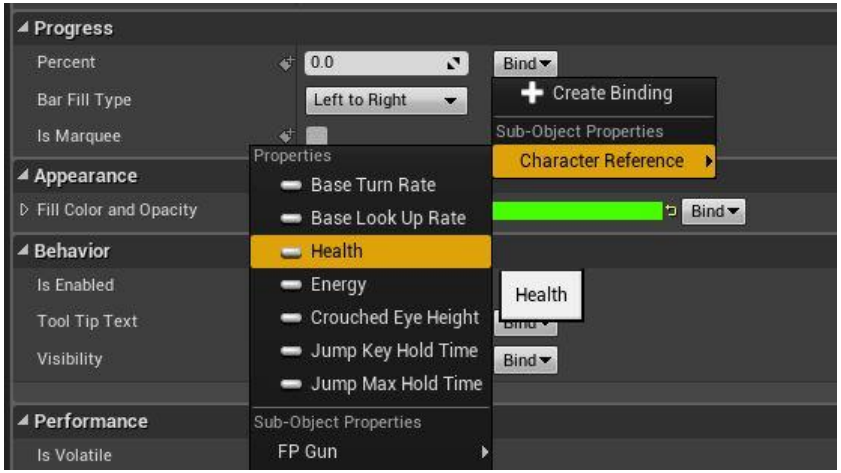




속성 바인딩

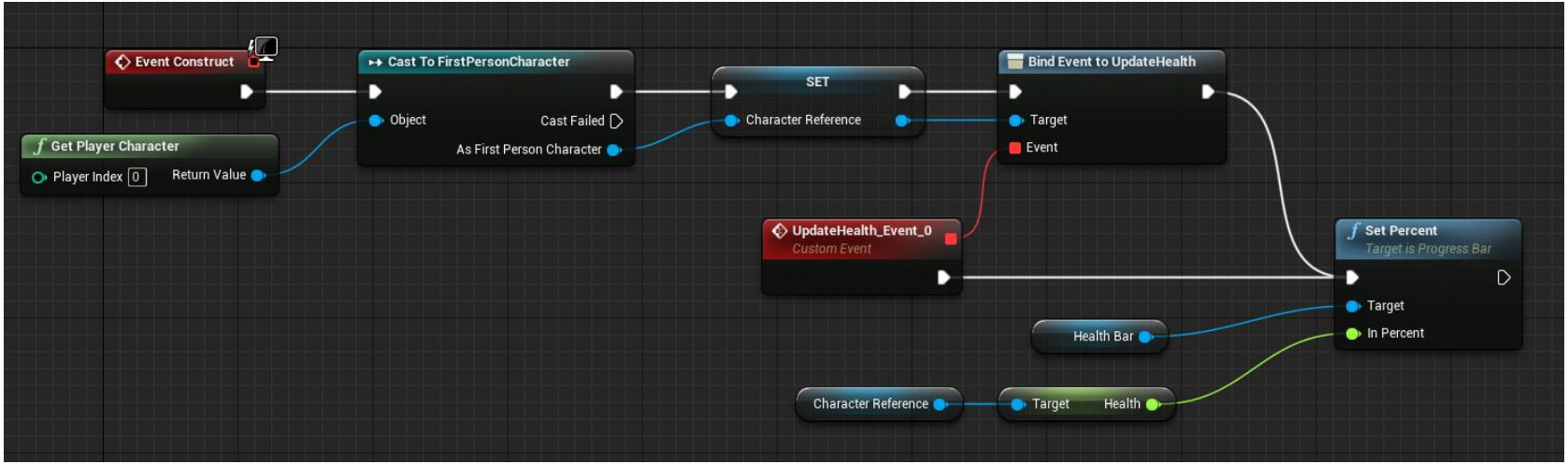
- 미리 액터의 Reference를 확보한 후, 액터의 변수를 직접 UI 요소와 연결.
- 함수 바인딩보다는 효율적임.

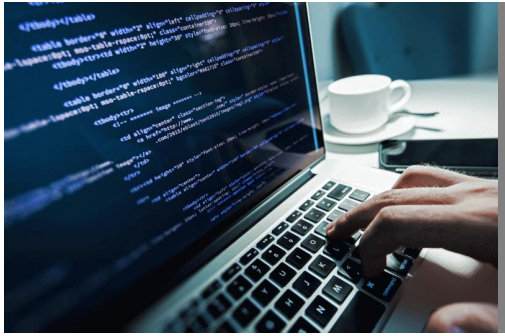




이벤트 기반 바인딩

- 액터에서 변수값이 변화가 있을 때만, UI 요소의 업데이트를 호출함.
- Event Dispatcher 를 이용하면 편리함.
- 가장 효율적임.



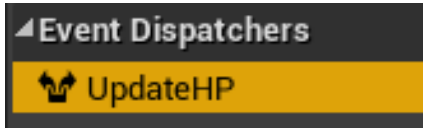


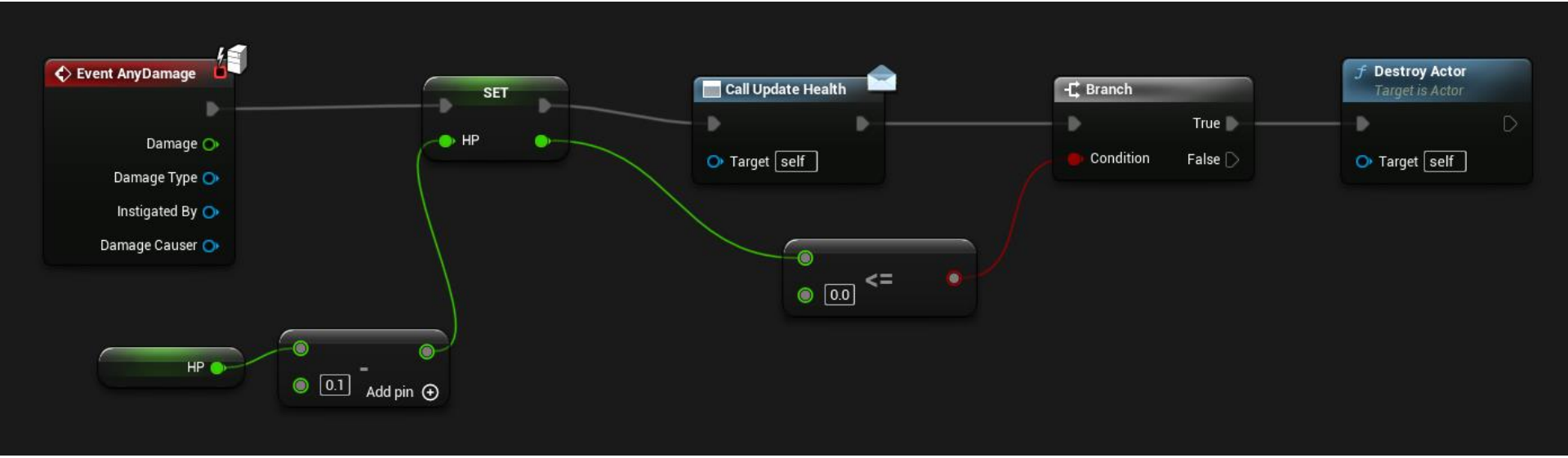
실습

이벤트 기반 바인딩 UI

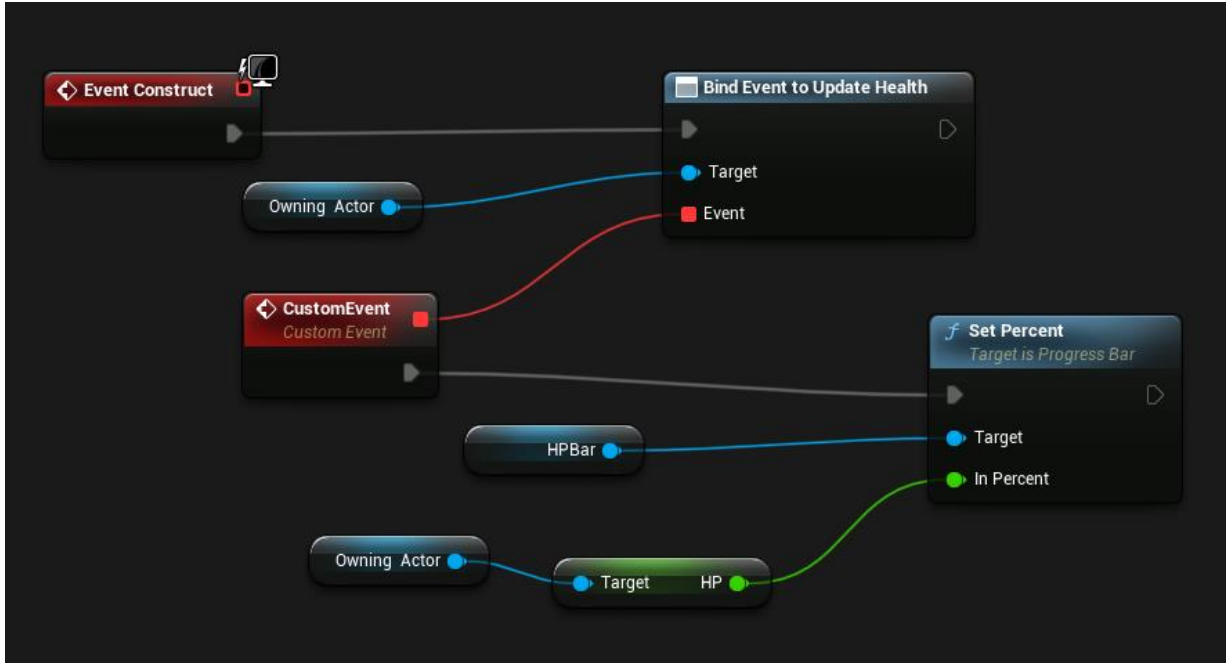
이벤트 디스패처 입력 파라미터

- 이벤트 디스패처를 통해 파라미터를 함께 전달할 수 있음.





Zombie UI


Event Construct
Bind Event to Update Health
Target
Owning Actor
Event
CustomEvent
Custom Event
Set Percent
Target is Progress Bar
HPBar
Target
In Percent
Owning Actor
Target
HP