

목차

1. BP_ChannelMsg 처리 1
2. BP_BoardBase 처리 1
3. BP_ChannelMsg 처리 2
4. SaveGameStats 처리
5. BP_ChannelMsg 처리 3

목차

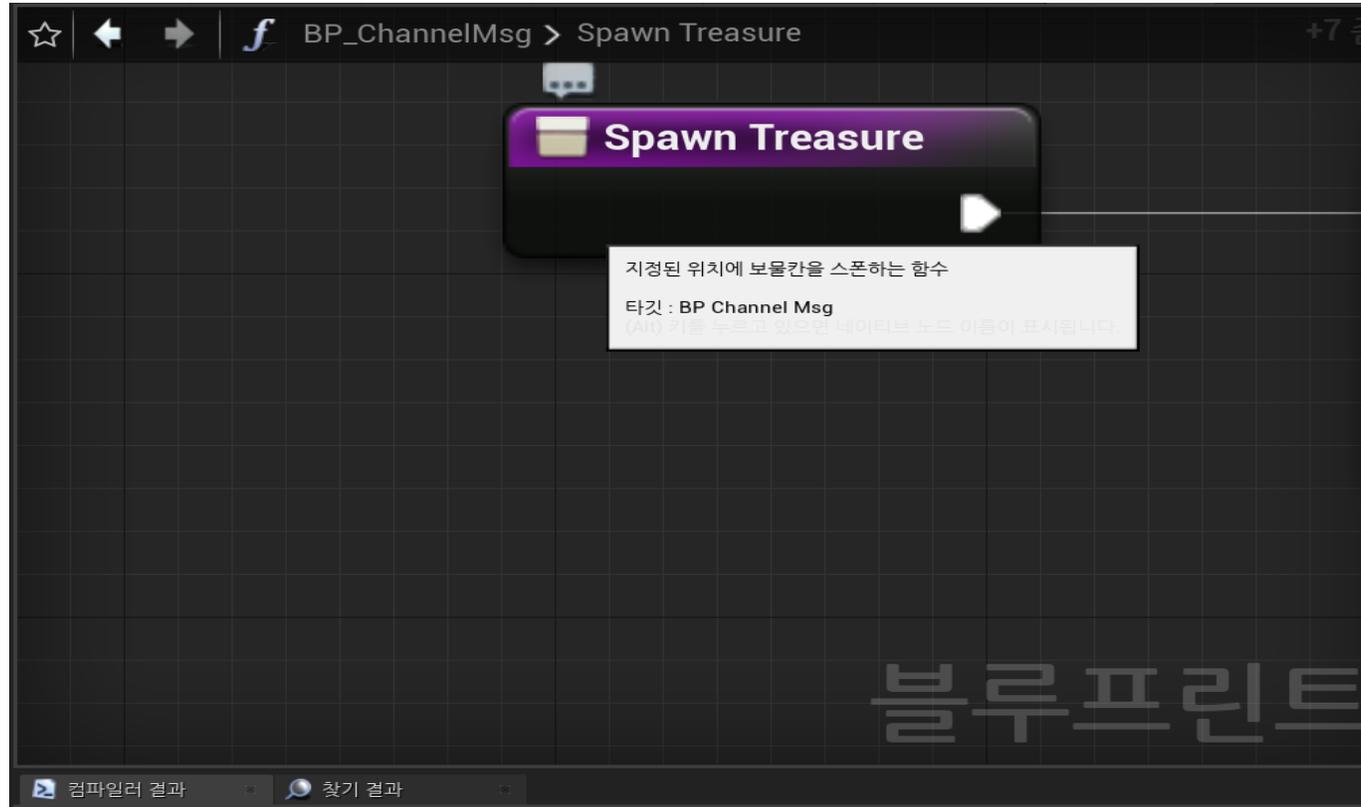
6. BP_BoardBase 처리 2
7. 위젯 블루프린트 BoardHeadUpDisplay를 생성한다.
8. PartyWidget_Slot 처리
9. PartyWidget 처리
10. BattlePlayController 처리

목차

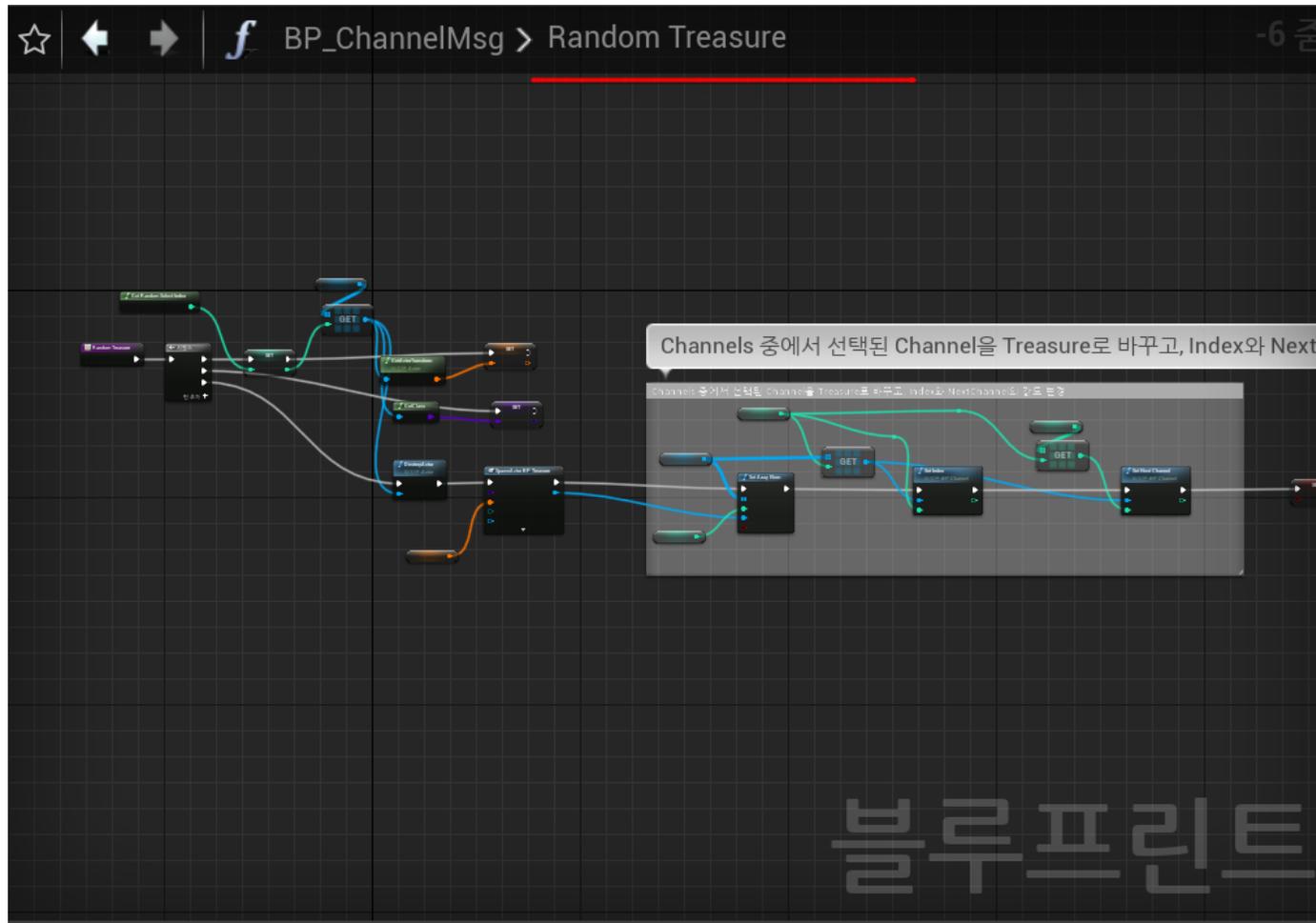
11. BattleGameMode 처리

12. BP_Treasure 처리

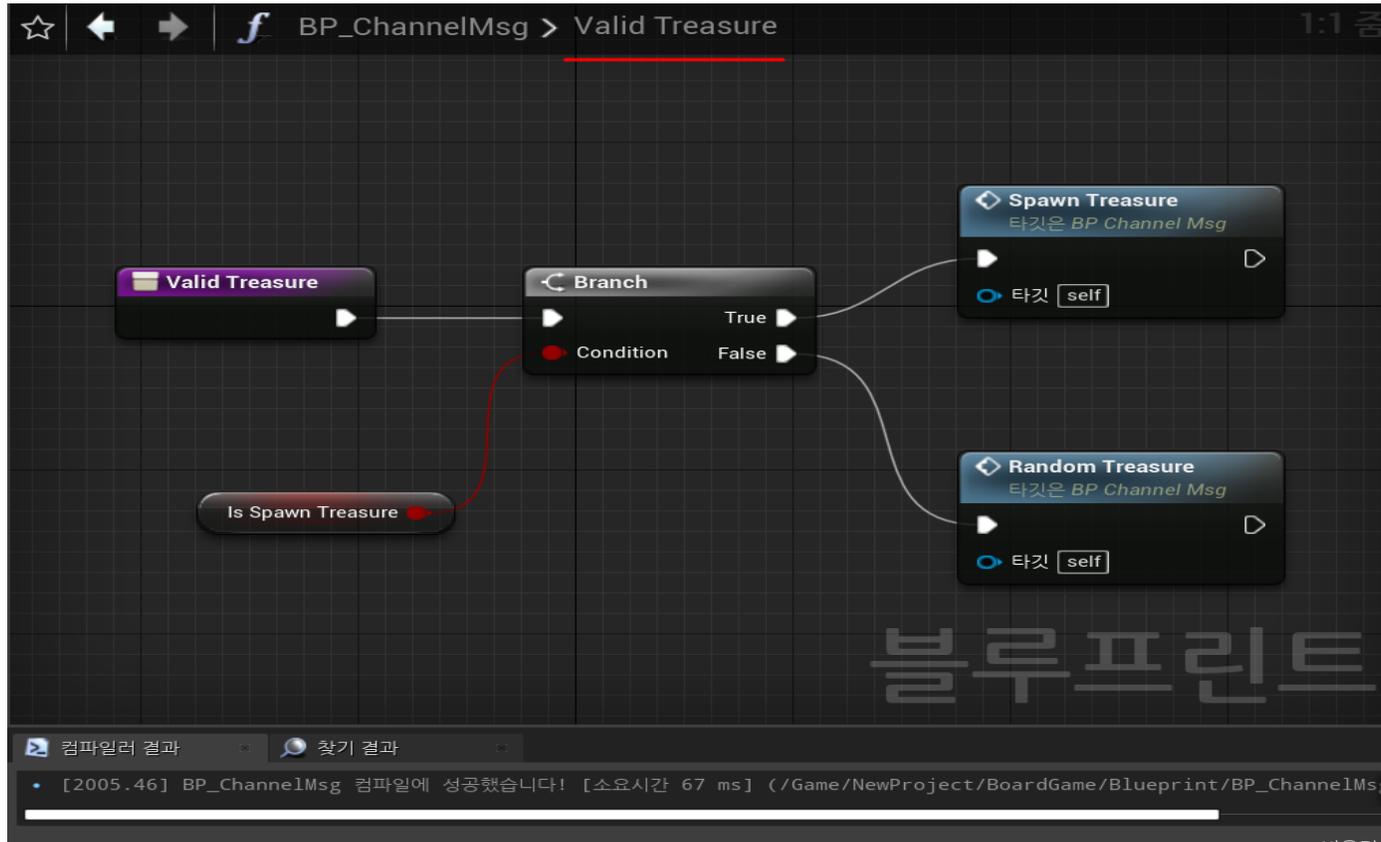
13. BP_BoardPlayer 처리



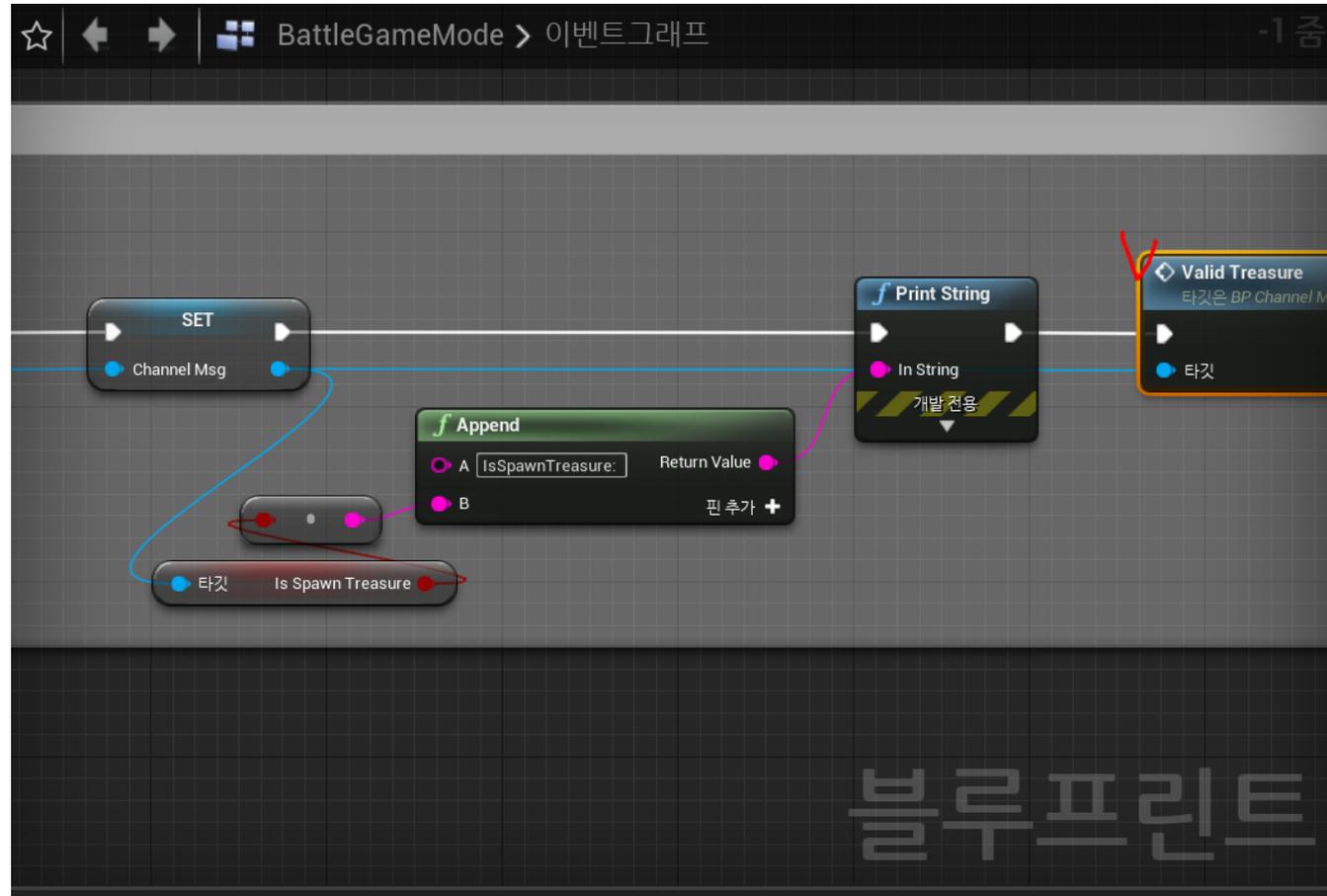
1. BP_ChannelMsg에 들어가서 SpawnTreasure 함수 생성



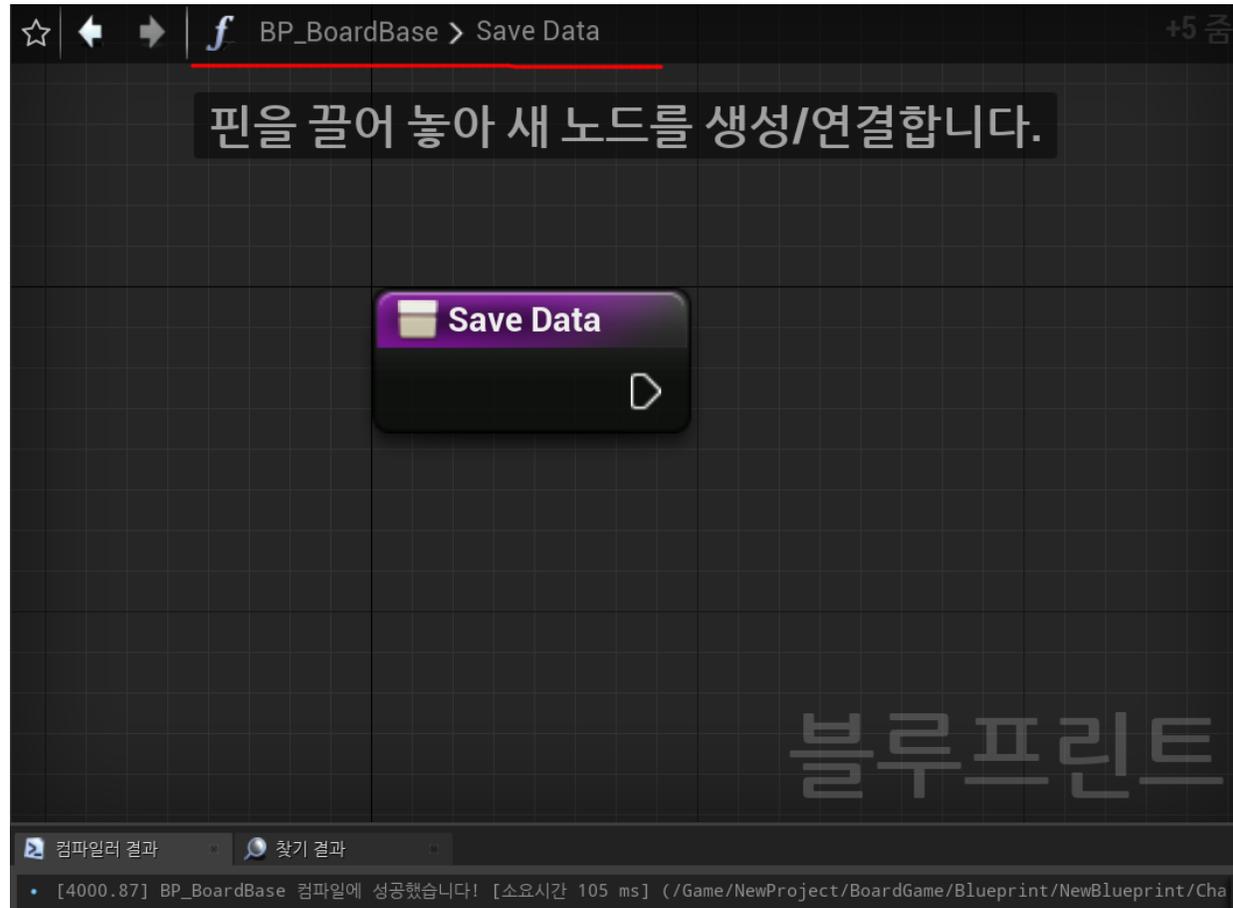
1.1 RandomTreasure 함수의 기능 전부 복사



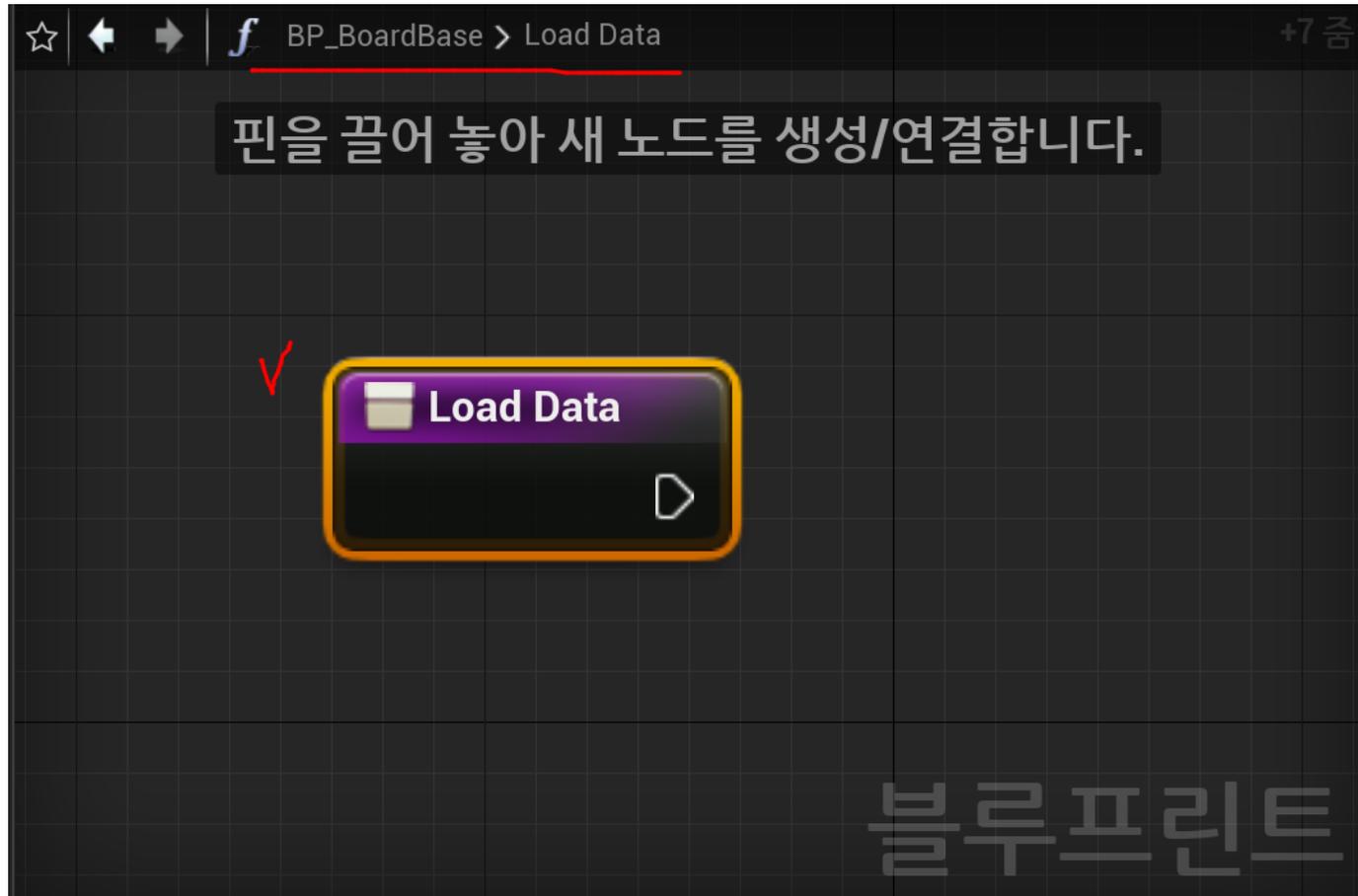
1.3 ValidTreasure 함수를 생성하고, isSpawnTreasure가 True이면 SpawnTreasure를 호출하고, False 이면 RandomTreasure를 호출한다.



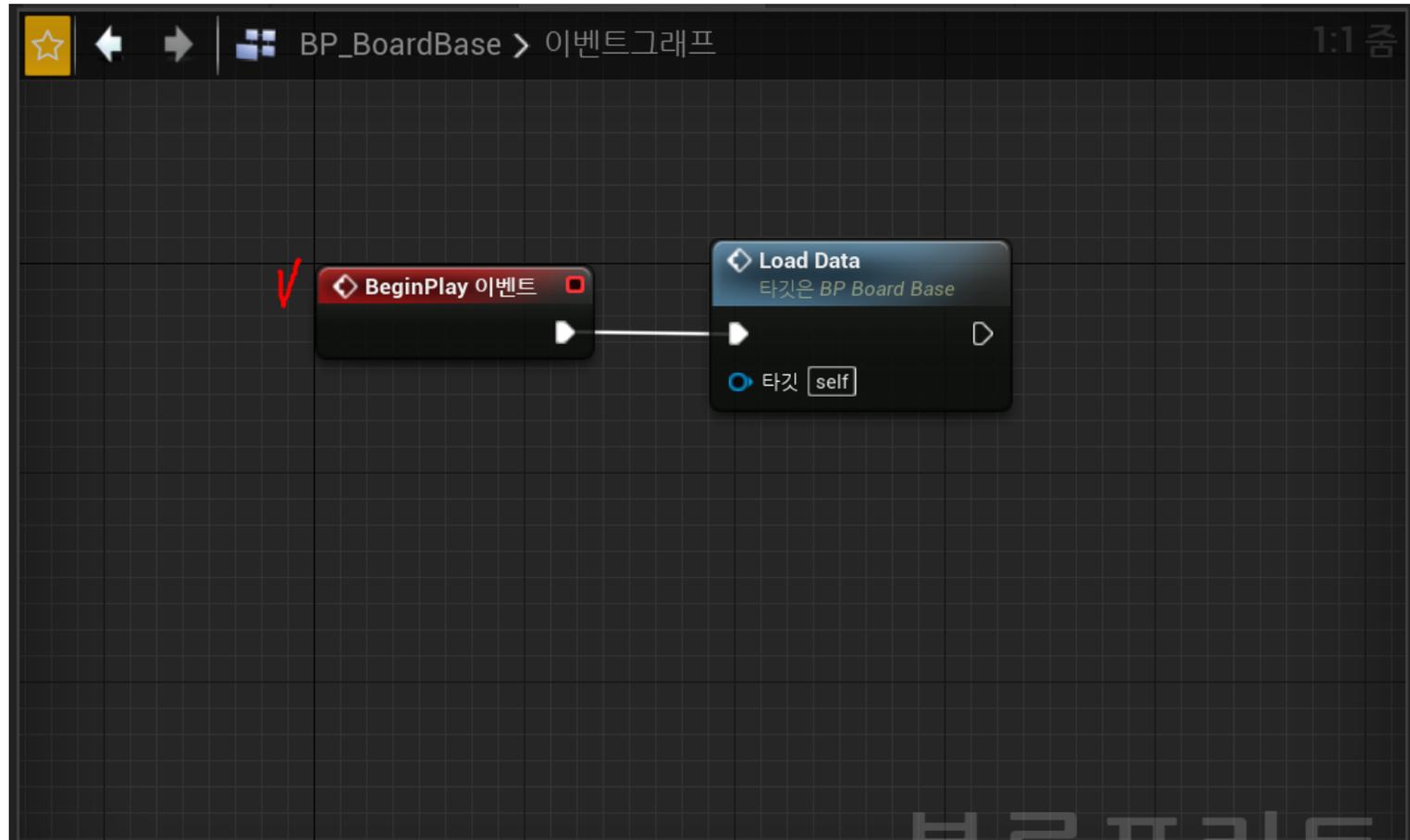
1.4 BattleGameMode에 들어가서 RandomTreasure 대신 ValidTreasure 함수를 호출한다.



2. BP_BoardBase에 들어가서 SaveData 함수를 생성한다.



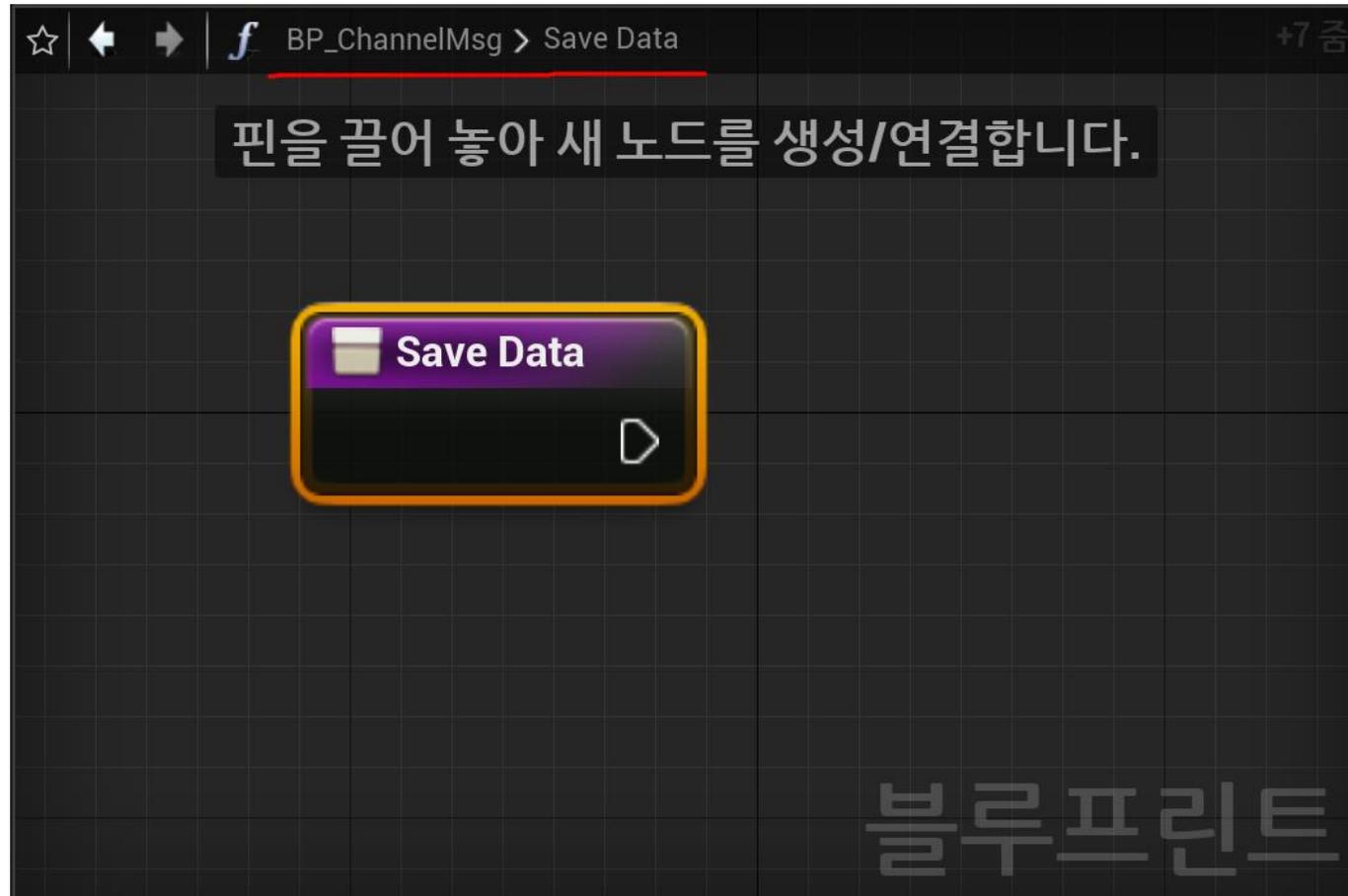
2.1 LoadData 함수를 생성한다.



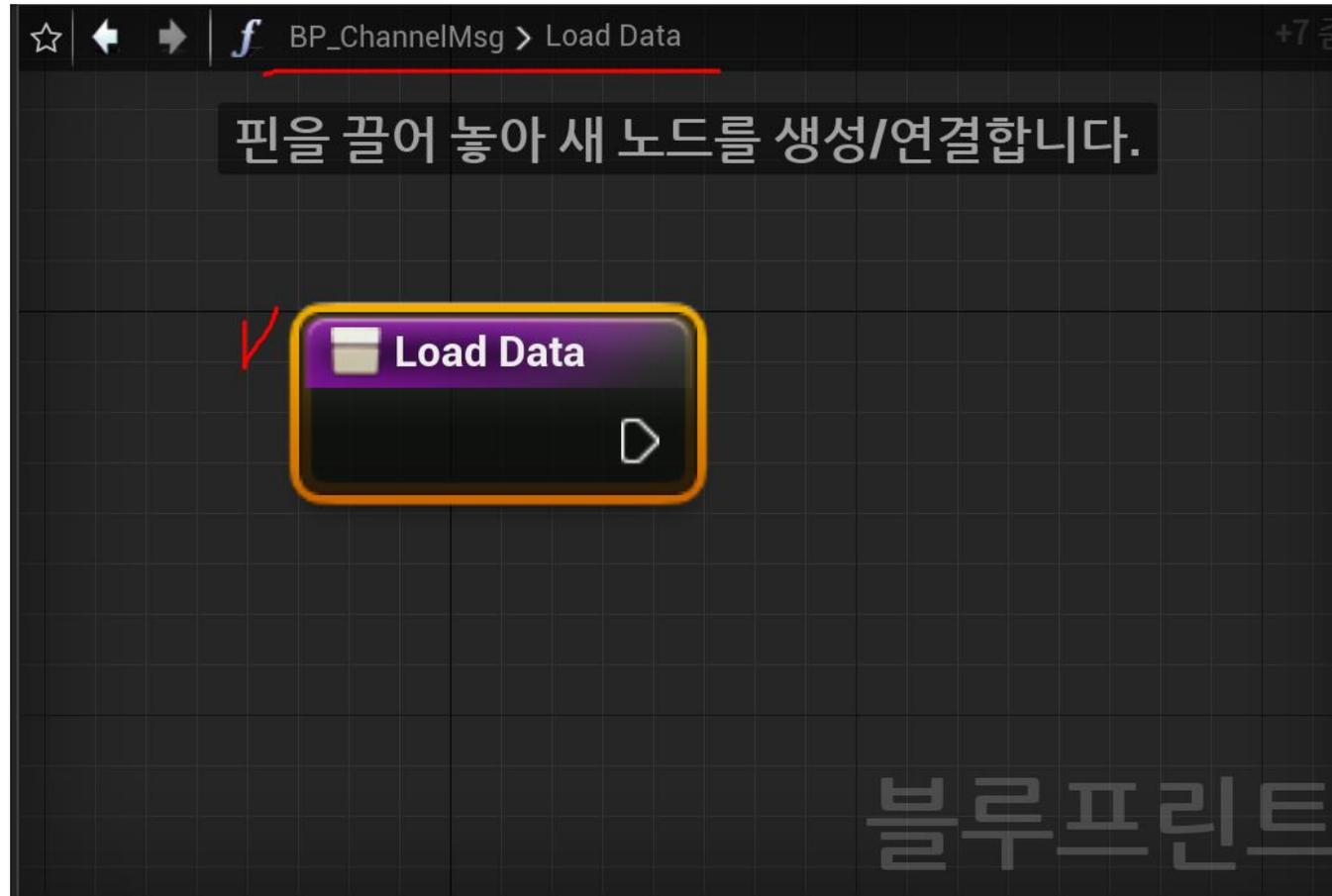
2.1.1 BeginPlay 이벤트에 들어가서 LoadData를 호출한다.



2.2 CombatComponent의 EndUnitTurn에 들어가서 SavaData 함수를 호출한다.



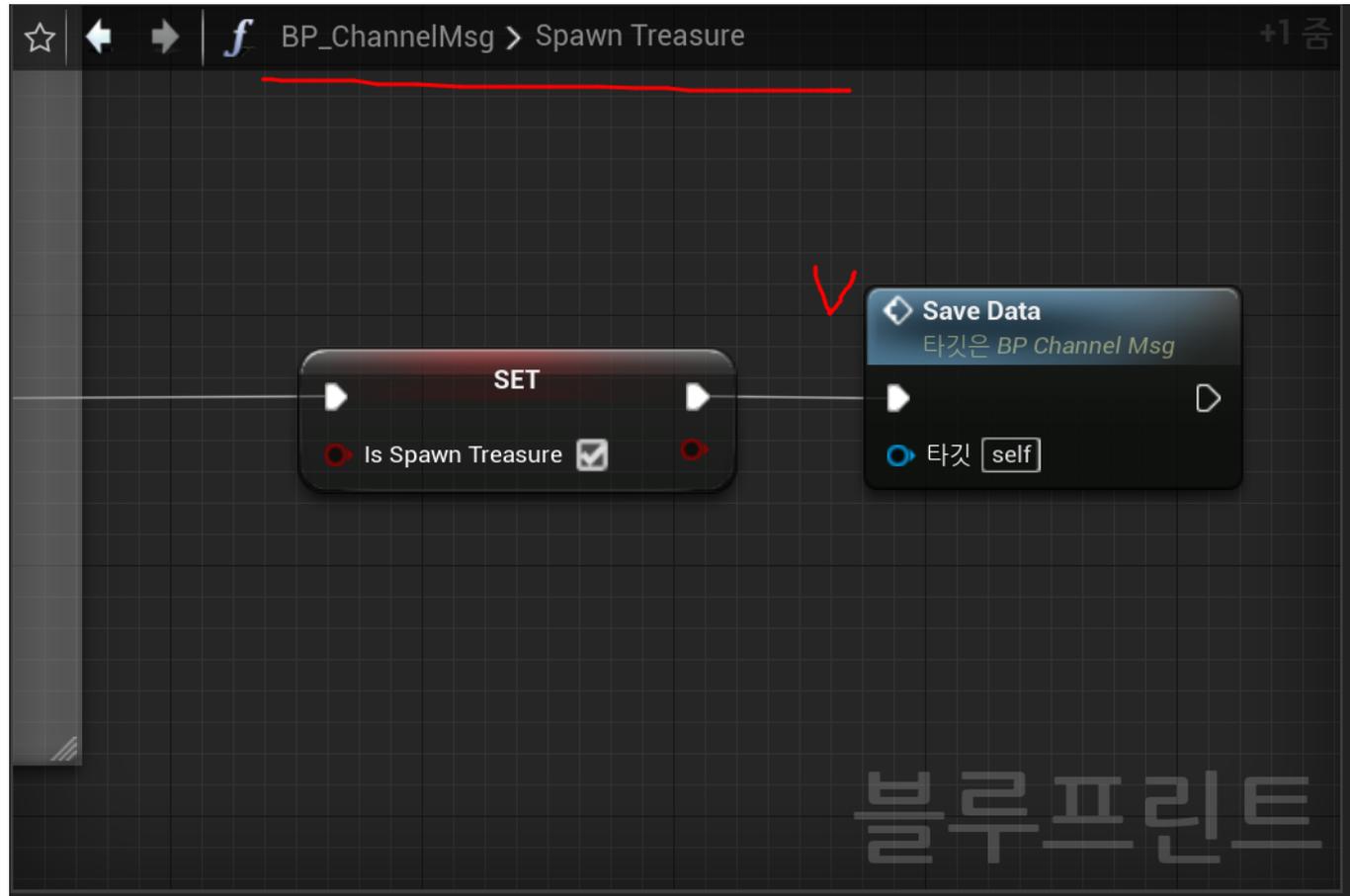
3. BP_ChannelMsg에 들어가서 SaveData 함수를 생성한다.



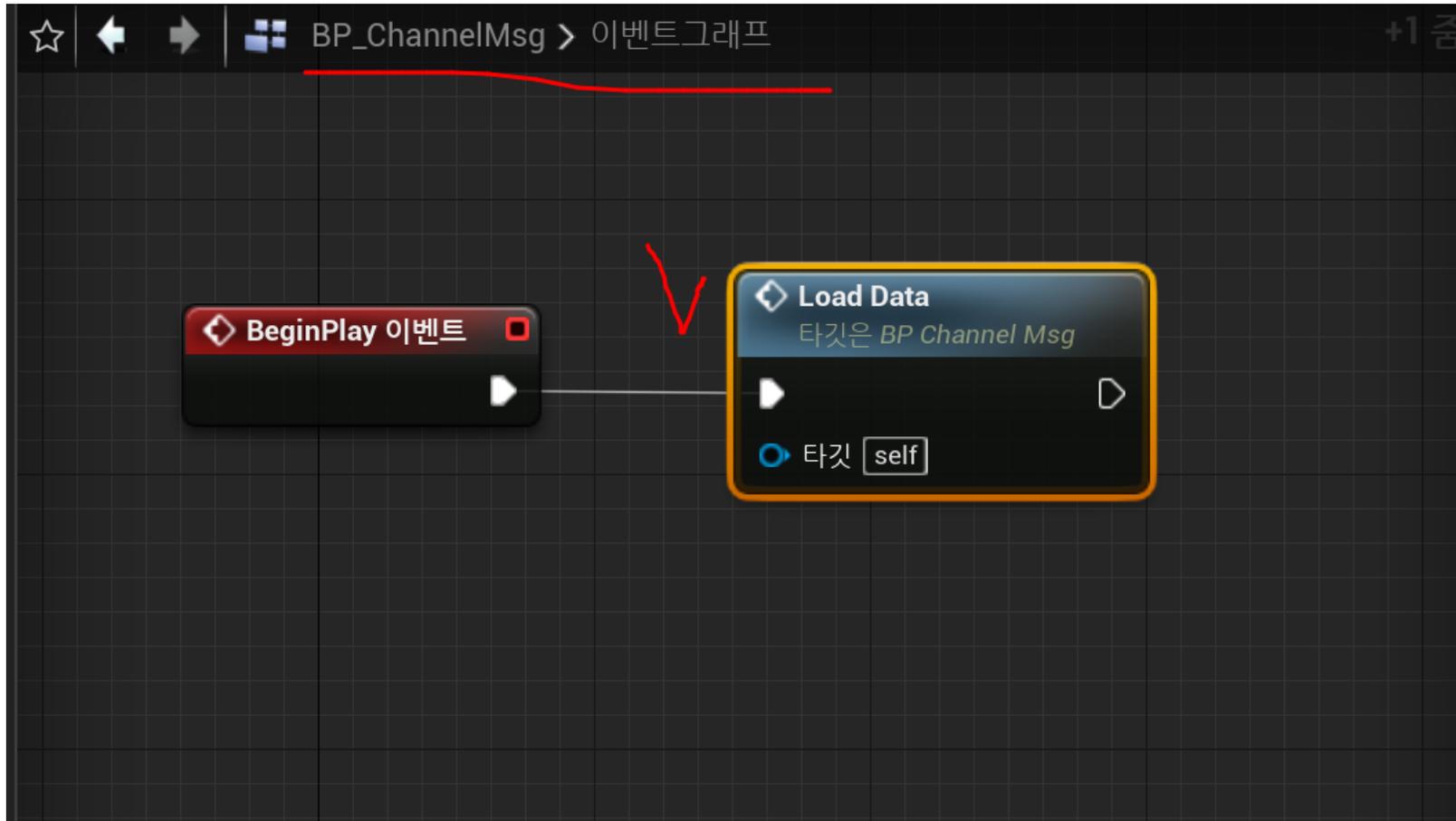
3.1 LoadData 함수를 생성한다.



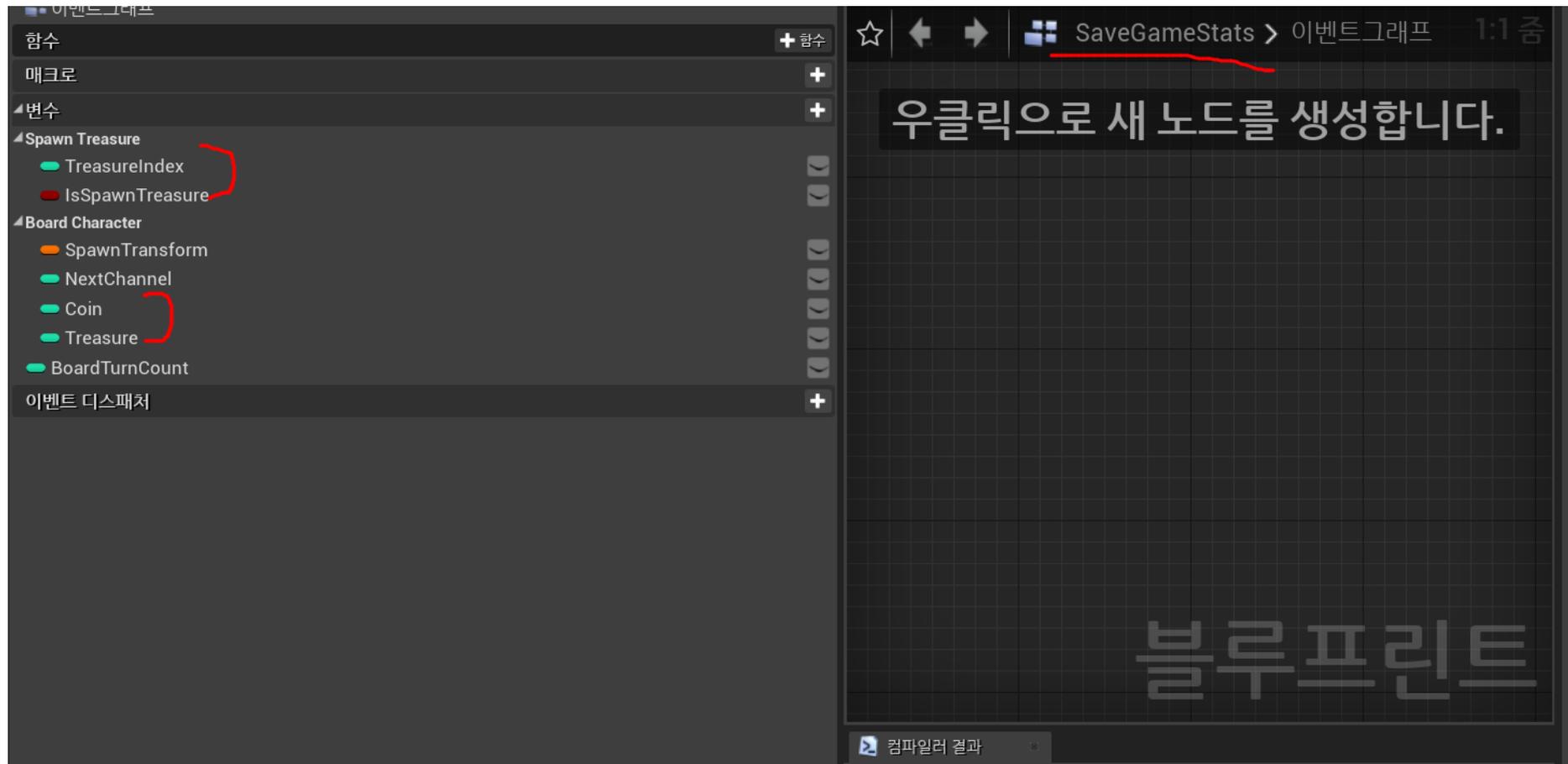
3.2 RandomTreasure 함수의 마지막에 SaveData 함수를 호출한다.



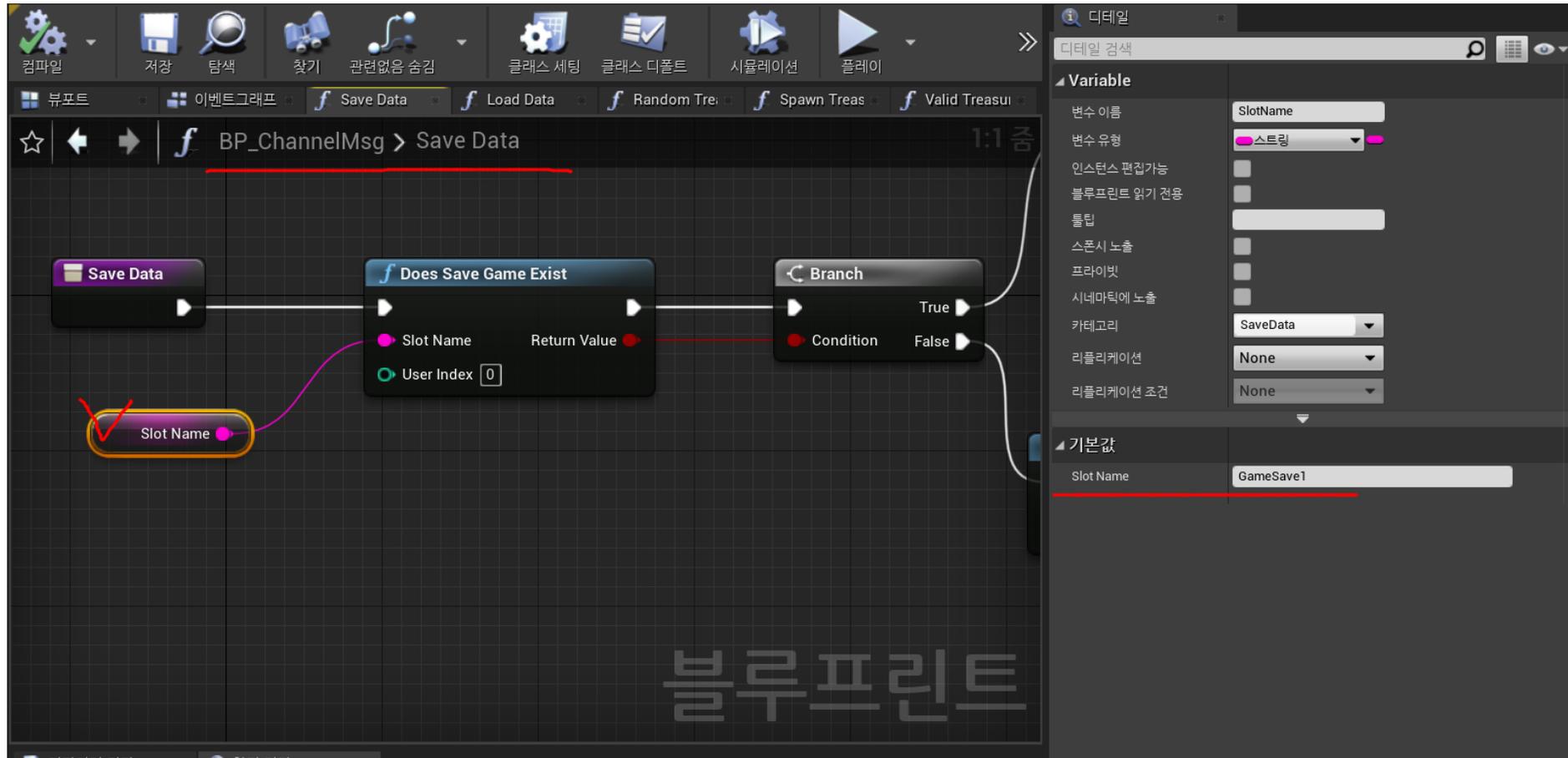
3.2.1 SpawnTreasure 함수의 마지막에 SaveData 함수 호출한다.



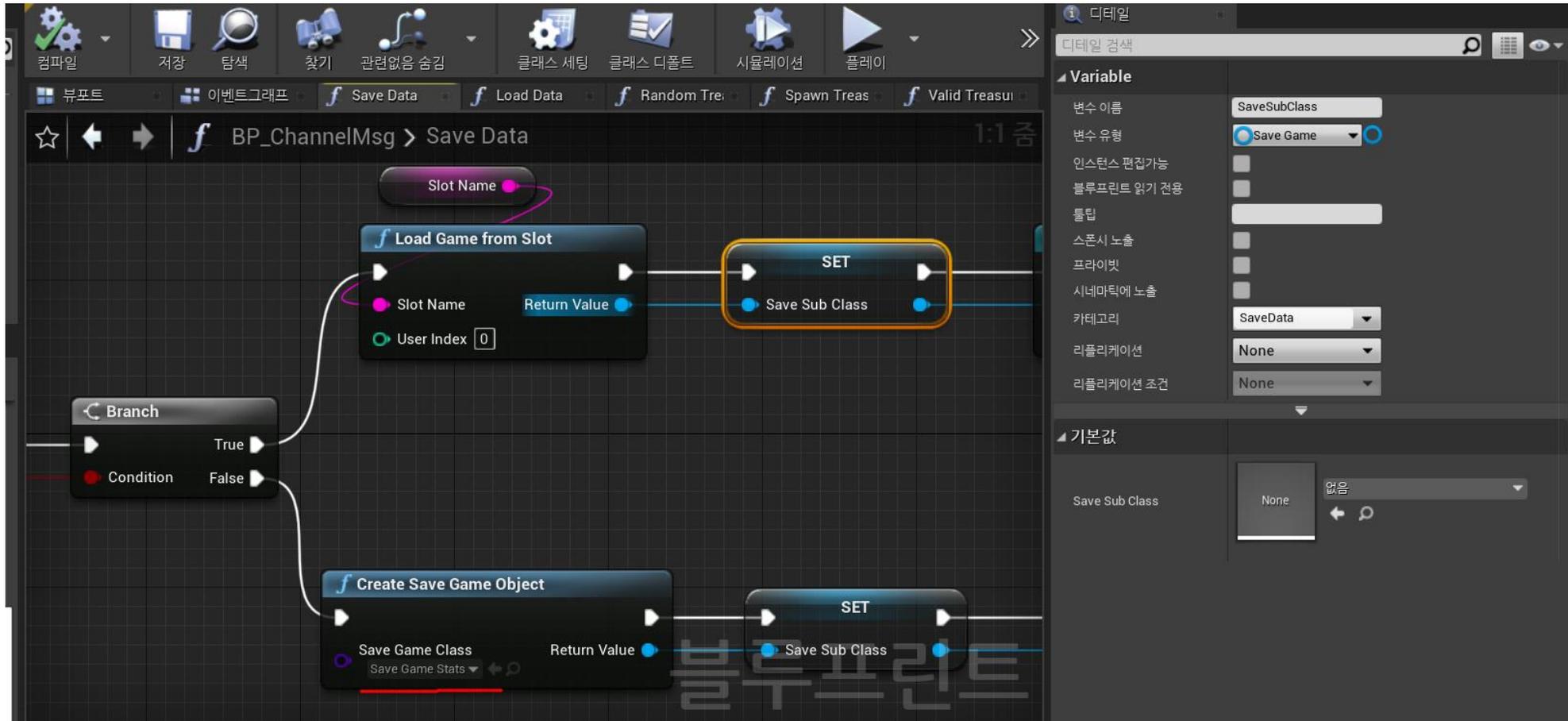
3.3 BeginPlay 이벤트에 들어가서 LoadData 함수를 호출한다.



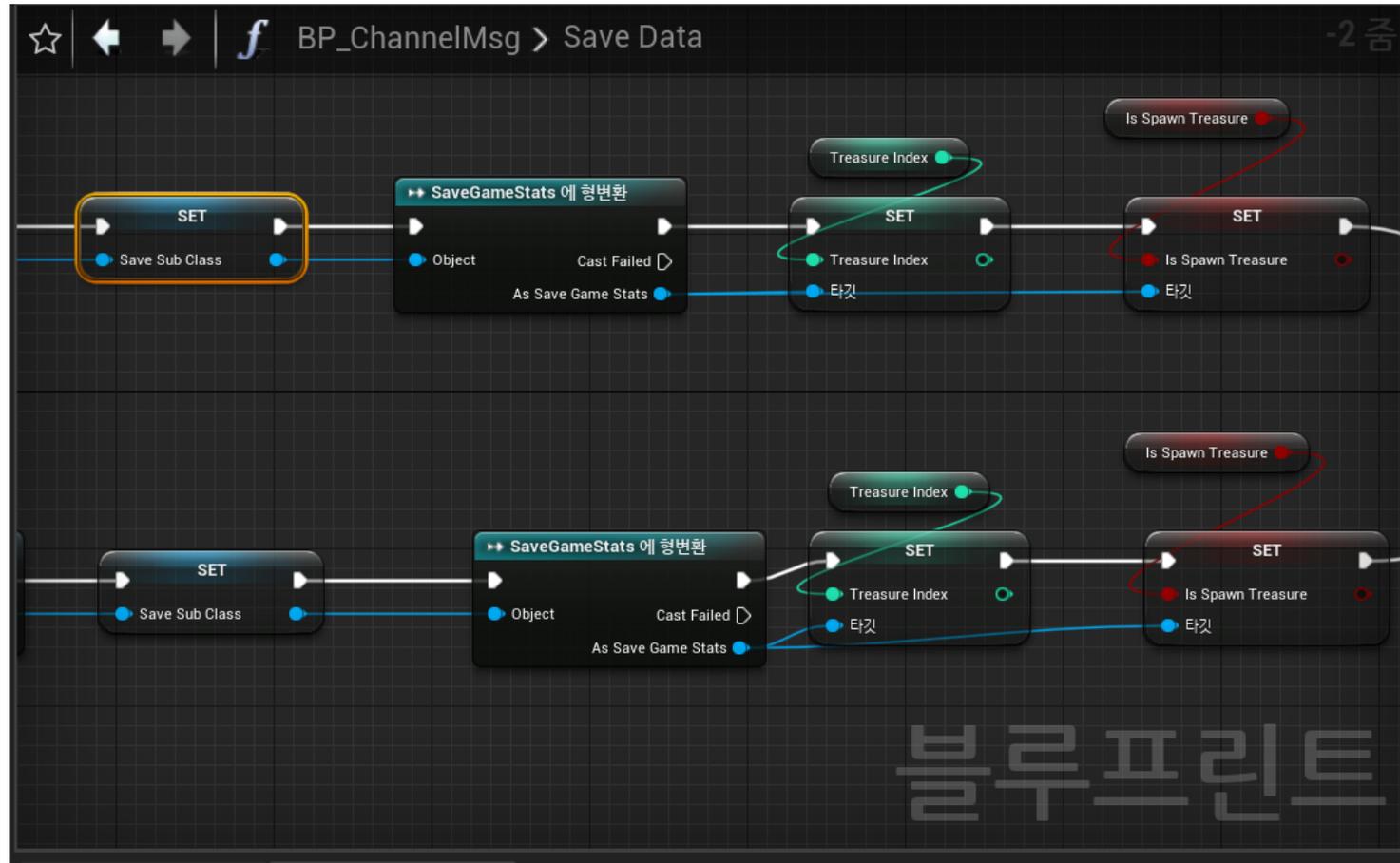
4. SaveGameStats에 들어가서 보물 스폰에 필요한 정보(보물 인덱스, 보물 스폰 여부), 보드 캐릭터에 필요한 정보(소지 코인, 소지 보물)를 저장하는 변수를 생성한다..



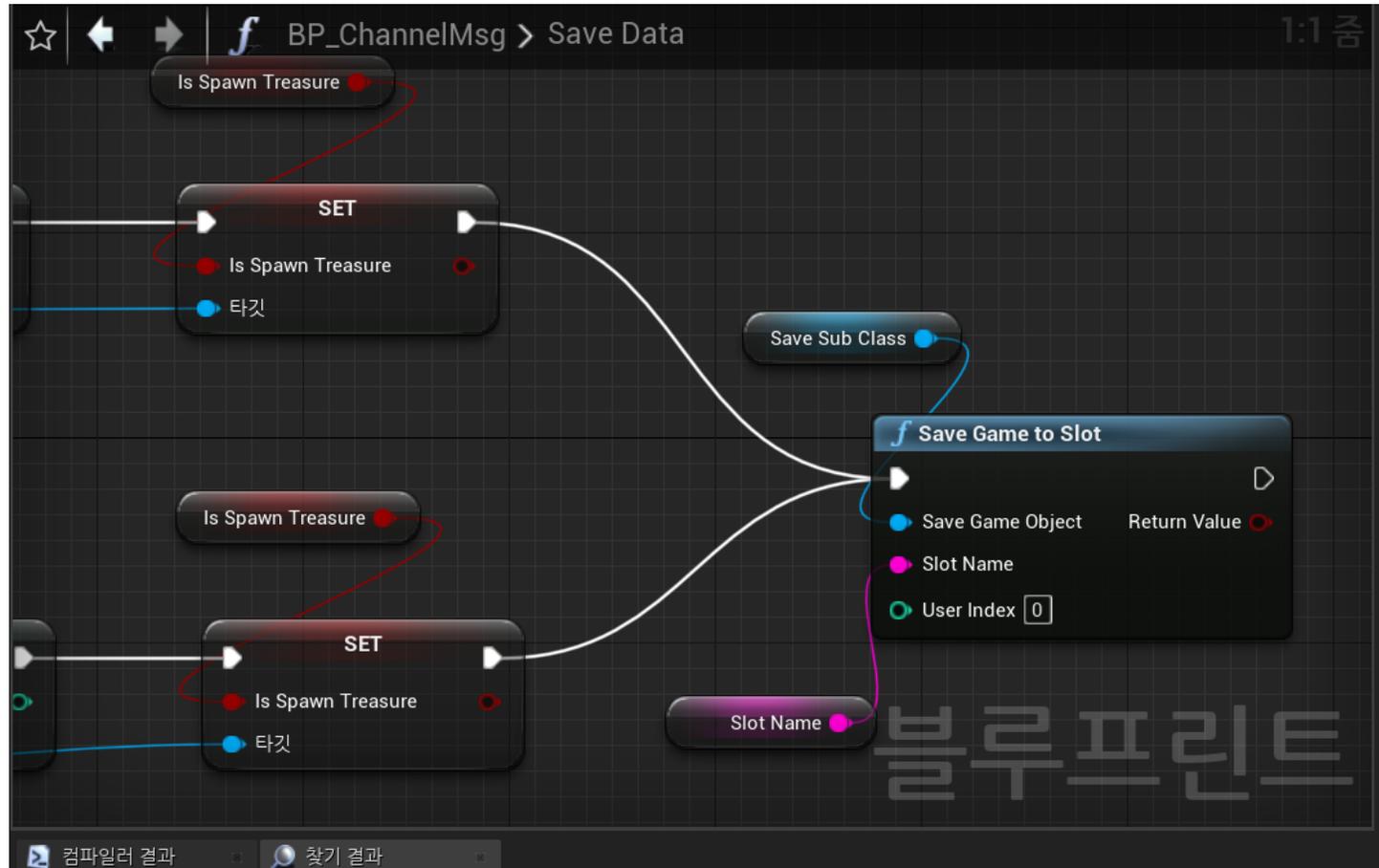
5. BP_ChannelMsg의 SaveData 함수에 들어가서 String 형 변수 SlotName을 생성하고, 디폴트 값을 GameSave1이라고 하고, SlotName의 SaveGame이 존재하는지 판단한다.



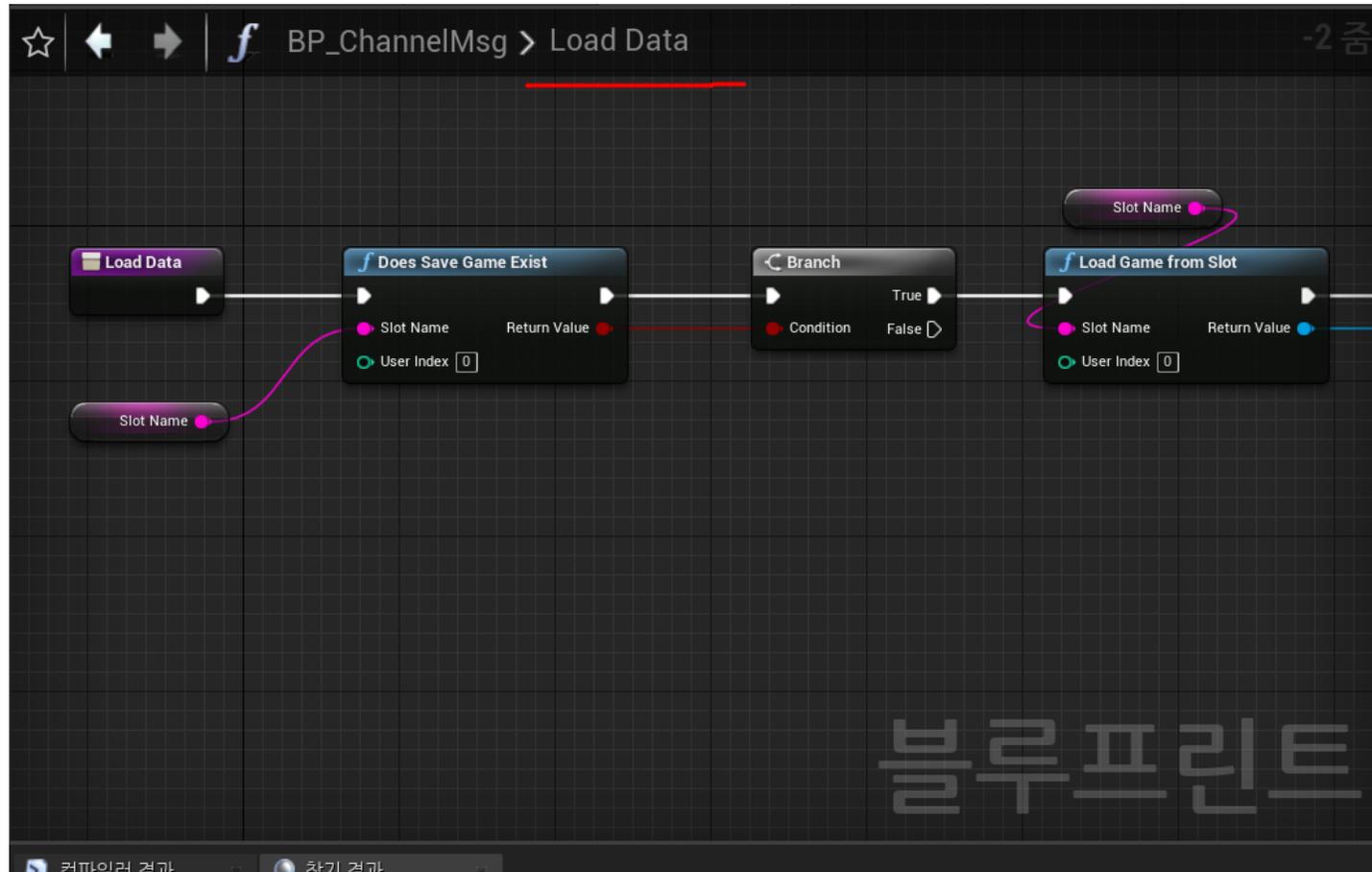
5.1 SlotName의 SaveGame이 존재하면 Load하고, 존재하지 않으면 SaveGameStats를 생성한다.
그리고 SaveSubClass 변수를 생성한다.



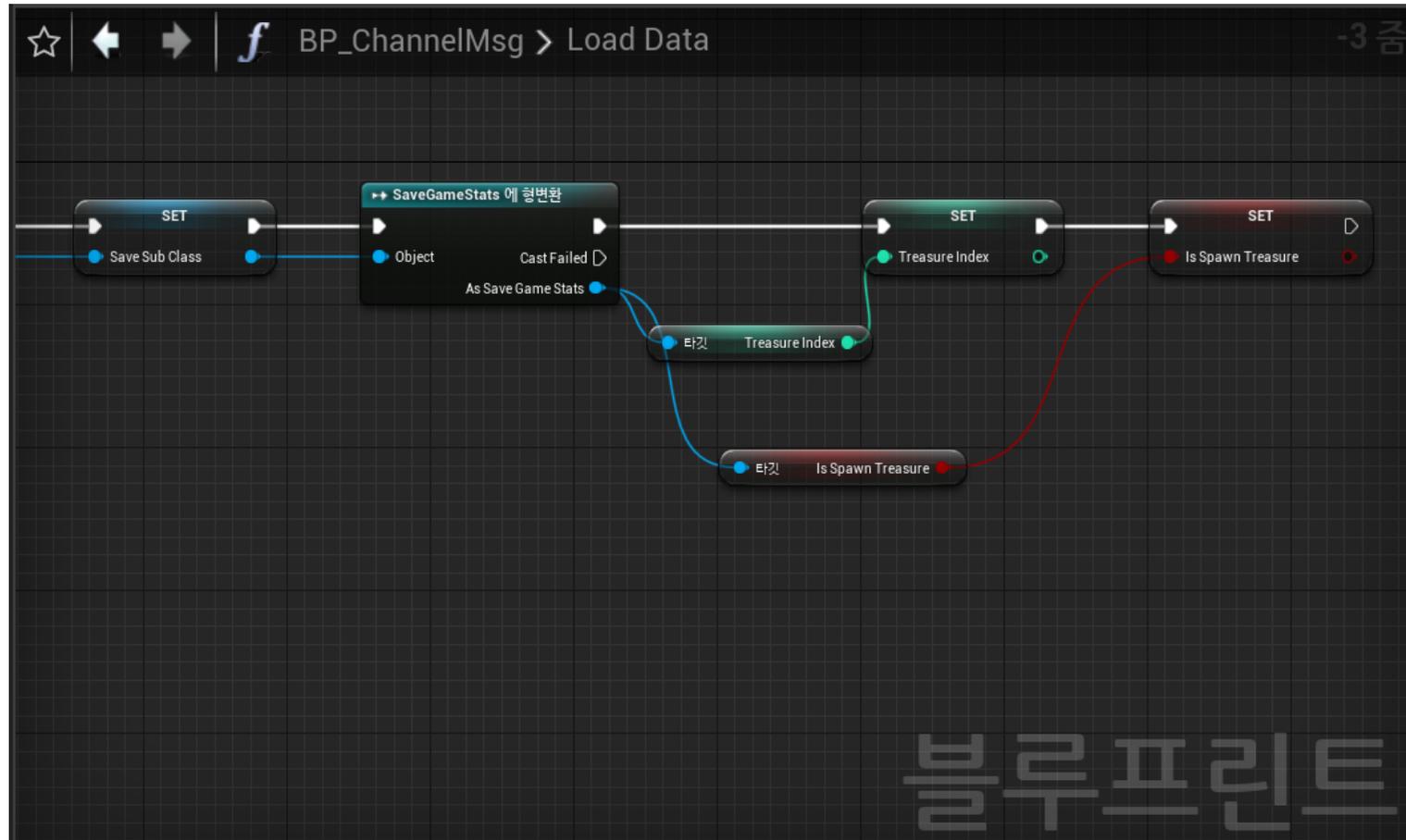
5.2 SaveGameStats의 TreasureIndex와 IsSpawnTreasure의 데이터를 저장한다.



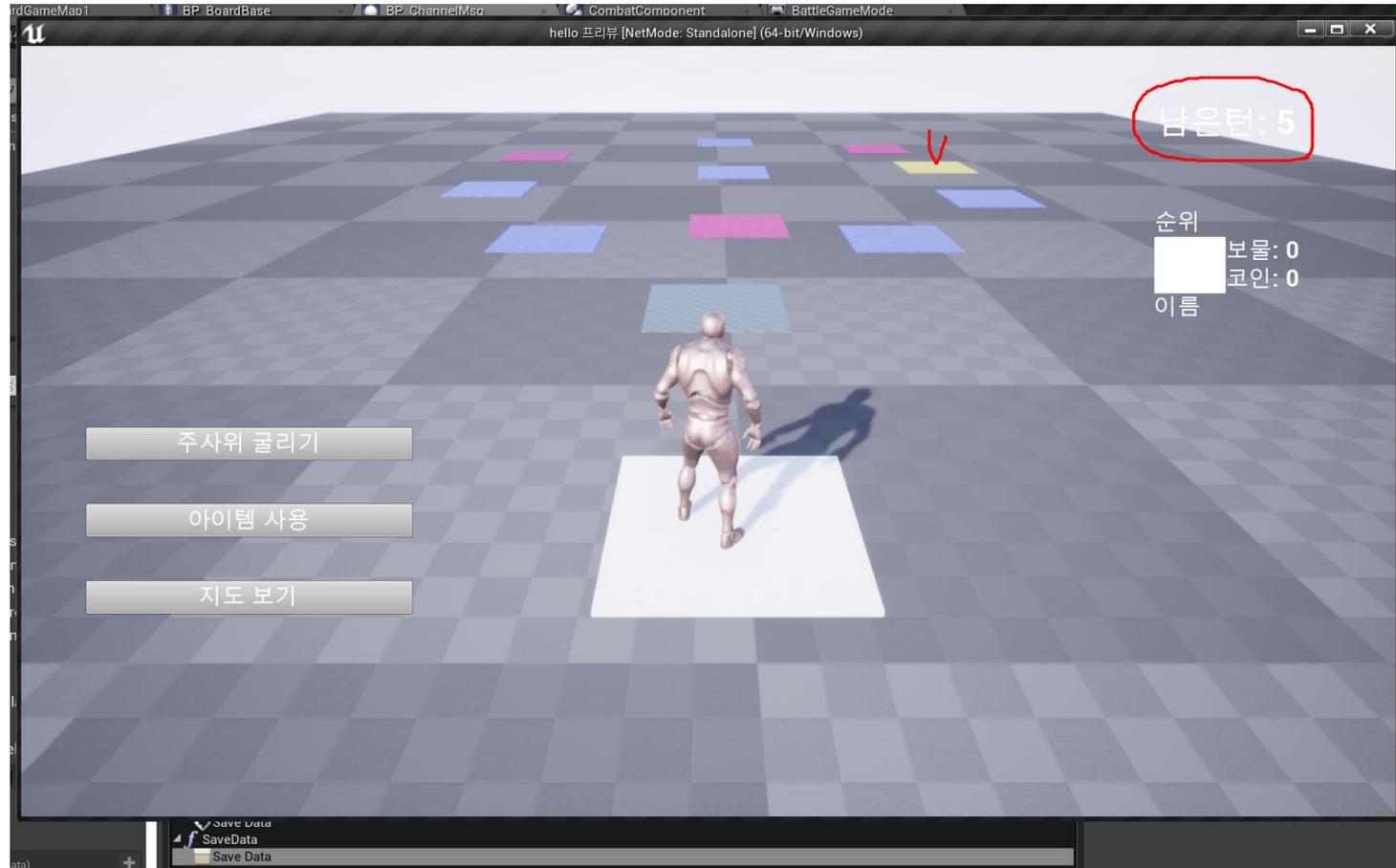
5.3 SlotName의 SaveGame Slot을 생성한다.



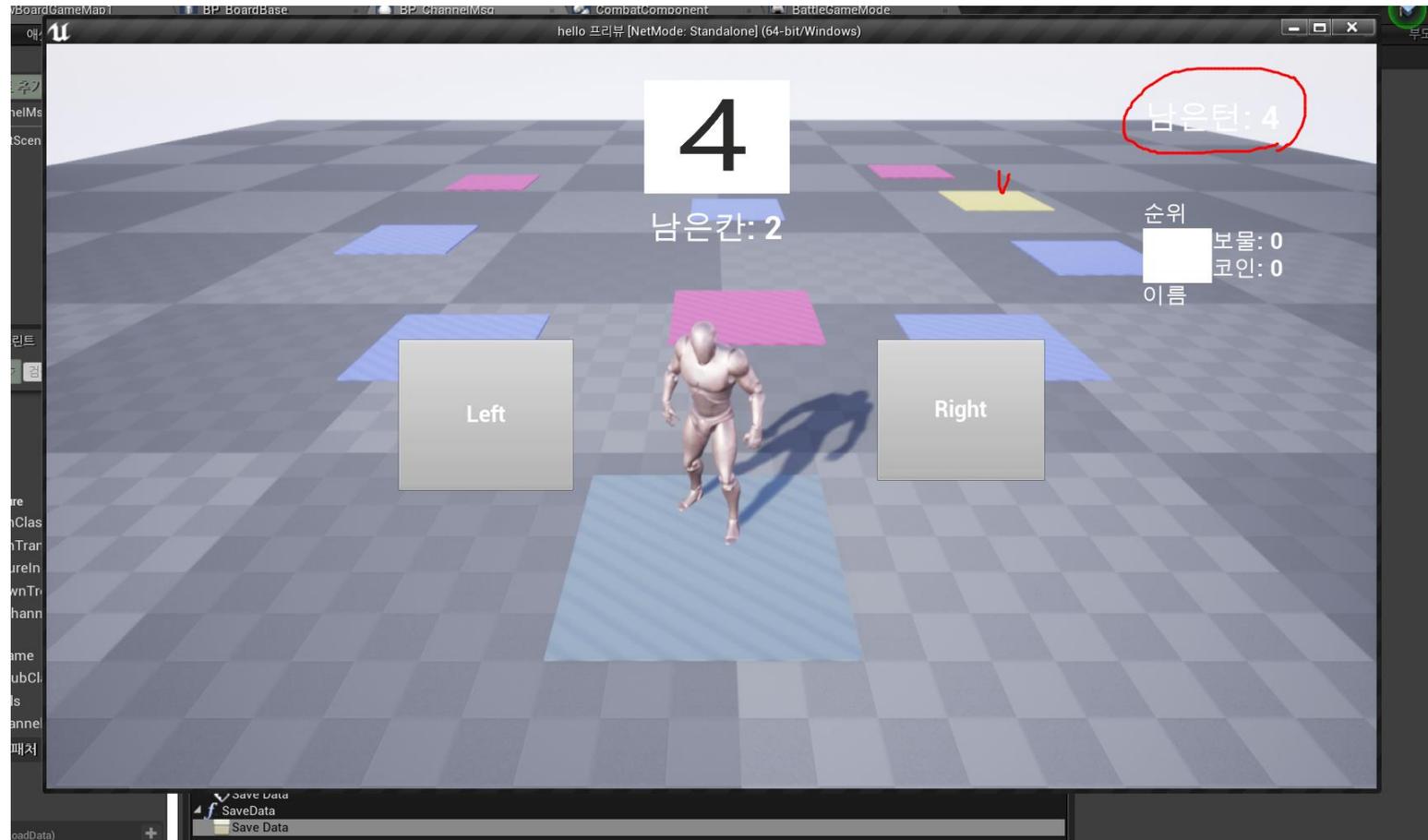
5.4 LoadData 함수에 들어가서 SlotName이 SaveGameSlot이 존재하는지 판단하고, Load한다.



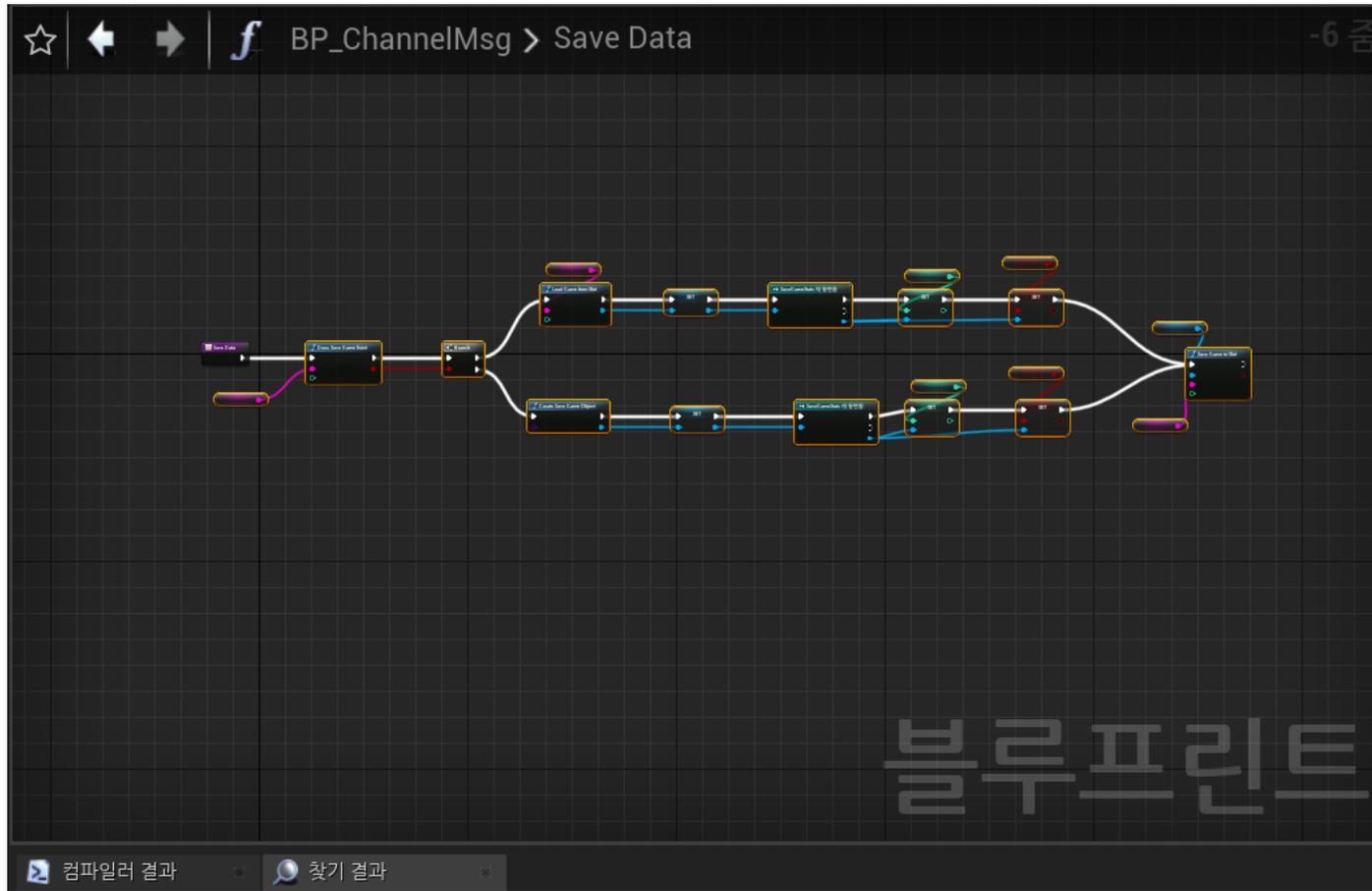
5.5 TreasureIndex와 IsSpawnTreasure를 Load한다.



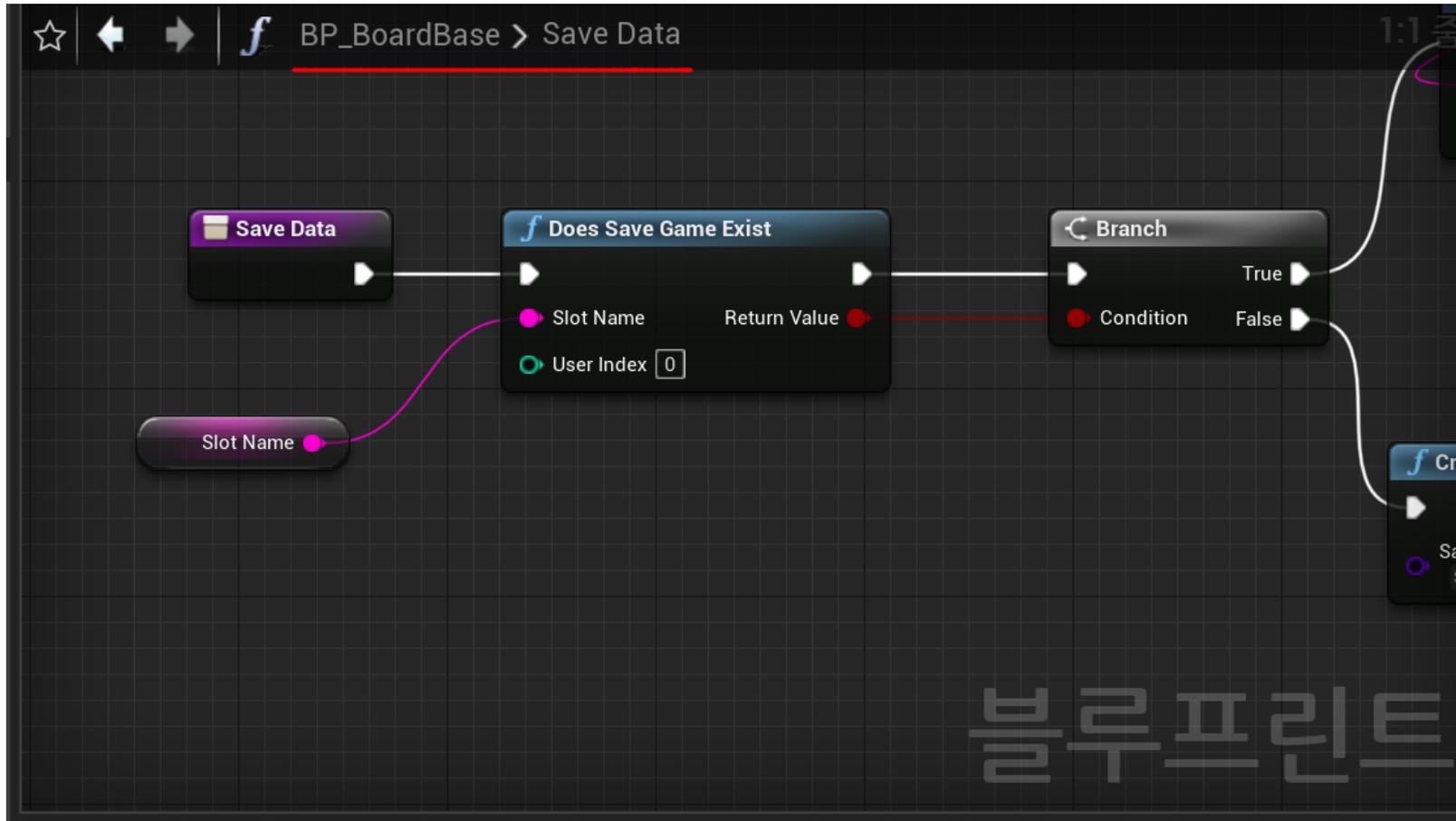
5.6 초반 실행 화면



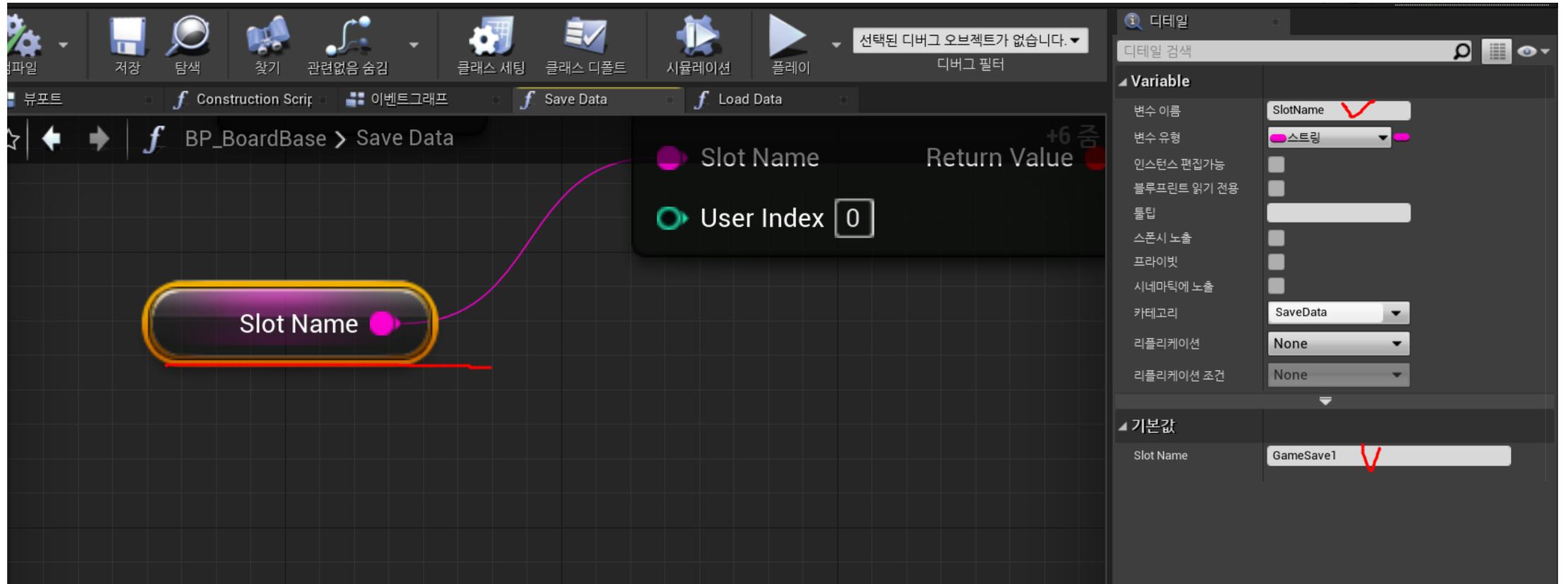
5.7 저장된 보물칸



5.8 SaveData 함수의 Save 기능을 복사한다.



6. BP_BoardBase의 SaveData에 들어가서 복사한 Save 기능을 붙여넣는다.



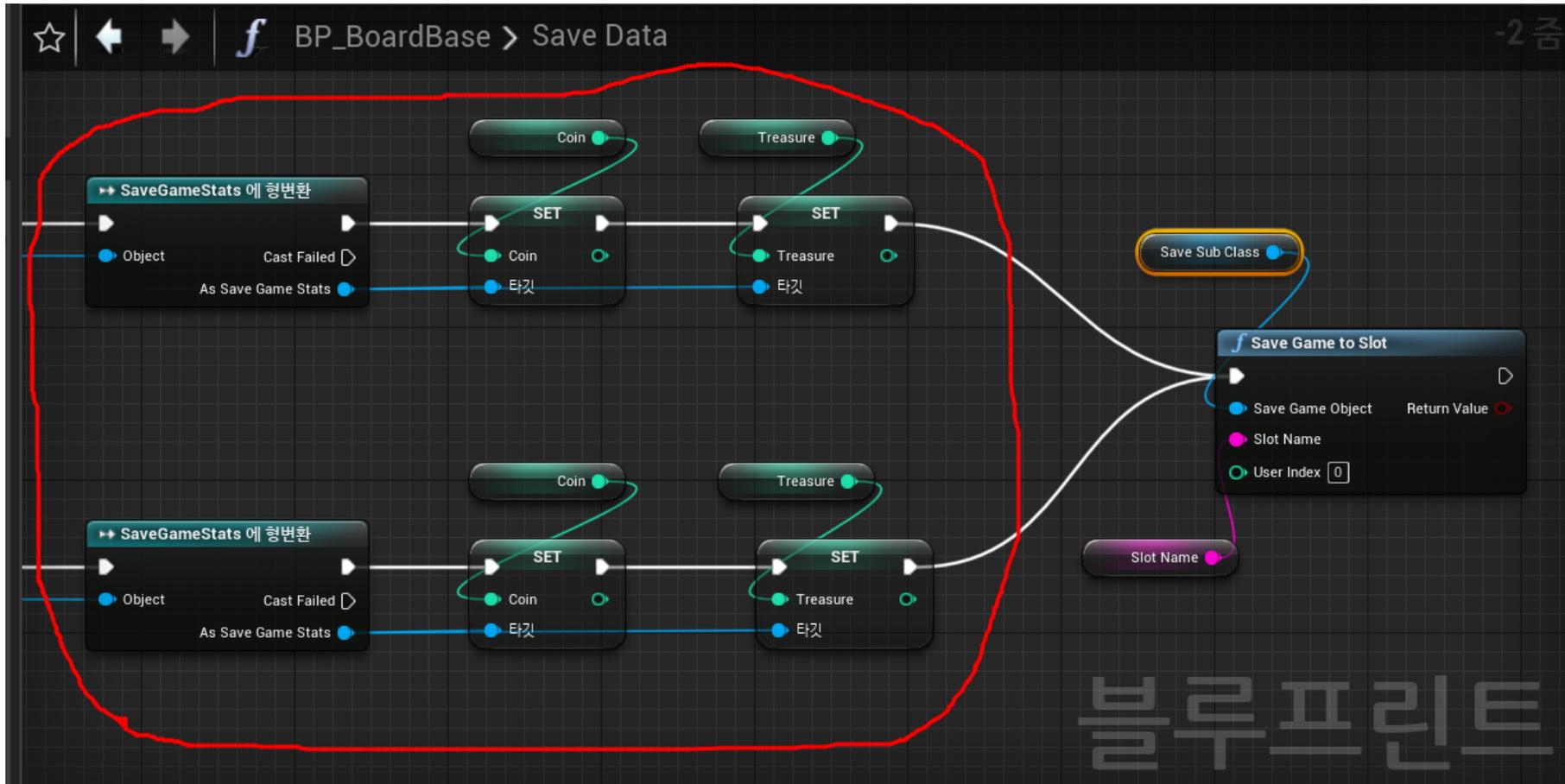
6.1 String형 변수 SlotName을 생성하고, 디폴트로 GameSave1로 한다.

The image shows a screenshot of the Unreal Engine Blueprints editor. The main workspace displays a Blueprint for a function named 'Save Data' within a 'BP_BoardBase' class. A 'Slot' node on the left has a 'Return Value' pin connected to the 'Save Sub Class' input of a 'SET' node. The 'SET' node is highlighted with an orange border and has an 'Object' input connected to a 'SaveGame' node. The right-hand side of the screen shows the 'Variable' panel for the 'SaveSubClass' variable, with the following settings:

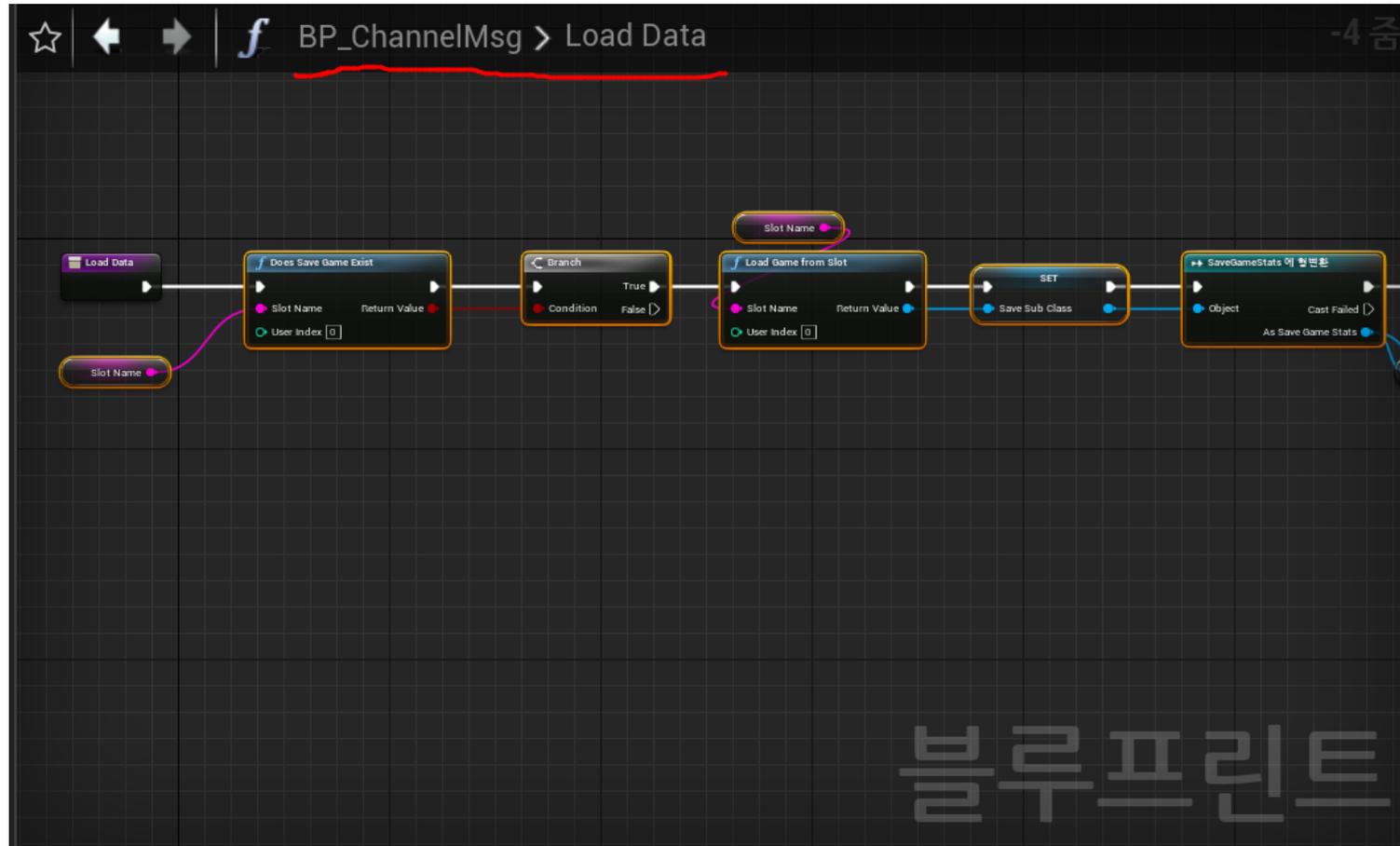
Property	Value
변수 이름	SaveSubClass
변수 유형	Save Game
인스턴스 편집가능	<input type="checkbox"/>
블루프린트 읽기 전용	<input type="checkbox"/>
틀림	
스폰시 노출	<input type="checkbox"/>
프라이빗	<input type="checkbox"/>
시네마틱에 노출	<input type="checkbox"/>
카테고리	SaveData
리플리케이션	None
리플리케이션 조건	None

Below the 'Variable' panel is the '기본값' (Default Value) section, where the 'Save Sub Class' property is set to '없음' (None).

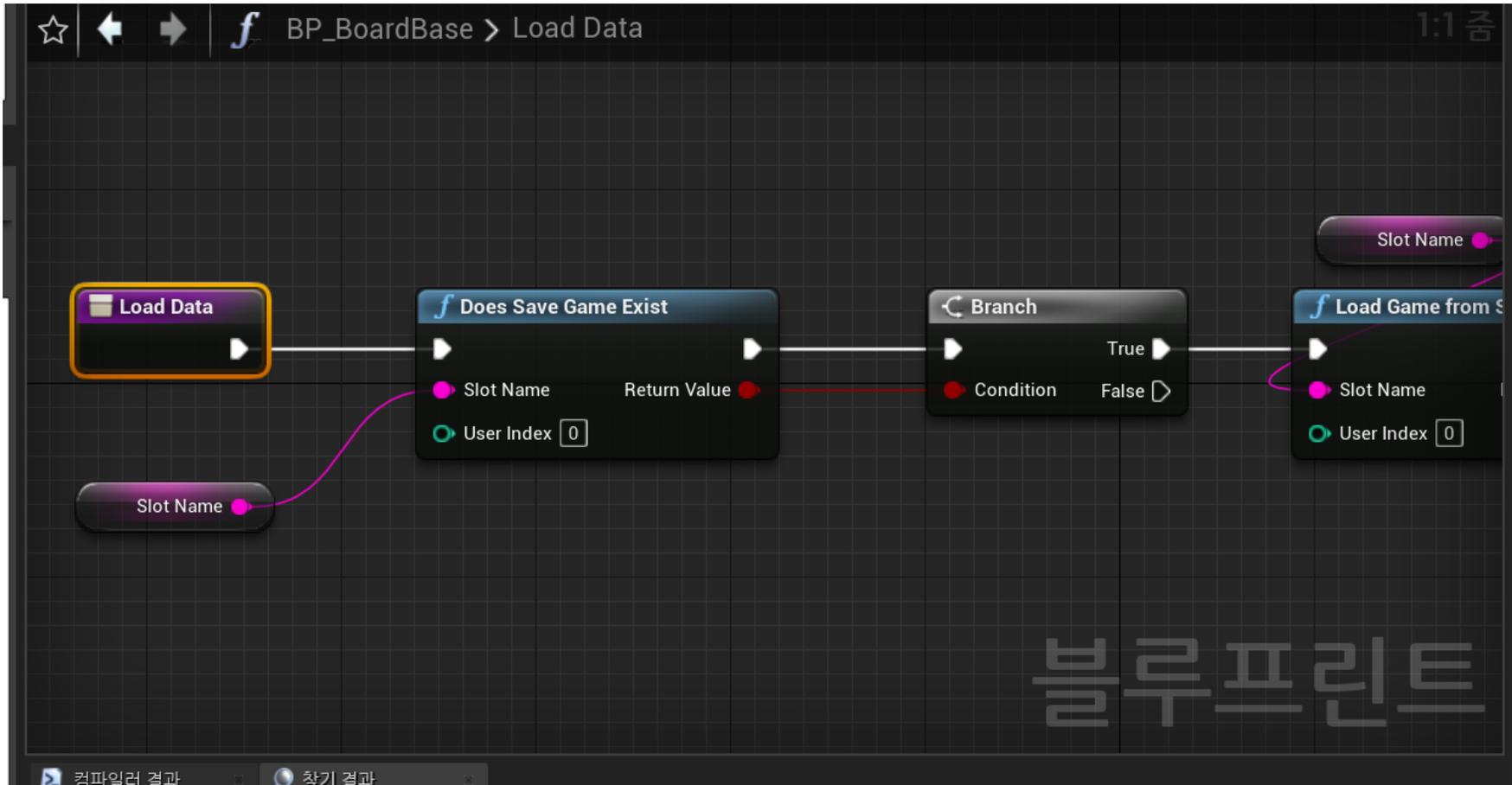
6.2 SaveGame형 SaveSubClass 변수를 생성한다.



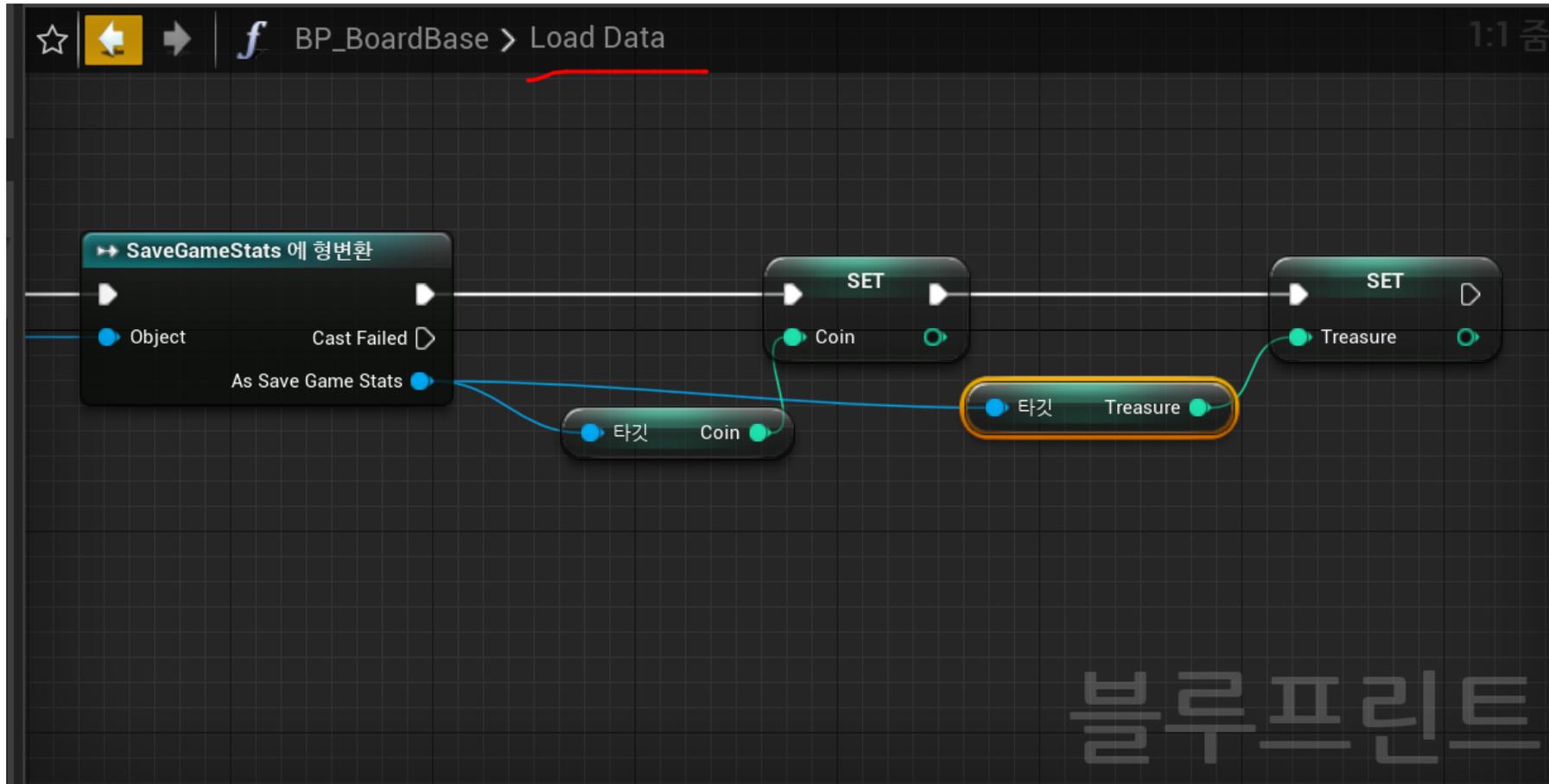
6.3 Coin과 Treasure를 저장한다.



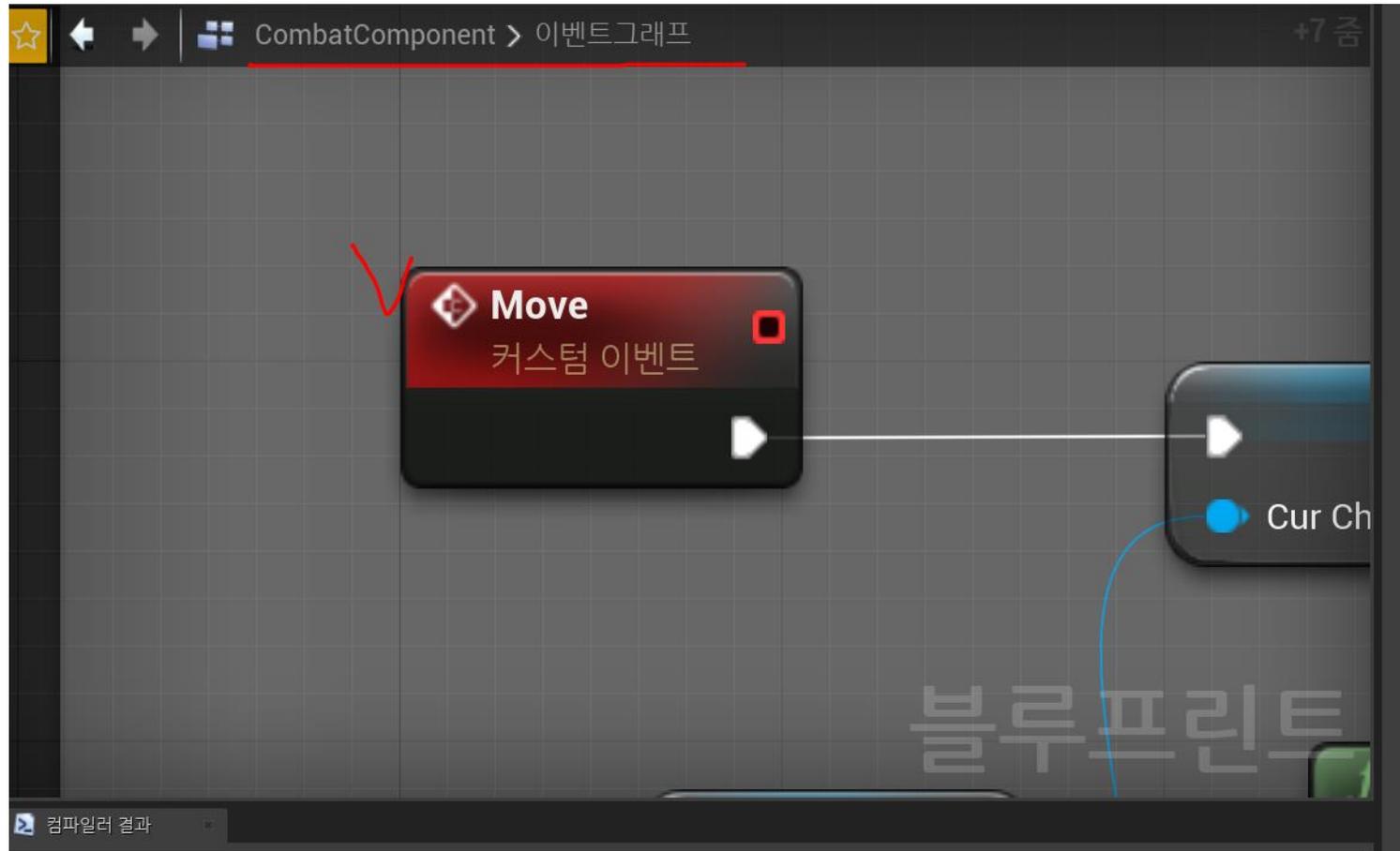
6.4 BP_ChannelMsg의 LoadData의 Load 기능을 복사한다.



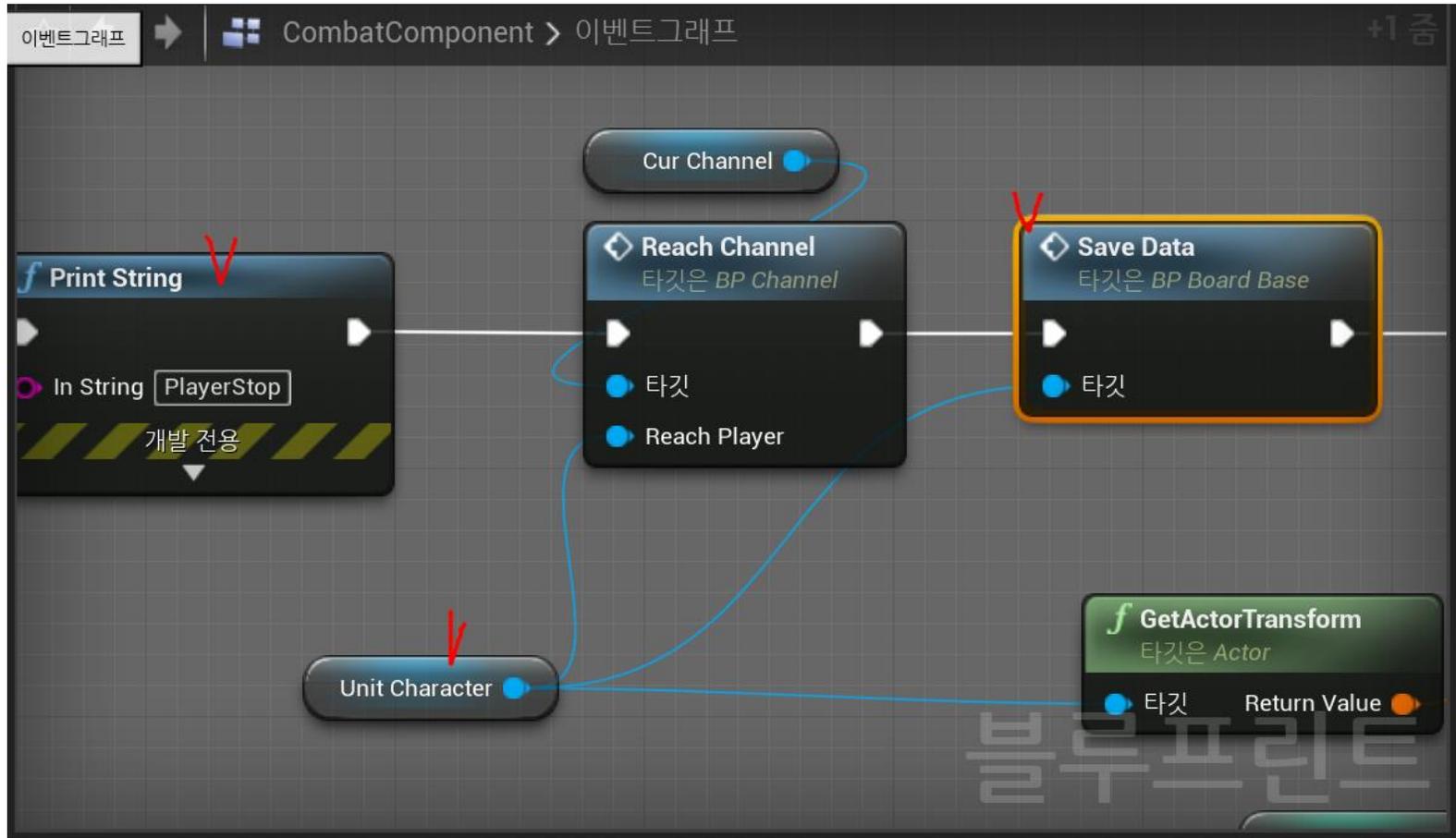
6.5 BP_BoardBase의 LoadData 함수에 Load 기능을 붙여넣는다.



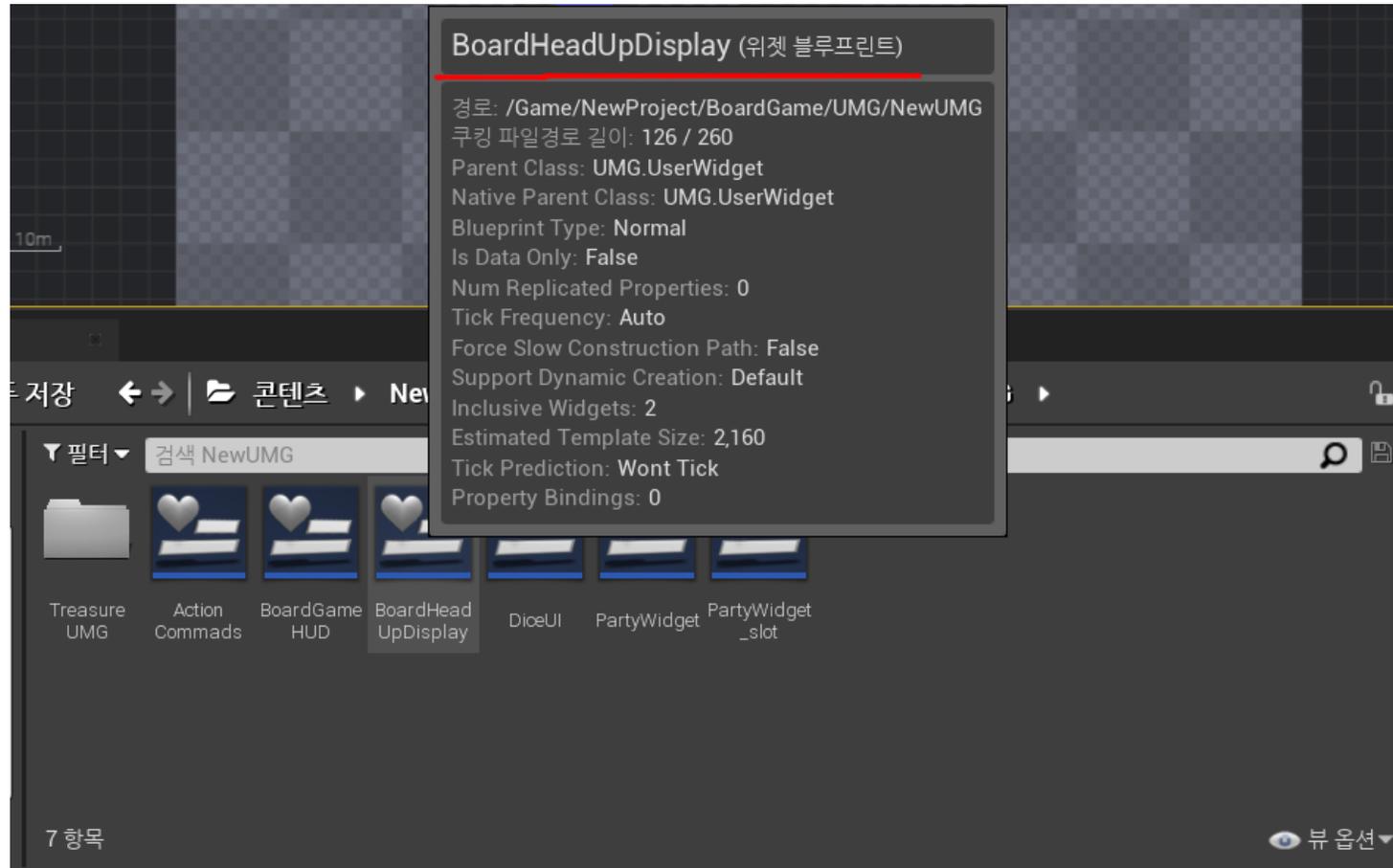
6.6 Coin과 Treasure를 Load 한다.



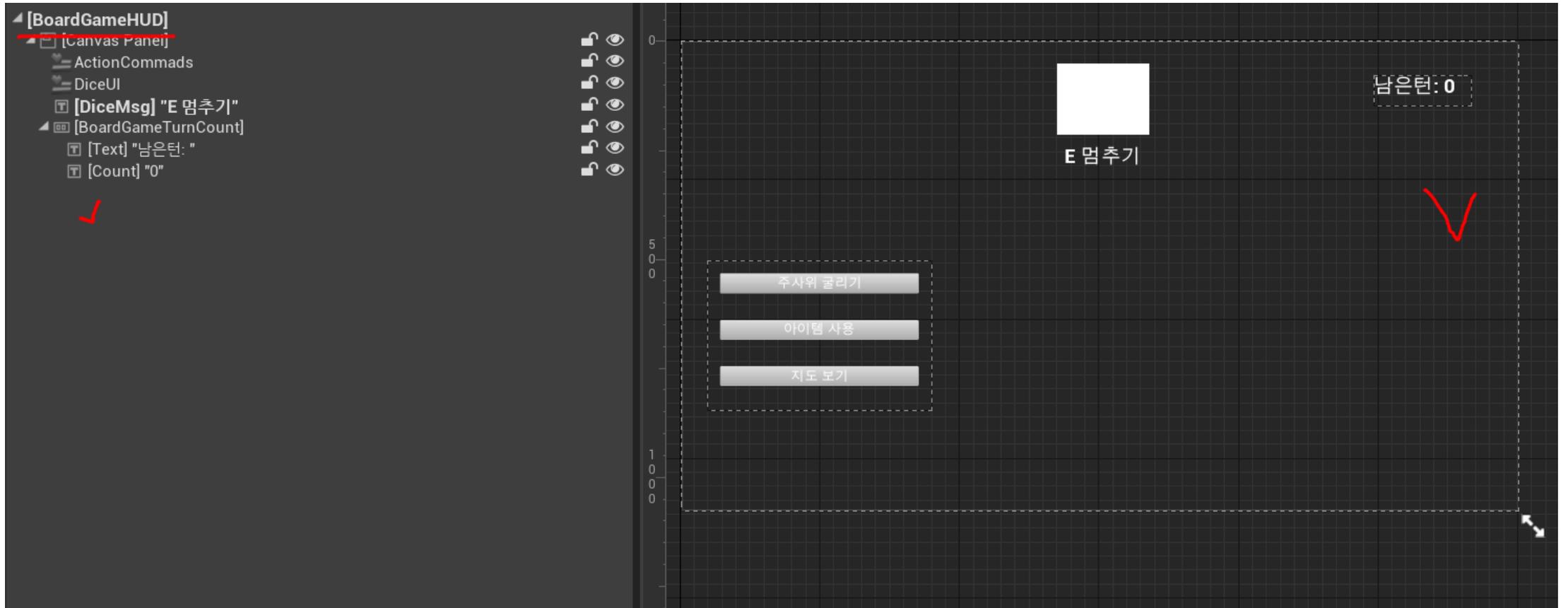
6.7 CombatComponent의 Move 이벤트에 들어간다.



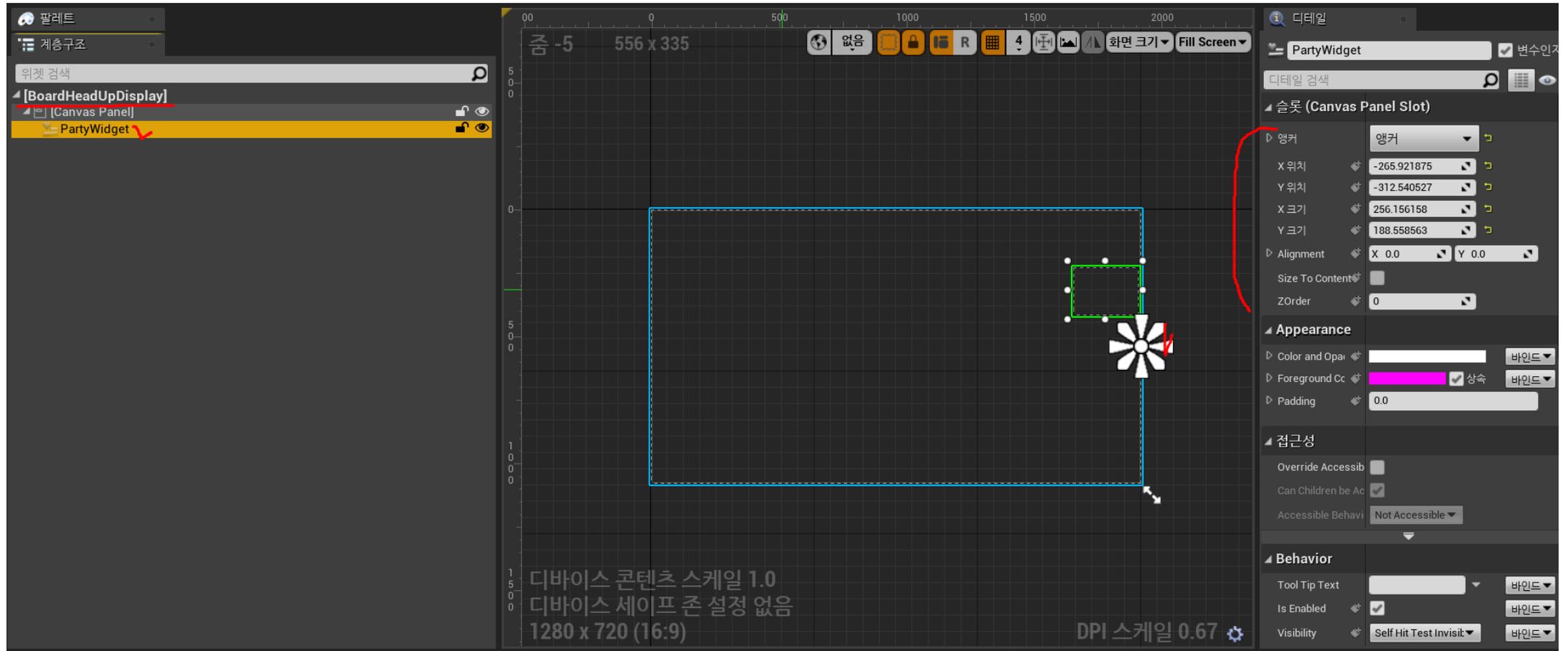
6.7.1 SaveData를 호출해서 Coin과 Treasure를 저장한다.



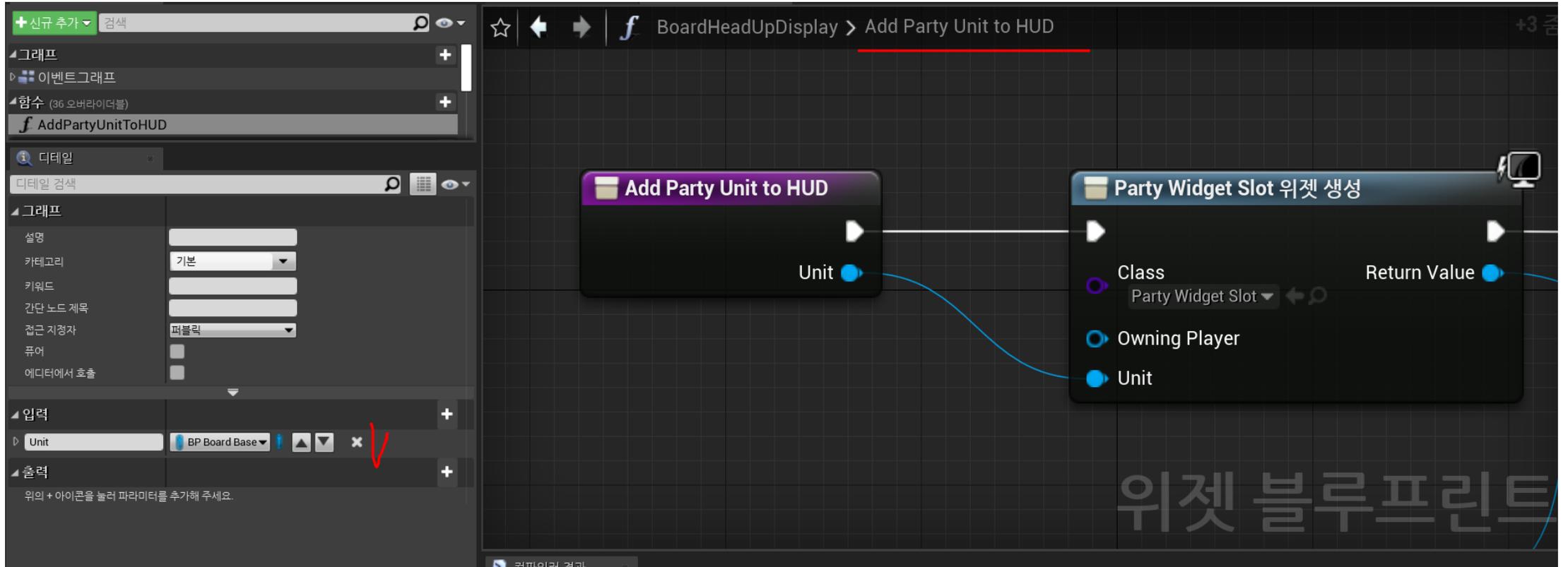
7. 위젯 블루프린트 BoardHeadUpDisplay를 생성한다.



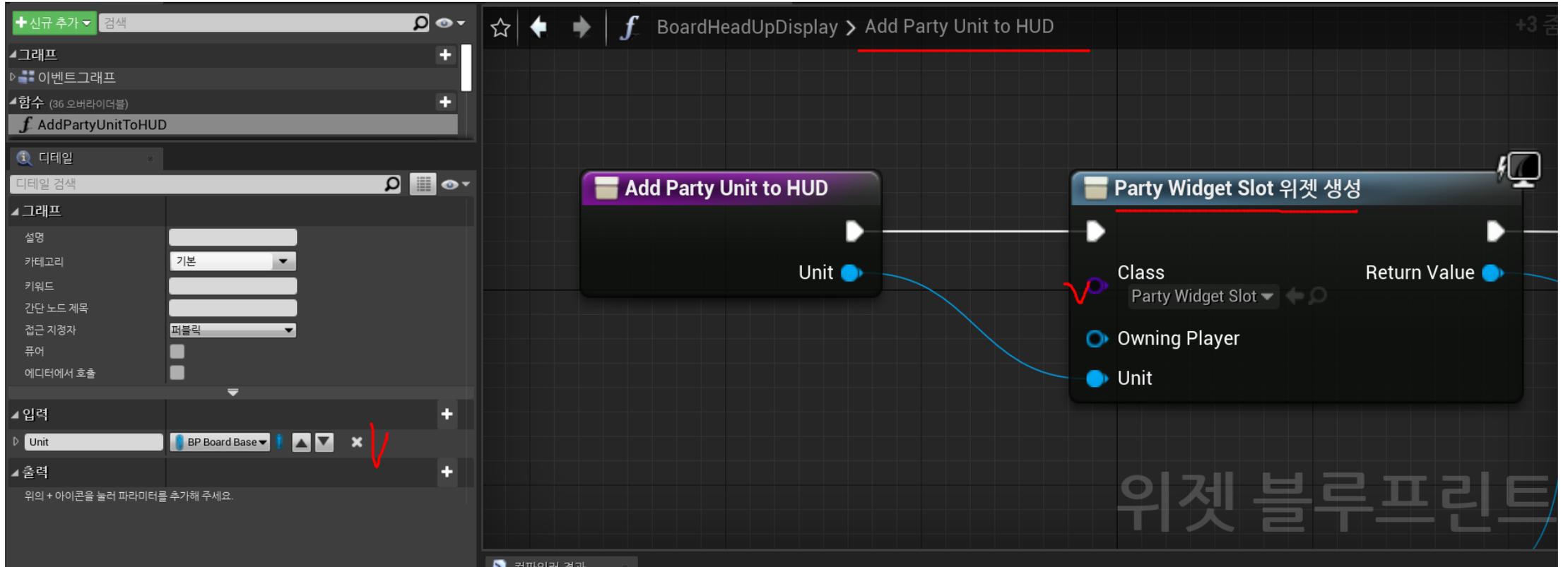
7.1 BoardGameHUD에 들어가서 PartyWidget을 제거한다.



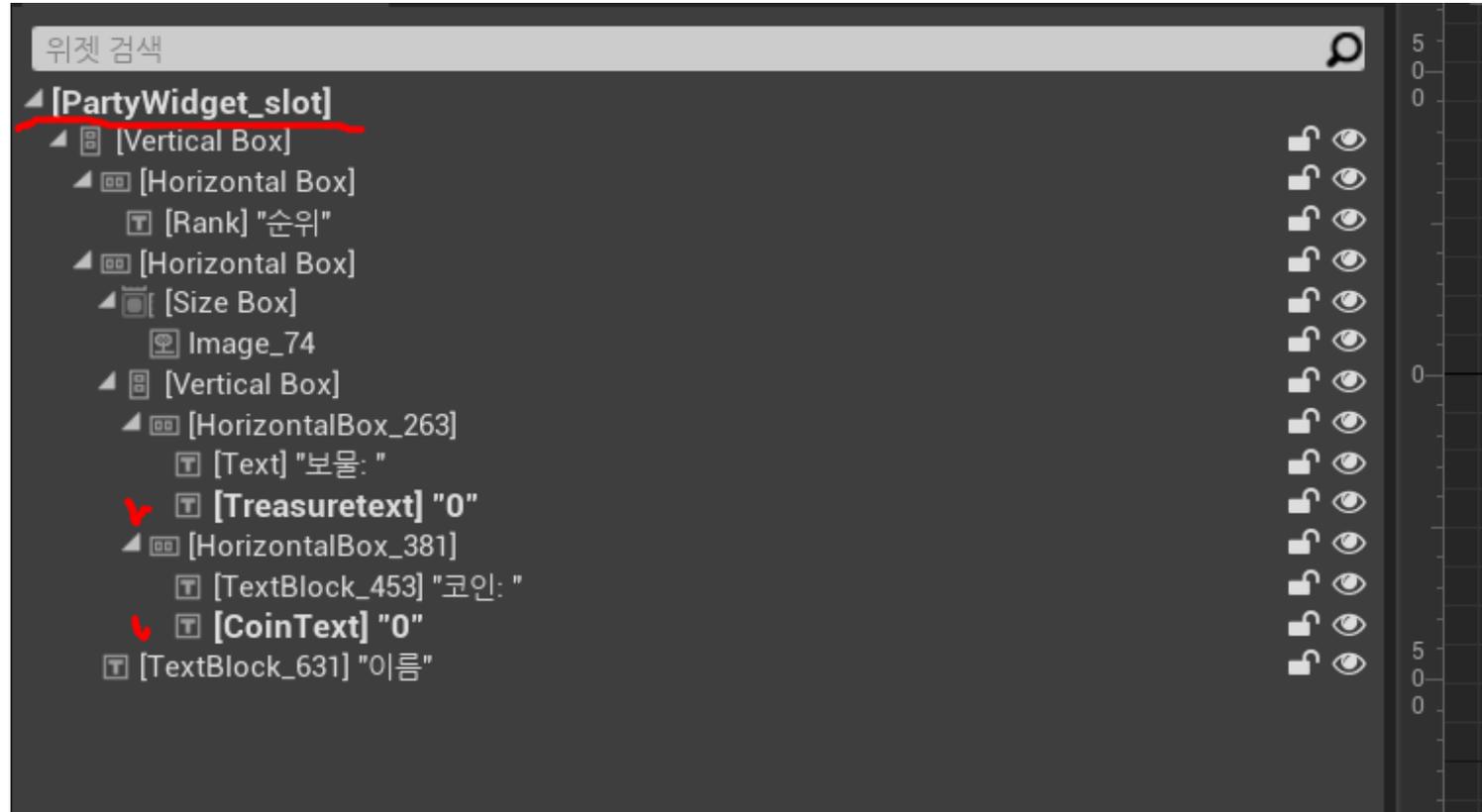
7.2 BoardHeadUpDisplay에 들어가서 PartyWidget을 추가한다.



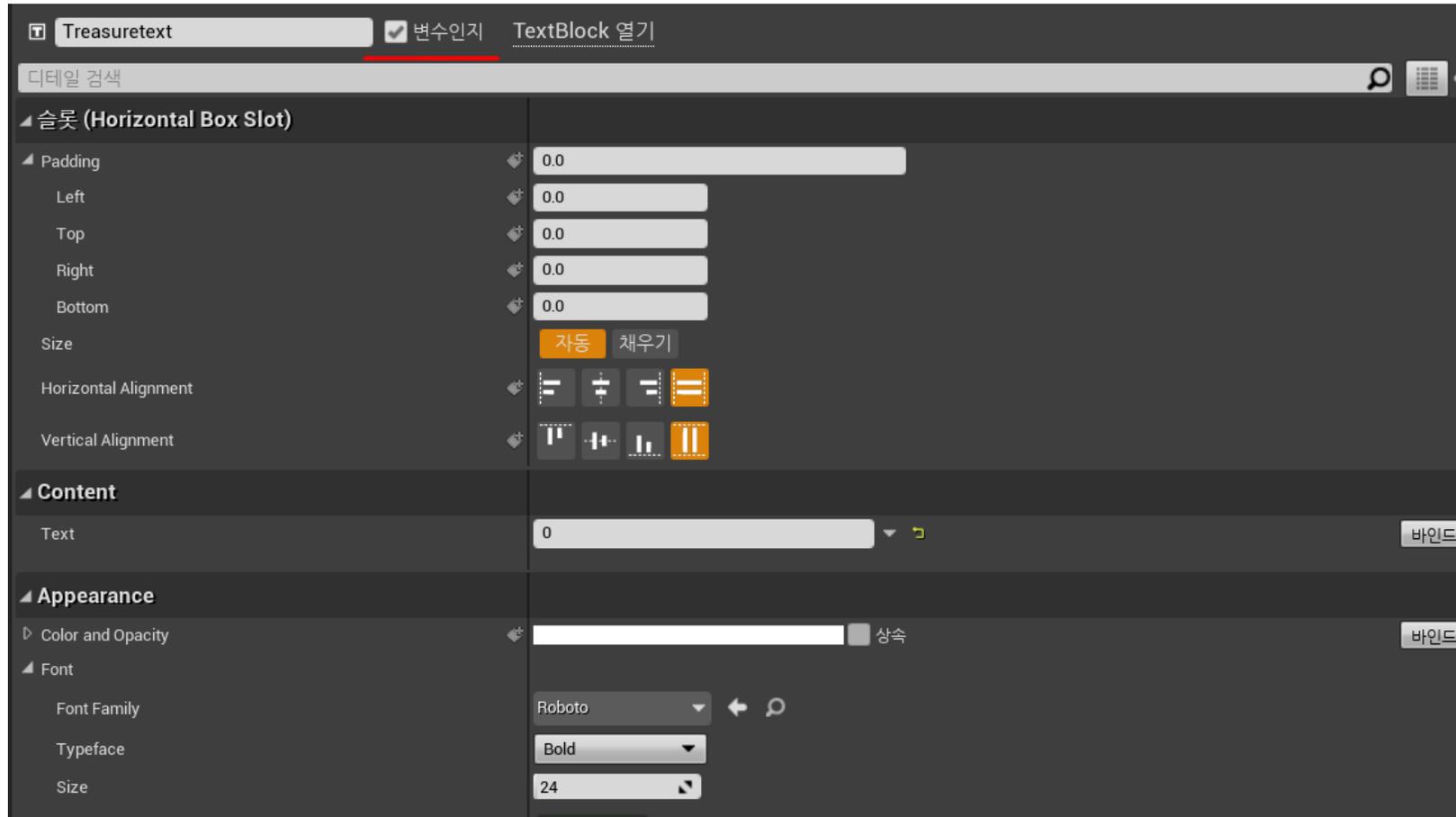
7.3 AddPartyUnittoHUD 함수를 생성하고, 입력을 BP_BoardBase Unit으로 한다.



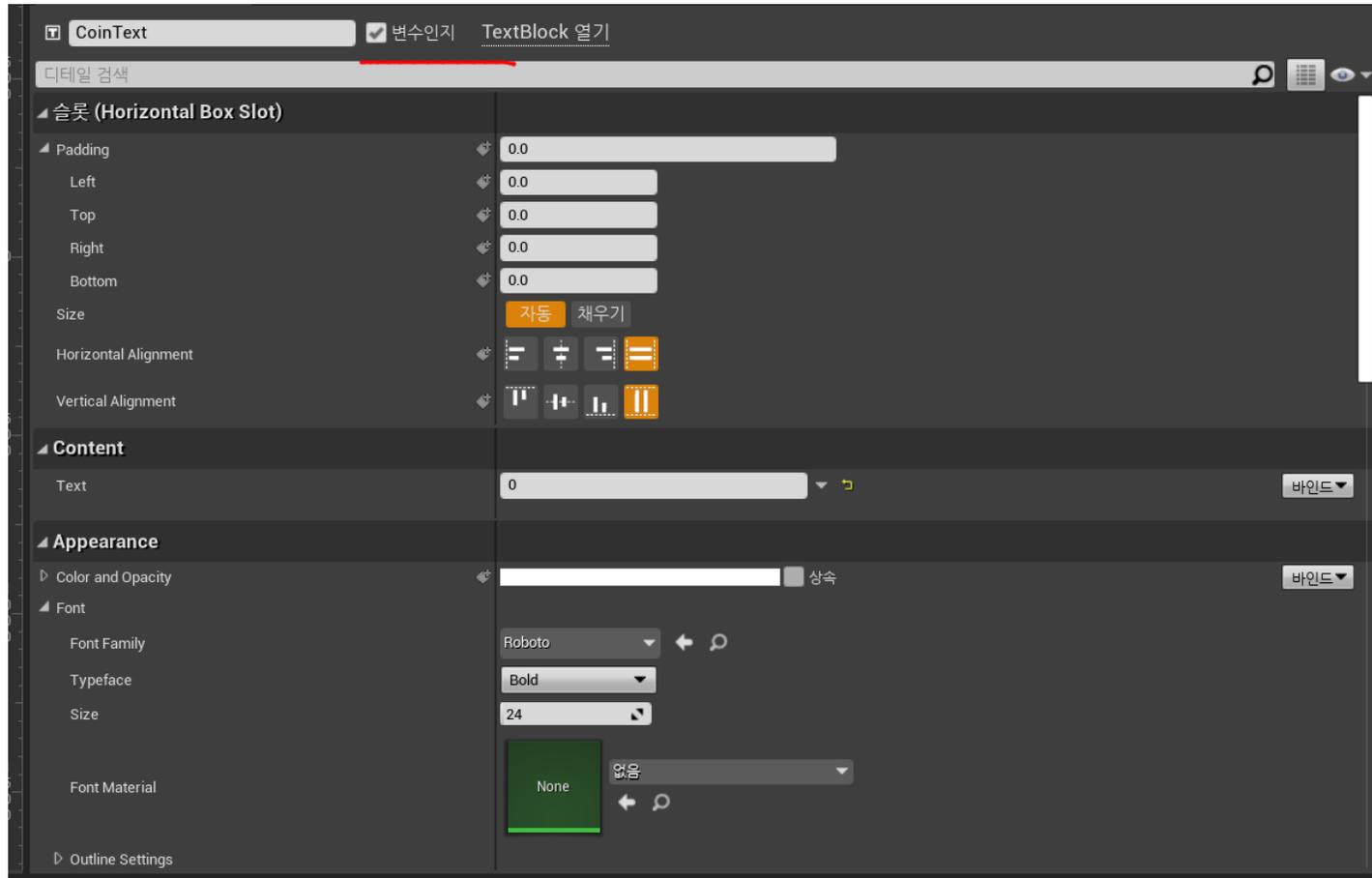
7.3.1 PartyWidget_Slot 위젯 생성을 한다.



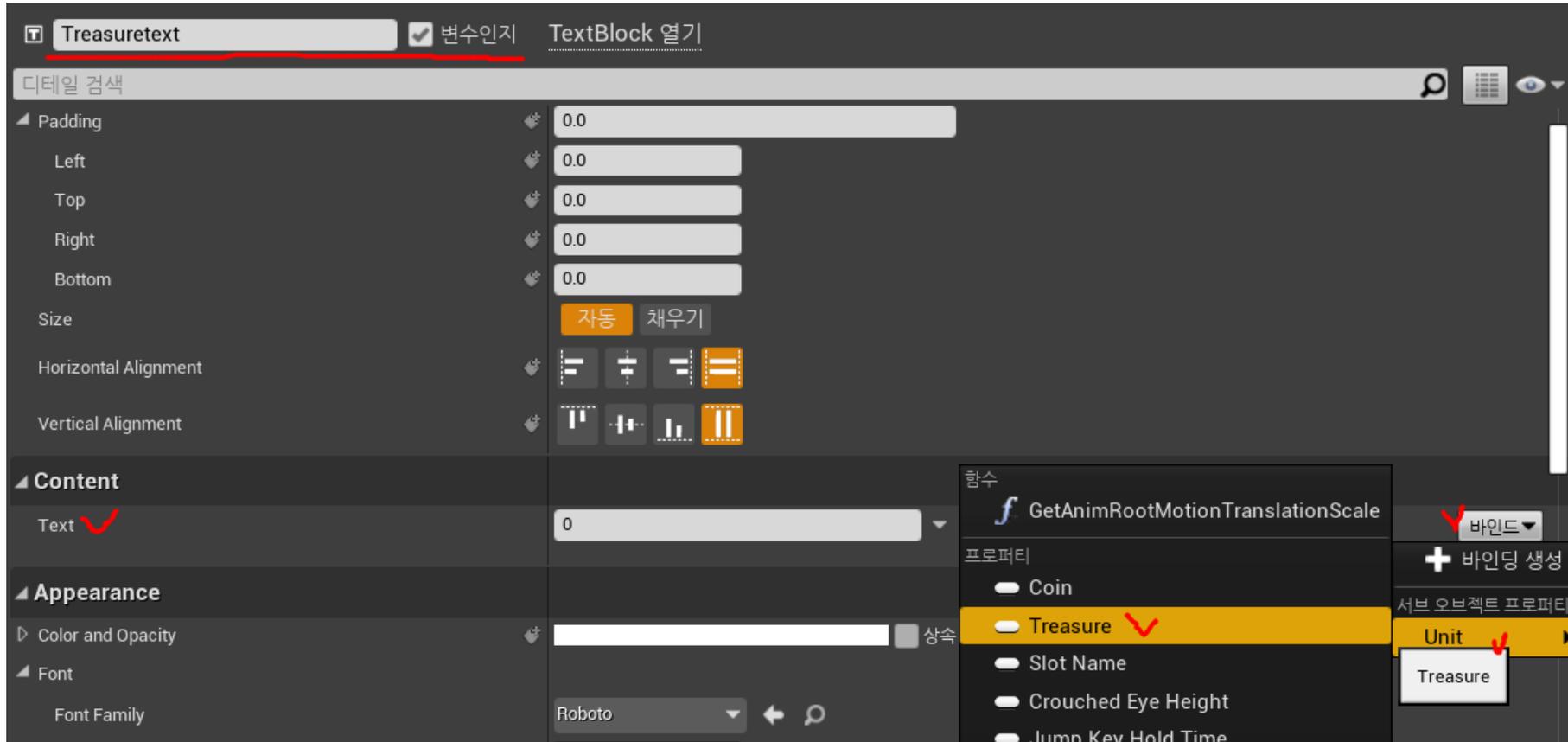
8. PartyWidget_Slot에 들어가서 Text 위젯의 이름을 각각 TreasureText, CoinText로 변경한다.



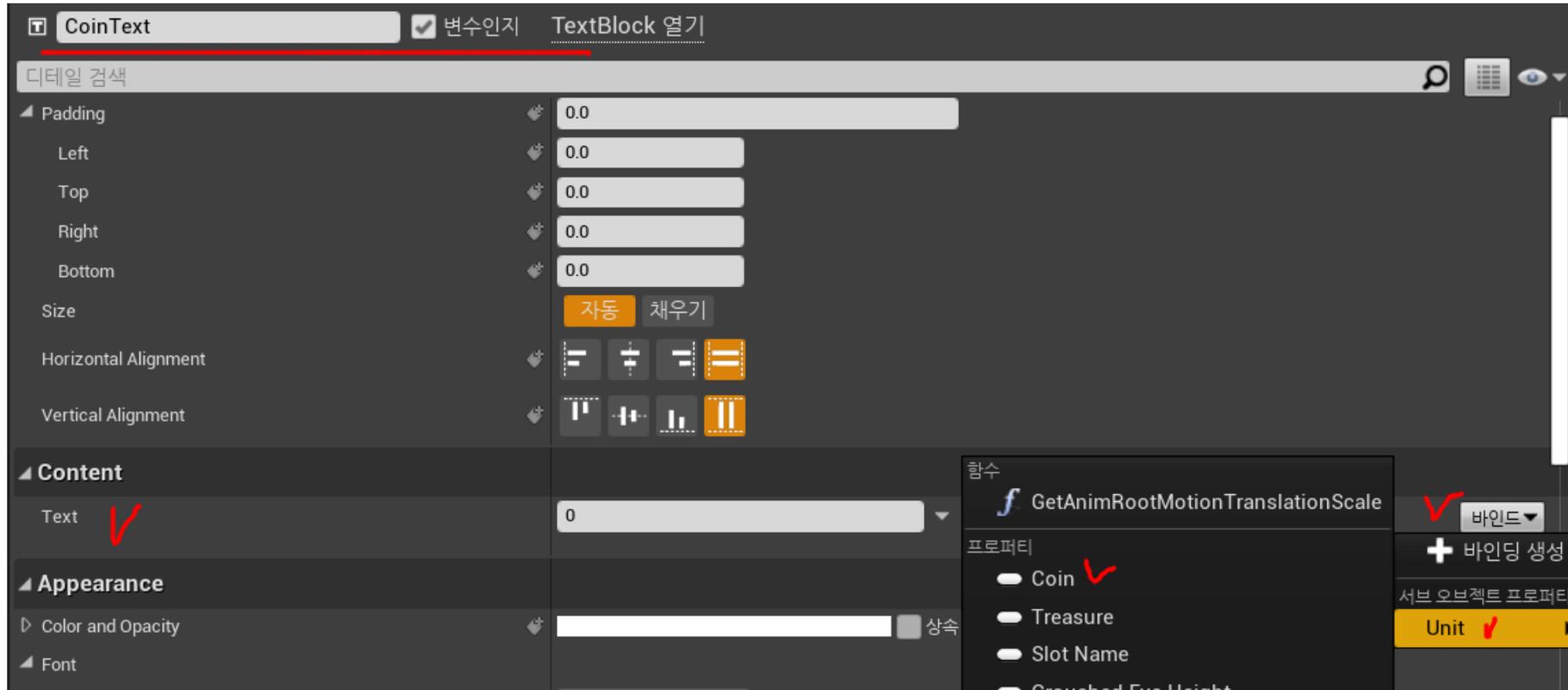
8.1 TreasureText를 변수로 만든다.



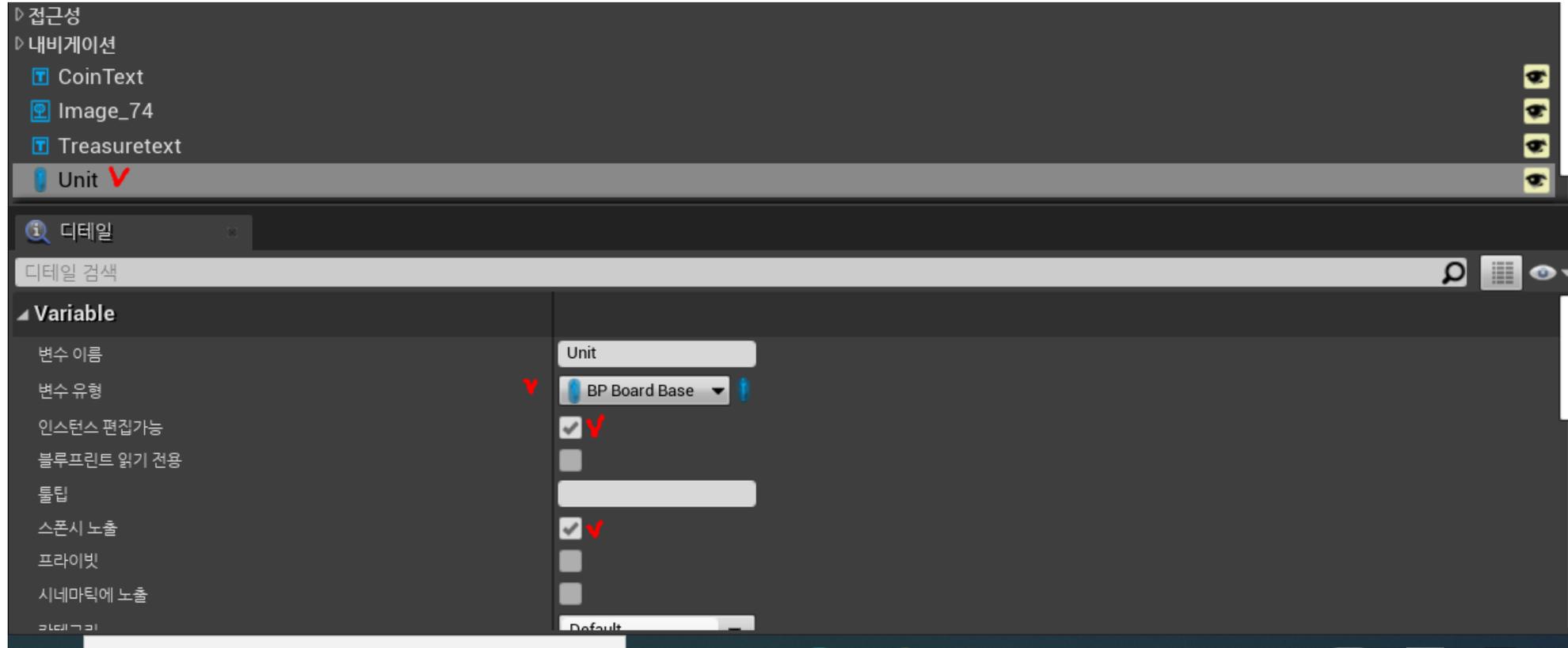
8.1.1 CoinText를 변수로 만든다.



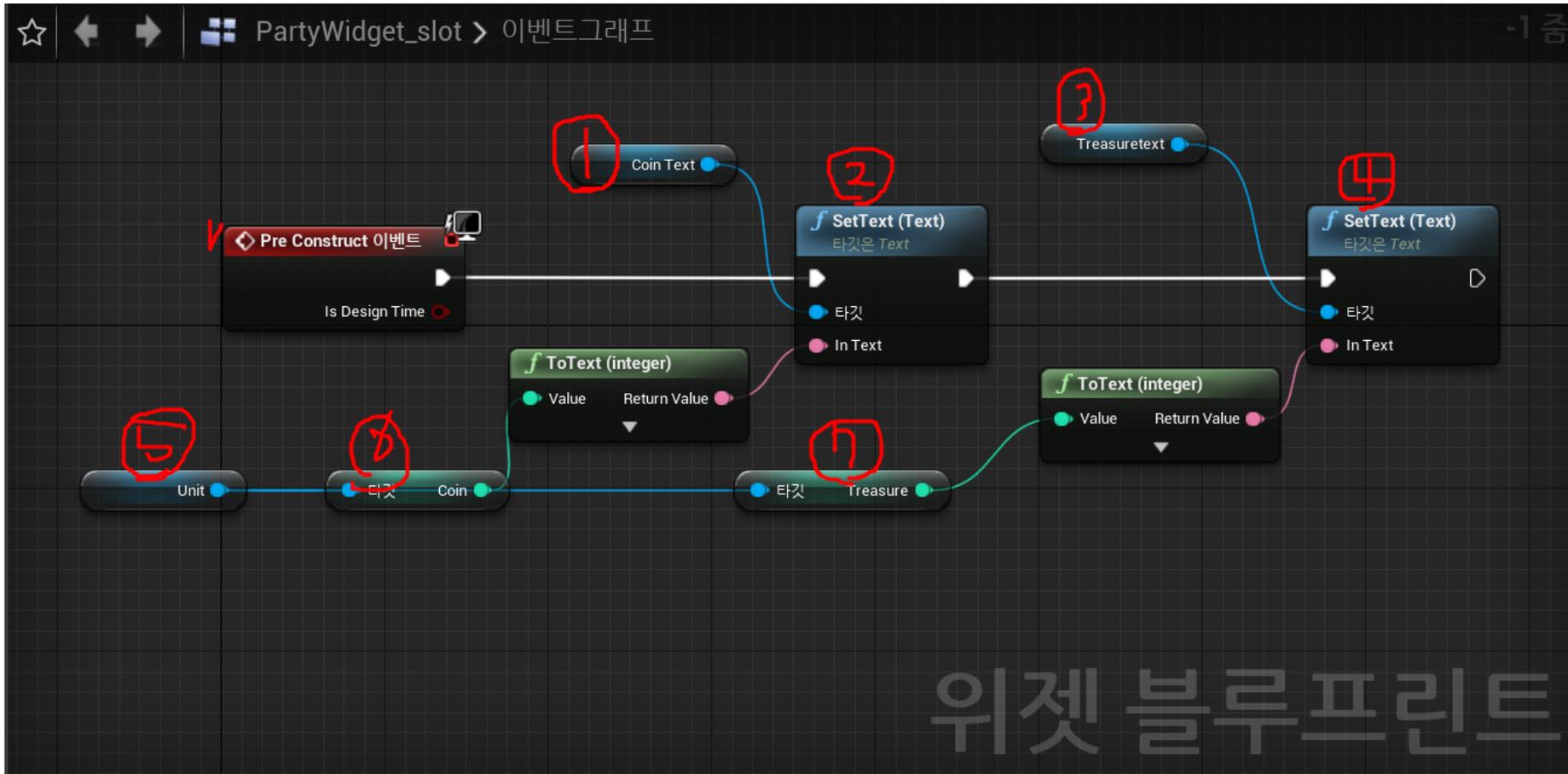
8.1.2 Treasuretext의 Text를 Unit의 Treasure에 바인딩한다.



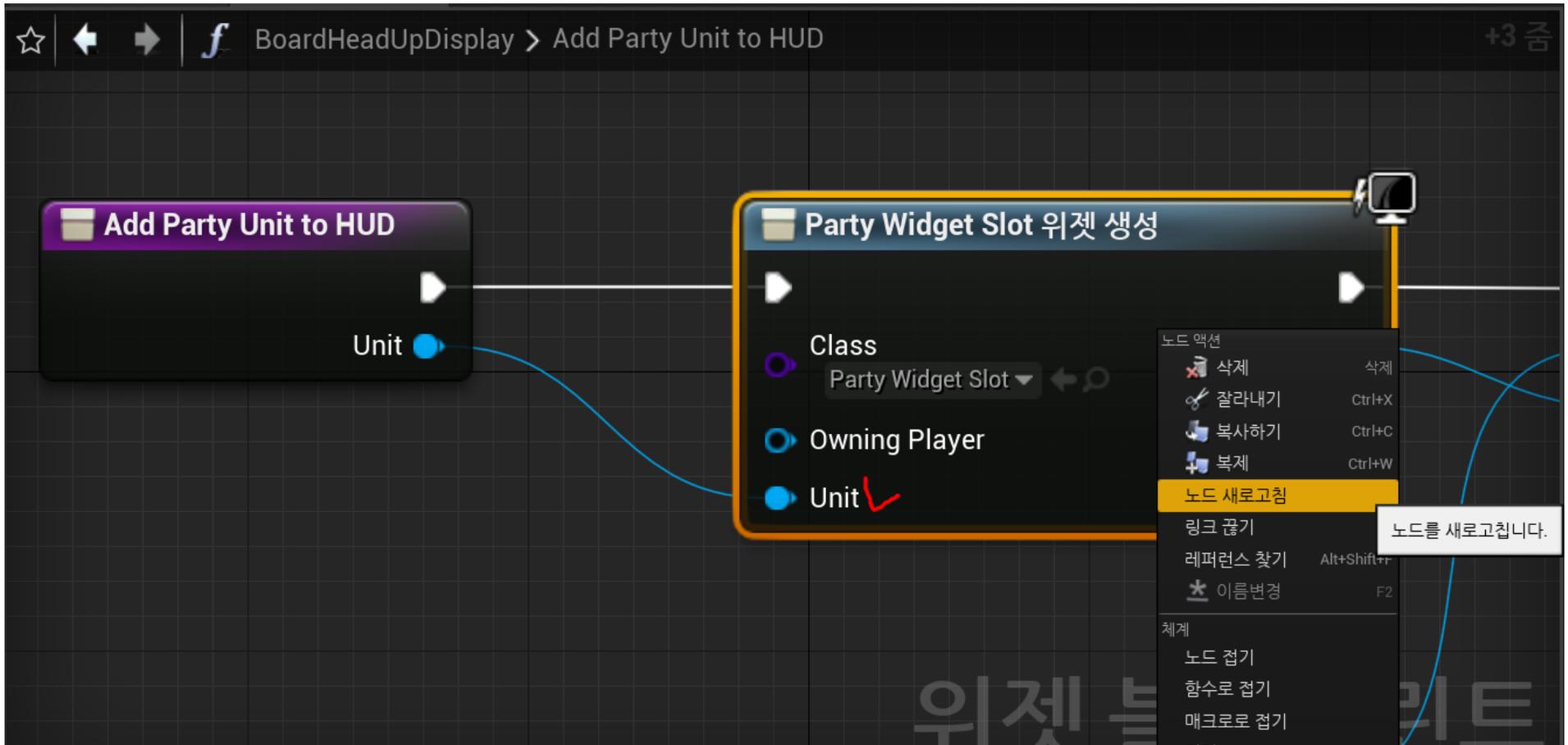
8.1.3 CoinText의 Text를 Unit의 Coin에 바인딩한다.



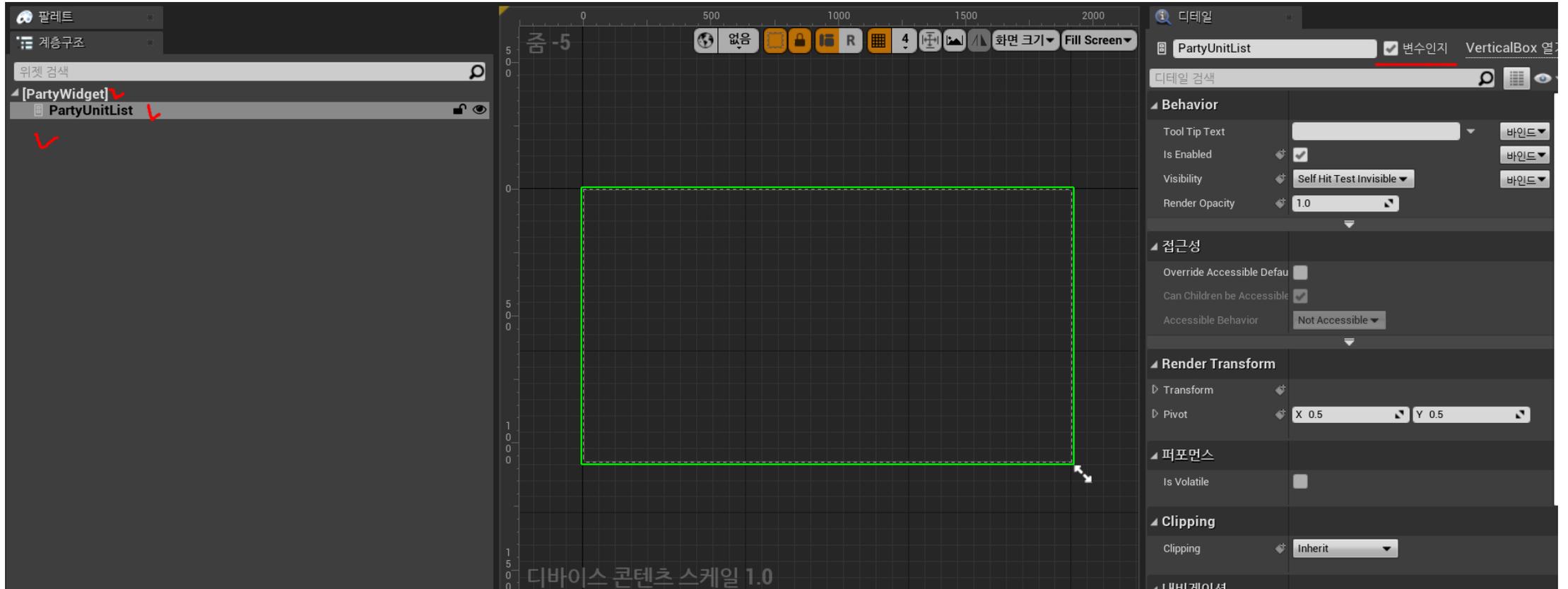
8.2 BP_BoardBase 형 변수 Unit을 생성하고, 인스턴스 편집, 스폰시 노출을 체크한다.



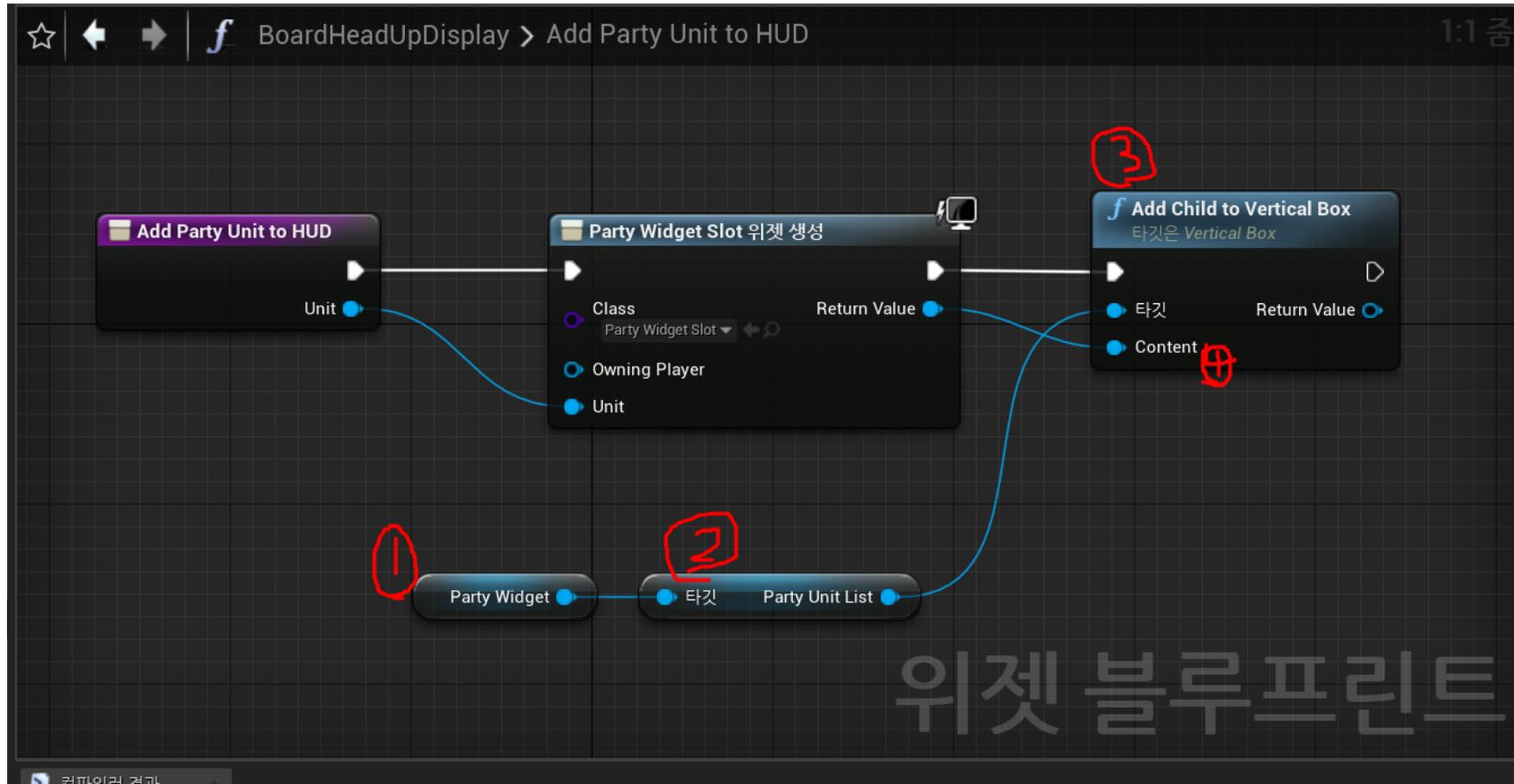
8.3 CoinText, TreasureText의 SetText 함수를 호출하고, Unit의 Coin과 Treasure를 Get해서 각각 연결한다.



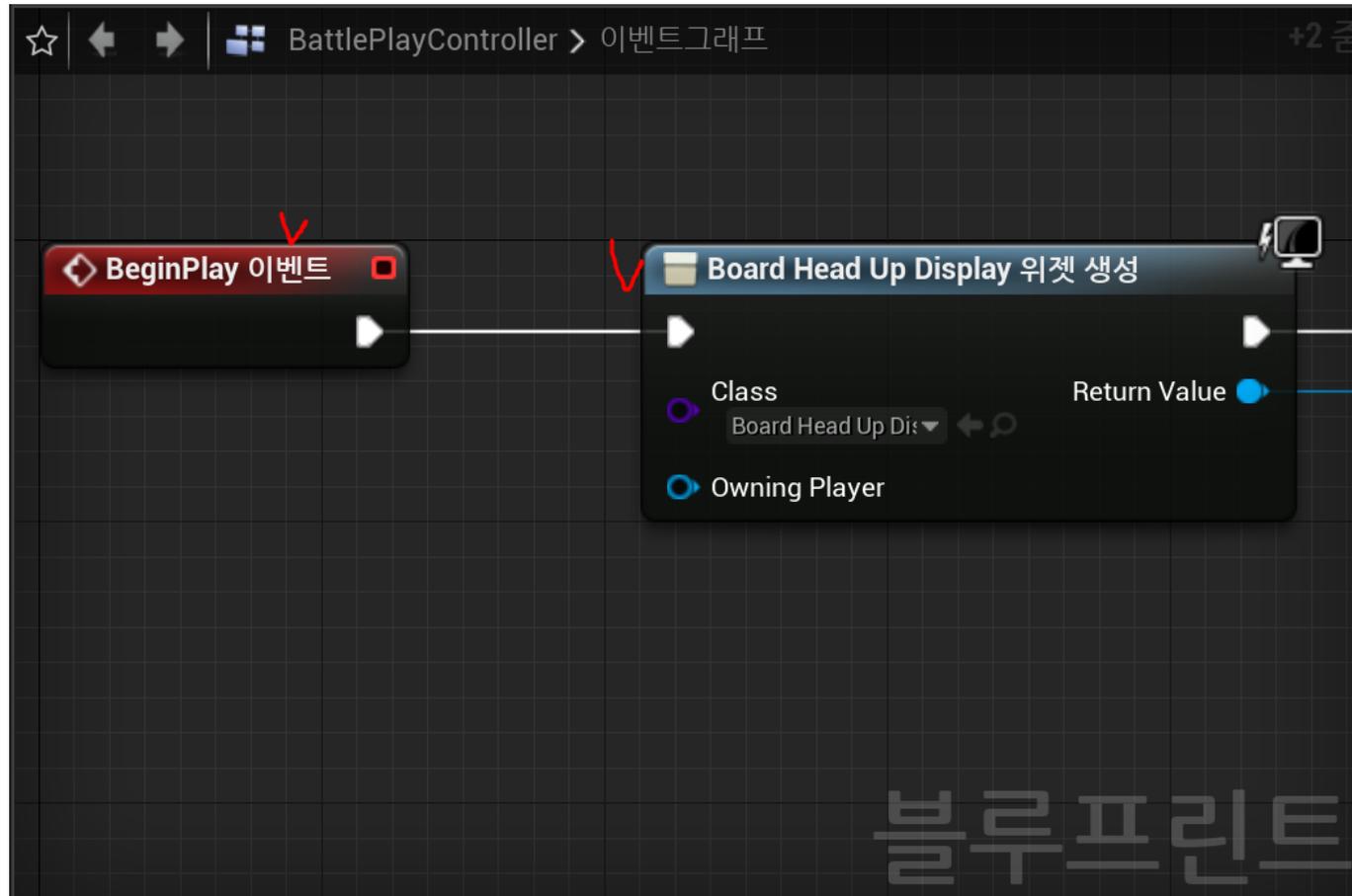
8.4 BoardHeadUpDisplay의 AddPartyUnittoHUD에 들어가서 PartyWidget_Slot 위젯 생성 노드를 새로고침해서 Unit을 연결한다.



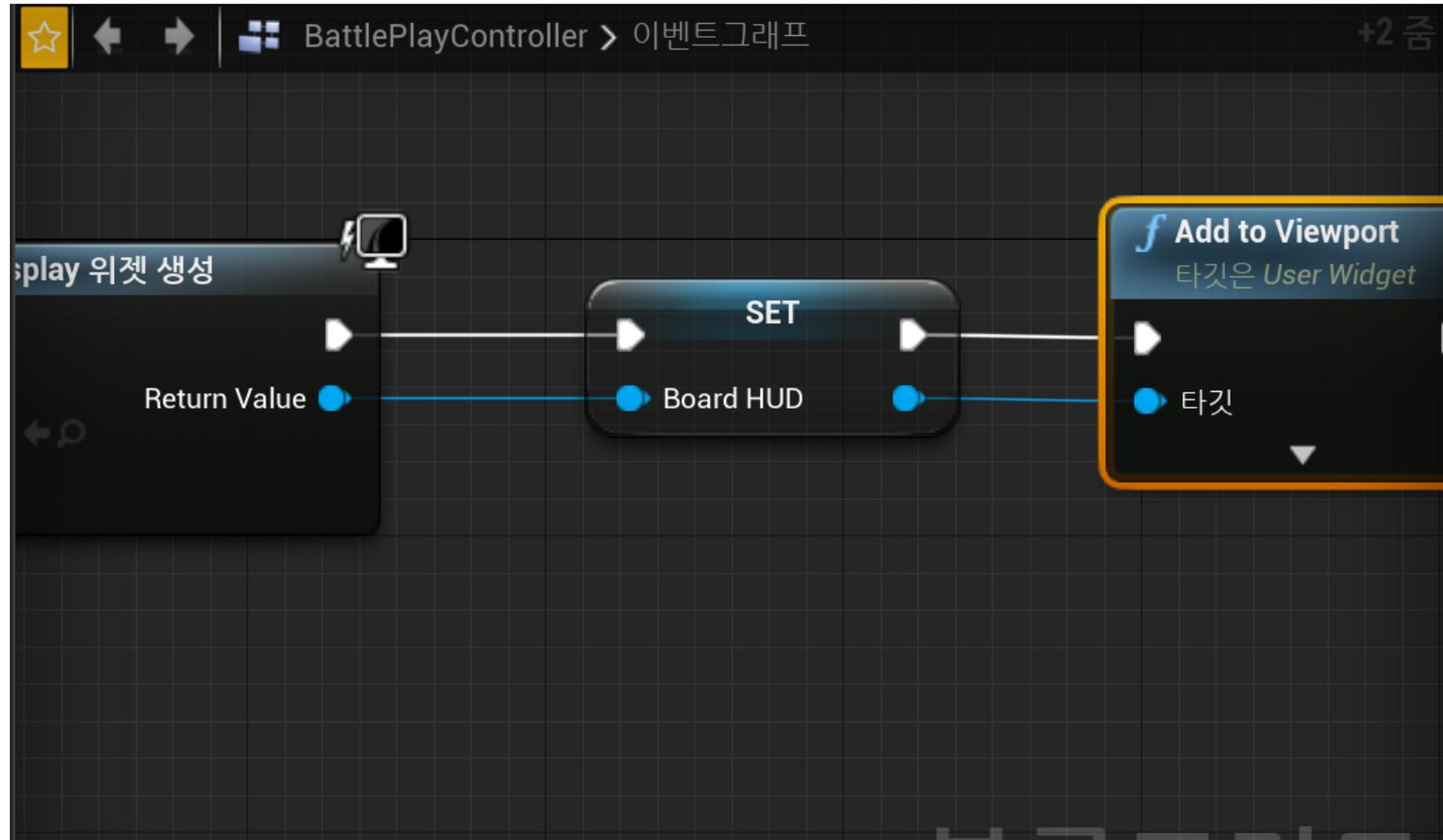
9. PartyWidget에 들어가서 VerticalBox를 PartyUnitList로 변경하고, 추가했던 PartyWidget_Slot을 제거한다. 또, PartyUnitList를 변수로 만든다.



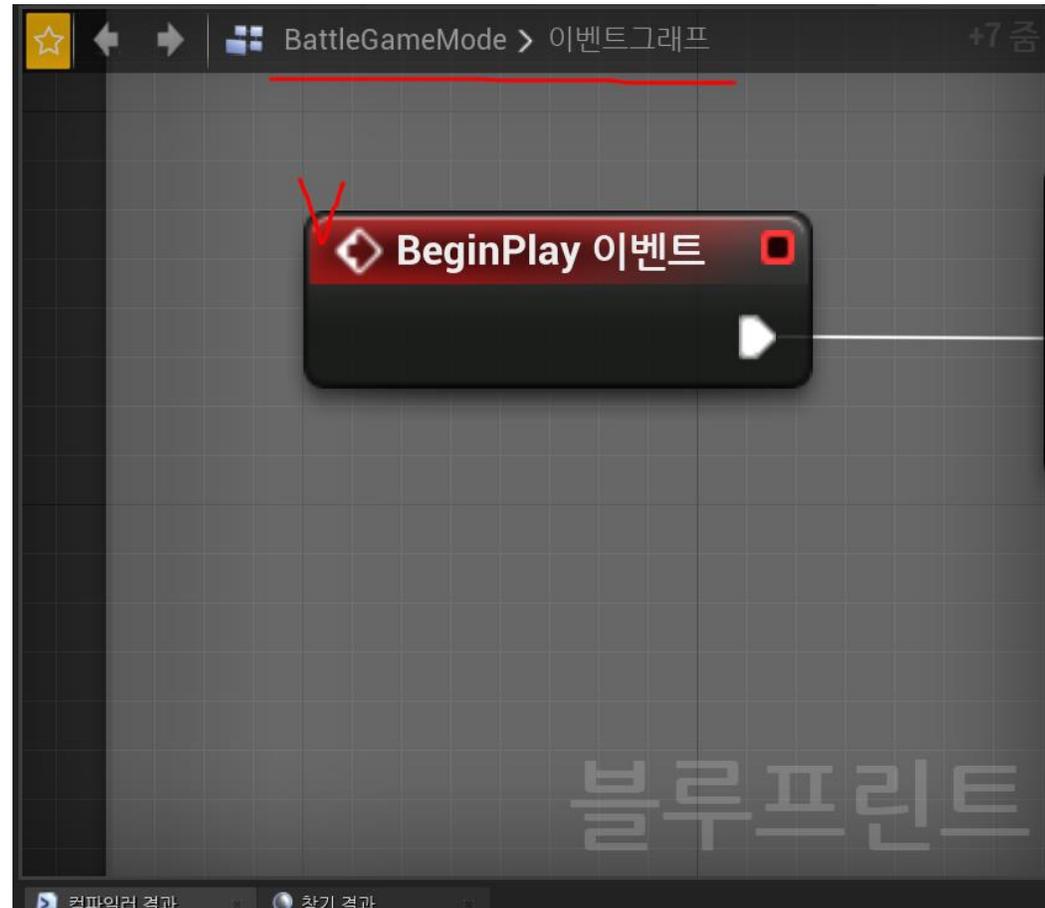
9.1 BoardHeadUpDisplay의 AddPartyUnittoHUD에 들어가서 PartyUnitList의 AddChildtoVecticalBox를 호출하고, Content에 PartyWidget_Slot을 넣는다.



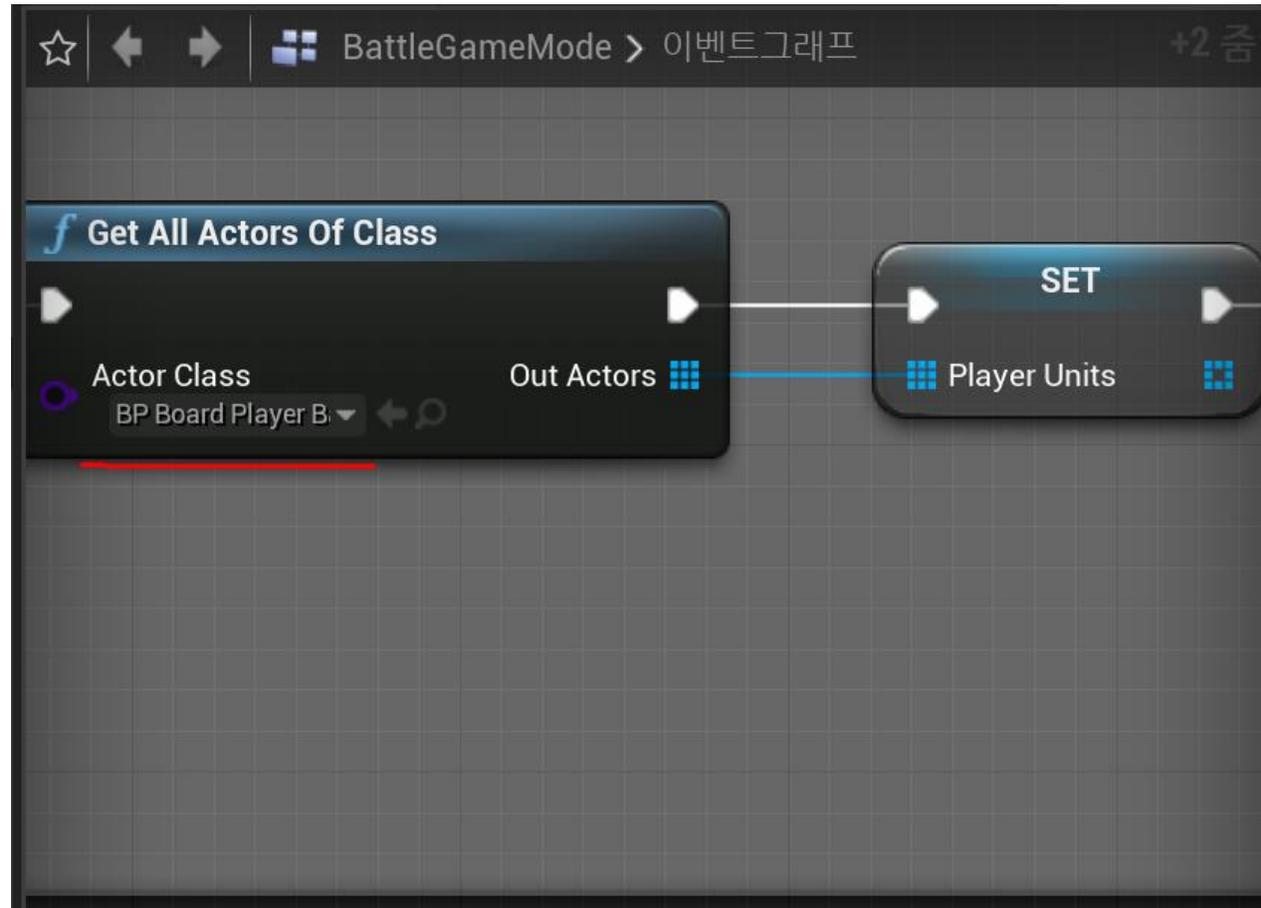
10. BattlePlayController에 들어가서 BoardHeadUpDisplay 위젯을 생성한다.



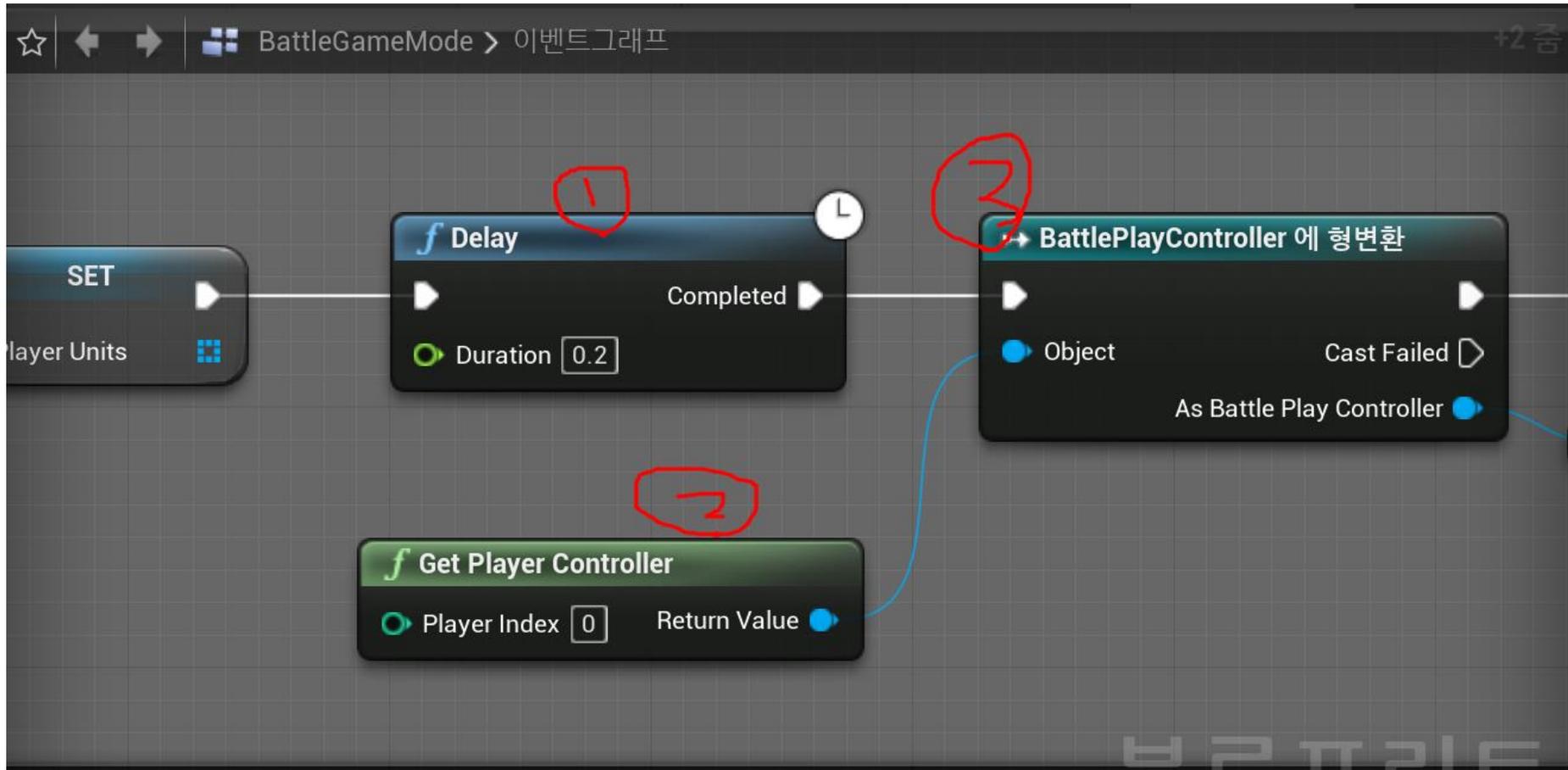
10.1 생성한 위젯을 변수로 승격해서 BoardHUD를 생성하고, AddtoViewport에 연결한다.



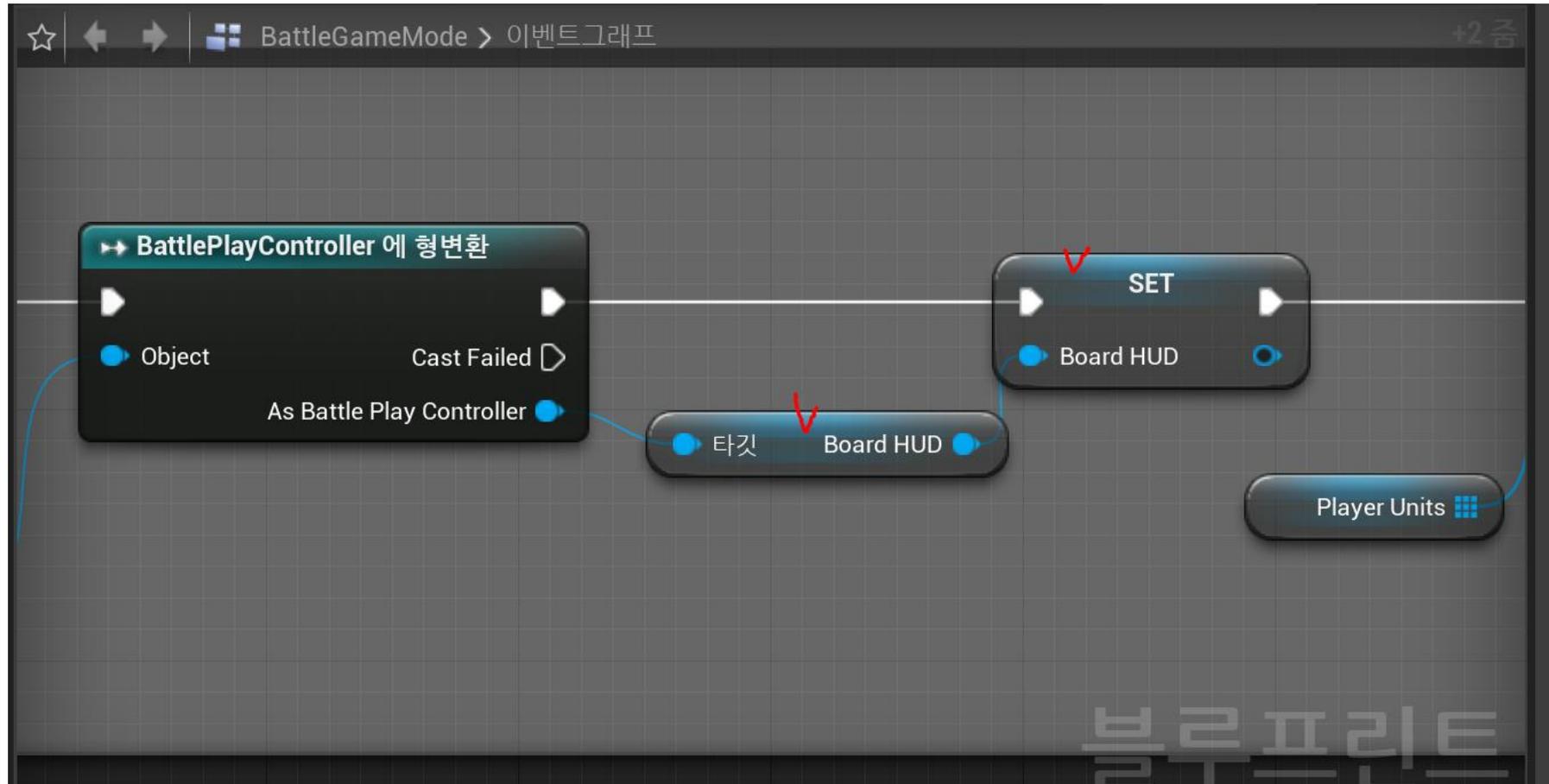
11. BattleGameMode의 BeginPlay 이벤트에 들어간다.



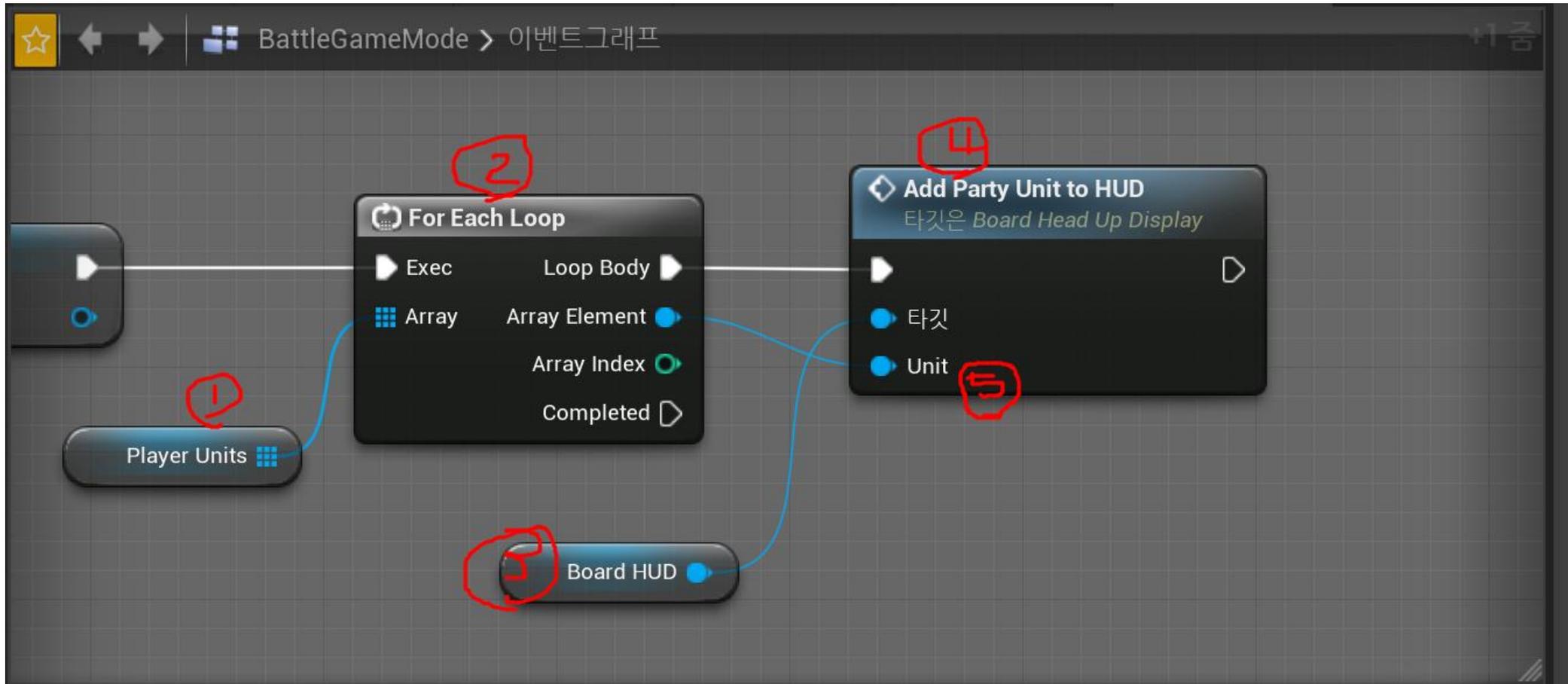
11.1 월드에 배치된 모든 BP_BoardPlayerBase를 가지고 와서 변수로 승격해서 PlayerUnits을 생성한다.



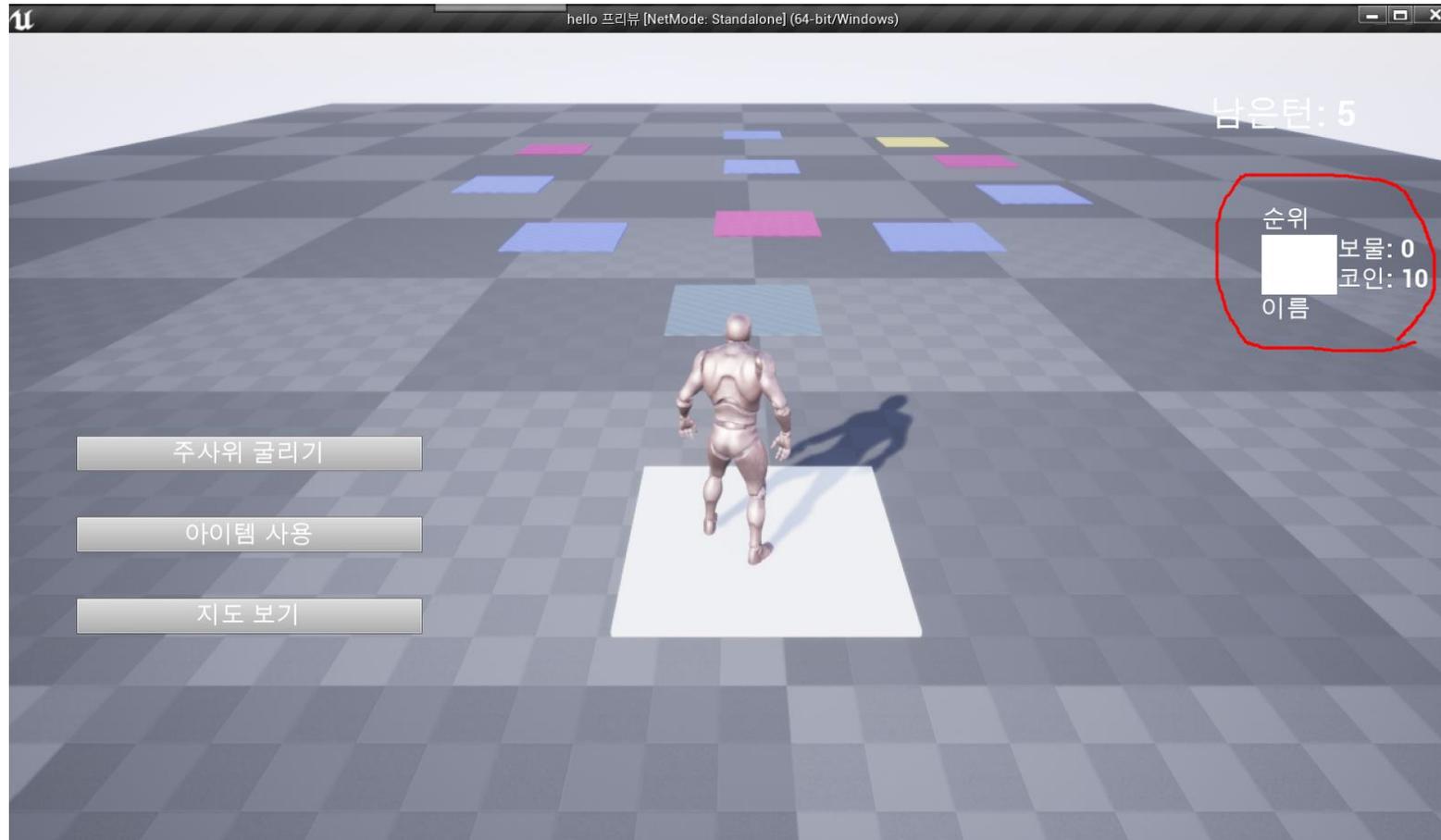
11.2 스폰하는 액터로 인한 오류를 방지하기 위해 딜레이를 주고, BattlePlayController를 Cast한다.



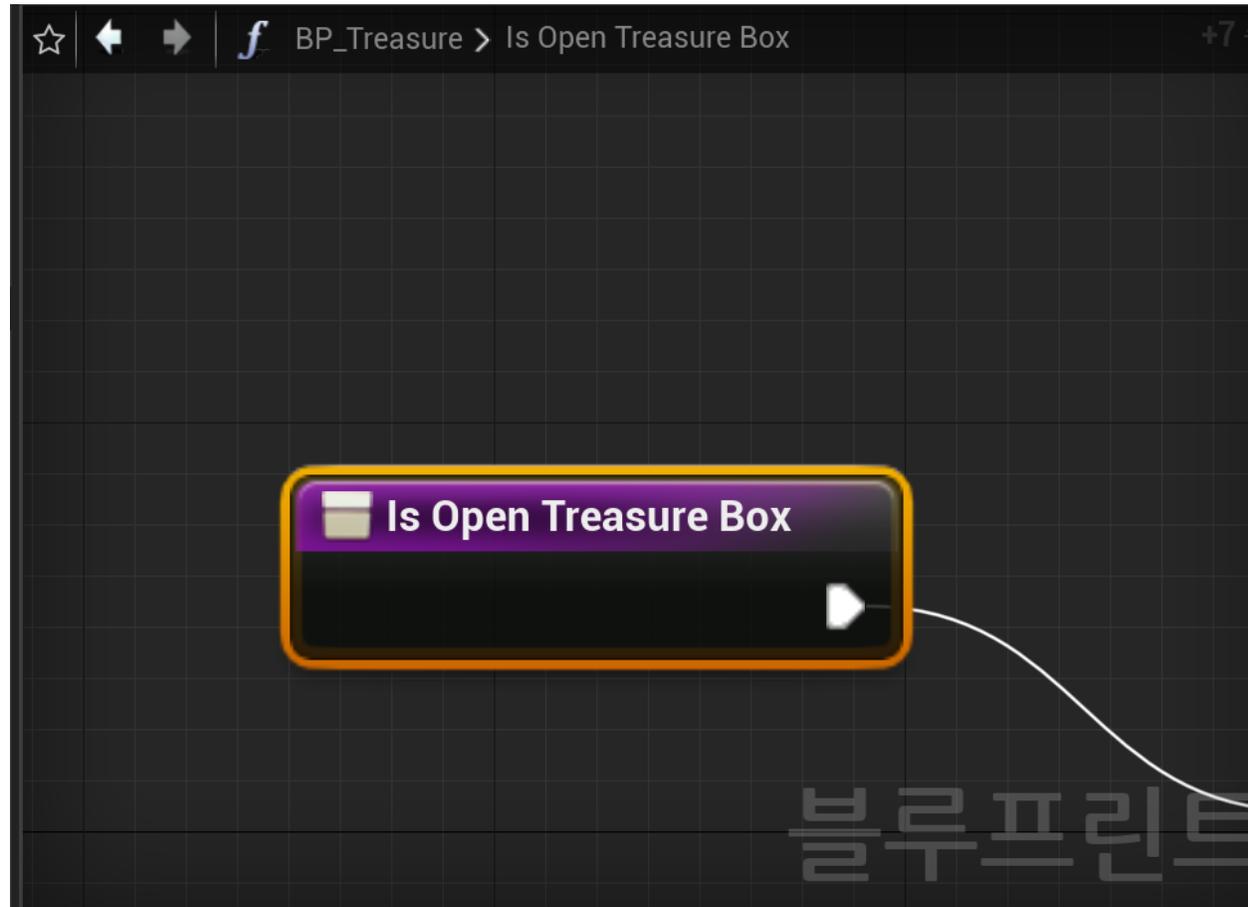
11.3 BattlePlayController의 BoardHUD를 GET하고, 변수로 승격해서 BoardHUD를 생성한다.



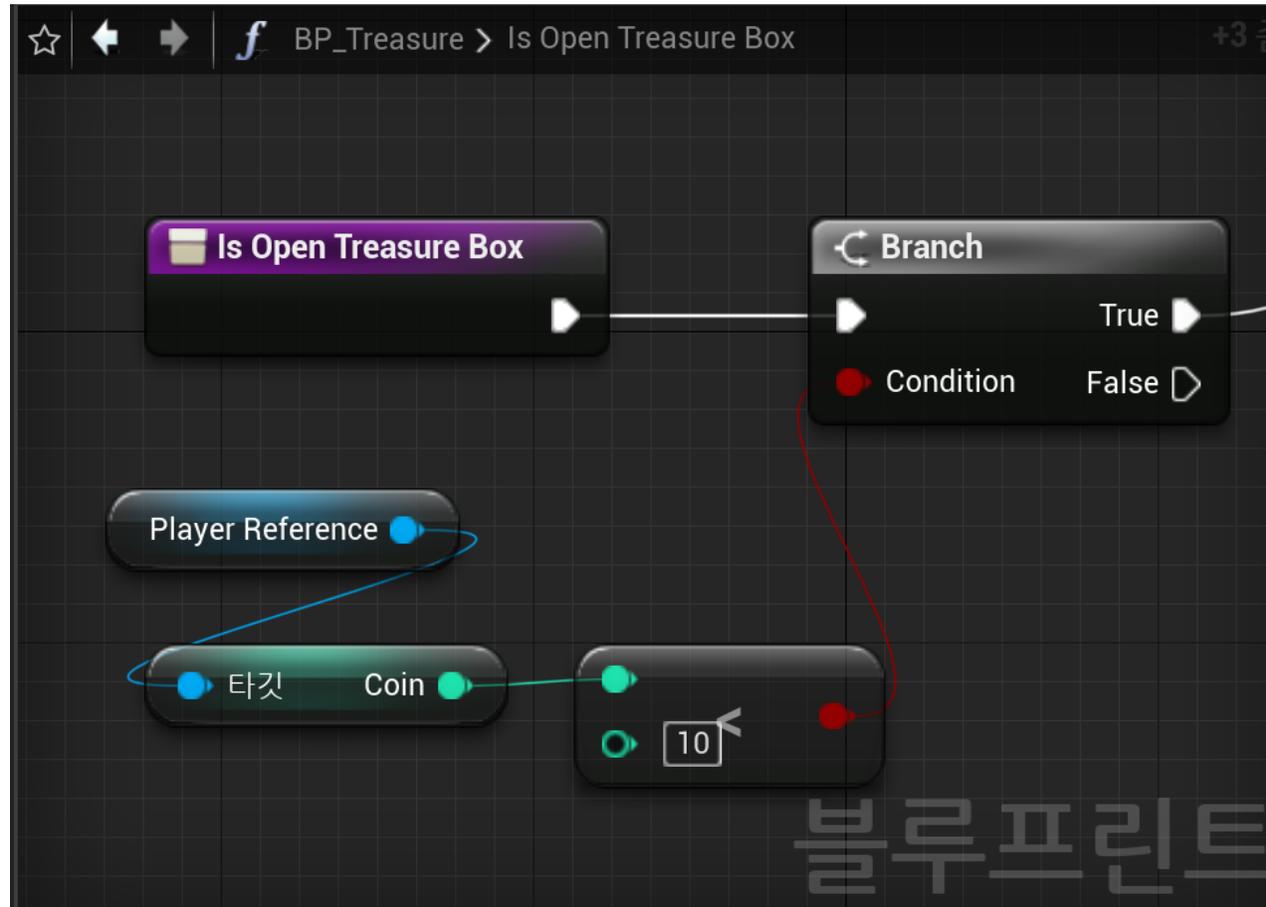
11.4 PlayerUnits의 element를 BoardHUD의 AddPartyUnittoHUD에 연결한다.



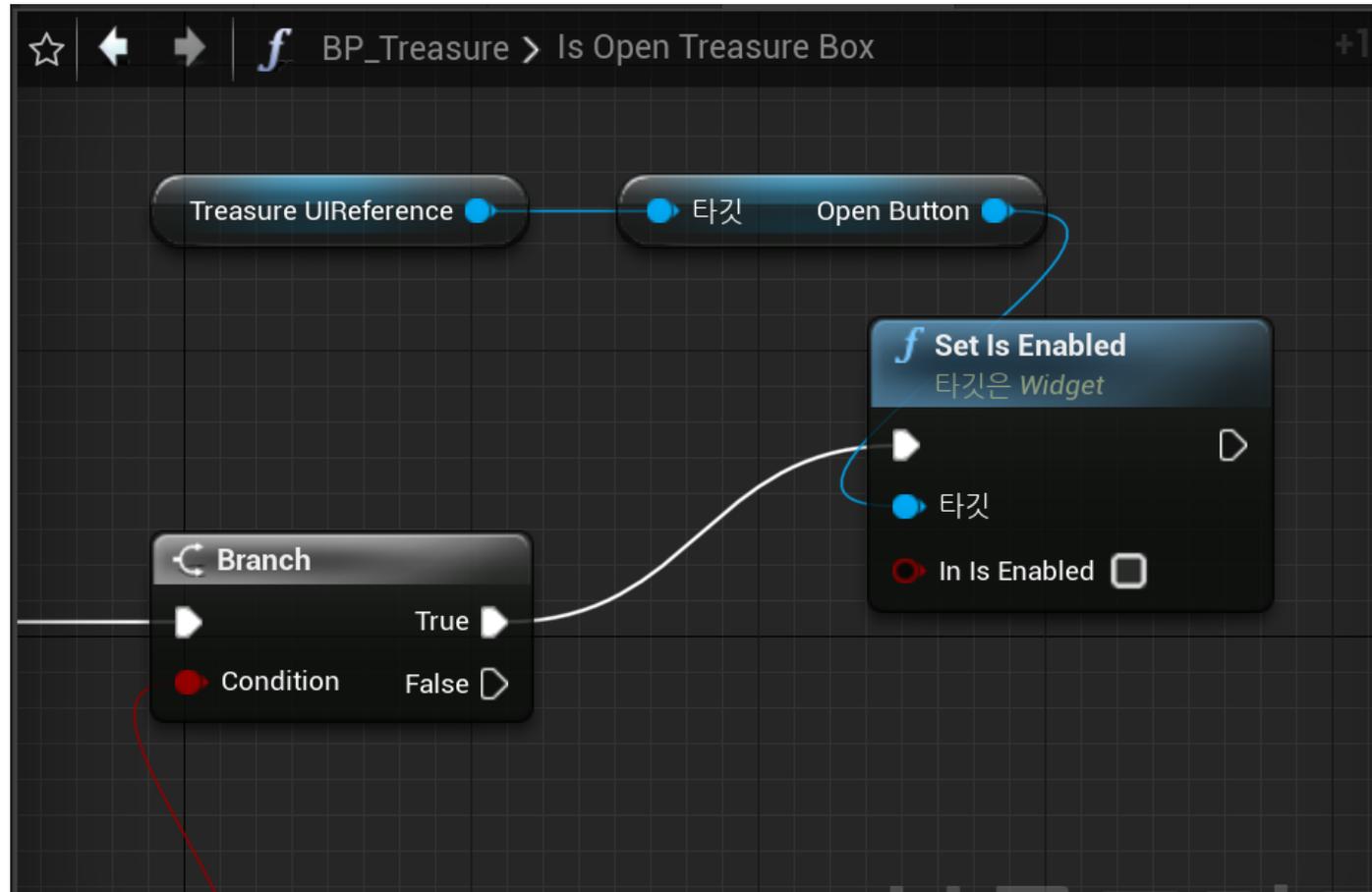
11.5 캐릭터의 정보가 연동된 결과 화면



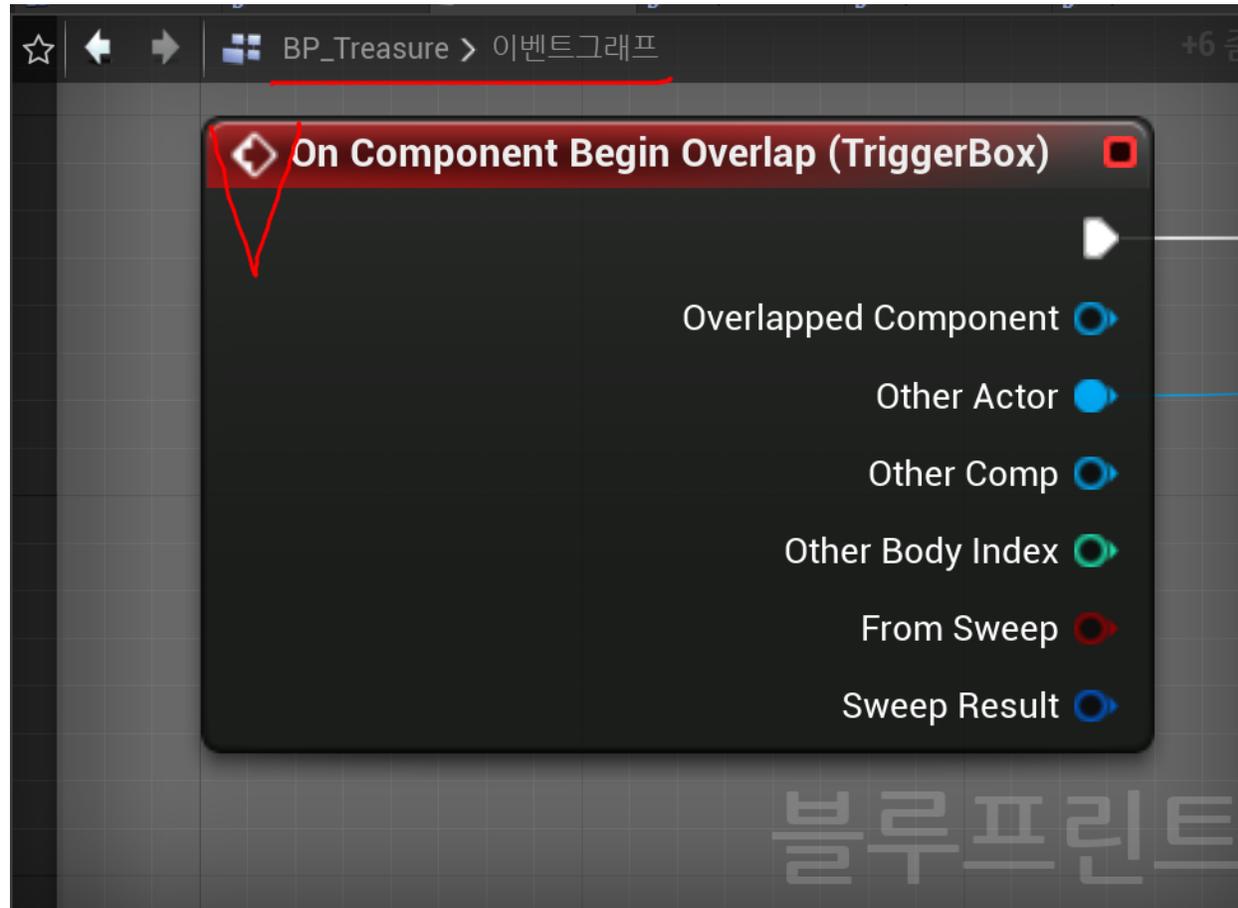
12. BP_Treasure에 들어가서 IsOpenTreasureBox 함수를 생성한다.



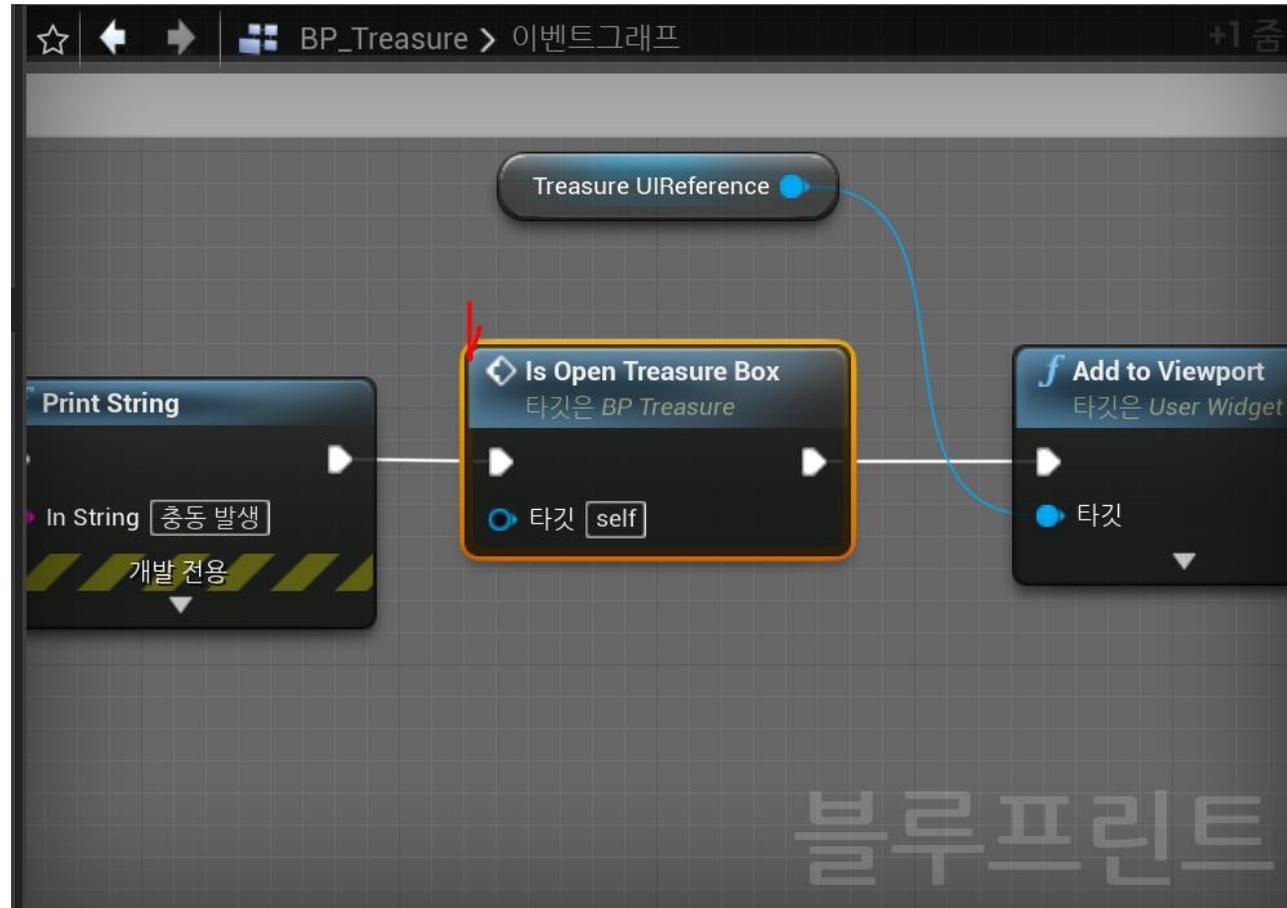
12.1 Player가 소유한 Coin이 10보다 작은지 판단한다.



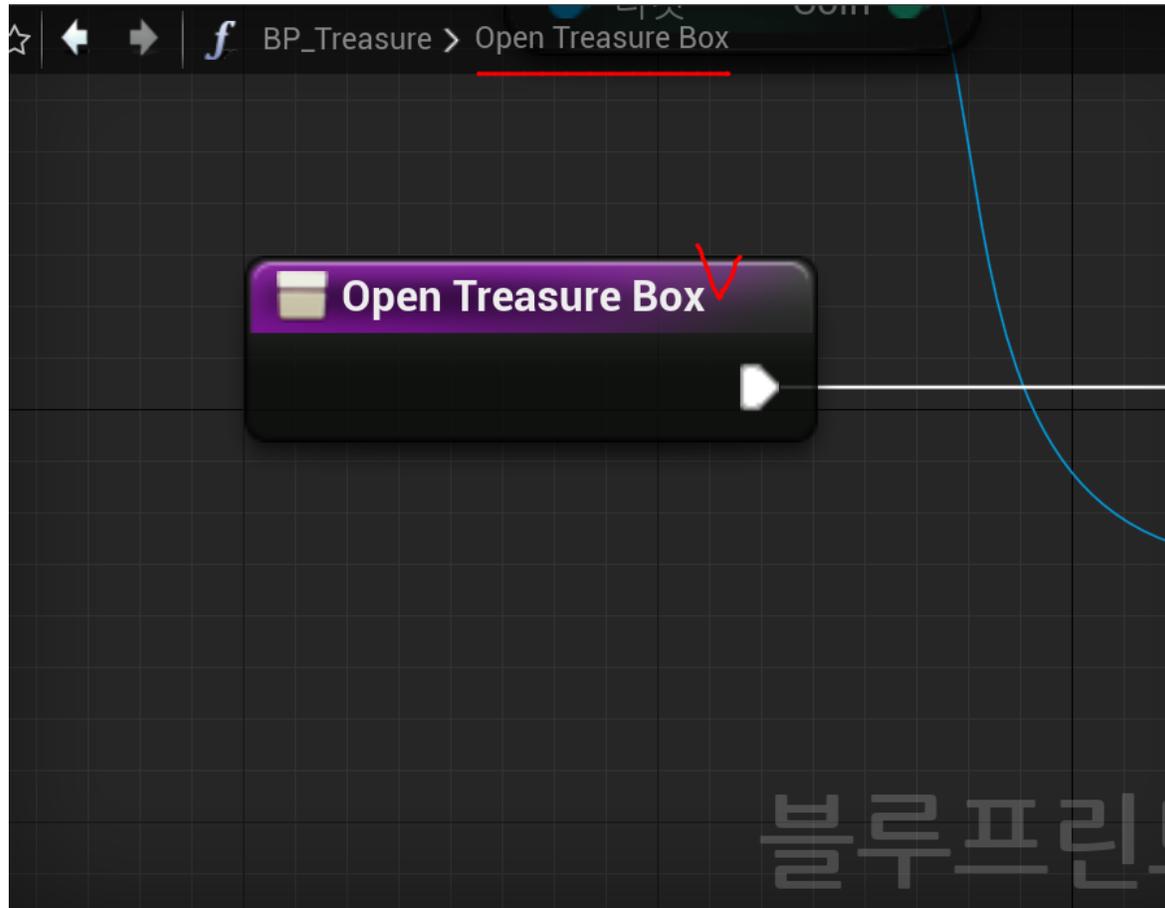
12.1.1 10보다 작으면 TreasureUI의 OpenButton을 비활성화 시킨다.



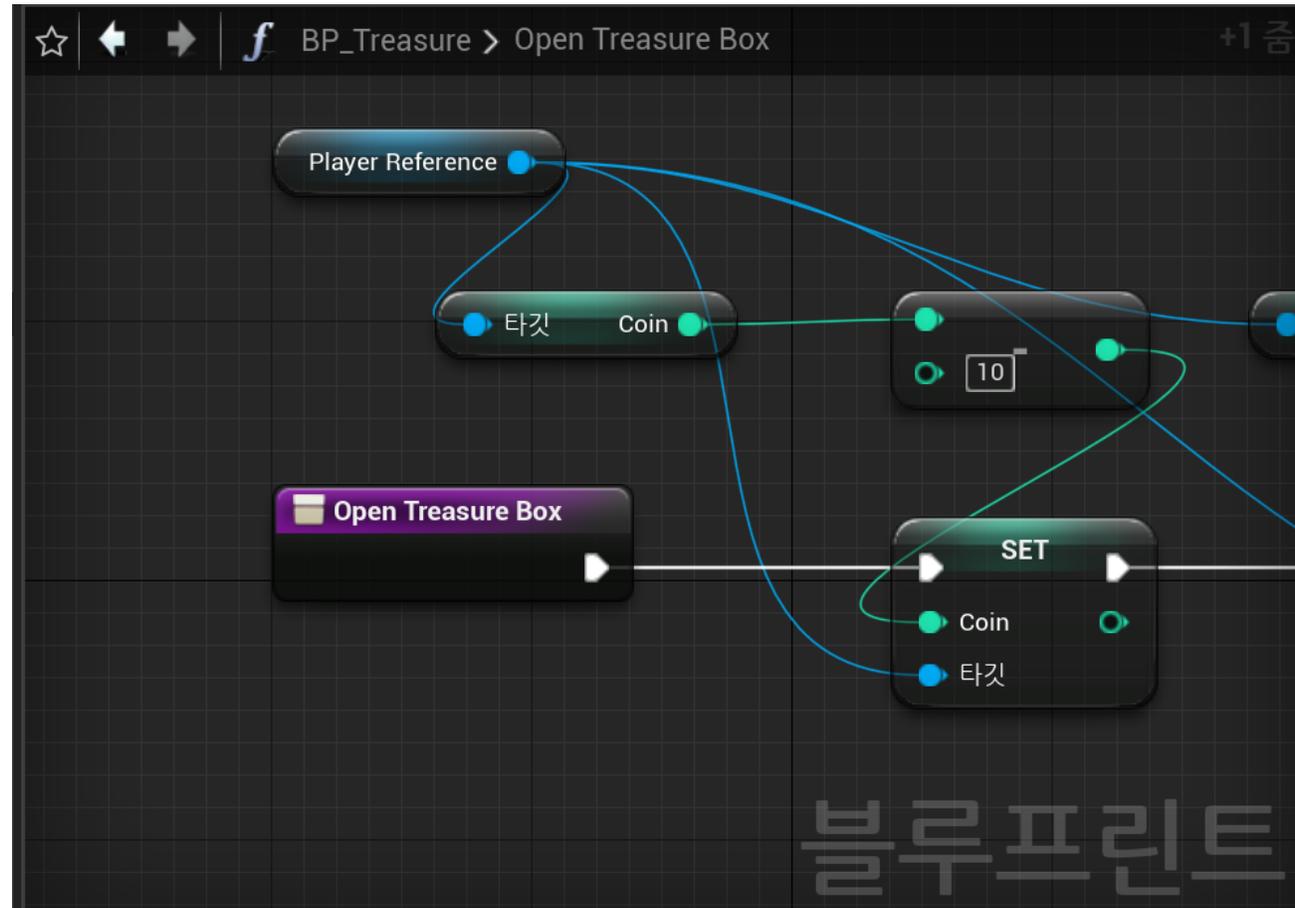
12.2 OnComponentBeginOverlap(TriggerBox) 이벤트에 들어간다.



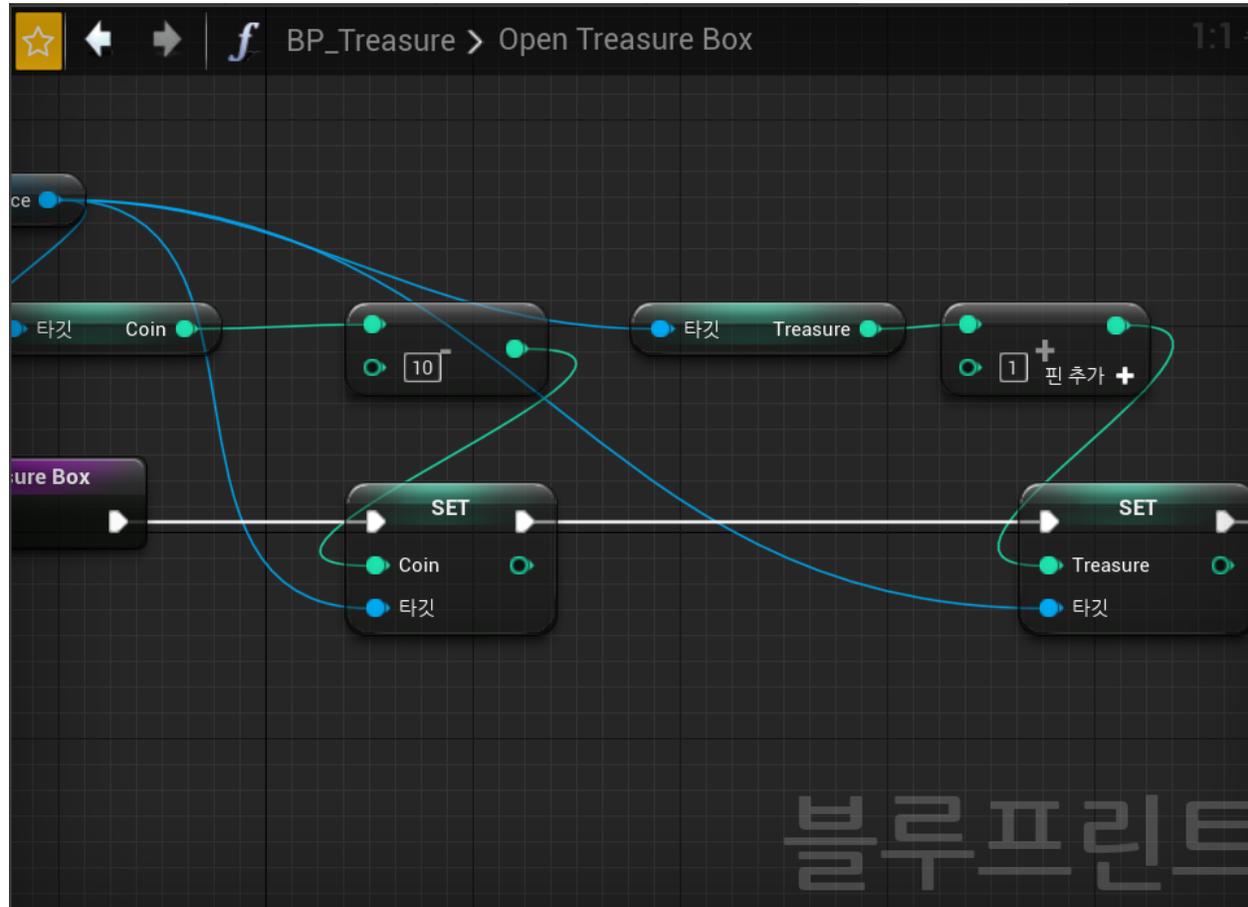
12.2.1 충돌이 발생하면 IsOpenTreasureBox를 호출한다.



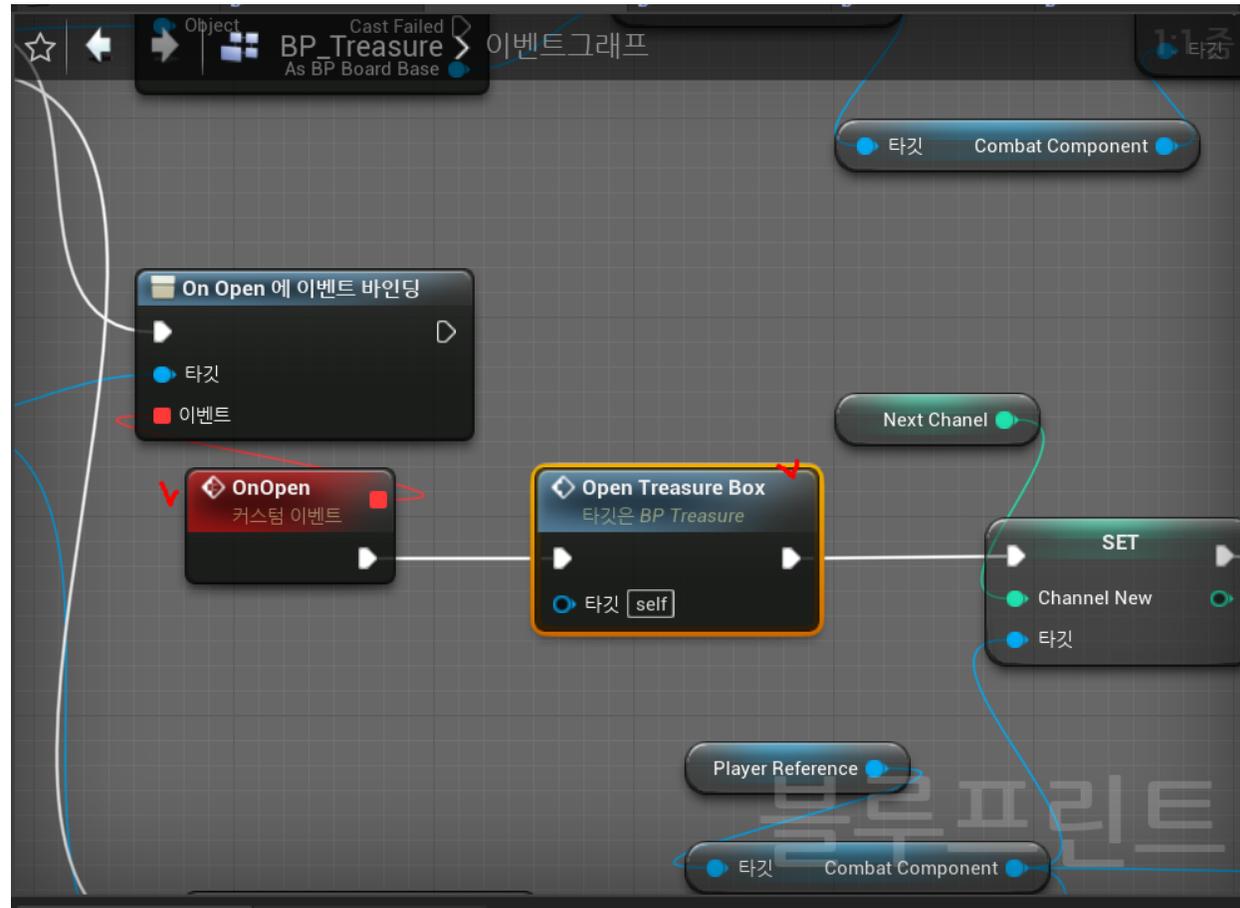
12.3 OpenTreasureBox 함수를 생성한다.



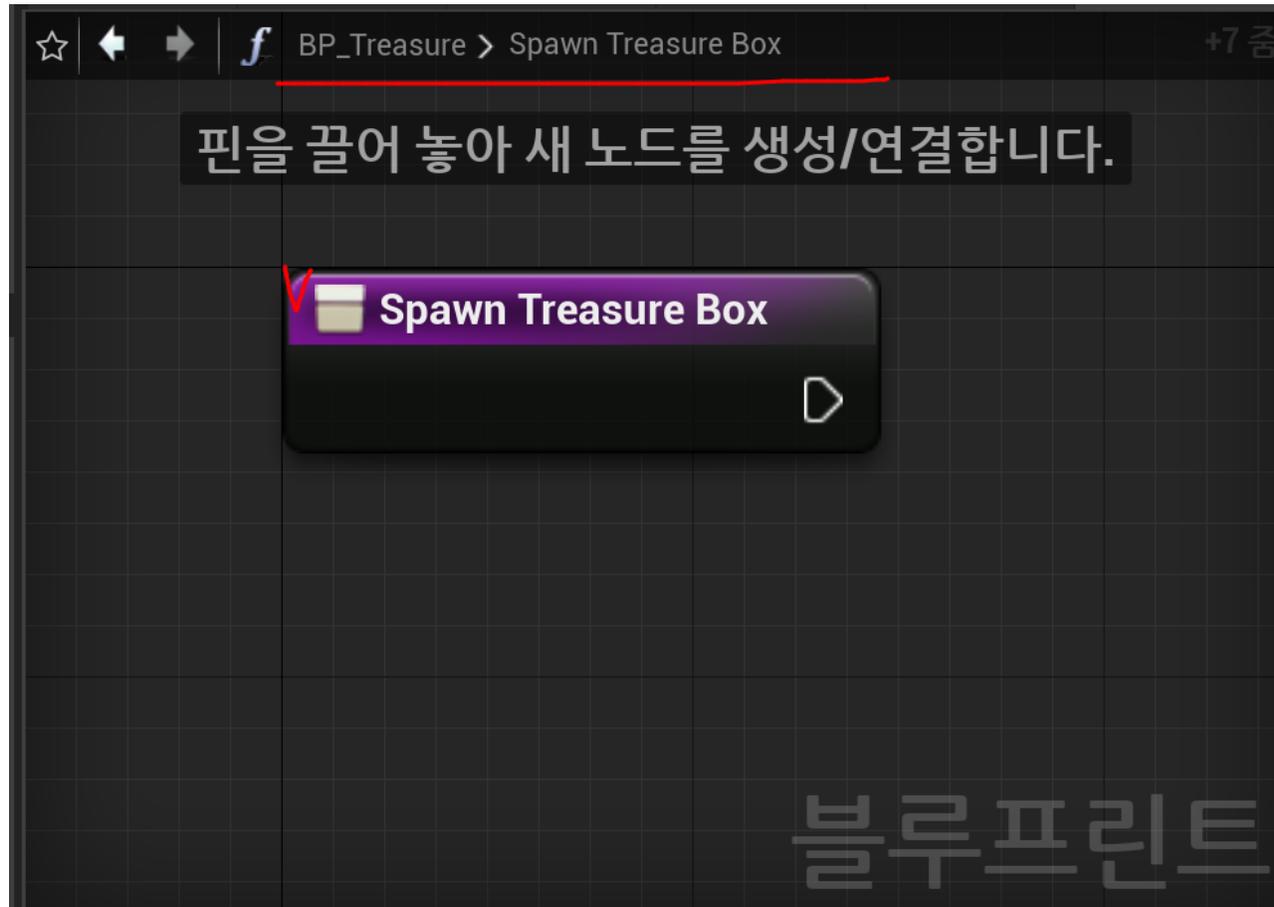
12.3.1 소유한 코인을 10 차감한다.



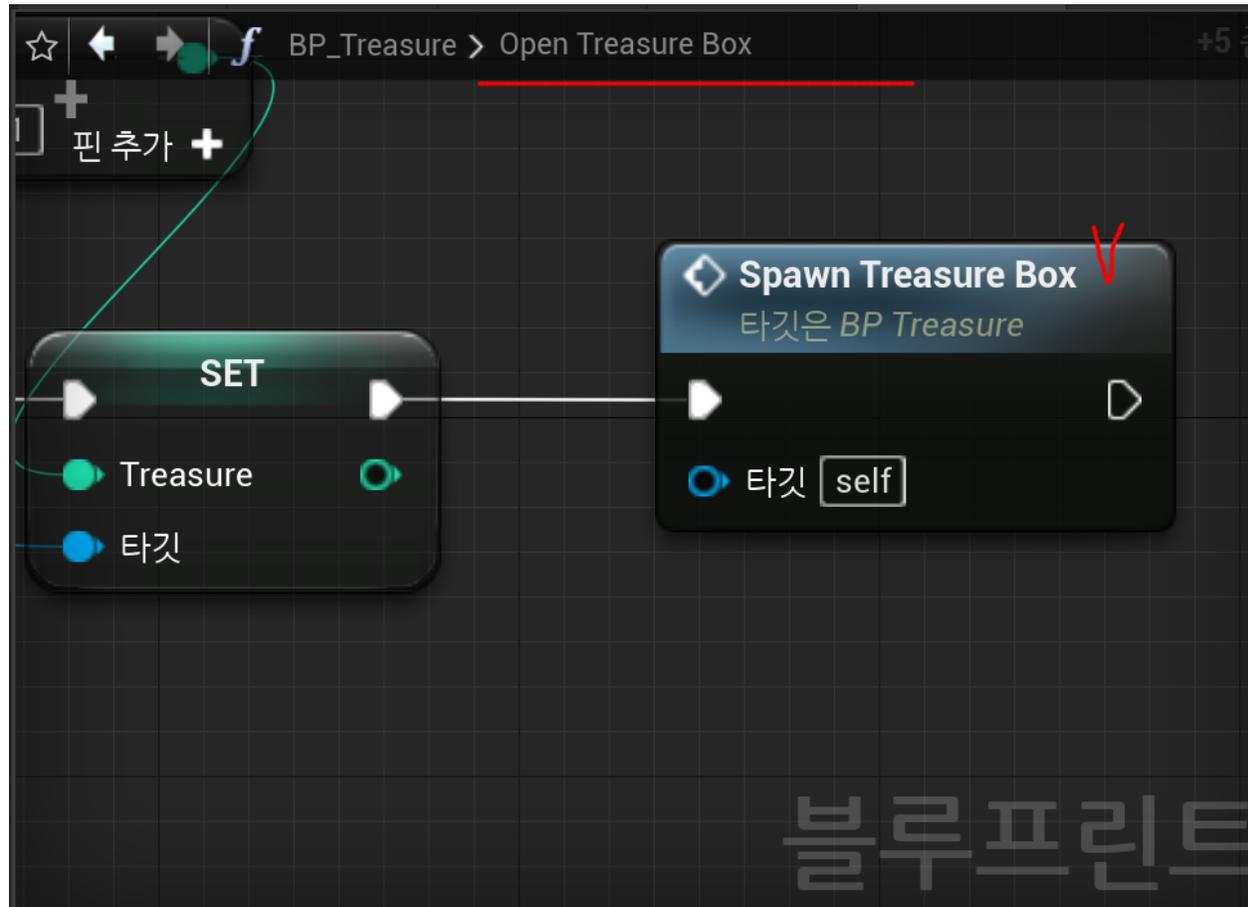
12.3.2 소유한 보물을 1 증가시킨다.



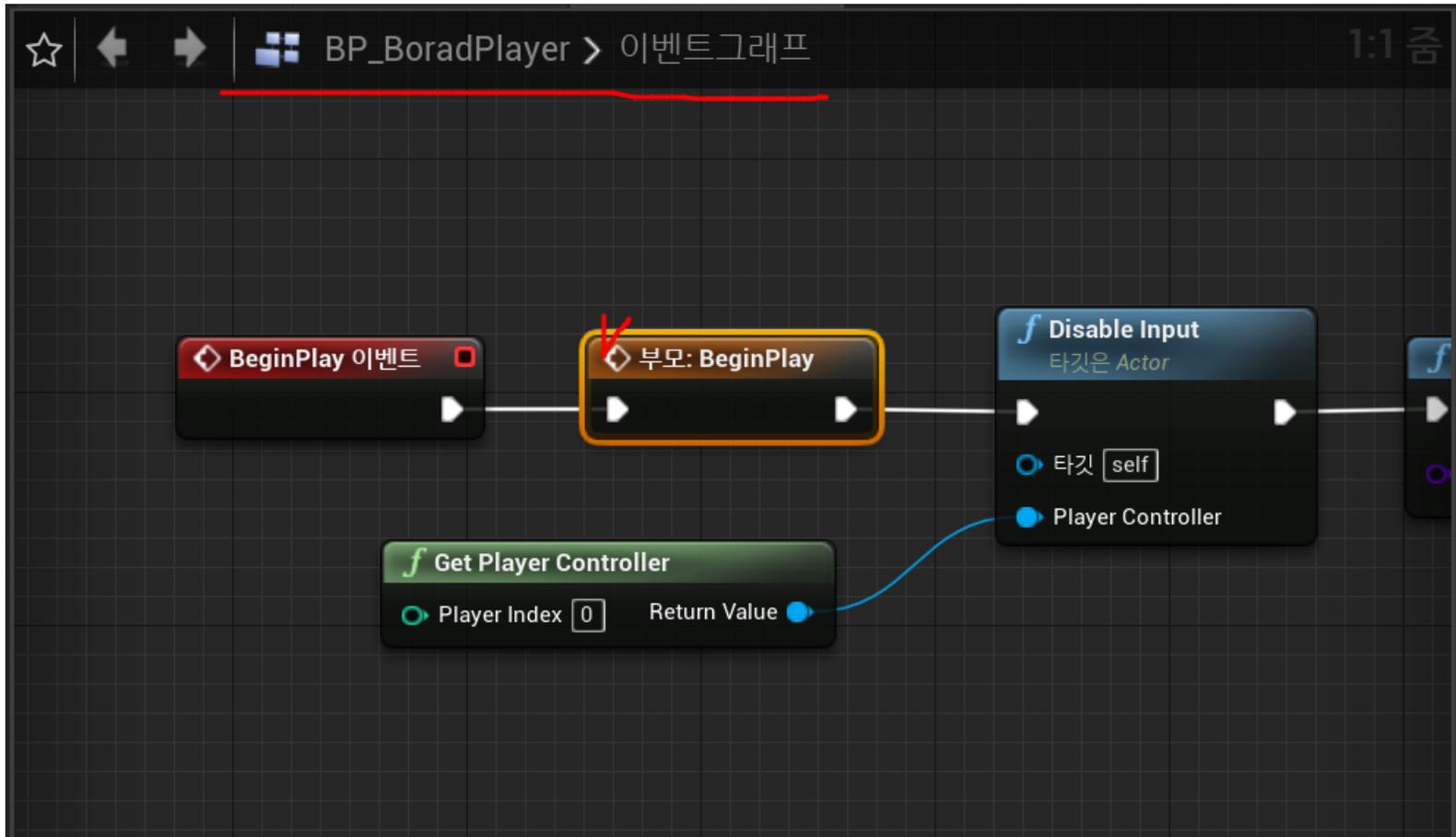
12.3.3 OnOpen 이벤트에 들어가서 OpenTreasureBox를 호출한다.



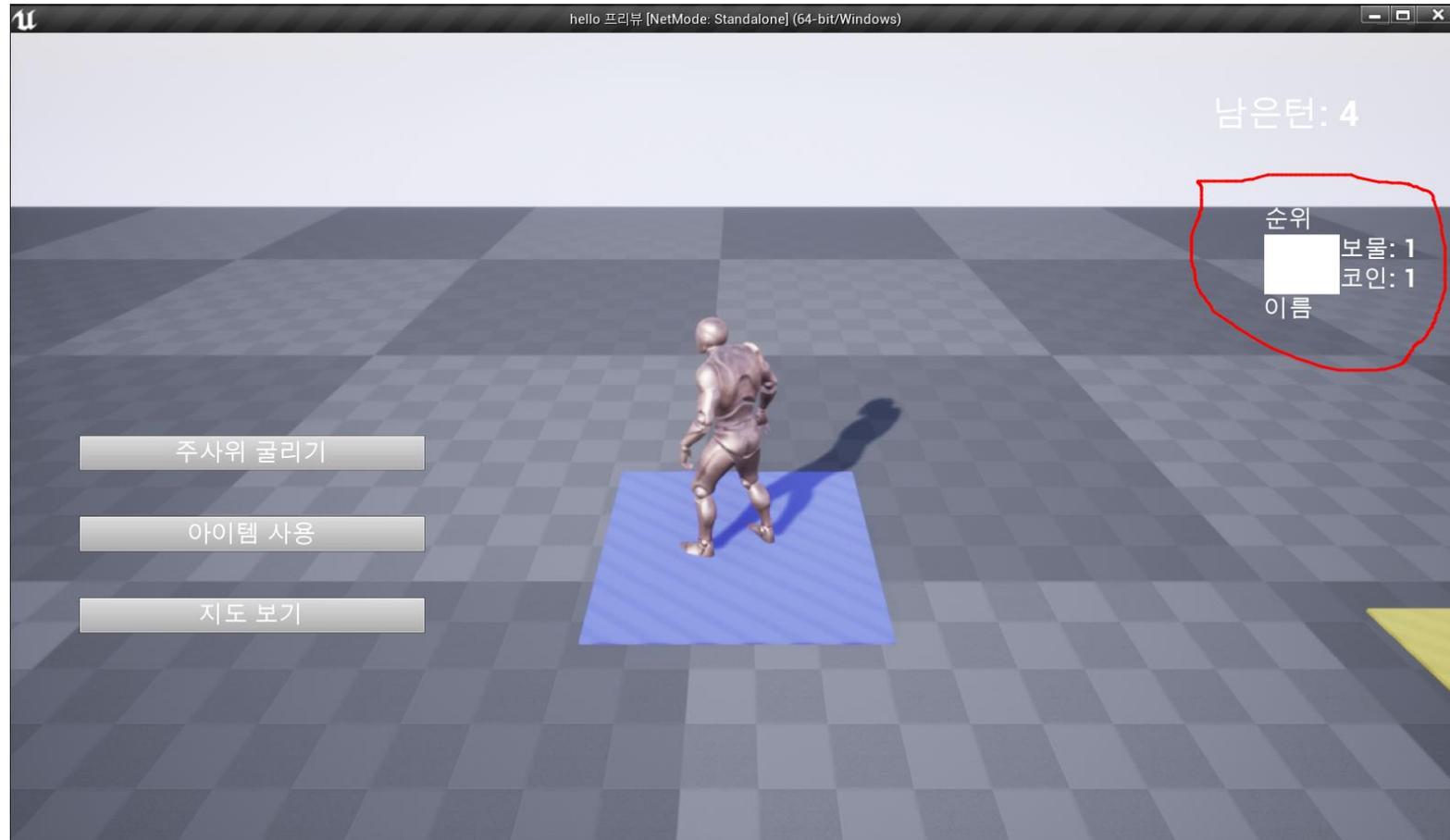
12.4 SpawnTreasureBox 함수를 생성한다.



12.4.1 OpenTreasureBox에 들어가서 SpawnTreasureBox를 호출한다.



13. BP_BoardPlayer의 BeginPlay 이벤트에 들어가서 부모 BeginPlay를 호출한다.



13.1 보물을 획득한 실행 결과