목차

1. **위젯 블루프린트 UI_HPBar 생성**

2. **GameInstance를 상속받은 ABGameInstance를 생성한다.**

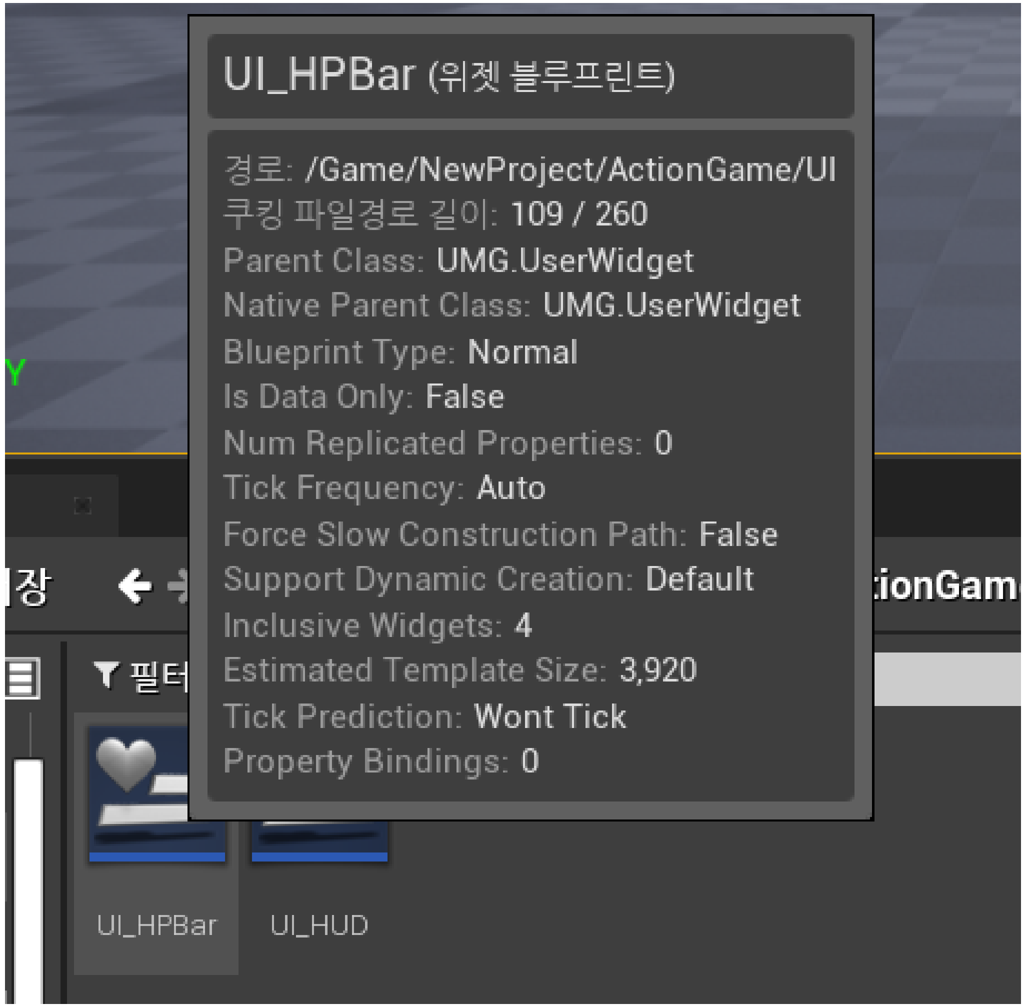
3. **ActorComponent를 상속받은 ABCharacterStatComponent를 생성한다.**

4. **PublicCharacter 처리**

5. **ABCharacterStatComponent 처리**

6. **UI_HPBar 처리**

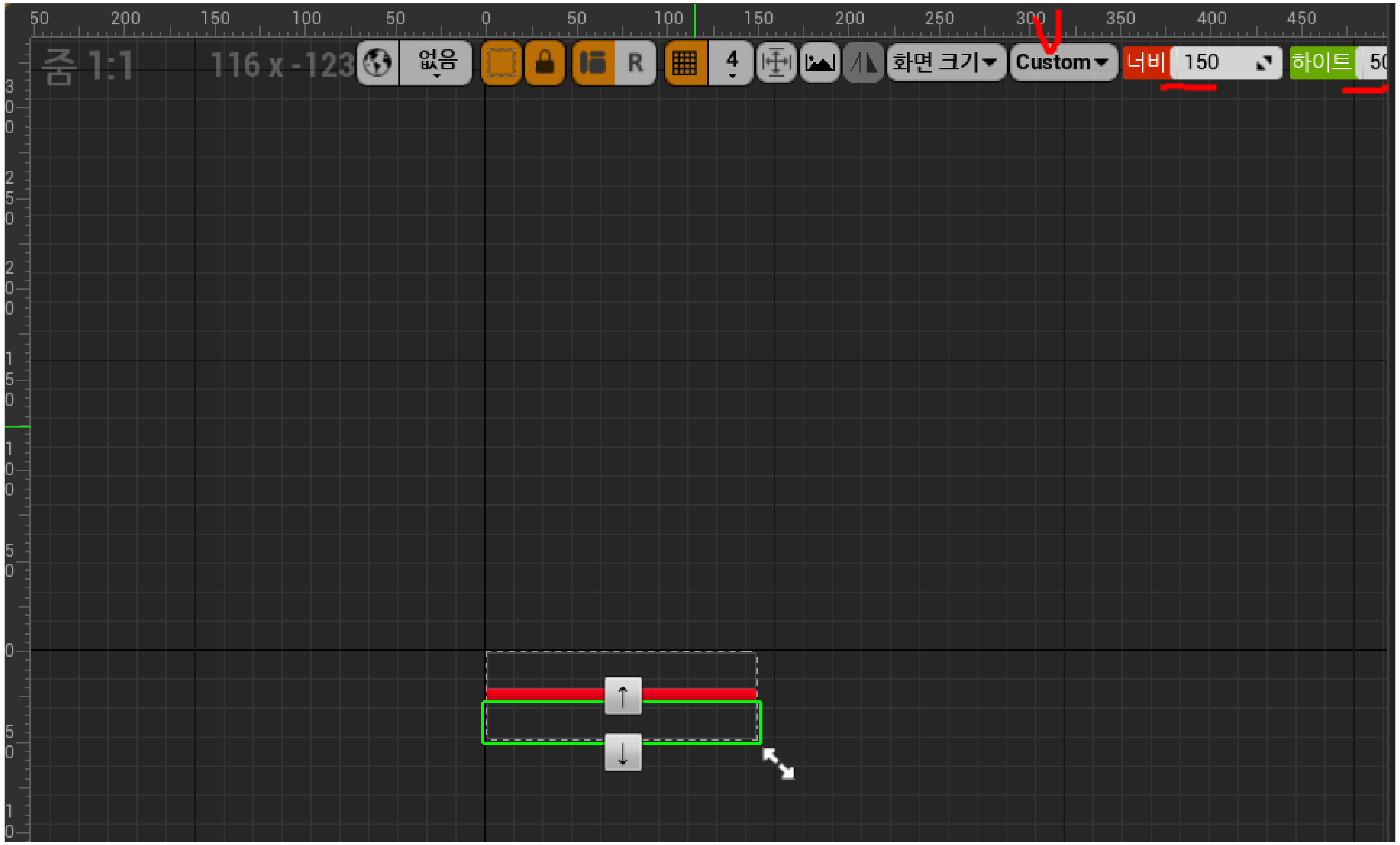
7. **EnemyCharacter 처리**



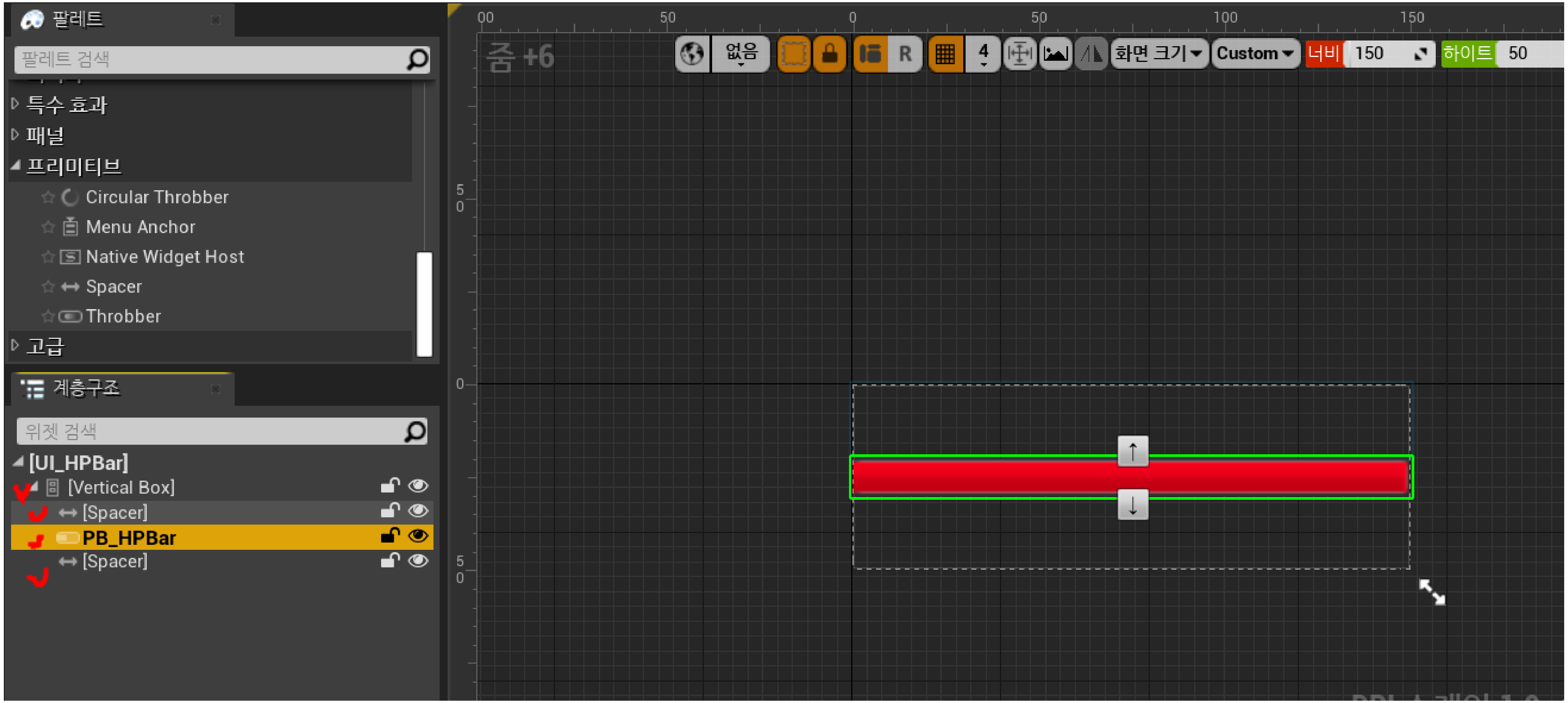
UI_HPBar (위젯 블루프린트)

경로: /Game/NewProject/ActionGame/UI
쿠킹 파일경로 길이: 109 / 260
Parent Class: UMG.UserWidget
Native Parent Class: UMG.UserWidget
Blueprint Type: Normal
Is Data Only: False
Num Replicated Properties: 0
Tick Frequency: Auto
Force Slow Construction Path: False
Support Dynamic Creation: Default
Inclusive Widgets: 4
Estimated Template Size: 3,920
Tick Prediction: Wont Tick
Property Bindings: 0

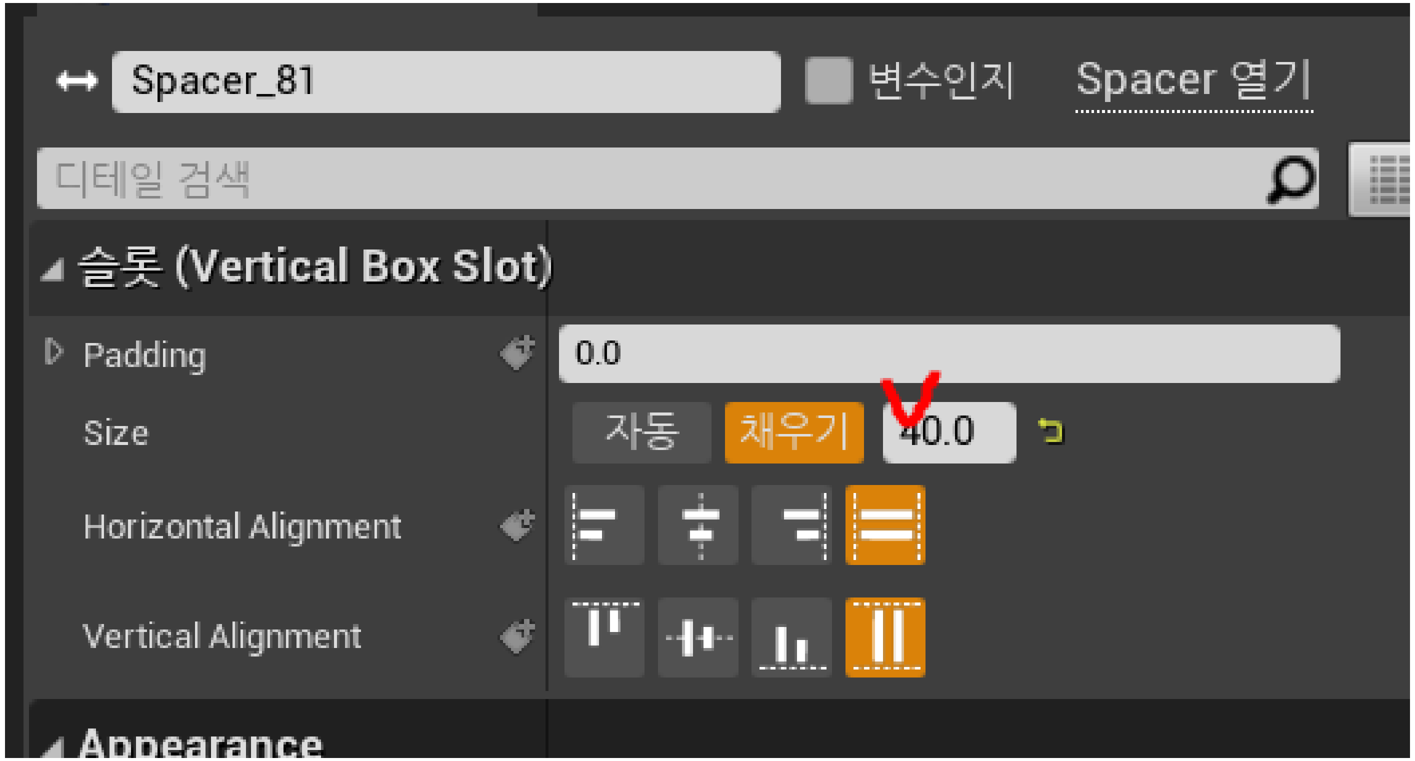
1. 위젯 블루프린트 UI_HPBar 생성



1.1 스크린의 크기를 Custom으로 바꾸고 150, 50으로 맞춘다.


팔레트
팔레트 검색
특수 효과
패널
프리미티브
Circular Throbber
Menu Anchor
Native Widget Host
Spacer
Throbber
고급
계층구조
위젯 검색
[UI_HPBar]
[Vertical Box]
[Spacer]
PB_HPBar
[Spacer]
줌 +6
없음
R
4
화면 크기
Custom
너비
150
하이트
50

1.2 위젯을 부착한다.



↔ Spacer_81 □ 변수인지

슬롯 (Vertical Box Slot)

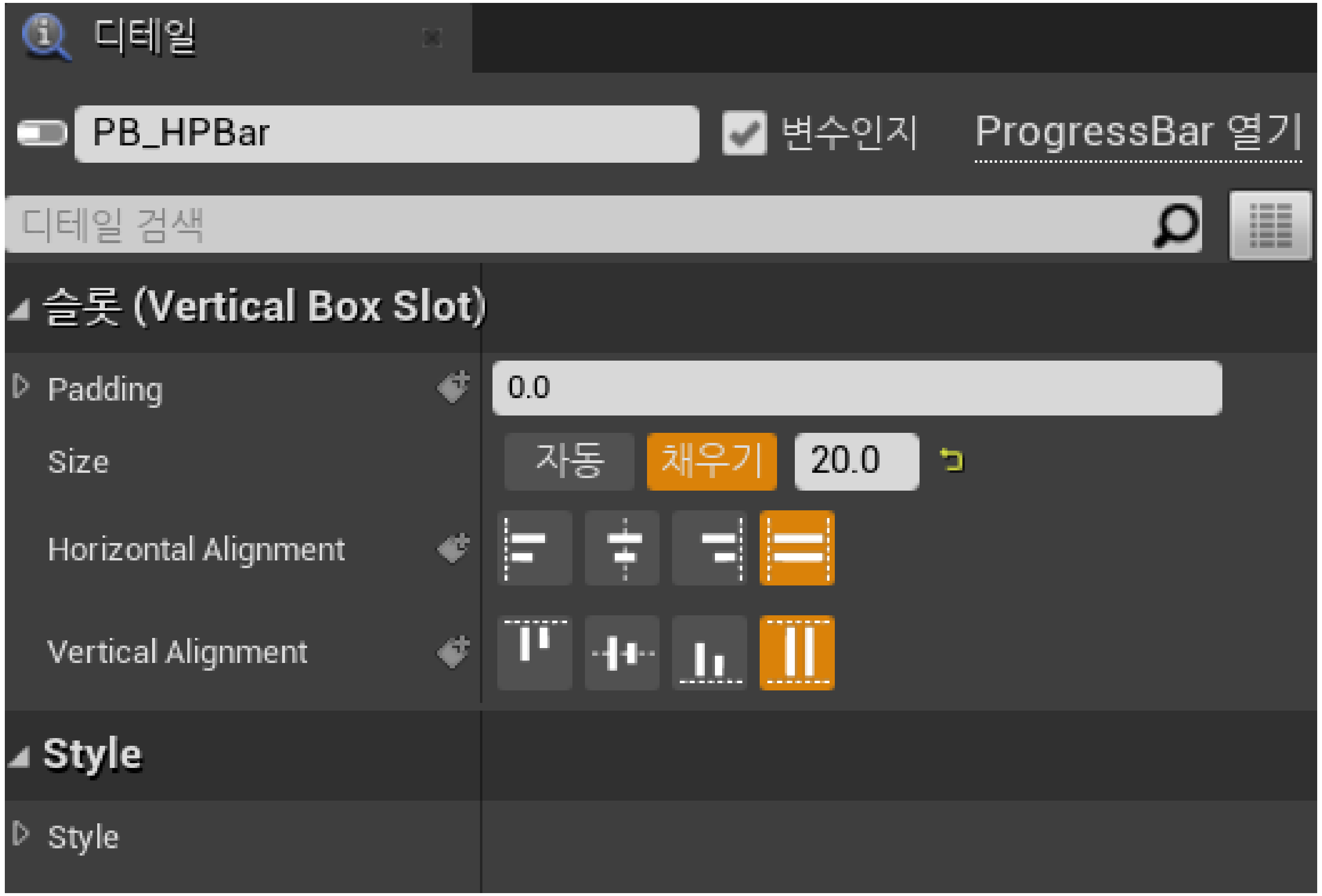
Padding 0.0

Size 자동 채우기 40.0

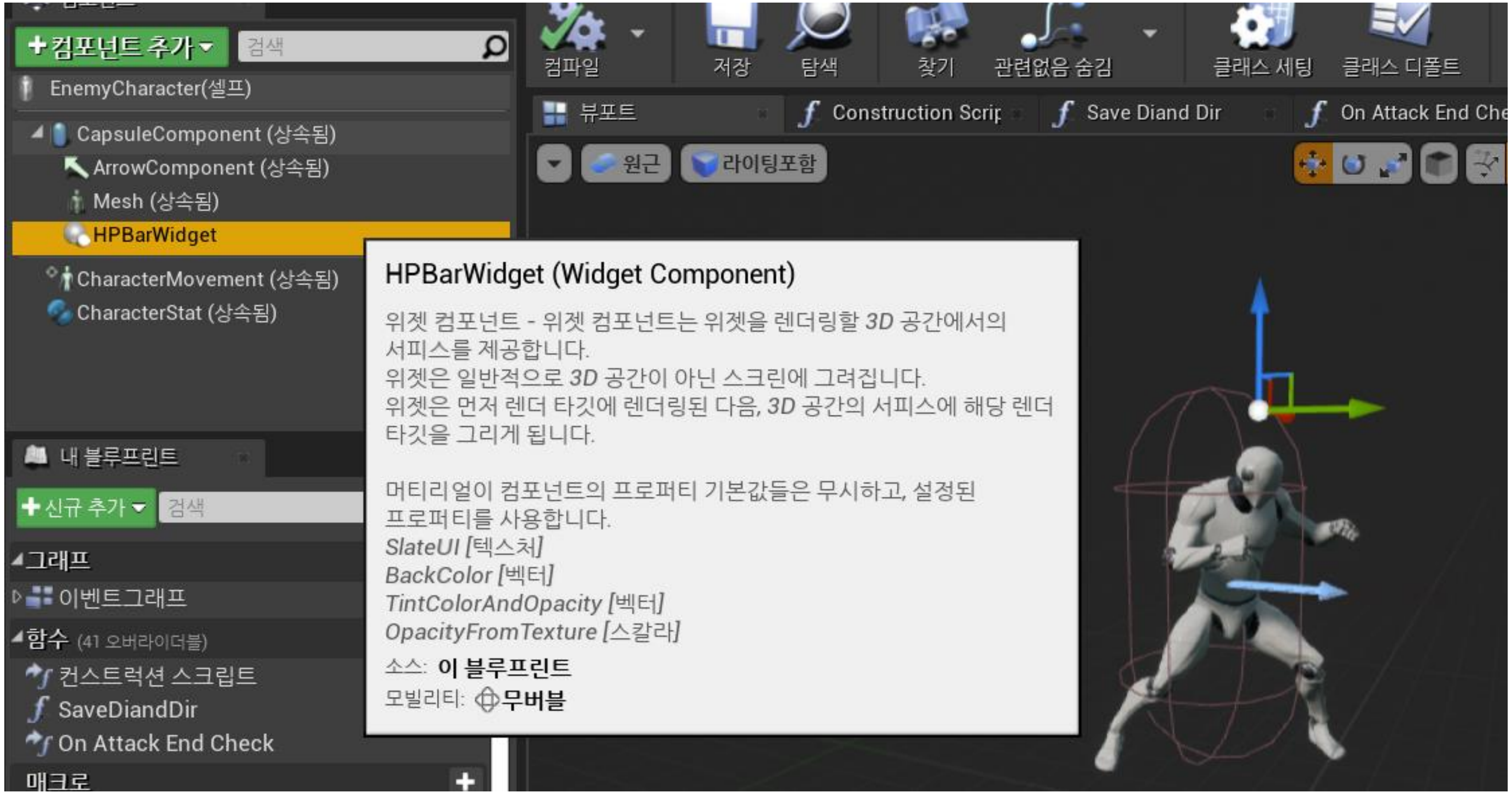
Horizontal Alignment

Vertical Alignment

1.3 Spacer의 size를 40으로 맞춘다.



1.4 PB_HPBar의 Size를 20으로 바꾼다.



HPBarWidget (Widget Component)

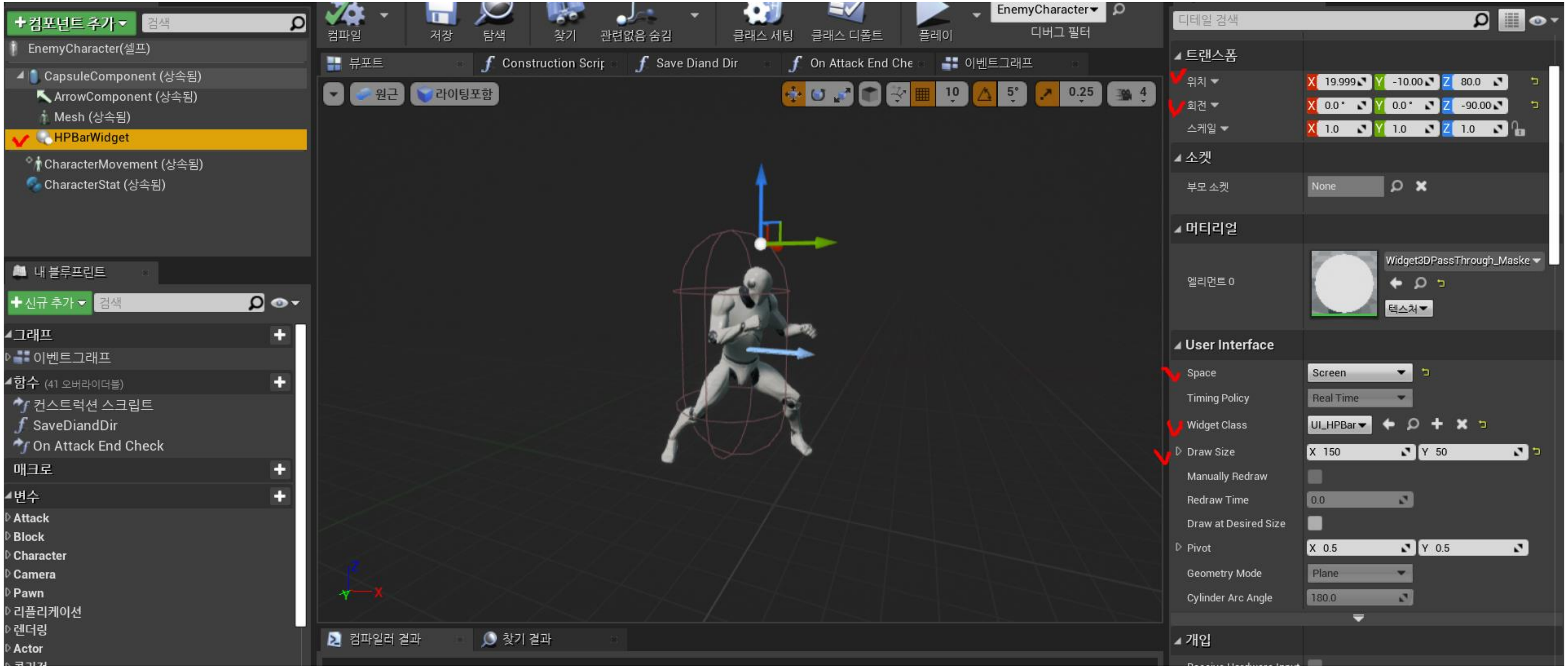
위젯 컴포넌트 - 위젯 컴포넌트는 위젯을 렌더링할 *3D* 공간에서의 서피스를 제공합니다.
위젯은 일반적으로 *3D* 공간이 아닌 스크린에 그려집니다.
위젯은 먼저 렌더 타깃에 렌더링된 다음, *3D* 공간의 서피스에 해당 렌더 타깃을 그리게 됩니다.

머티리얼이 컴포넌트의 프로퍼티 기본값들은 무시하고, 설정된 프로퍼티를 사용합니다.
SlateUI [텍스처]
BackColor [벡터]
TintColorAndOpacity [벡터]
OpacityFromTexture [스칼라]

소스: **이 블루프린트**

모빌리티: **무버블**

1.5 EnemyCharacter에 들어가서 WidgetComponent HPBarWidget을 추가한다



EnemyCharacter(셀프)
- CapsuleComponent (상속됨)
  - ArrowComponent (상속됨)
  - Mesh (상속됨)
  - ✓ HPBarWidget
- CharacterMovement (상속됨)
- CharacterStat (상속됨)

트랜스폼
- ✓ 위치: X 19.999 Y -10.00 Z 80.0
- ✓ 회전: X 0.0° Y 0.0° Z -90.00°
- 스케일: X 1.0 Y 1.0 Z 1.0

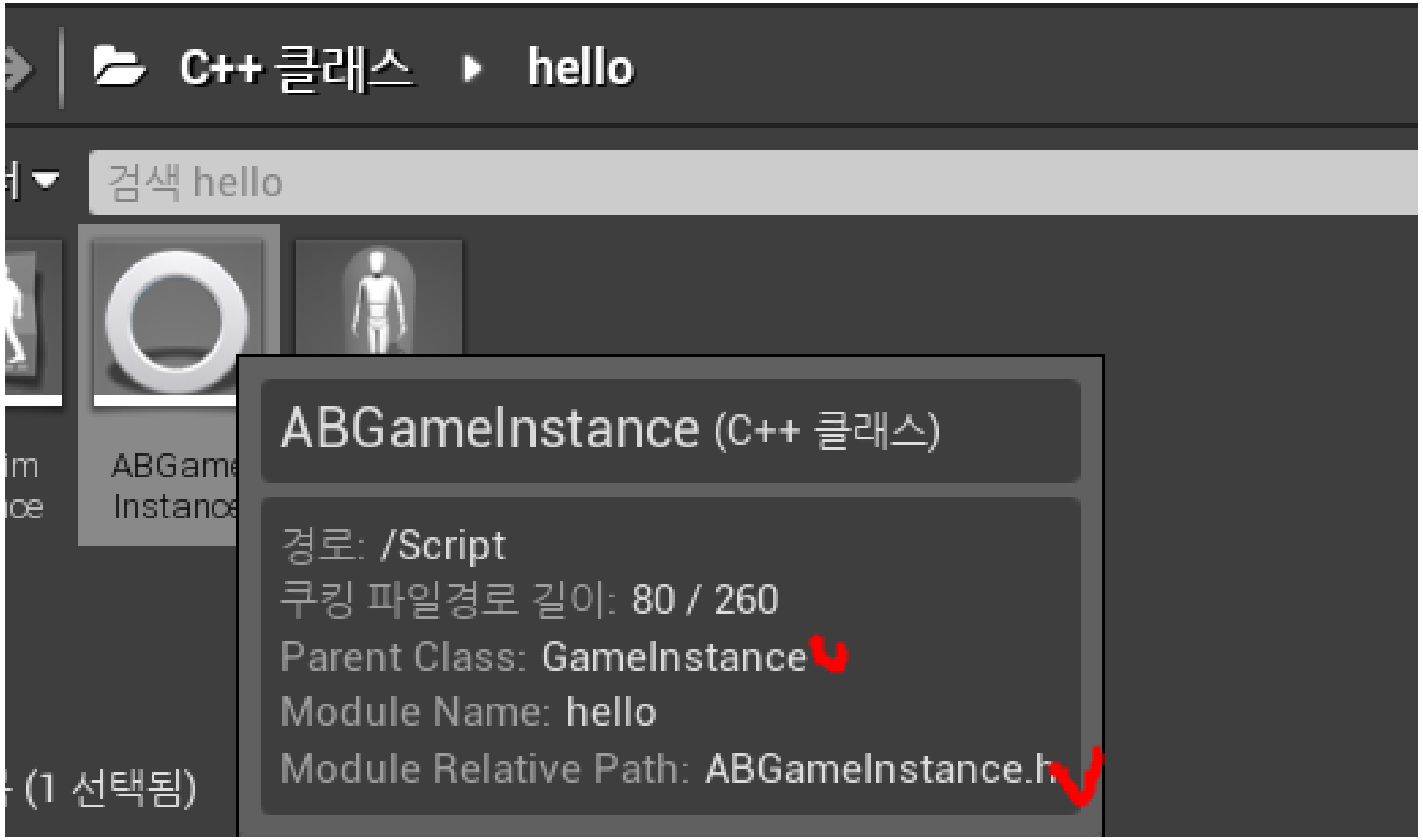
소켓
- 부모 소켓: None

머티리얼
- 엘리먼트 0: Widget3DPassThrough_Maske
- 텍스처

User Interface
- ✓ Space: Screen
- Timing Policy: Real Time
- ✓ Widget Class: UI_HPBar
- ✓ Draw Size: X 150 Y 50
- Manually Redraw: □
- Redraw Time: 0.0
- Draw at Desired Size: □
- Pivot: X 0.5 Y 0.5
- Geometry Mode: Plane
- Cylinder Arc Angle: 180.0

개입

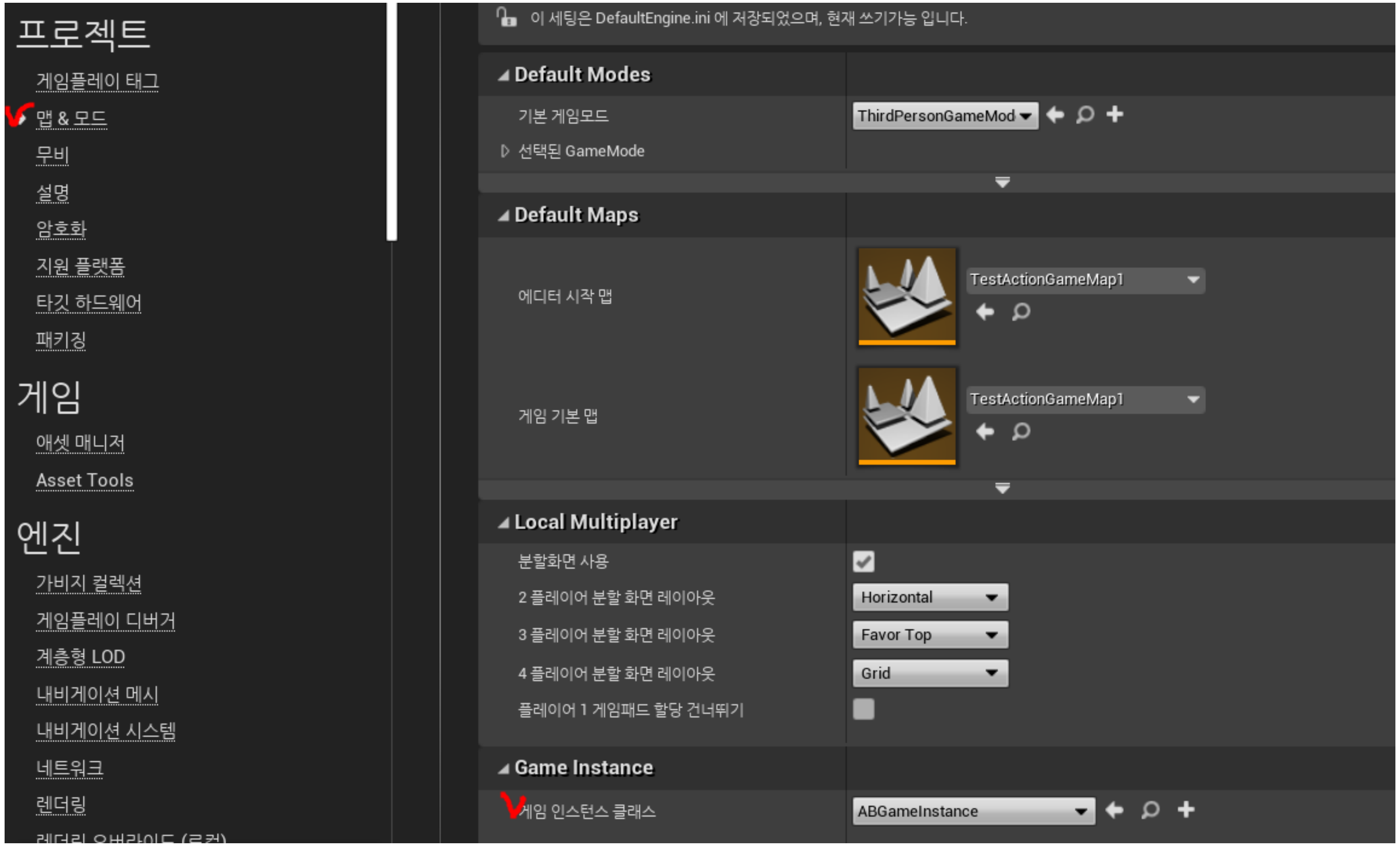
1.6 HPBarWidget 설정 변경



ABGameInstance (C++ 클래스)

경로: /Script
쿠킹 파일경로 길이: 80 / 260
Parent Class: GameInstance
Module Name: hello
Module Relative Path: ABGameInstance.h

2. GameInstance를 상속받은 ABGameInstance를 생성한다.



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

Default Modes

기본 게임모드: ThirdPersonGameMod

선택된 GameMode

Default Maps

에디터 시작 맵: TestActionGameMap1

게임 기본 맵: TestActionGameMap1

Local Multiplayer

분할화면 사용: ☑

2 플레이어 분할 화면 레이아웃: Horizontal

3 플레이어 분할 화면 레이아웃: Favor Top

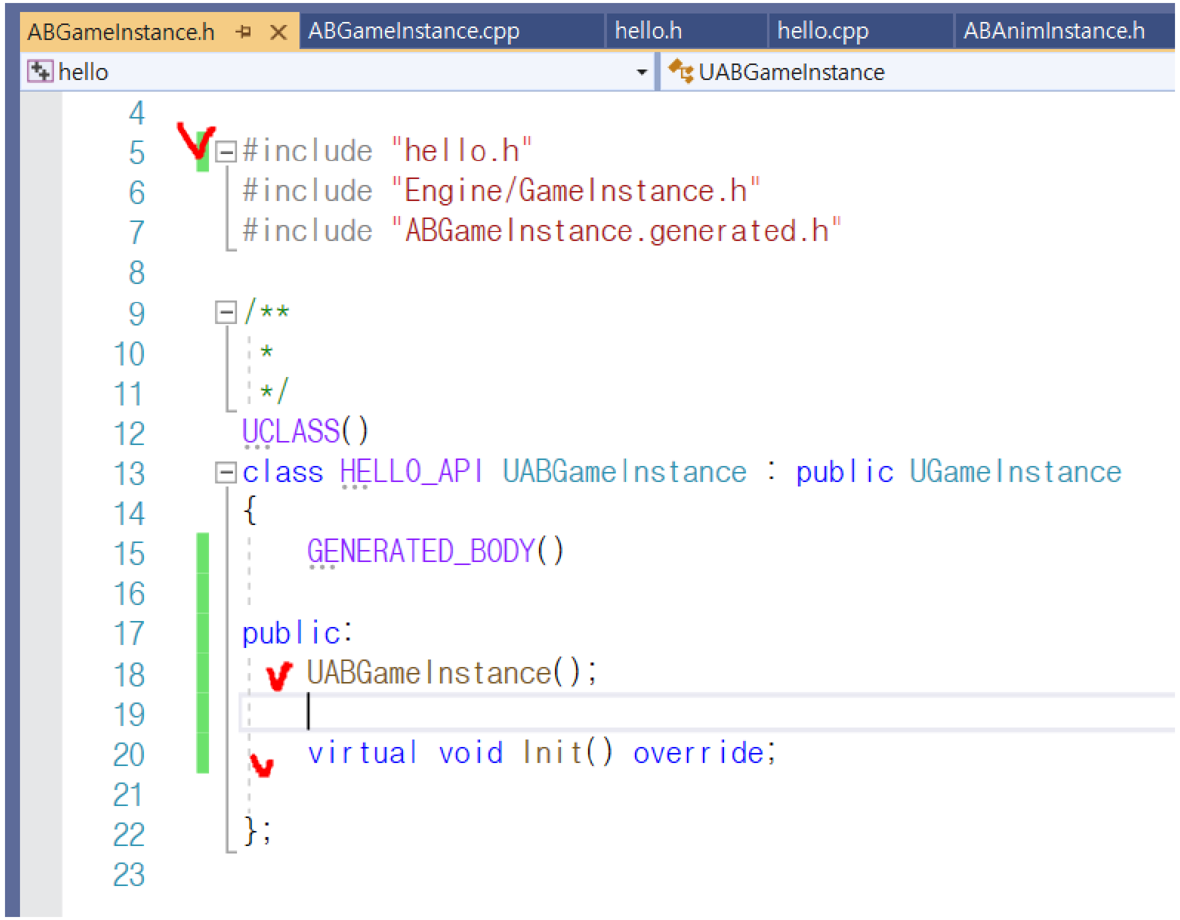
4 플레이어 분할 화면 레이아웃: Grid

플레이어 1 게임패드 할당 건너뛰기: □

Game Instance

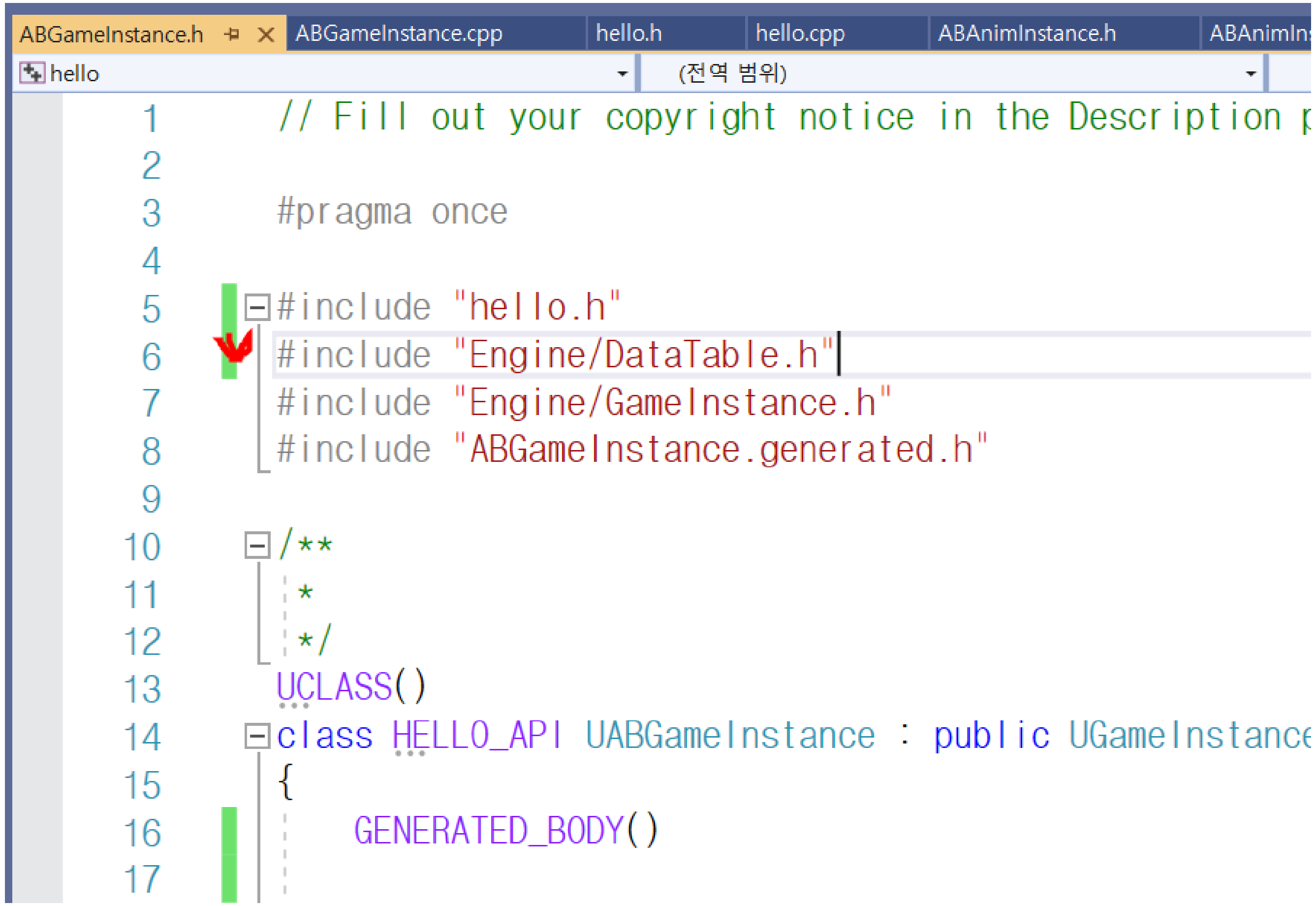
게임 인스턴스 클래스: ABGameInstance

2.1 게임 인스턴스 클래스에 ABGameInstance를 넣는다.



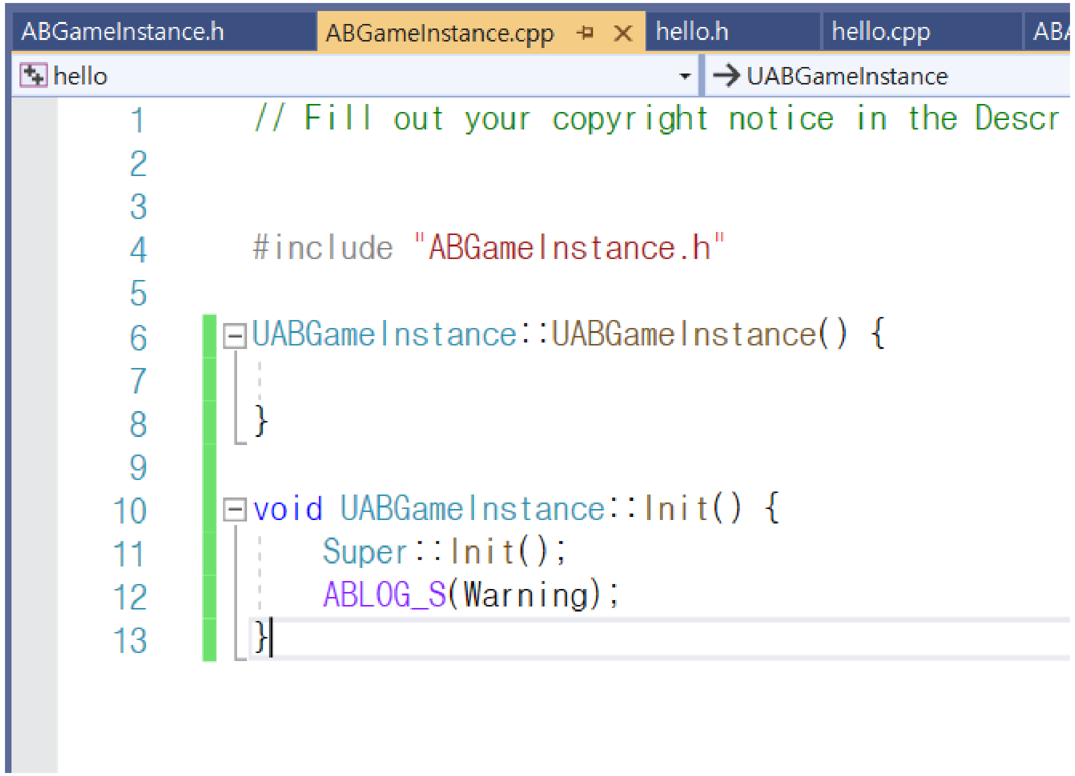
```cpp

#include "hello.h"
#include "Engine/GameInstance.h"
#include "ABGameInstance.generated.h"

/**
 *
 */
UCLASS()
class HELLO_API UABGameInstance : public UGameInstance
{
    GENERATED_BODY()

public:
    UABGameInstance();

    virtual void Init() override;

};

```

2.2 ABGameInstance에 들어가서 생성자와 Init()를 추가한다



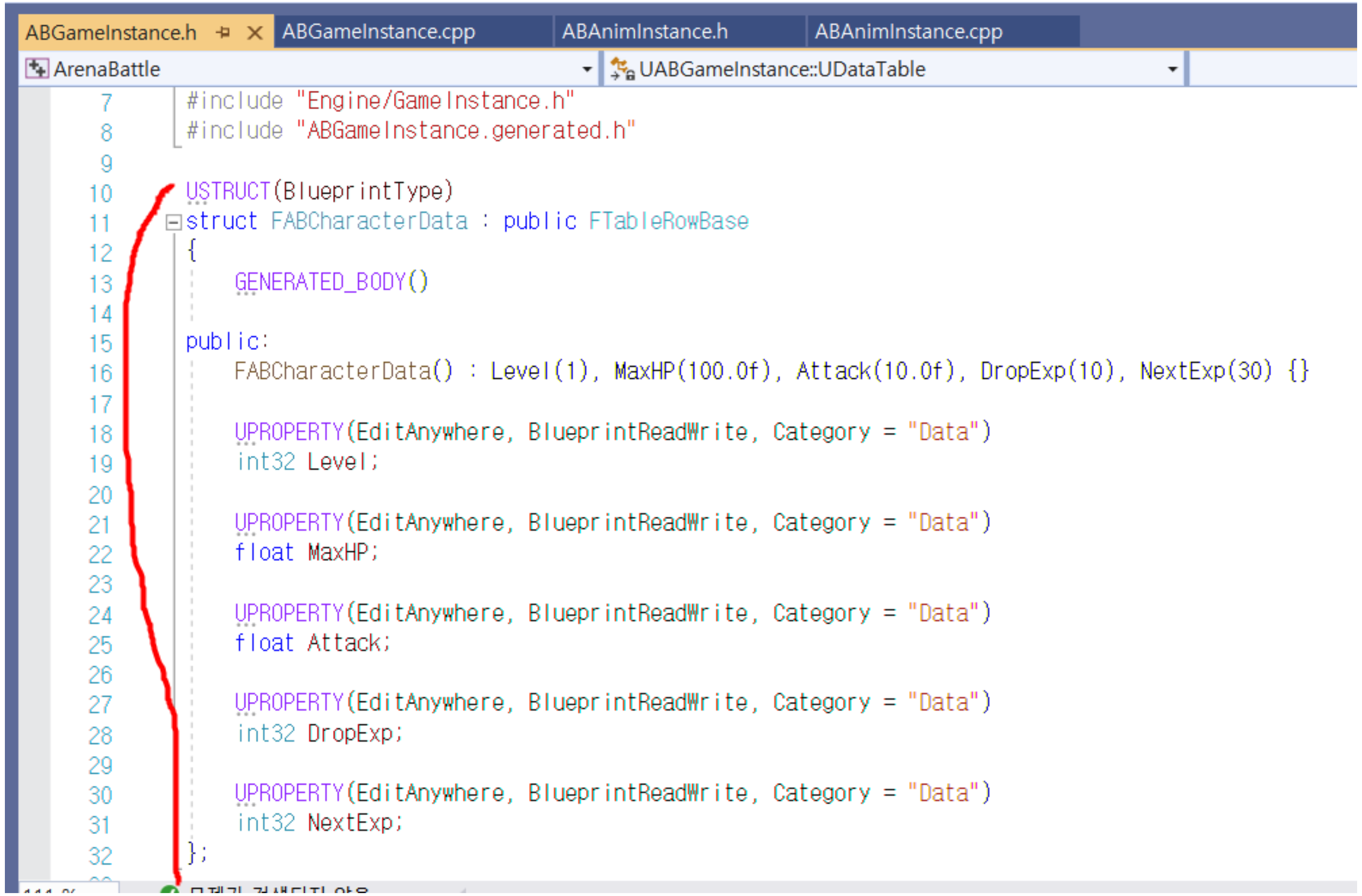
```
// Fill out your copyright notice in the Description p

#pragma once

#include "hello.h"
#include "Engine/DataTable.h"
#include "Engine/GameInstance.h"
#include "ABGameInstance.generated.h"

/**
 *
 */
UCLASS()
class HELLO_API UABGameInstance : public UGameInstance
{
	GENERATED_BODY()

```

2.3 DataTable을 인크루드한다.



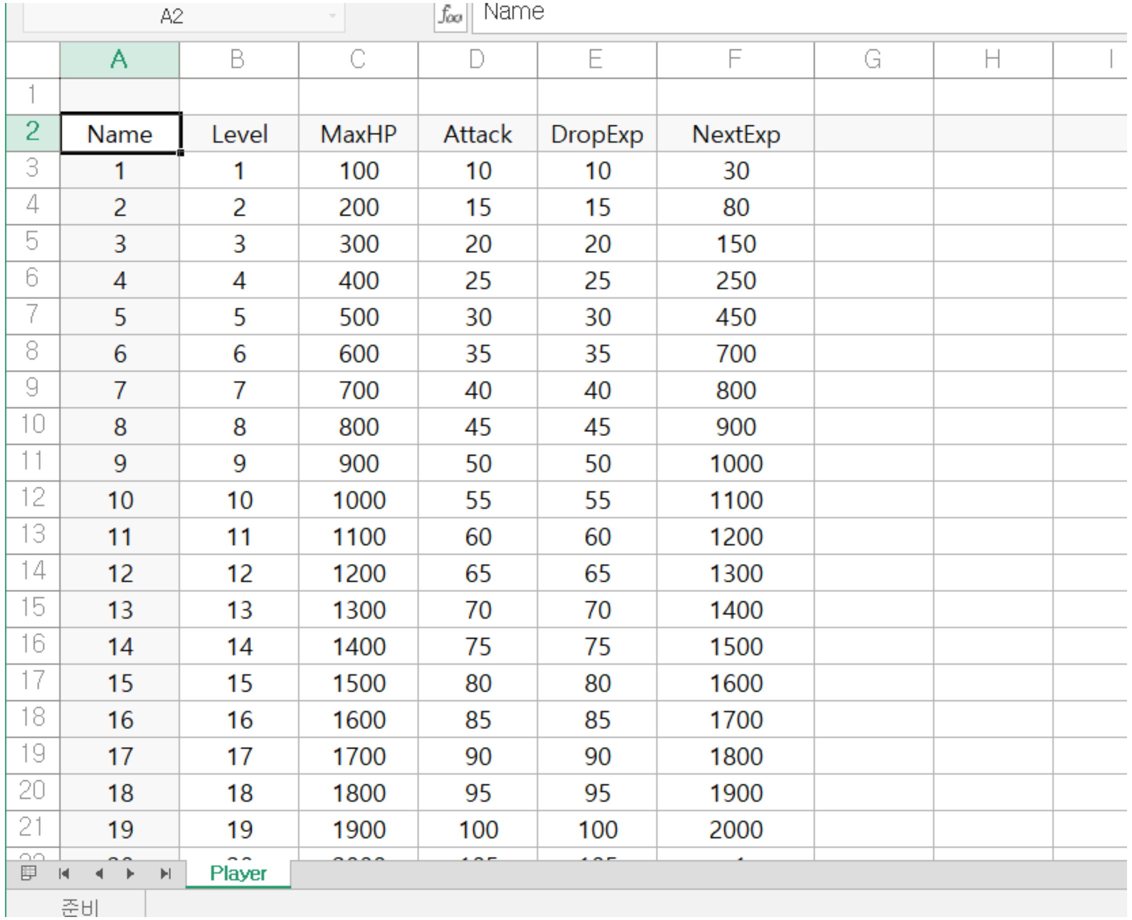
```
// Fill out your copyright notice in the Descr


#include "ABGameInstance.h"

UABGameInstance::UABGameInstance() {

}

void UABGameInstance::Init() {
	Super::Init();
	ABLOG_S(Warning);
}
```

2.3 구현에 들어가서 생성자와 함수를 구현한다.



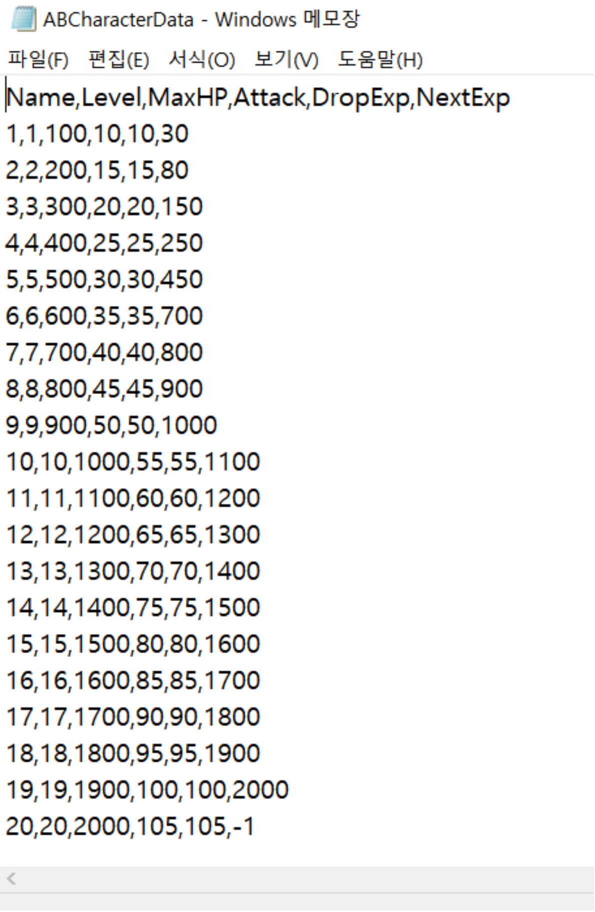
```cpp
#include "Engine/GameInstance.h"
#include "ABGameInstance.generated.h"

USTRUCT(BlueprintType)
struct FABCharacterData : public FTableRowBase
{
	GENERATED_BODY()

public:
	FABCharacterData() : Level(1), MaxHP(100.0f), Attack(10.0f), DropExp(10), NextExp(30) {}

	UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "Data")
	int32 Level;

	UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "Data")
	float MaxHP;

	UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "Data")
	float Attack;

	UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "Data")
	int32 DropExp;

	UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "Data")
	int32 NextExp;
};
```

2.4 struct FABCharacterData를 추가한다



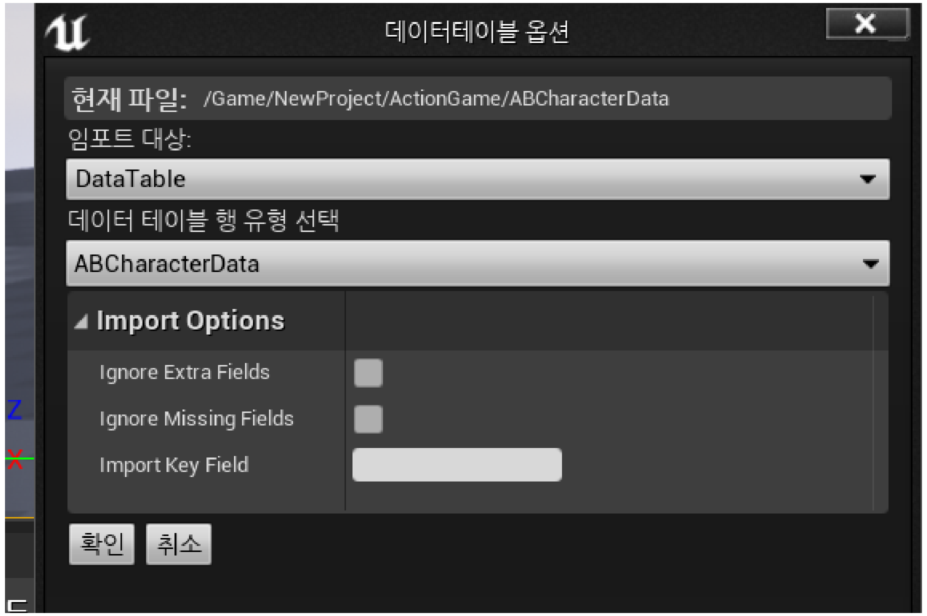
	A	B	C	D	E	F	G	H	I
1									
2	Name	Level	MaxHP	Attack	DropExp	NextExp			
3	1	1	100	10	10	30			
4	2	2	200	15	15	80			
5	3	3	300	20	20	150			
6	4	4	400	25	25	250			
7	5	5	500	30	30	450			
8	6	6	600	35	35	700			
9	7	7	700	40	40	800			
10	8	8	800	45	45	900			
11	9	9	900	50	50	1000			
12	10	10	1000	55	55	1100			
13	11	11	1100	60	60	1200			
14	12	12	1200	65	65	1300			
15	13	13	1300	70	70	1400			
16	14	14	1400	75	75	1500			
17	15	15	1500	80	80	1600			
18	16	16	1600	85	85	1700			
19	17	17	1700	90	90	1800			
20	18	18	1800	95	95	1900			
21	19	19	1900	100	100	2000			

2.5 미리 만들어둔 데이터테이블을 CSV 파일로 저장한다.



```
Name,Level,MaxHP,Attack,DropExp,NextExp
1,1,100,10,10,30
2,2,200,15,15,80
3,3,300,20,20,150
4,4,400,25,25,250
5,5,500,30,30,450
6,6,600,35,35,700
7,7,700,40,40,800
8,8,800,45,45,900
9,9,900,50,50,1000
10,10,1000,55,55,1100
11,11,1100,60,60,1200
12,12,1200,65,65,1300
13,13,1300,70,70,1400
14,14,1400,75,75,1500
15,15,1500,80,80,1600
16,16,1600,85,85,1700
17,17,1700,90,90,1800
18,18,1800,95,95,1900
19,19,1900,100,100,2000
20,20,2000,105,105,-1
```

2.6 CSV로 저장된 결과



데이터테이블 옵션

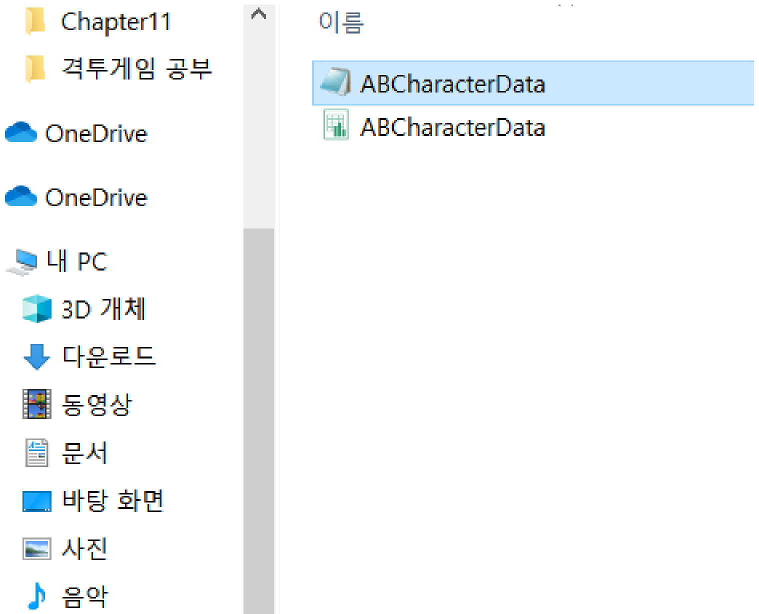
현재 파일: /Game/NewProject/ActionGame/ABCharacterData

임포트 대상: DataTable

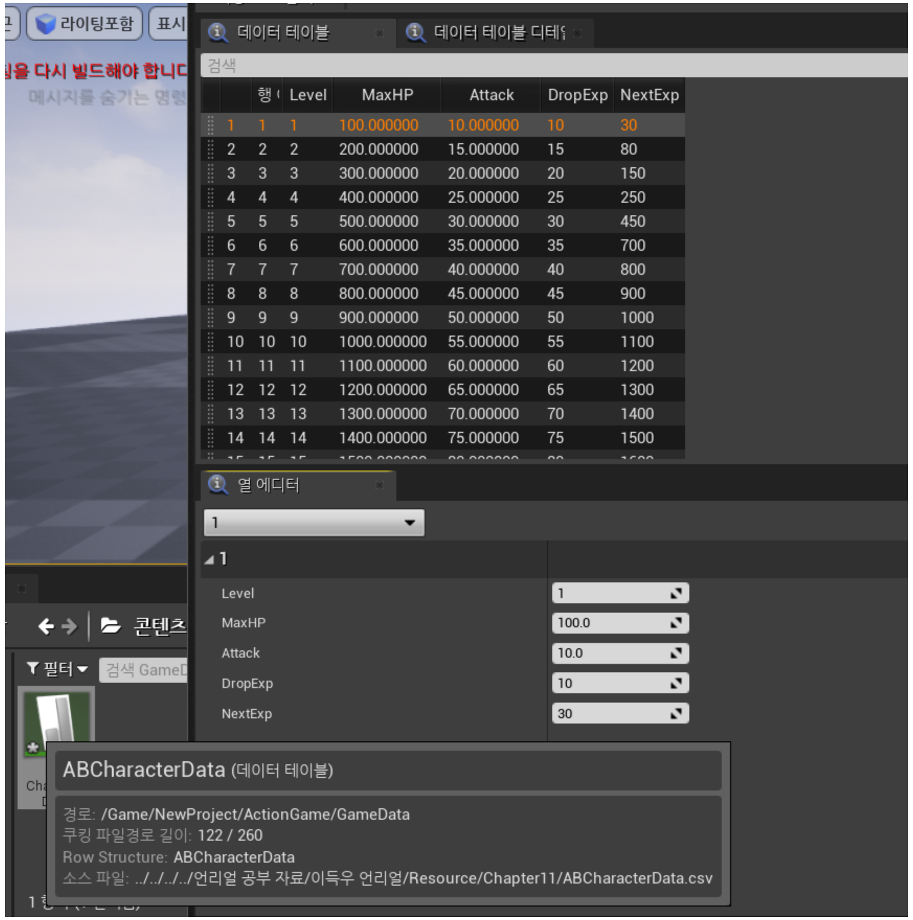
데이터 테이블 행 유형 선택: ABCharacterData

Import Options

- Ignore Extra Fields □
- Ignore Missing Fields □
- Import Key Field



2.7 ABCharacterData를 에디터에 임포트하고 구조체 ABCharcterData를 데이터테이블 유형으로 한다.



	행	Level	MaxHP	Attack	DropExp	NextExp
1	1	1	100.000000	10.000000	10	30
2	2	2	200.000000	15.000000	15	80
3	3	3	300.000000	20.000000	20	150
4	4	4	400.000000	25.000000	25	250
5	5	5	500.000000	30.000000	30	450
6	6	6	600.000000	35.000000	35	700
7	7	7	700.000000	40.000000	40	800
8	8	8	800.000000	45.000000	45	900
9	9	9	900.000000	50.000000	50	1000
10	10	10	1000.000000	55.000000	55	1100
11	11	11	1100.000000	60.000000	60	1200
12	12	12	1200.000000	65.000000	65	1300
13	13	13	1300.000000	70.000000	70	1400
14	14	14	1400.000000	75.000000	75	1500

1

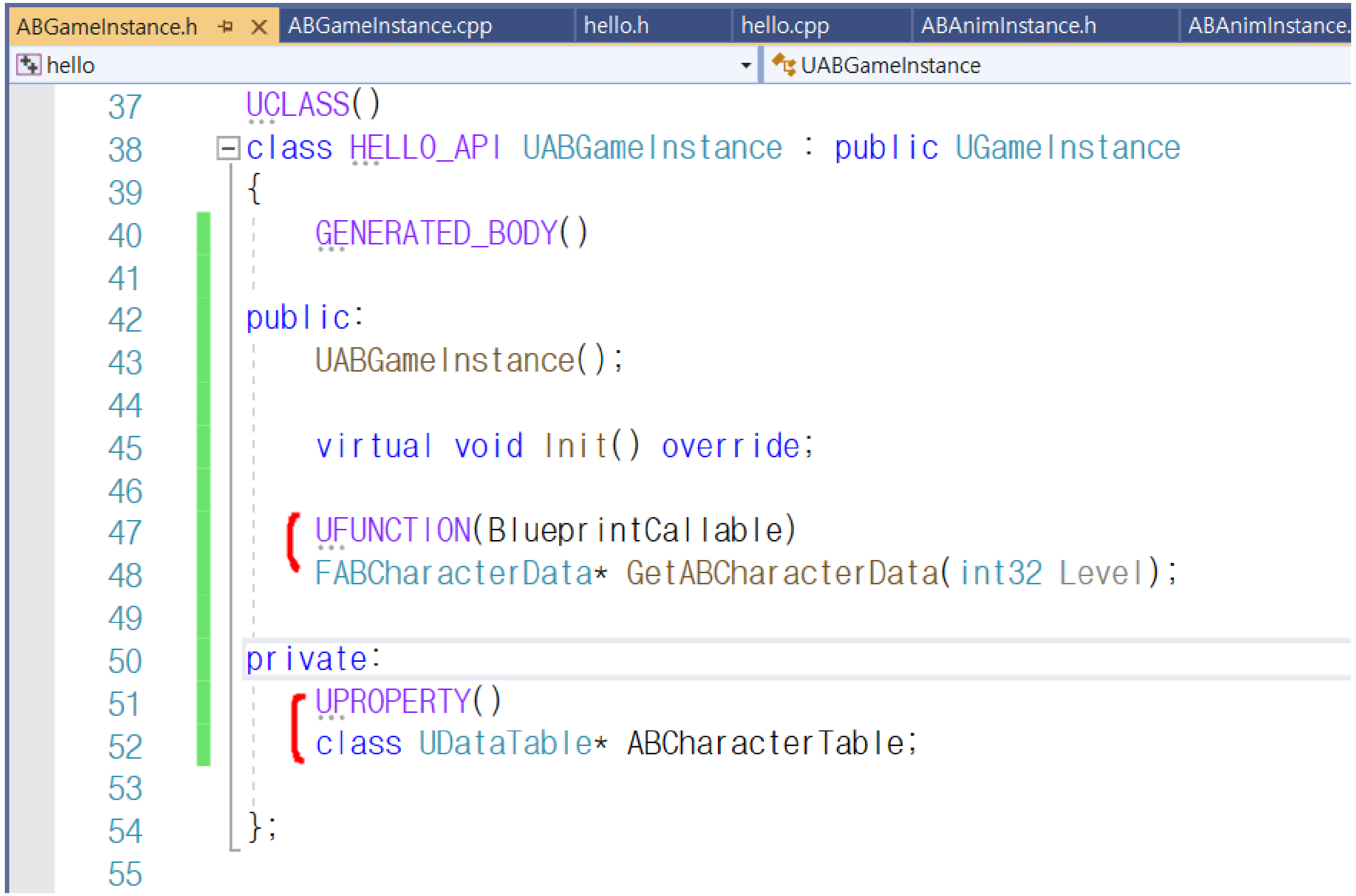
1

Level	1
MaxHP	100.0
Attack	10.0
DropExp	10
NextExp	30

ABCharacterData (데이터 테이블)

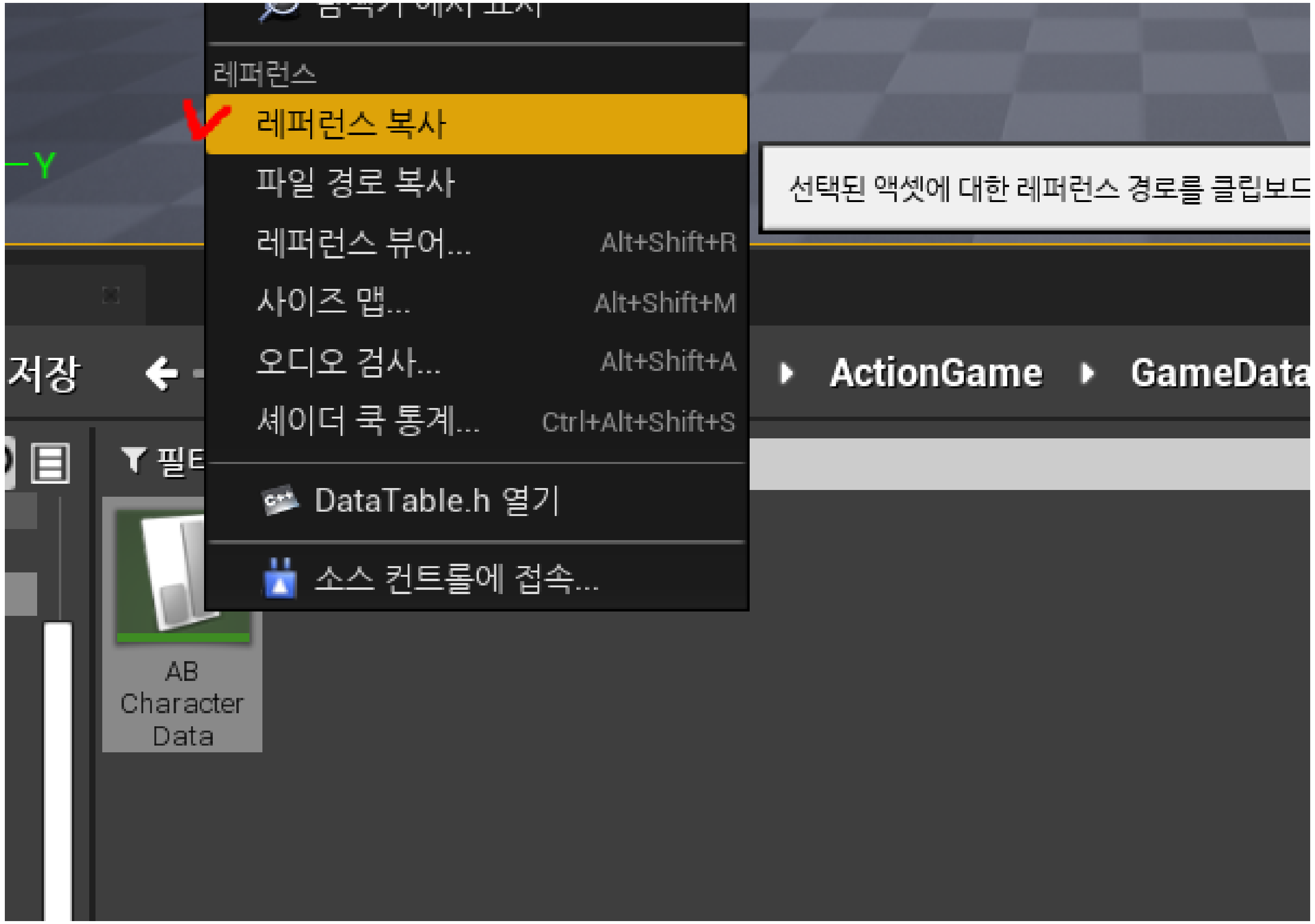
경로: /Game/NewProject/ActionGame/GameData
쿠킹 파일경로 길이: 122 / 260
Row Structure: ABCharacterData
소스 파일: ../../../../언리얼 공부 자료/이득우 언리얼/Resource/Chapter11/ABCharacterData.csv

2.8 데이터테이블이 생성된 결과

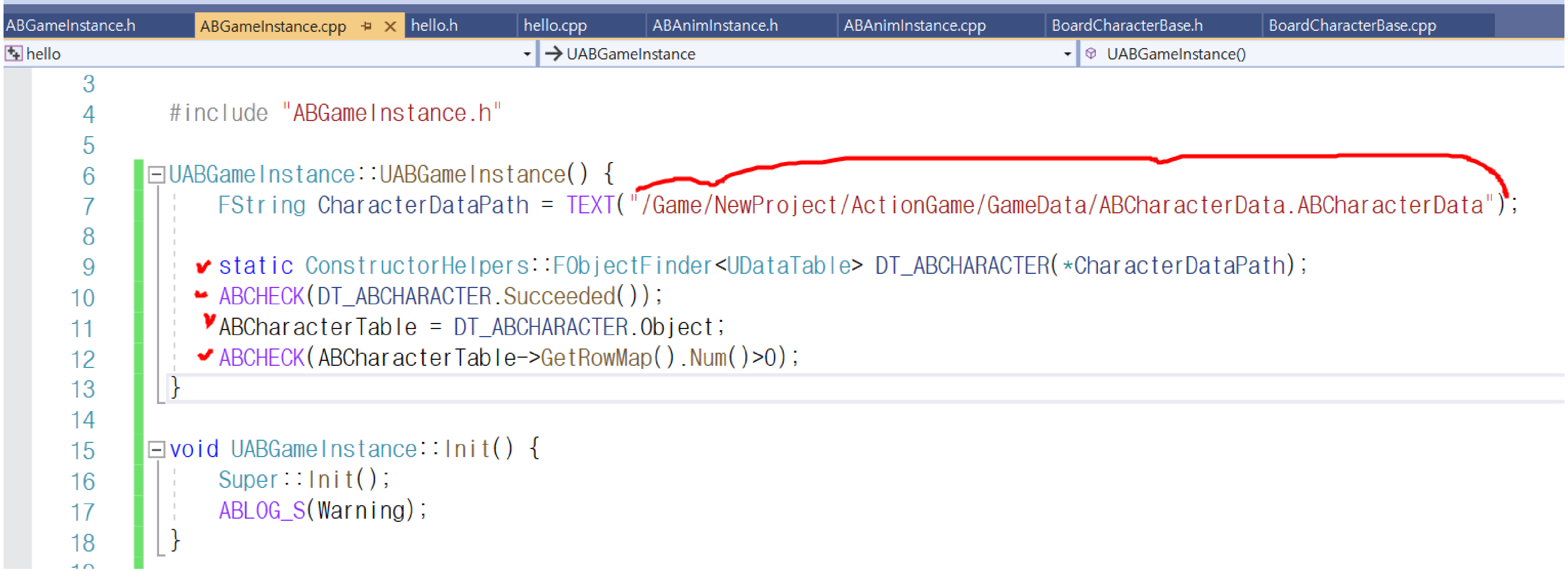


```cpp
UCLASS()
class HELLO_API UABGameInstance : public UGameInstance
{
    GENERATED_BODY()

public:
    UABGameInstance();

    virtual void Init() override;

    UFUNCTION(BlueprintCallable)
    FABCharacterData* GetABCharacterData(int32 Level);

private:
    UPROPERTY()
    class UDataTable* ABCharacterTable;

};
```

2.9 GetABCharacterData 함수와 UDataTable 포인터 ABCharacterTable을 선언

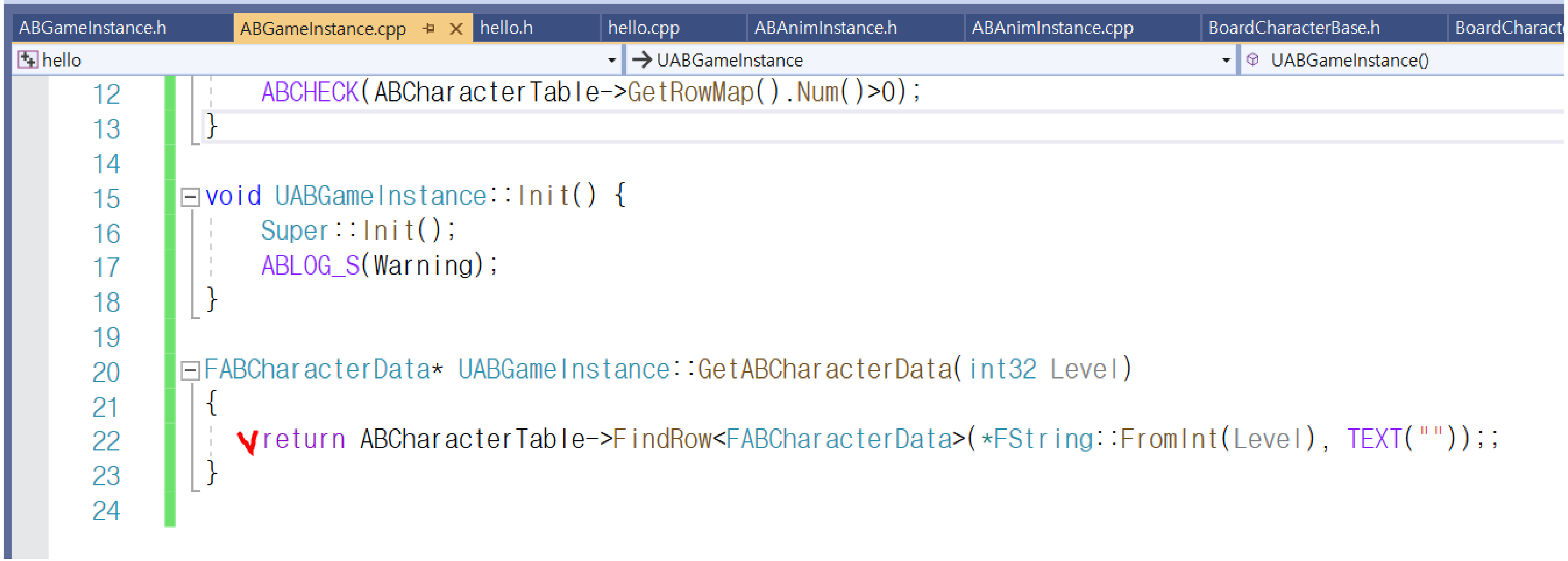


2.10 ABCharacterData의 레퍼런스 복사를 한다.



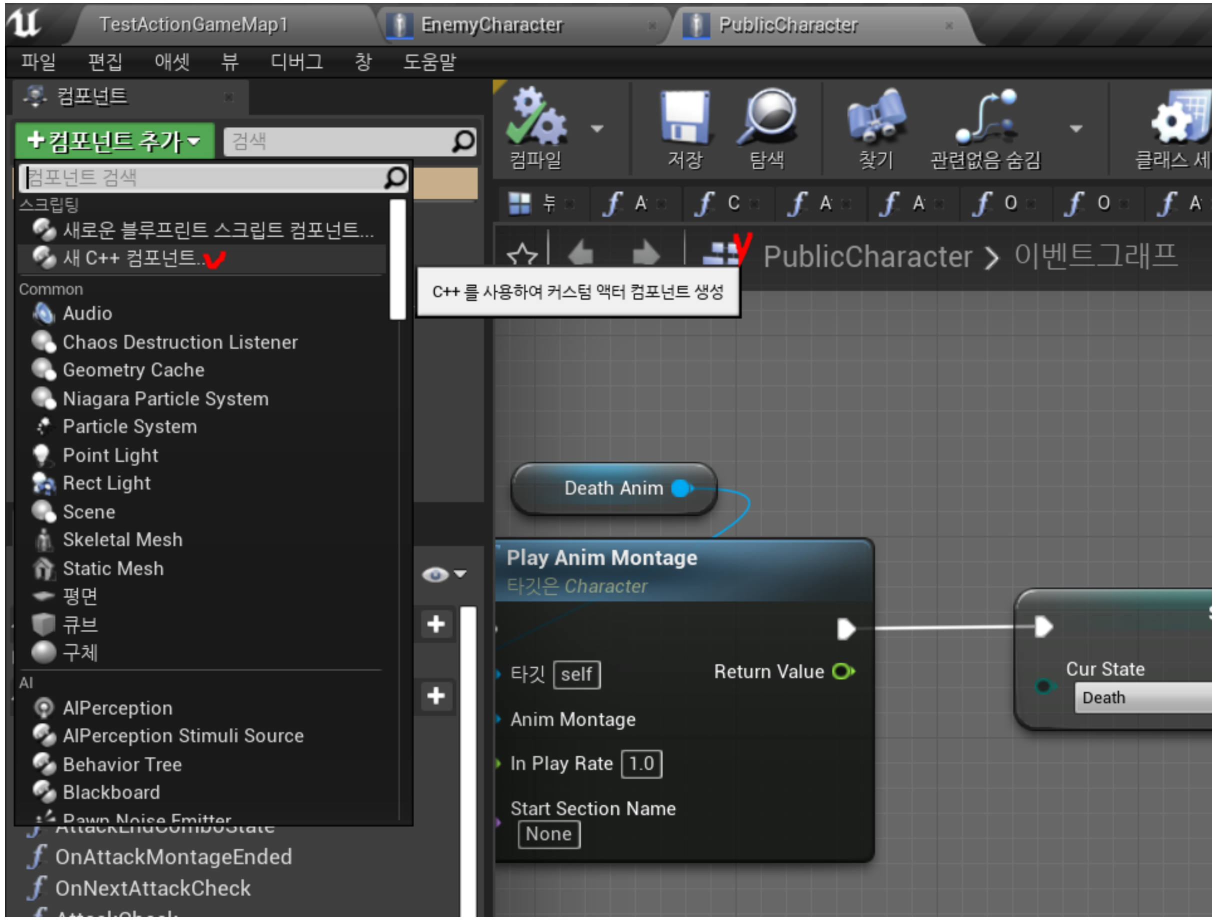
```

#include "ABGameInstance.h"

UABGameInstance::UABGameInstance() {
    FString CharacterDataPath = TEXT("/Game/NewProject/ActionGame/GameData/ABCharacterData.ABCharacterData");

    static ConstructorHelpers::FObjectFinder<UDataTable> DT_ABCHARACTER(*CharacterDataPath);
    ABCHECK(DT_ABCHARACTER.Succeeded());
    ABCharacterTable = DT_ABCHARACTER.Object;
    ABCHECK(ABCharacterTable->GetRowMap().Num()>0);
}

void UABGameInstance::Init() {
    Super::Init();
    ABLOG_S(Warning);
}
```

2.11 ABCharacterData의 레퍼런스를 가지고 와서 ABCharacterTable에 넣는다.



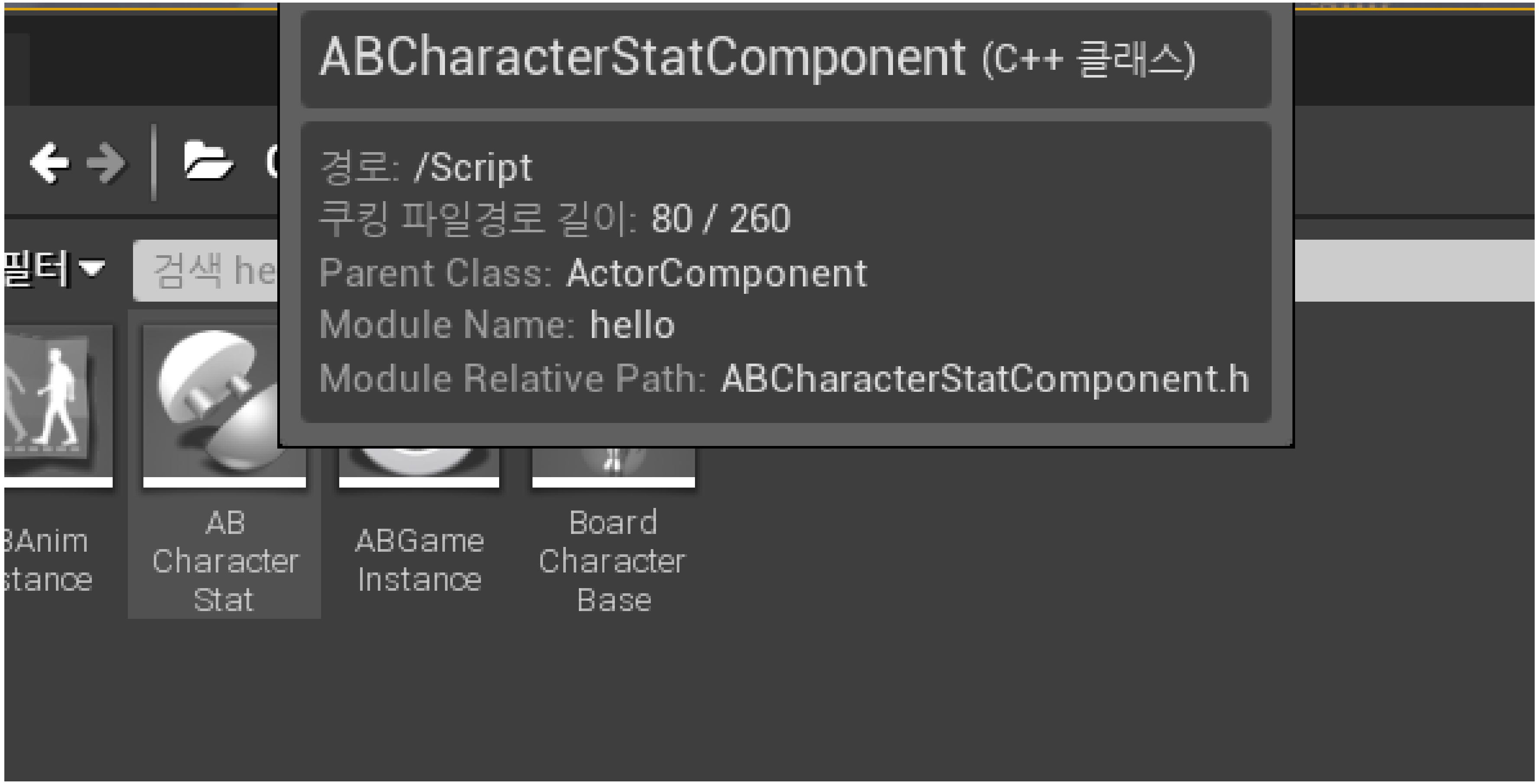
```
    ABCHECK(ABCharacterTable->GetRowMap().Num()>0);
}

void UABGameInstance::Init() {
    Super::Init();
    ABLOG_S(Warning);
}

FABCharacterData* UABGameInstance::GetABCharacterData(int32 Level)
{
    return ABCharacterTable->FindRow<FABCharacterData>(*FString::FromInt(Level), TEXT(""));;
}
```

2.12 GetABCharacterData 함수 구현



C++ 를 사용하여 커스텀 액터 컴포넌트 생성

3. PublicCharacter에 들어가서 컴포턴트 추가에 새 C++ 컴포넌트를 클릭한다.



ABCharacterStatComponent (C++ 클래스)

경로: /Script

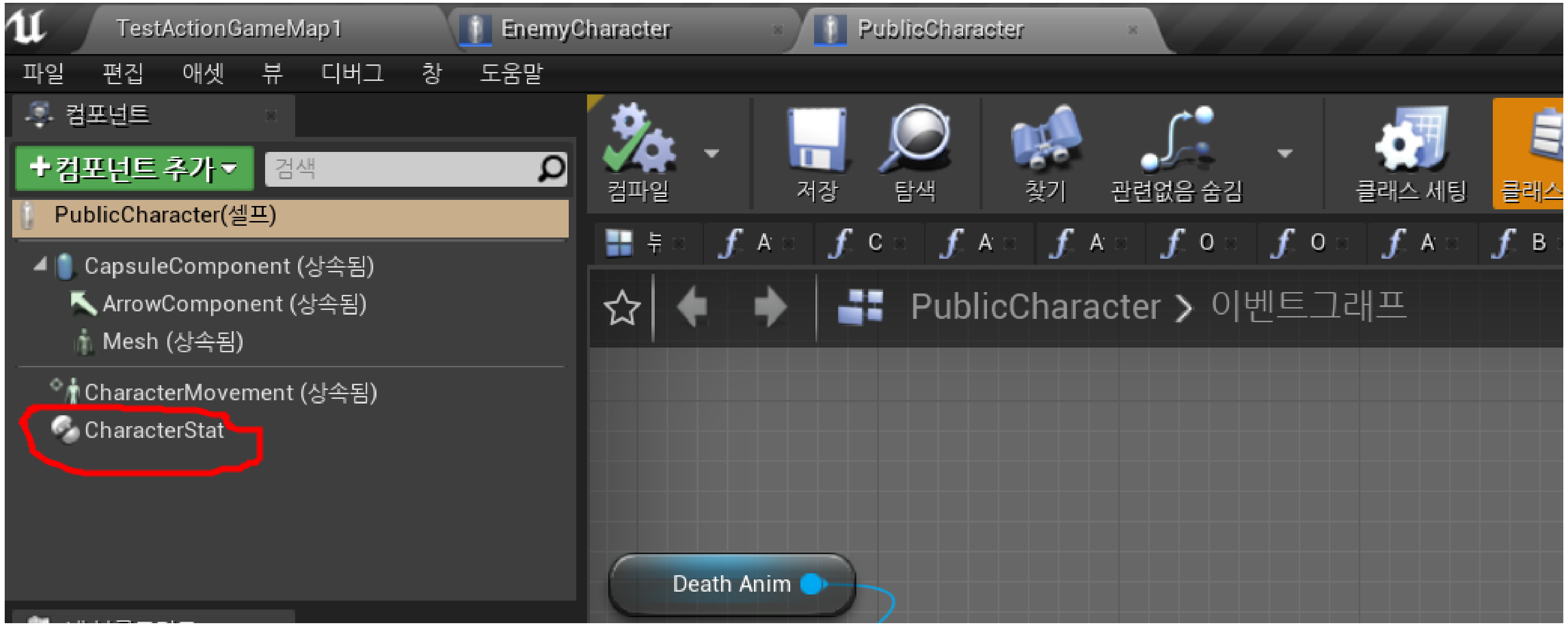
쿠킹 파일경로 길이: 80 / 260

Parent Class: ActorComponent

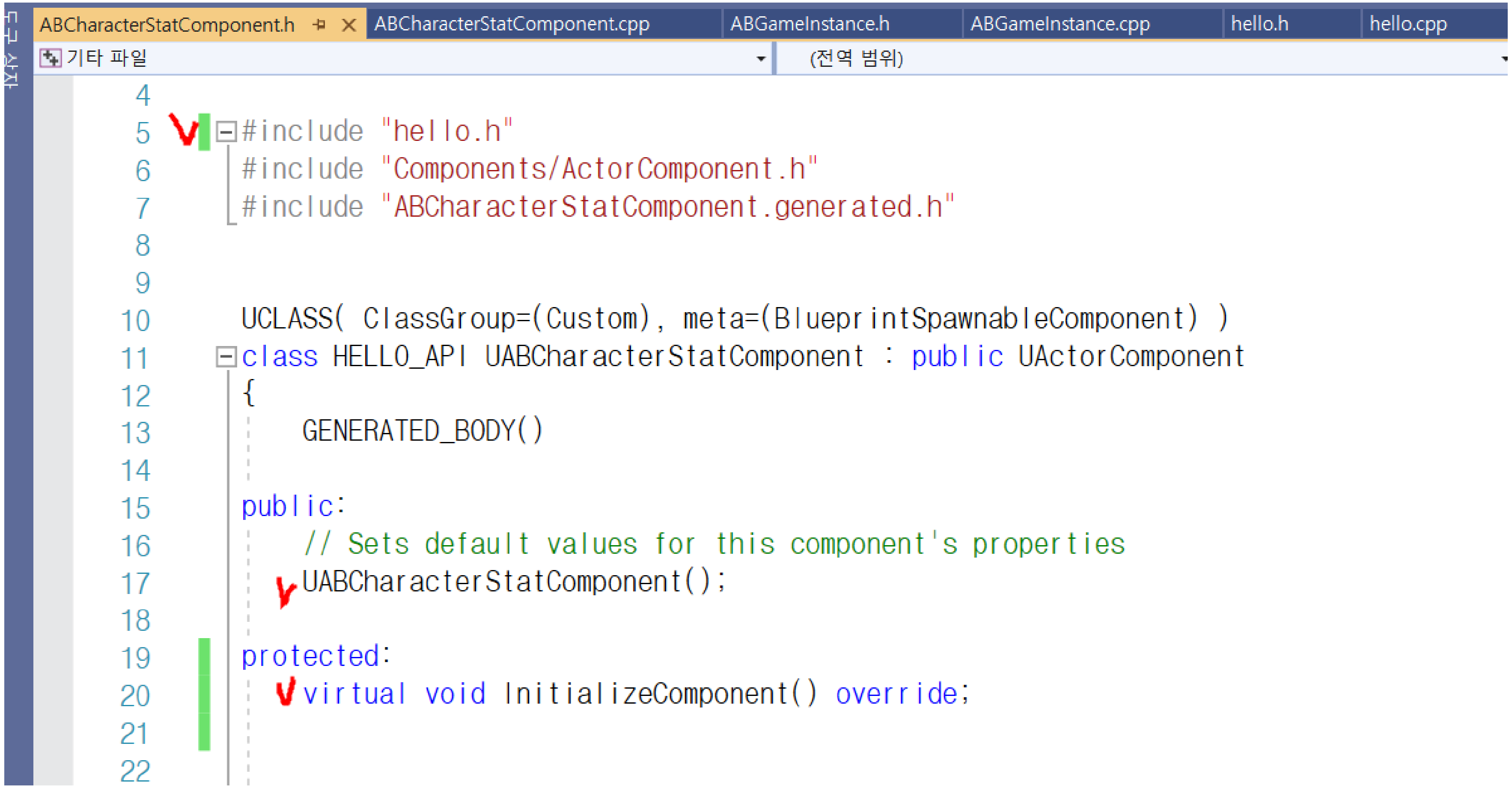
Module Name: hello

Module Relative Path: ABCharacterStatComponent.h

3.1 ActorComponent를 상속받은 ABCharacterStatComponent가 생성된다.

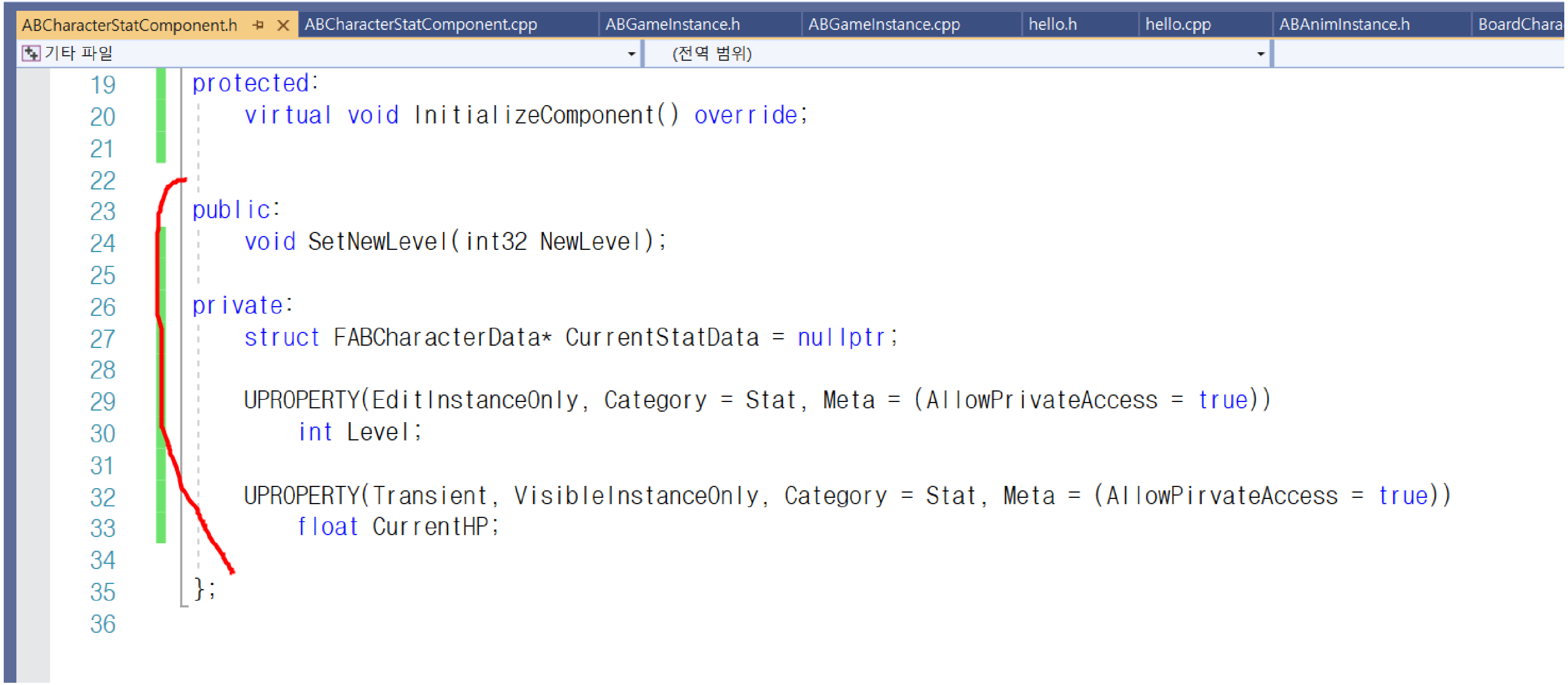


3.1.1 ABCharacterStatComponent를 컴포넌트에 추가하고 CharacterStat로 한다.



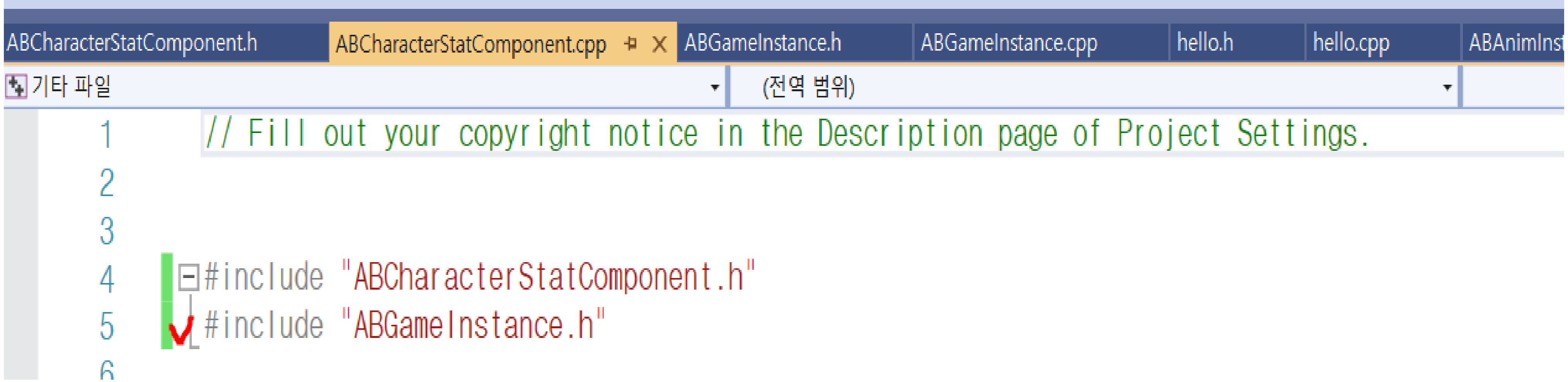
```cpp

#include "hello.h"
#include "Components/ActorComponent.h"
#include "ABCharacterStatComponent.generated.h"


UCLASS( ClassGroup=(Custom), meta=(BlueprintSpawnableComponent) )
class HELLO_API UABCharacterStatComponent : public UActorComponent
{
    GENERATED_BODY()

public:
    // Sets default values for this component's properties
    UABCharacterStatComponent();

protected:
    virtual void InitializeComponent() override;

```

3.2 프로젝트 헤더를 include 하고, 생성자, InitializeComponent 함수를 선언한다.



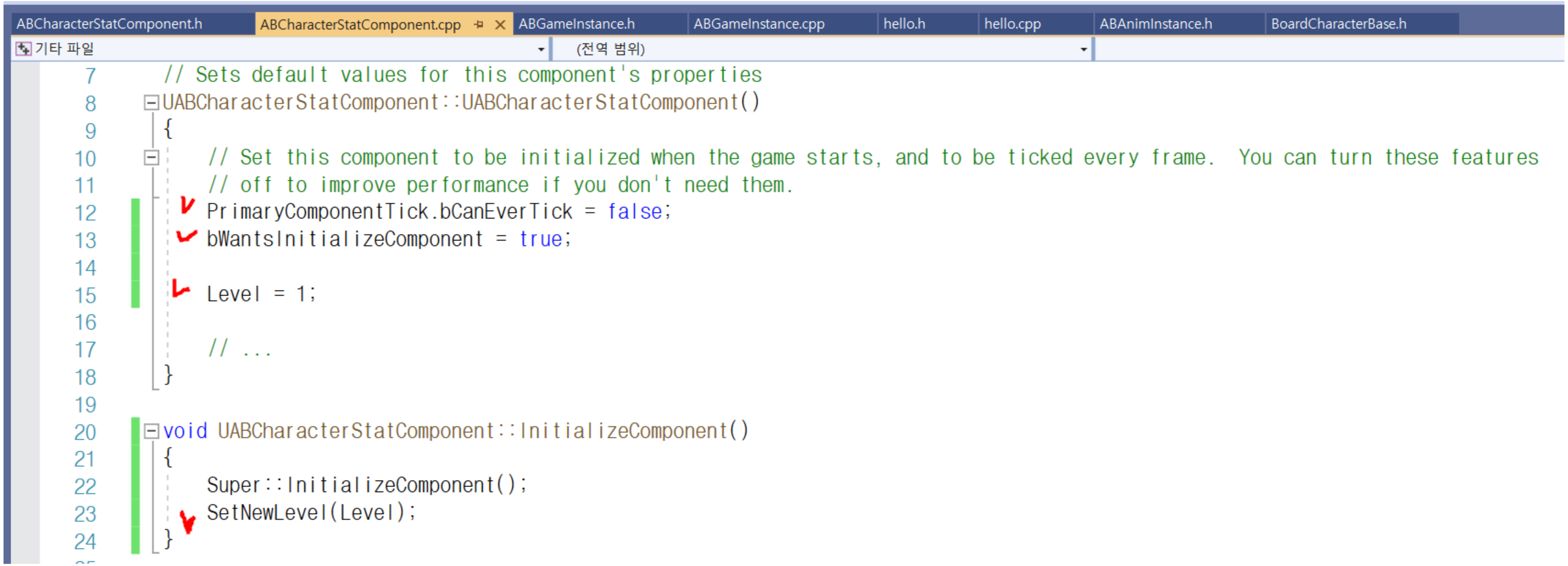
```
protected:
    virtual void InitializeComponent() override;


public:
    void SetNewLevel(int32 NewLevel);

private:
    struct FABCharacterData* CurrentStatData = nullptr;

    UPROPERTY(EditInstanceOnly, Category = Stat, Meta = (AllowPrivateAccess = true))
        int Level;

    UPROPERTY(Transient, VisibleInstanceOnly, Category = Stat, Meta = (AllowPirvateAccess = true))
        float CurrentHP;

};
```

3.3 SetNewLevel 함수와 변수들을 선언한다.



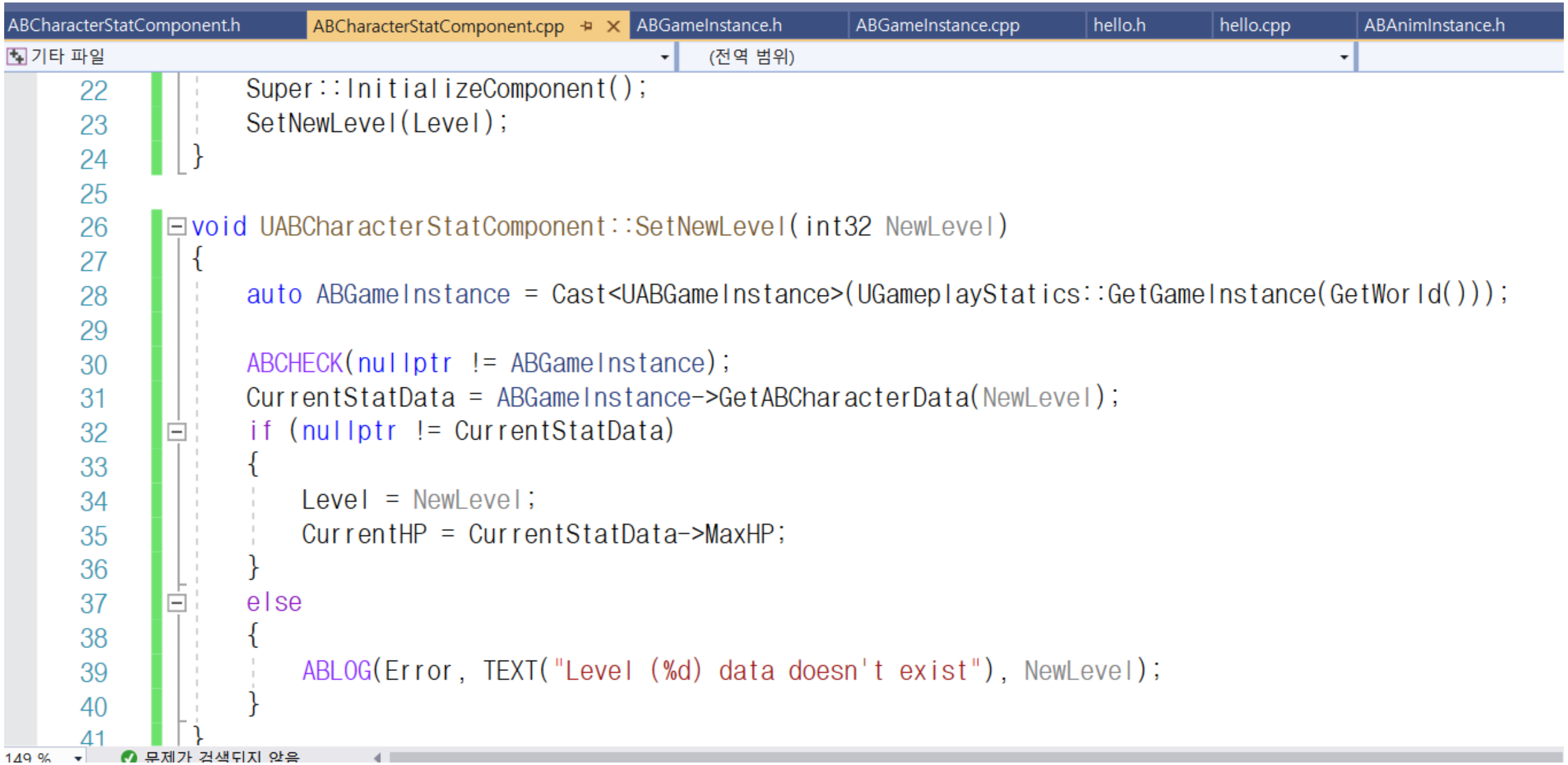
```
// Fill out your copyright notice in the Description page of Project Settings.


#include "ABCharacterStatComponent.h"
#include "ABGameInstance.h"
```

3.3.1 구현부로 너어가서 ABGameInstance 헤더를 추가한다.



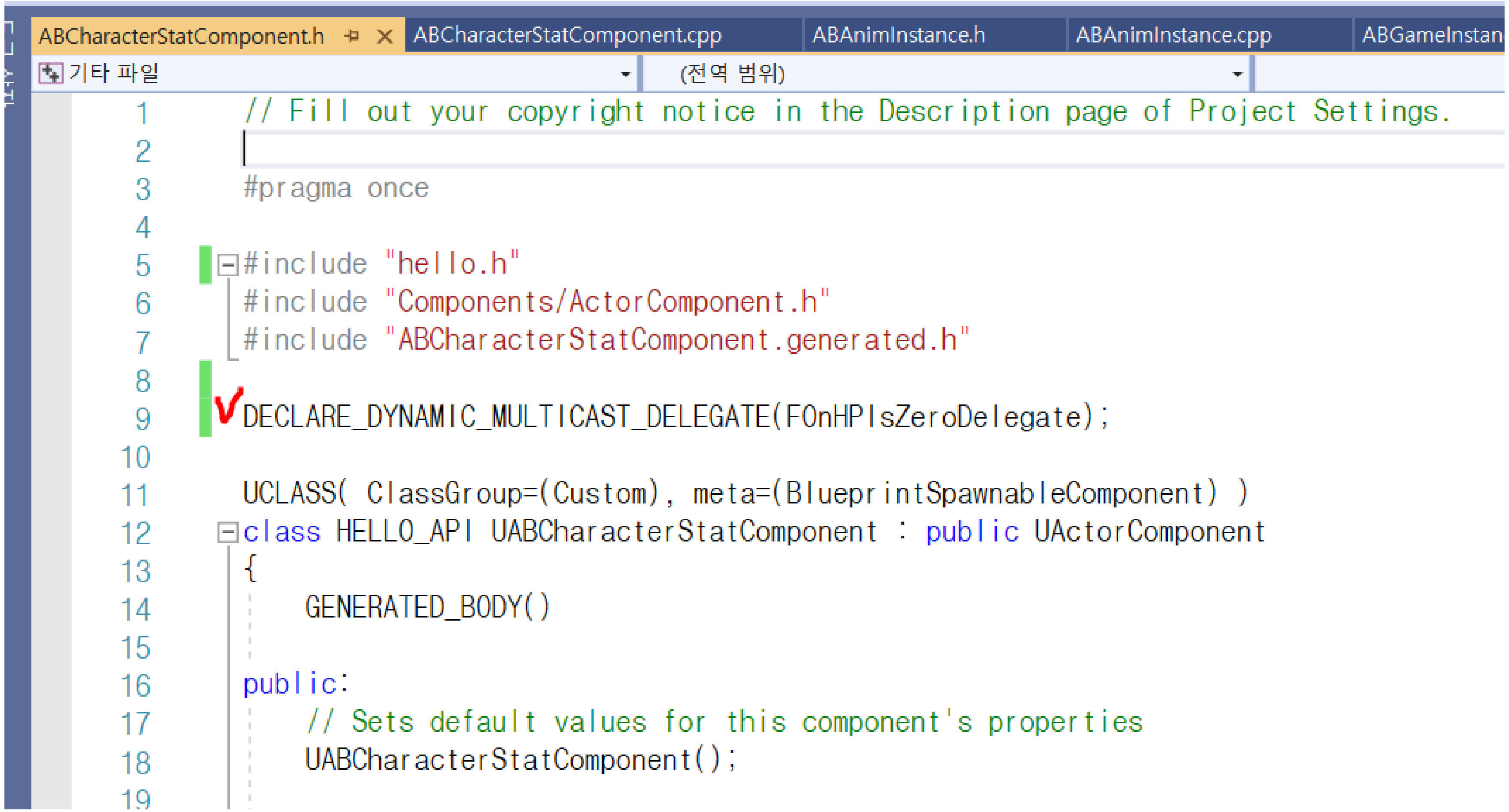
```cpp
// Sets default values for this component's properties
UABCharacterStatComponent::UABCharacterStatComponent()
{
	// Set this component to be initialized when the game starts, and to be ticked every frame.  You can turn these features
	// off to improve performance if you don't need them.
	PrimaryComponentTick.bCanEverTick = false;
	bWantsInitializeComponent = true;

	Level = 1;

	// ...
}

void UABCharacterStatComponent::InitializeComponent()
{
	Super::InitializeComponent();
	SetNewLevel(Level);
}
```

3.4 생성자에서 변수를 초기화 하고, InitializeComponent 함수에 SetNewLevel을 호출한다.



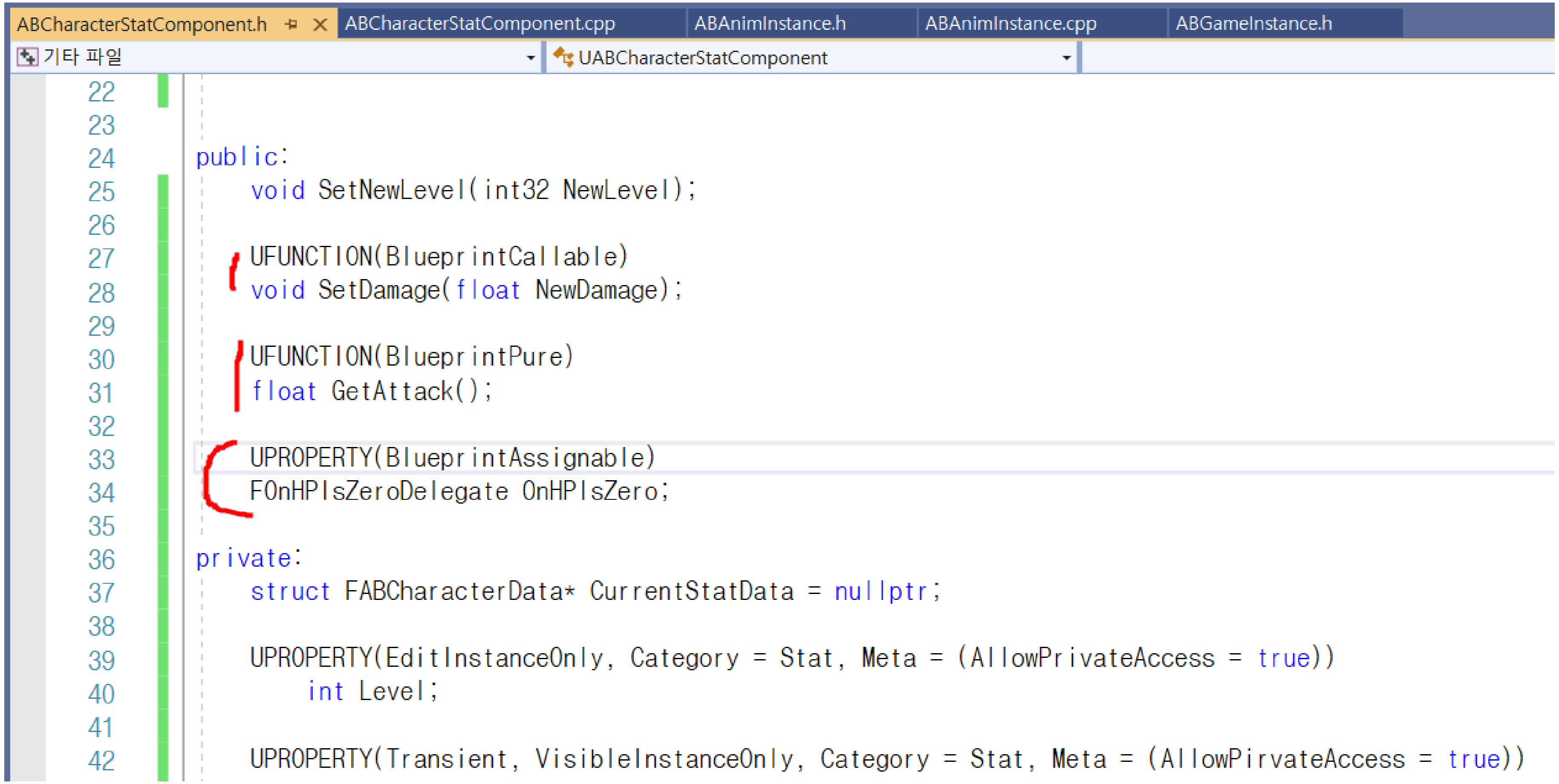
```
	Super::InitializeComponent();
	SetNewLevel(Level);
}

void UABCharacterStatComponent::SetNewLevel(int32 NewLevel)
{
	auto ABGameInstance = Cast<UABGameInstance>(UGameplayStatics::GetGameInstance(GetWorld()));

	ABCHECK(nullptr != ABGameInstance);
	CurrentStatData = ABGameInstance->GetABCharacterData(NewLevel);
	if (nullptr != CurrentStatData)
	{
		Level = NewLevel;
		CurrentHP = CurrentStatData->MaxHP;
	}
	else
	{
		ABLOG(Error, TEXT("Level (%d) data doesn't exist"), NewLevel);
	}
}
```

3.5 SetNewLevel 함수를 구현한다.



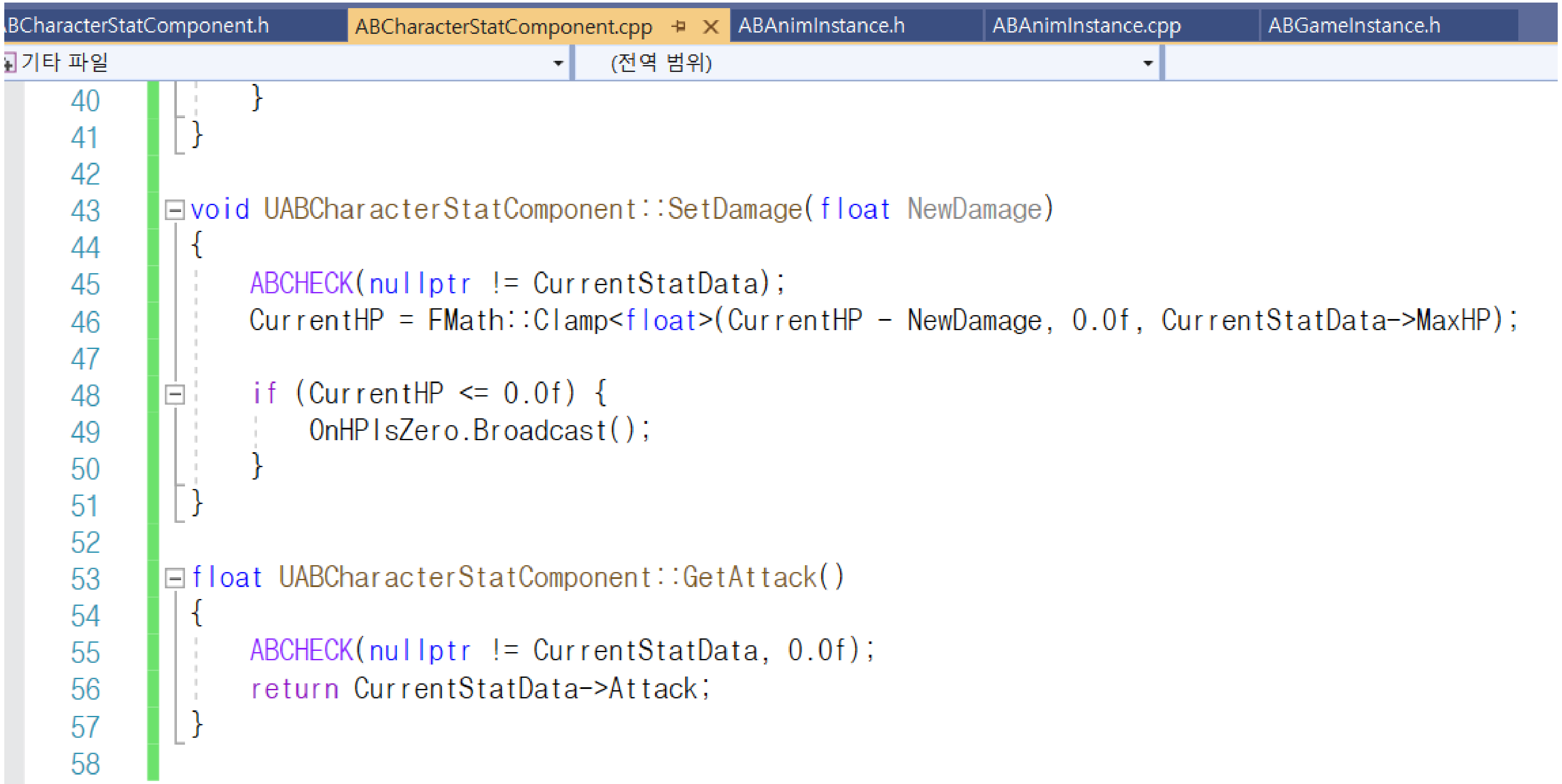
```
// Fill out your copyright notice in the Description page of Project Settings.

#pragma once

#include "hello.h"
#include "Components/ActorComponent.h"
#include "ABCharacterStatComponent.generated.h"

DECLARE_DYNAMIC_MULTICAST_DELEGATE(FOnHPIsZeroDelegate);

UCLASS( ClassGroup=(Custom), meta=(BlueprintSpawnableComponent) )
class HELLO_API UABCharacterStatComponent : public UActorComponent
{
	GENERATED_BODY()

public:
	// Sets default values for this component's properties
	UABCharacterStatComponent();

```

3.6 다이나믹 델리게이트 OnHPIsZeroDelegate 선언


ABCharacterStatComponent.h
ABCharacterStatComponent.cpp
ABAnimInstance.h
ABAnimInstance.cpp
ABGameInstance.h
기타 파일
UABCharacterStatComponent

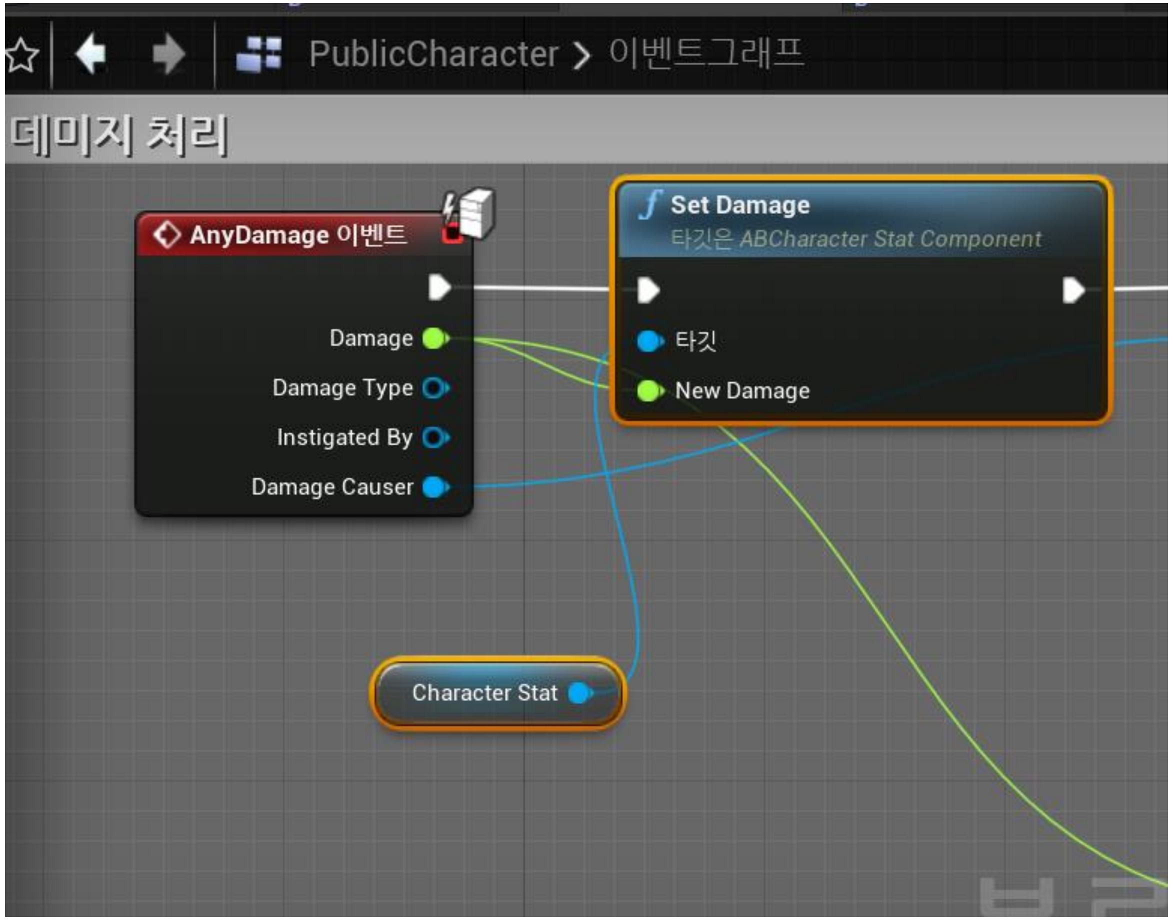
```cpp


public:
    void SetNewLevel(int32 NewLevel);

    UFUNCTION(BlueprintCallable)
    void SetDamage(float NewDamage);

    UFUNCTION(BlueprintPure)
    float GetAttack();

    UPROPERTY(BlueprintAssignable)
    FOnHPIsZeroDelegate OnHPIsZero;

private:
    struct FABCharacterData* CurrentStatData = nullptr;

    UPROPERTY(EditInstanceOnly, Category = Stat, Meta = (AllowPrivateAccess = true))
        int Level;

    UPROPERTY(Transient, VisibleInstanceOnly, Category = Stat, Meta = (AllowPirvateAccess = true))
```

3.7 SetDamge, GetAttack 함수 선언

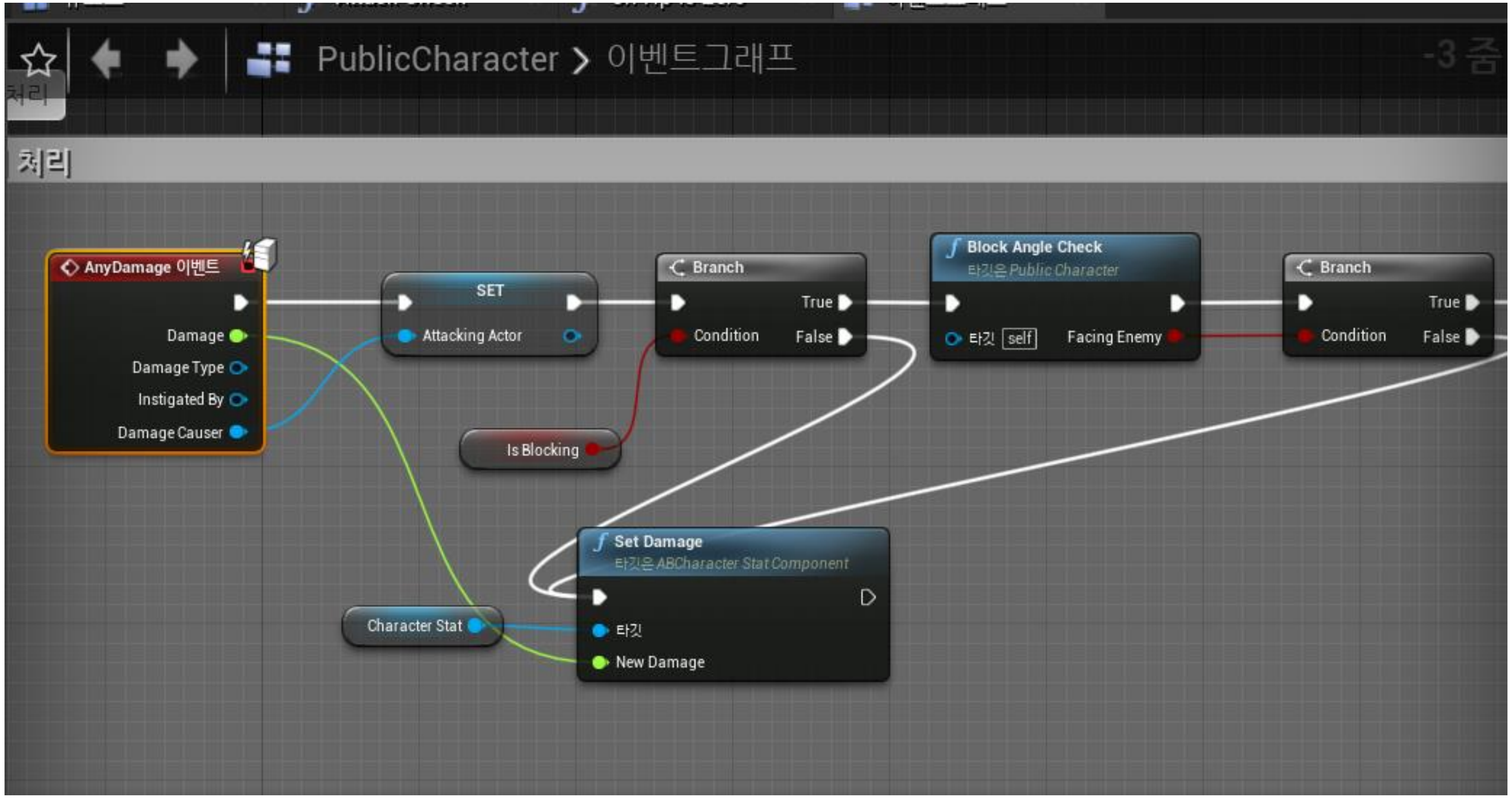


```
        }
}

void UABCharacterStatComponent::SetDamage(float NewDamage)
{
    ABCHECK(nullptr != CurrentStatData);
    CurrentHP = FMath::Clamp<float>(CurrentHP - NewDamage, 0.0f, CurrentStatData->MaxHP);

    if (CurrentHP <= 0.0f) {
        OnHPIsZero.Broadcast();
    }
}

float UABCharacterStatComponent::GetAttack()
{
    ABCHECK(nullptr != CurrentStatData, 0.0f);
    return CurrentStatData->Attack;
}
```

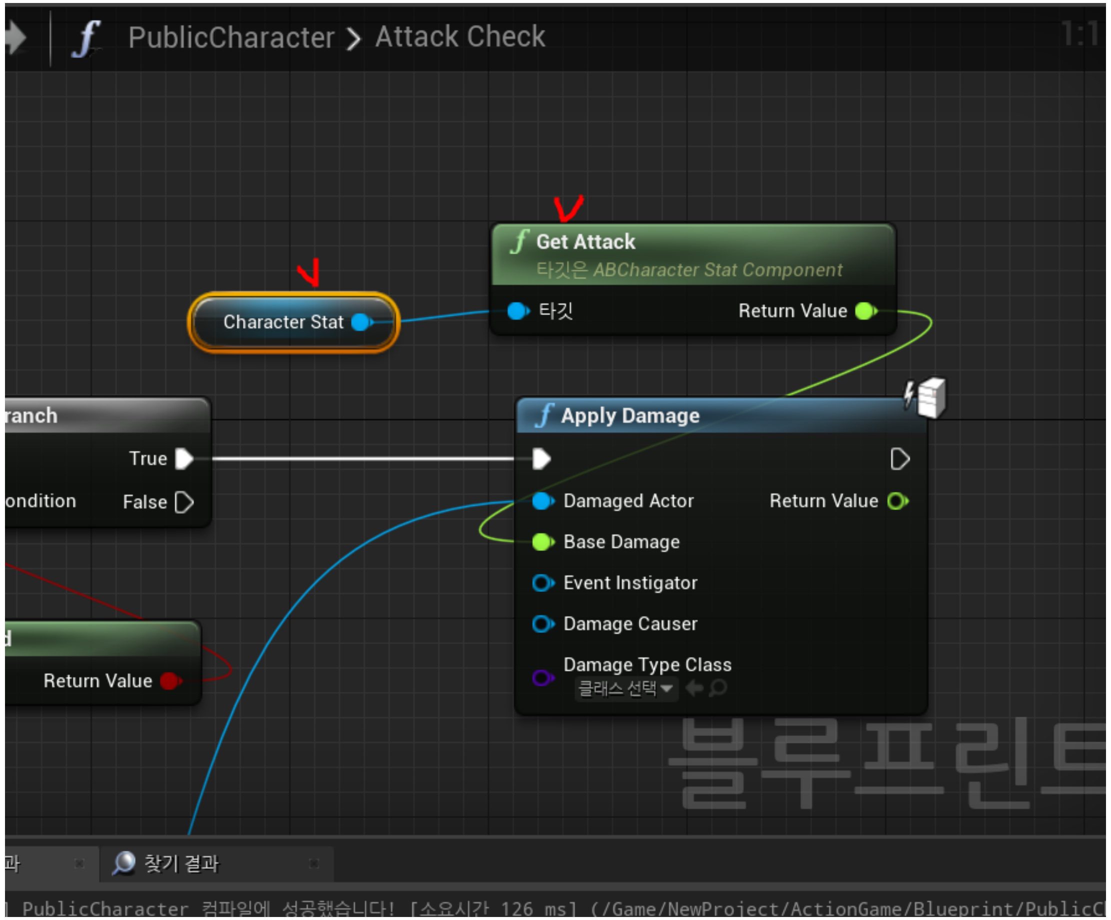
3.8 구현으로 넘어가서 SetDamage, GetAttack 함수 구현



4. PublicCharacter에 들어가서 CharacterStat의 SetDamage를 호출한다.

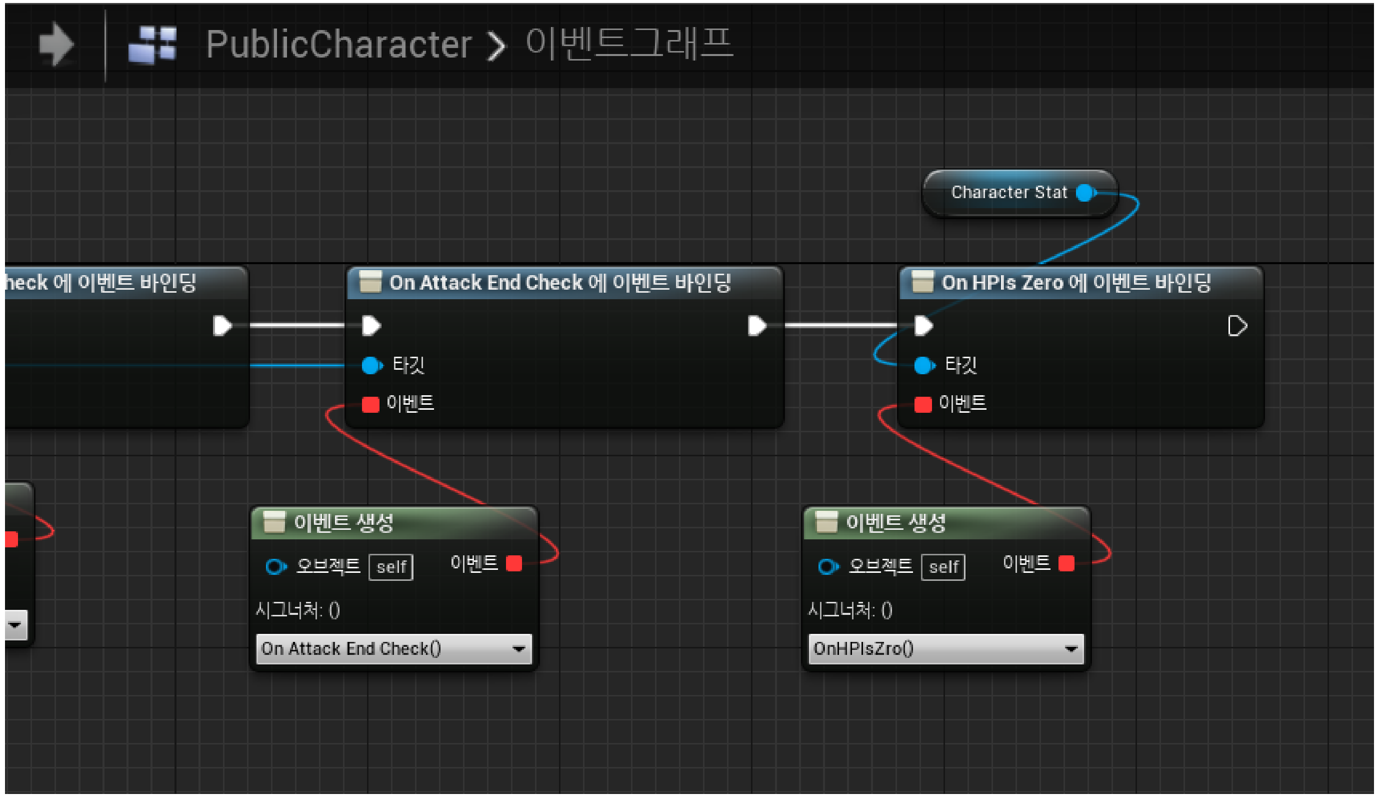


4.0 AnyDamage 이벤트를 정리한 결과

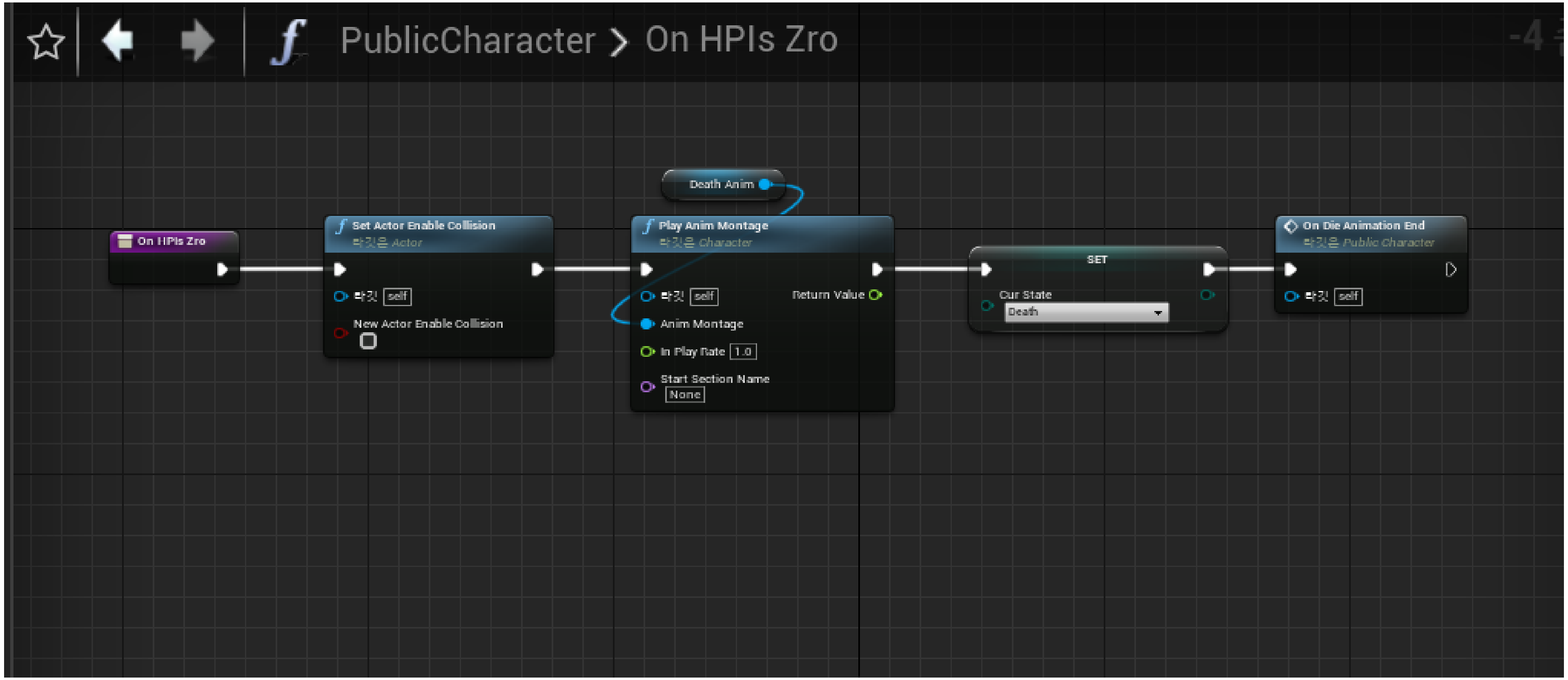


```
] PublicCharacter 컴파일에 성공했습니다! [소요시간 126 ms] (/Game/NewProject/ActionGame/Blueprint/PublicC
```

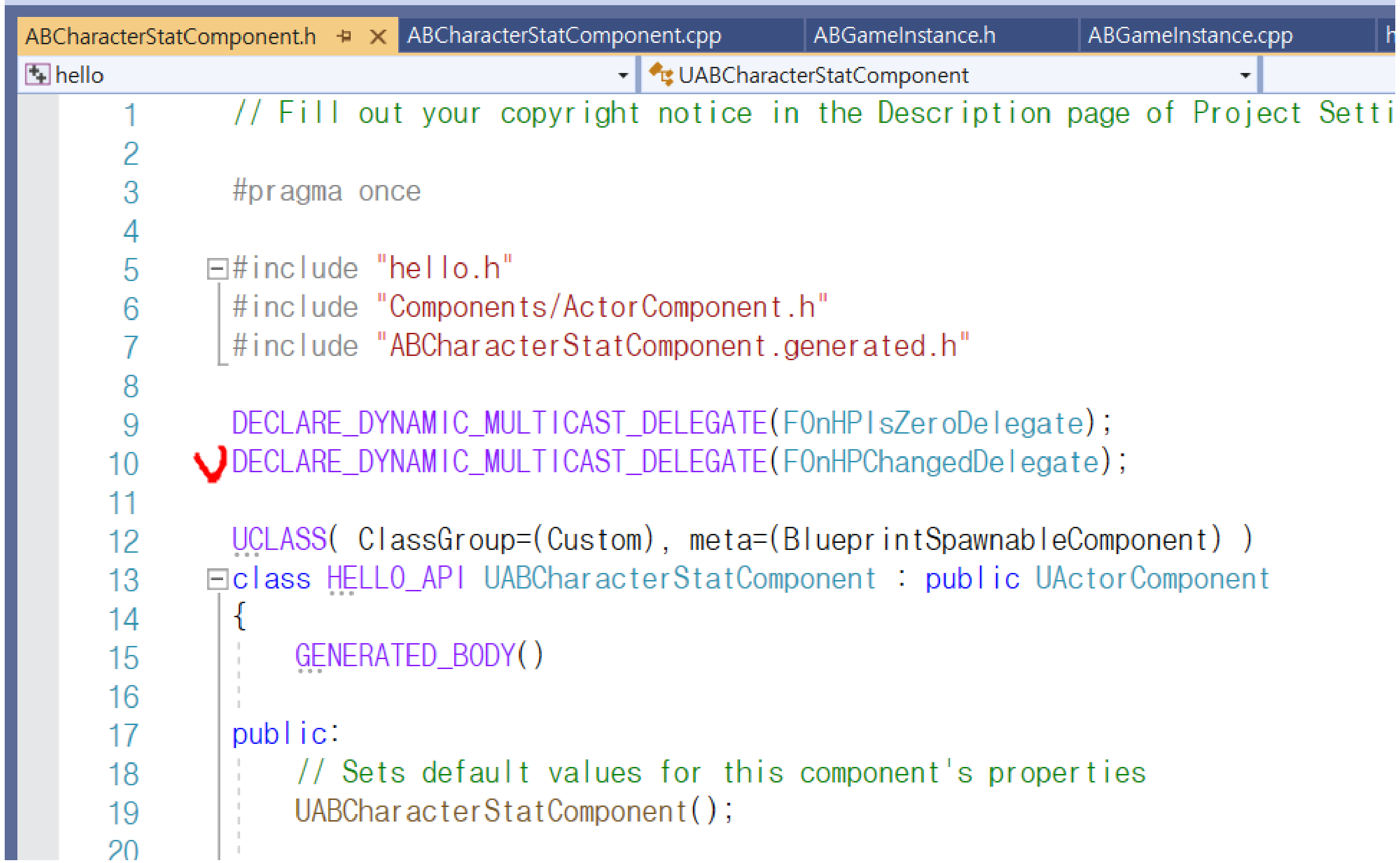
4.1 AttackCheck 함수로 넘어가서 CharacterStat의 GetAttack 함수를 호출한다.



4.2 CharacterStat의 OnHPIsZero를 바인하고 함수를 생성한다.

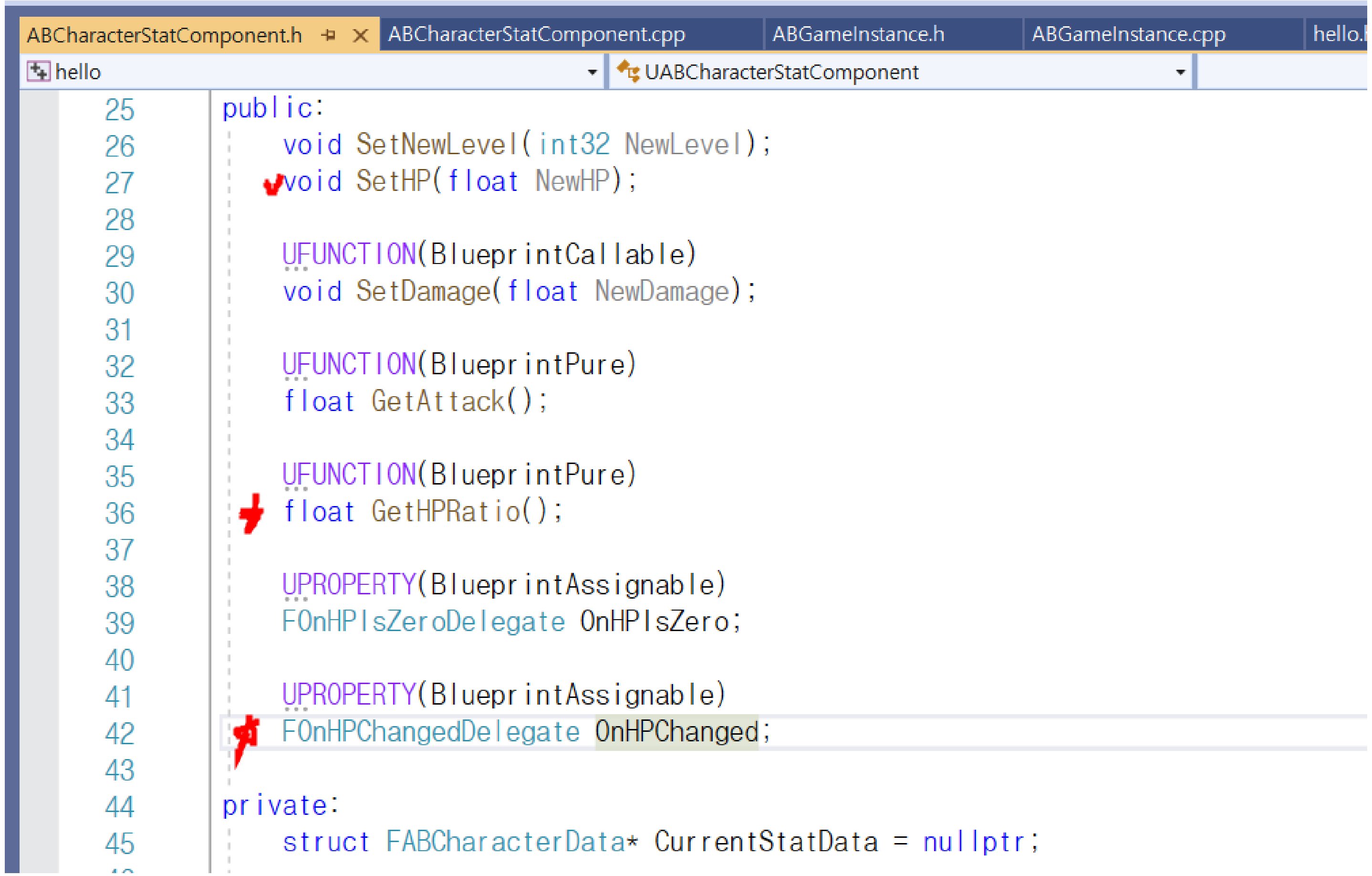


4.3 OnHPisZero 함수로 넘어가서 죽음 처리를 한다.


ABCharacterStatComponent.h
ABCharacterStatComponent.cpp
ABGameInstance.h
ABGameInstance.cpp
hello
UABCharacterStatComponent

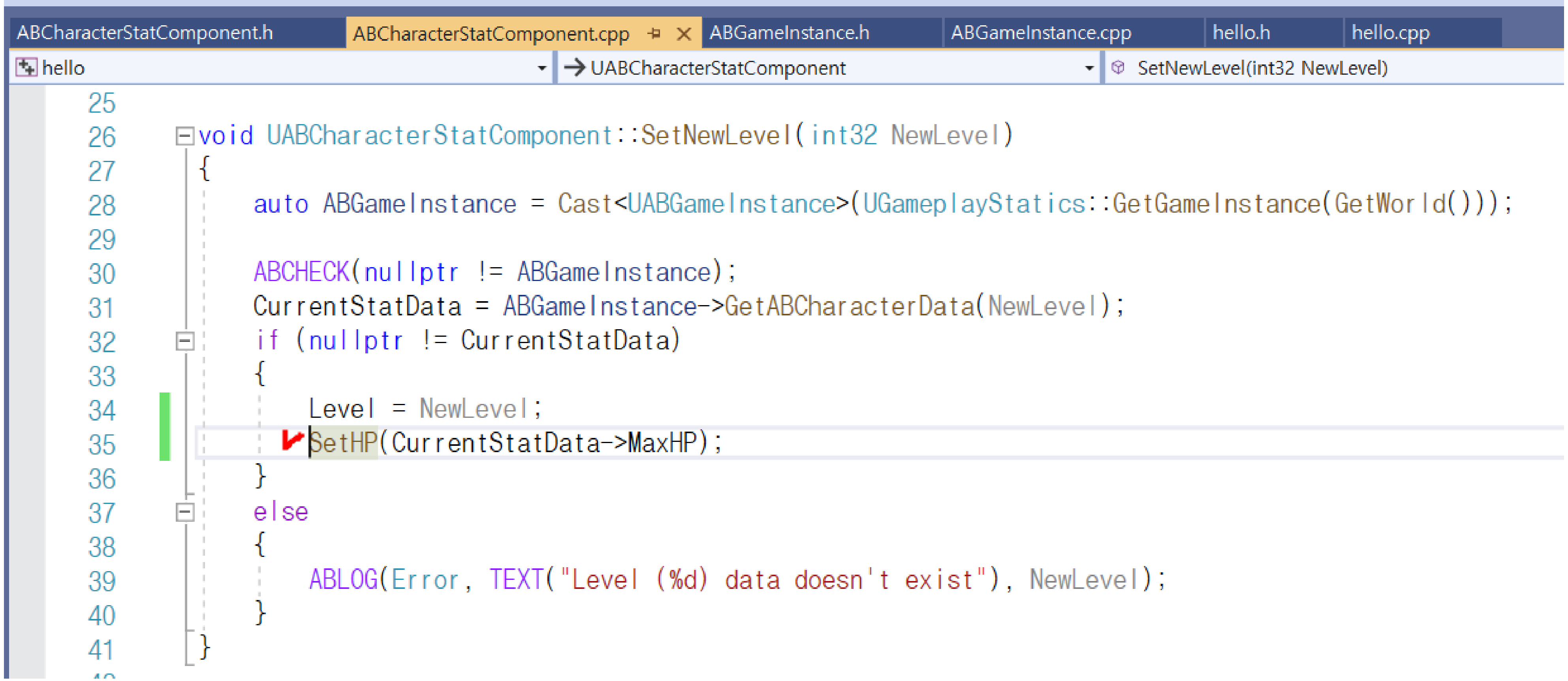
```cpp
// Fill out your copyright notice in the Description page of Project Setti

#pragma once

#include "hello.h"
#include "Components/ActorComponent.h"
#include "ABCharacterStatComponent.generated.h"

DECLARE_DYNAMIC_MULTICAST_DELEGATE(FOnHPIsZeroDelegate);
DECLARE_DYNAMIC_MULTICAST_DELEGATE(FOnHPChangedDelegate);

UCLASS( ClassGroup=(Custom), meta=(BlueprintSpawnableComponent) )
class HELLO_API UABCharacterStatComponent : public UActorComponent
{
	GENERATED_BODY()

public:
	// Sets default values for this component's properties
	UABCharacterStatComponent();

```

5. 다이나믹 델리게이트 OnHPChangedDelegate 선언


ABCharacterStatComponent.h
ABCharacterStatComponent.cpp
ABGameInstance.h
ABGameInstance.cpp
hello.
hello
UABCharacterStatComponent

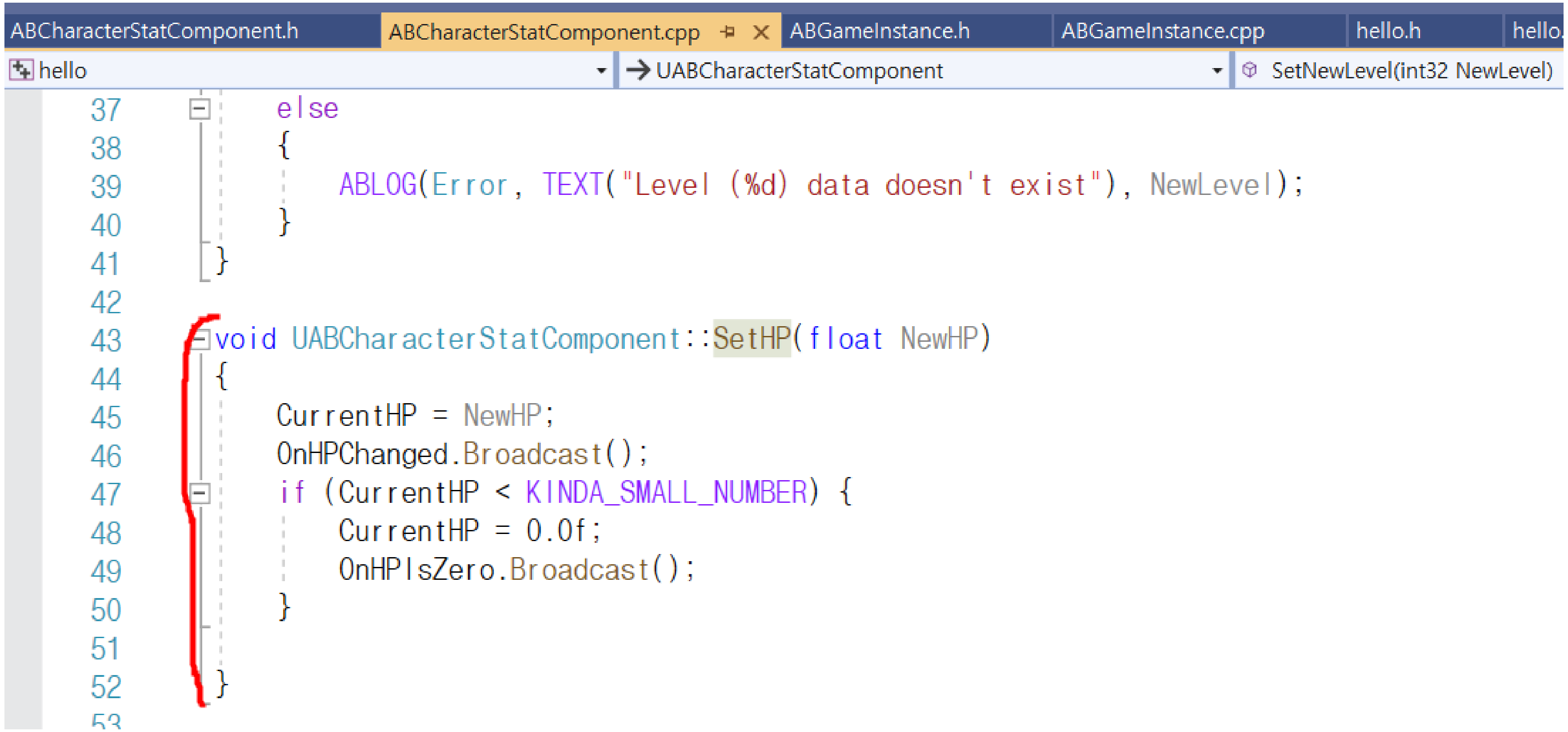
```
public:
    void SetNewLevel(int32 NewLevel);
    void SetHP(float NewHP);

    UFUNCTION(BlueprintCallable)
    void SetDamage(float NewDamage);

    UFUNCTION(BlueprintPure)
    float GetAttack();

    UFUNCTION(BlueprintPure)
    float GetHPRatio();

    UPROPERTY(BlueprintAssignable)
    FOnHPIsZeroDelegate OnHPIsZero;

    UPROPERTY(BlueprintAssignable)
    FOnHPChangedDelegate OnHPChanged;

private:
    struct FABCharacterData* CurrentStatData = nullptr;
```

5.1 SetHP, GetHPRatio 함수 선언, OnHPChanged를 선언


ABCharacterStatComponent.h
ABCharacterStatComponent.cpp
ABGameInstance.h
ABGameInstance.cpp
hello.h
hello.cpp
hello
UABCharacterStatComponent
SetNewLevel(int32 NewLevel)

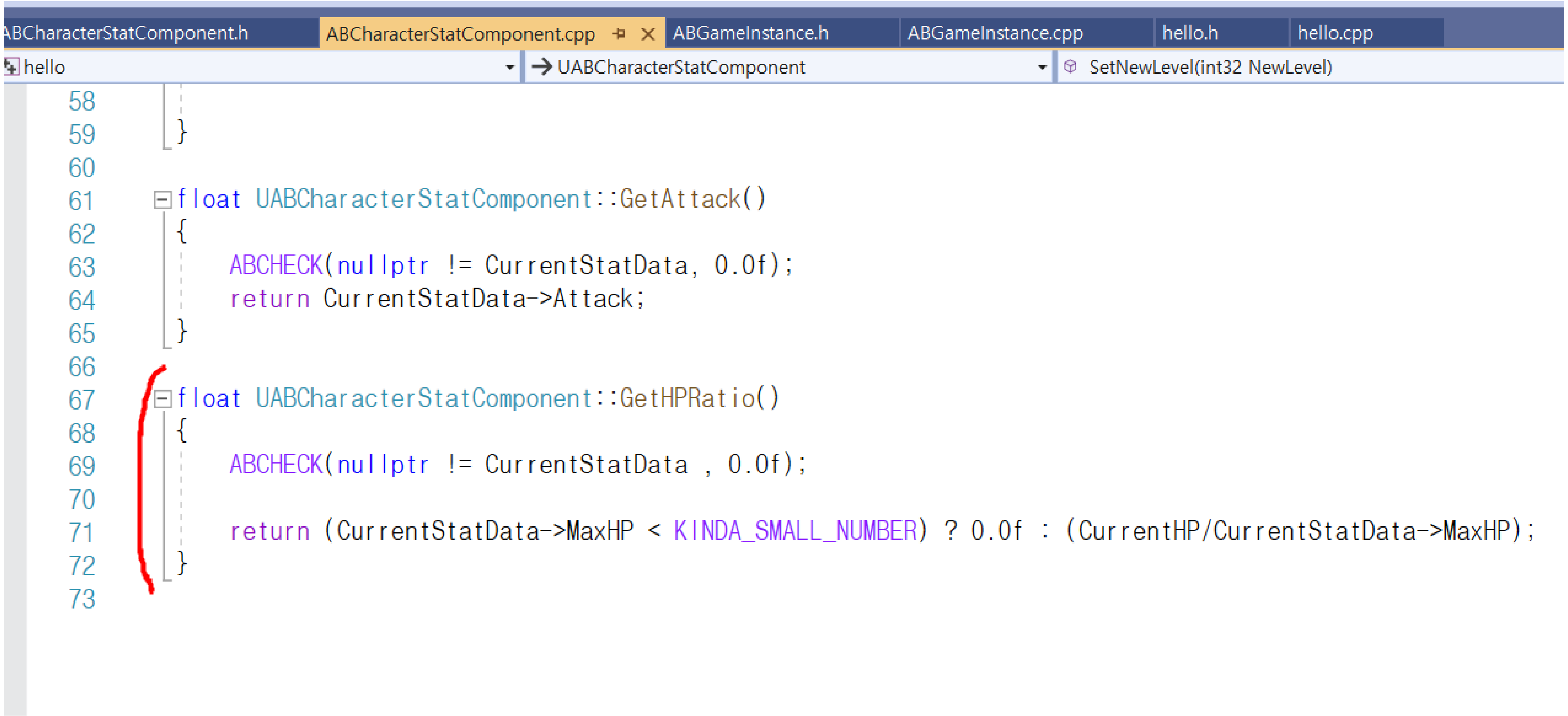
```

void UABCharacterStatComponent::SetNewLevel(int32 NewLevel)
{
    auto ABGameInstance = Cast<UABGameInstance>(UGameplayStatics::GetGameInstance(GetWorld()));

    ABCHECK(nullptr != ABGameInstance);
    CurrentStatData = ABGameInstance->GetABCharacterData(NewLevel);
    if (nullptr != CurrentStatData)
    {
        Level = NewLevel;
        SetHP(CurrentStatData->MaxHP);
    }
    else
    {
        ABLOG(Error, TEXT("Level (%d) data doesn't exist"), NewLevel);
    }
}
```

5.2 SetNewLevel 함수에 SetHP 함수 호출


ABCharacterStatComponent.h
ABCharacterStatComponent.cpp
ABGameInstance.h
ABGameInstance.cpp
hello.h
hello.
hello
UABCharacterStatComponent
SetNewLevel(int32 NewLevel)

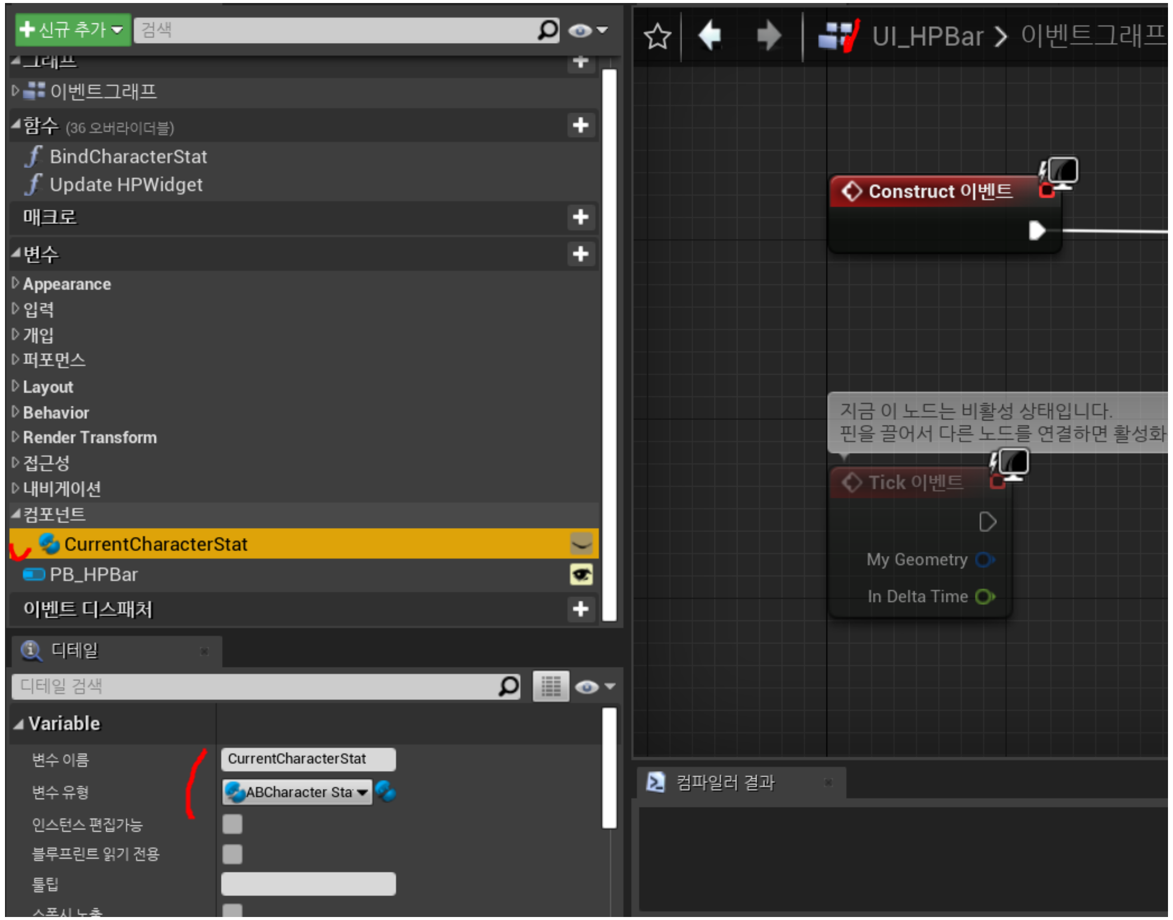
```cpp
    else
    {
        ABLOG(Error, TEXT("Level (%d) data doesn't exist"), NewLevel);
    }
}

void UABCharacterStatComponent::SetHP(float NewHP)
{
    CurrentHP = NewHP;
    OnHPChanged.Broadcast();
    if (CurrentHP < KINDA_SMALL_NUMBER) {
        CurrentHP = 0.0f;
        OnHPIsZero.Broadcast();
    }

}
```

5.3 SetHP 함수 구현


ABCharacterStatComponent.h
ABCharacterStatComponent.cpp
ABGameInstance.h
ABGameInstance.cpp
hello.h
hello.cpp
hello
UABCharacterStatComponent
SetNewLevel(int32 NewLevel)

```
}

float UABCharacterStatComponent::GetAttack()
{
    ABCHECK(nullptr != CurrentStatData, 0.0f);
    return CurrentStatData->Attack;
}

float UABCharacterStatComponent::GetHPRatio()
{
    ABCHECK(nullptr != CurrentStatData , 0.0f);

    return (CurrentStatData->MaxHP < KINDA_SMALL_NUMBER) ? 0.0f : (CurrentHP/CurrentStatData->MaxHP);
}
```

5.4 GetHPRatio 함수 구현



변수 이름: CurrentCharacterStat

변수 유형: ABCharacter Sta

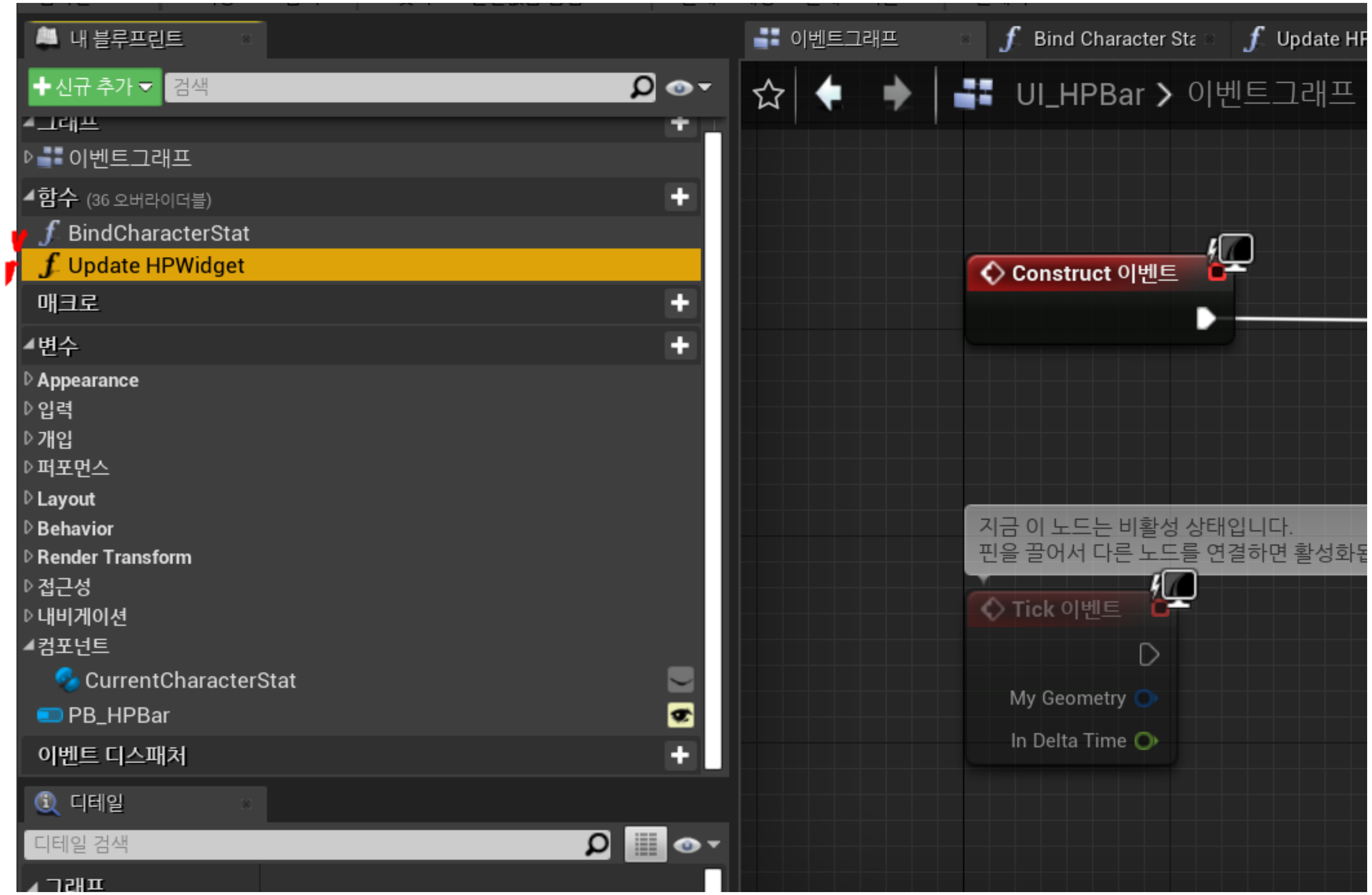
인스턴스 편집가능: □

블루프린트 읽기 전용: □

툴팁:

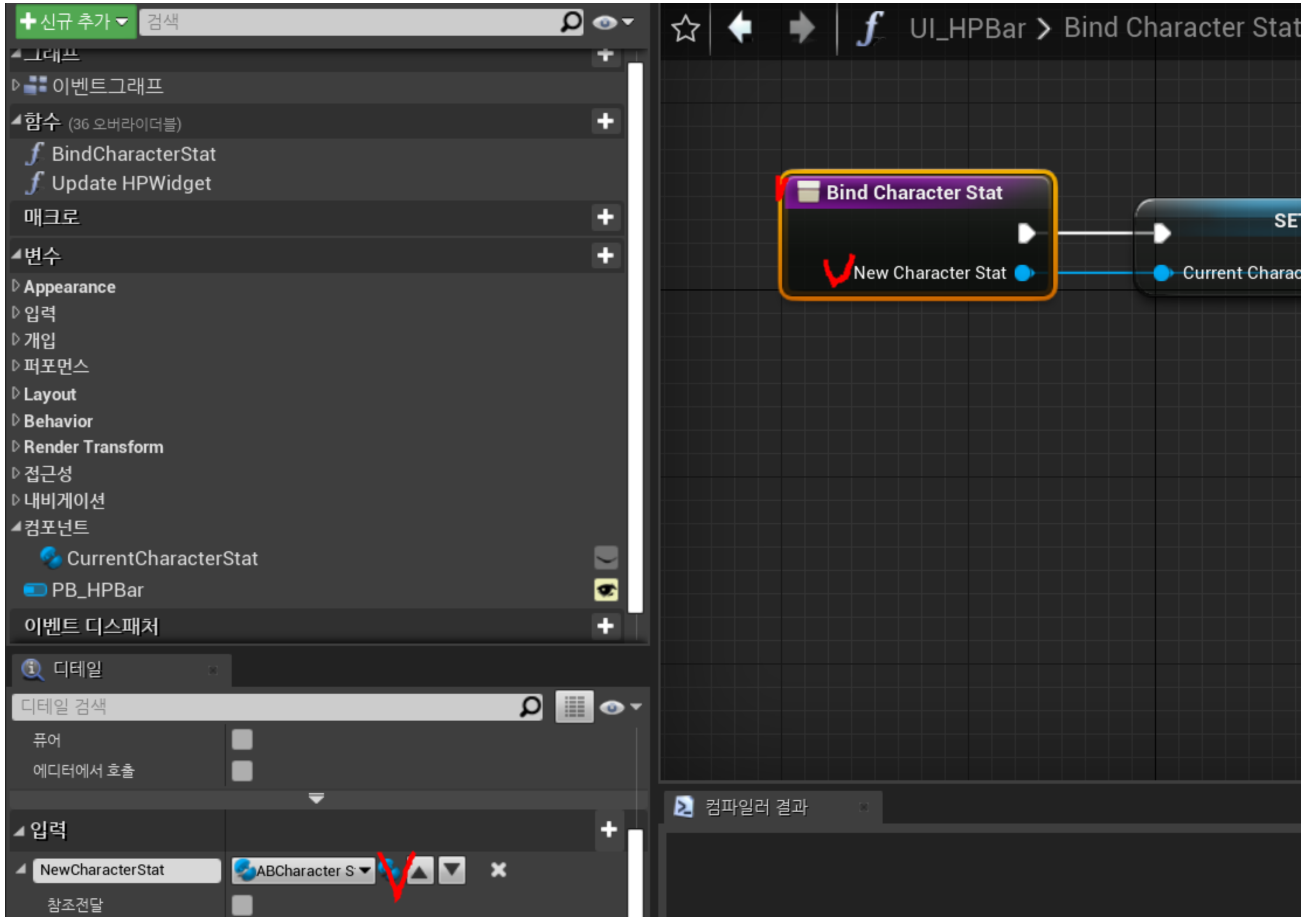
지금 이 노드는 비활성 상태입니다.
핀을 끌어서 다른 노드를 연결하면 활성화

6. UI_HPBar에 들어가서 ABCharacterStatComponent 변수 CurrentCharacterStat 선언



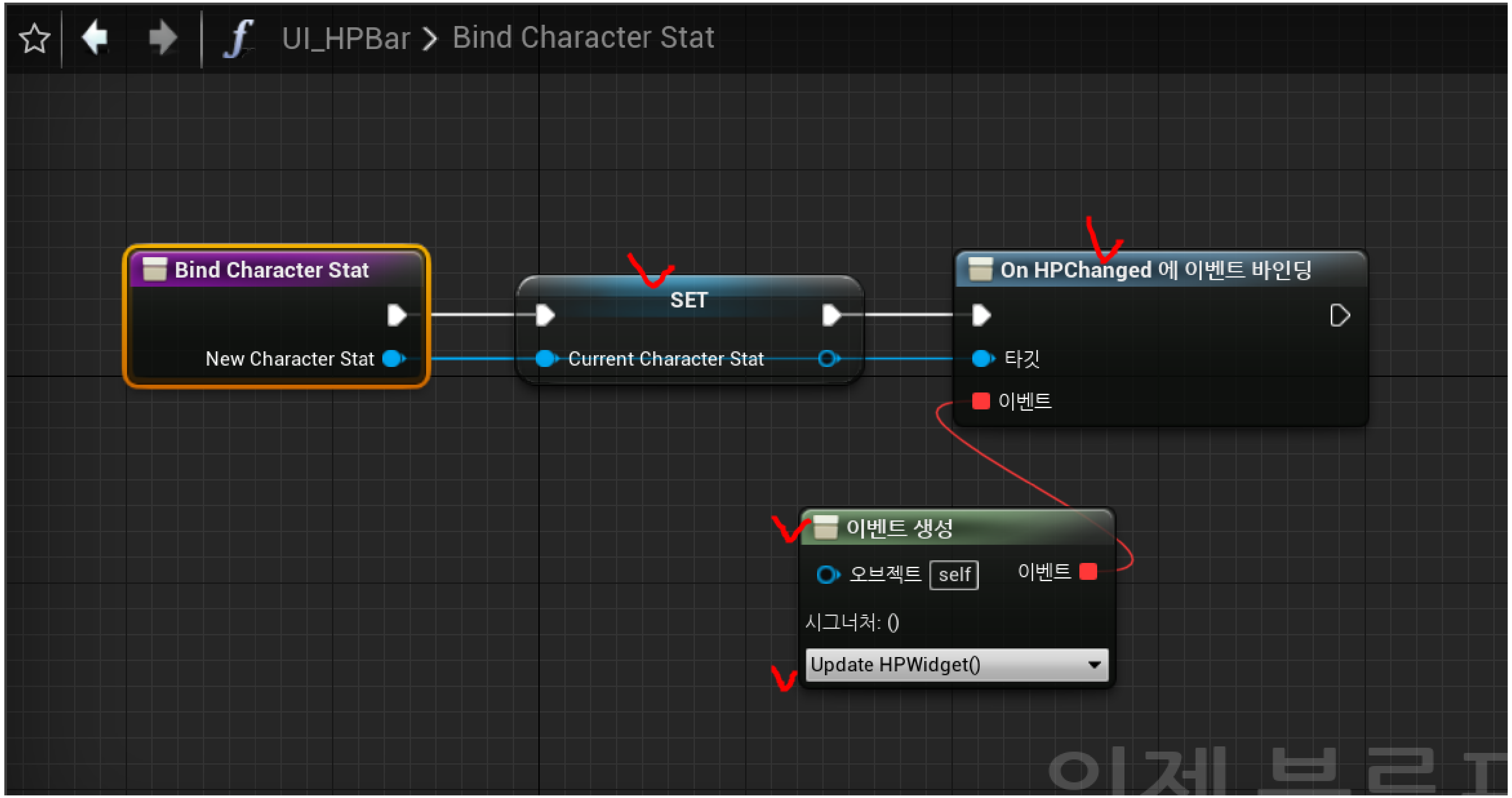
지금 이 노드는 비활성 상태입니다.
핀을 끌어서 다른 노드를 연결하면 활성화됨

6.1 BindCharacterStat, UpdateHPWidget 함수 선언

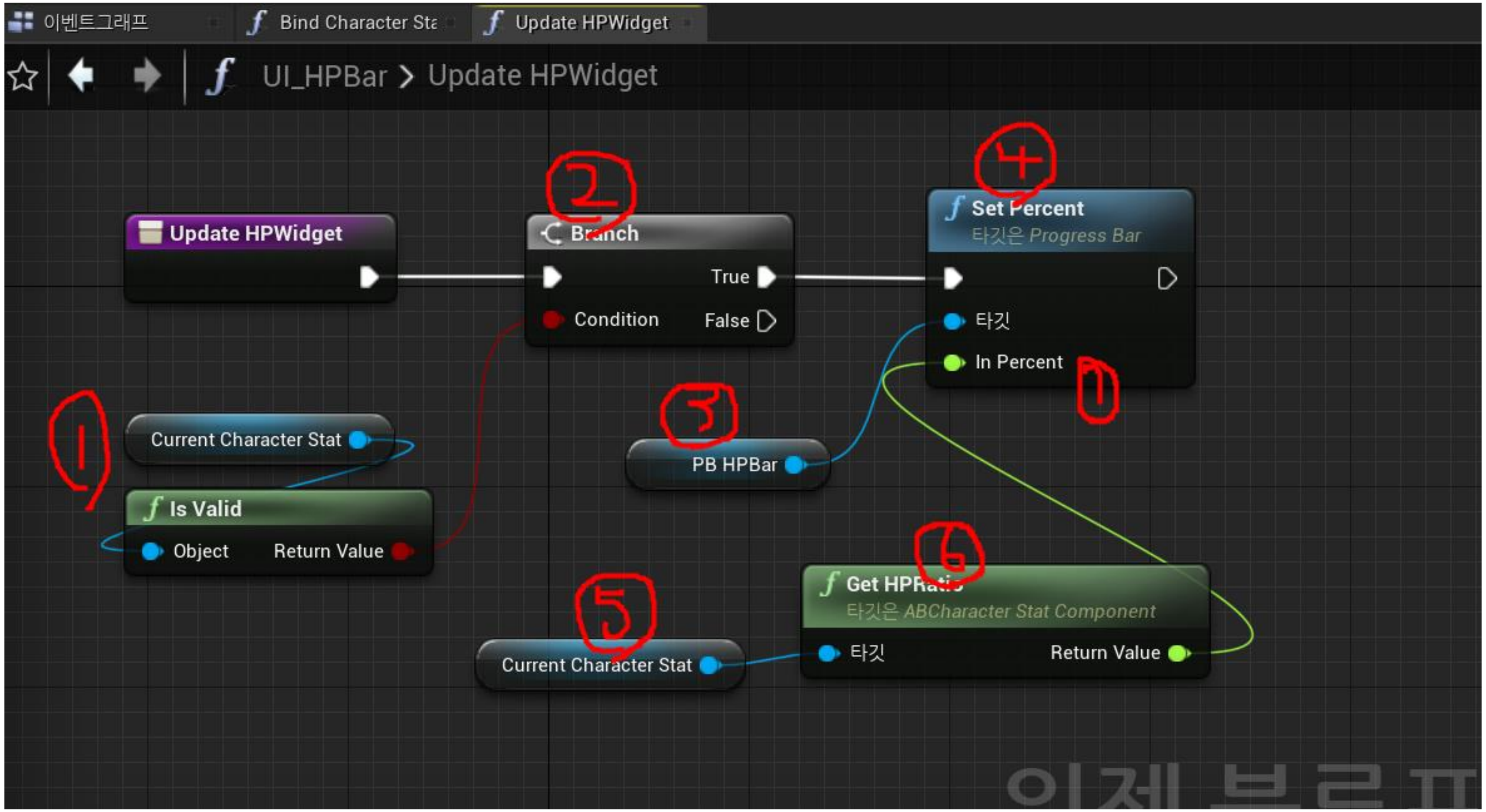


NewCharacterStat ABCharacter S

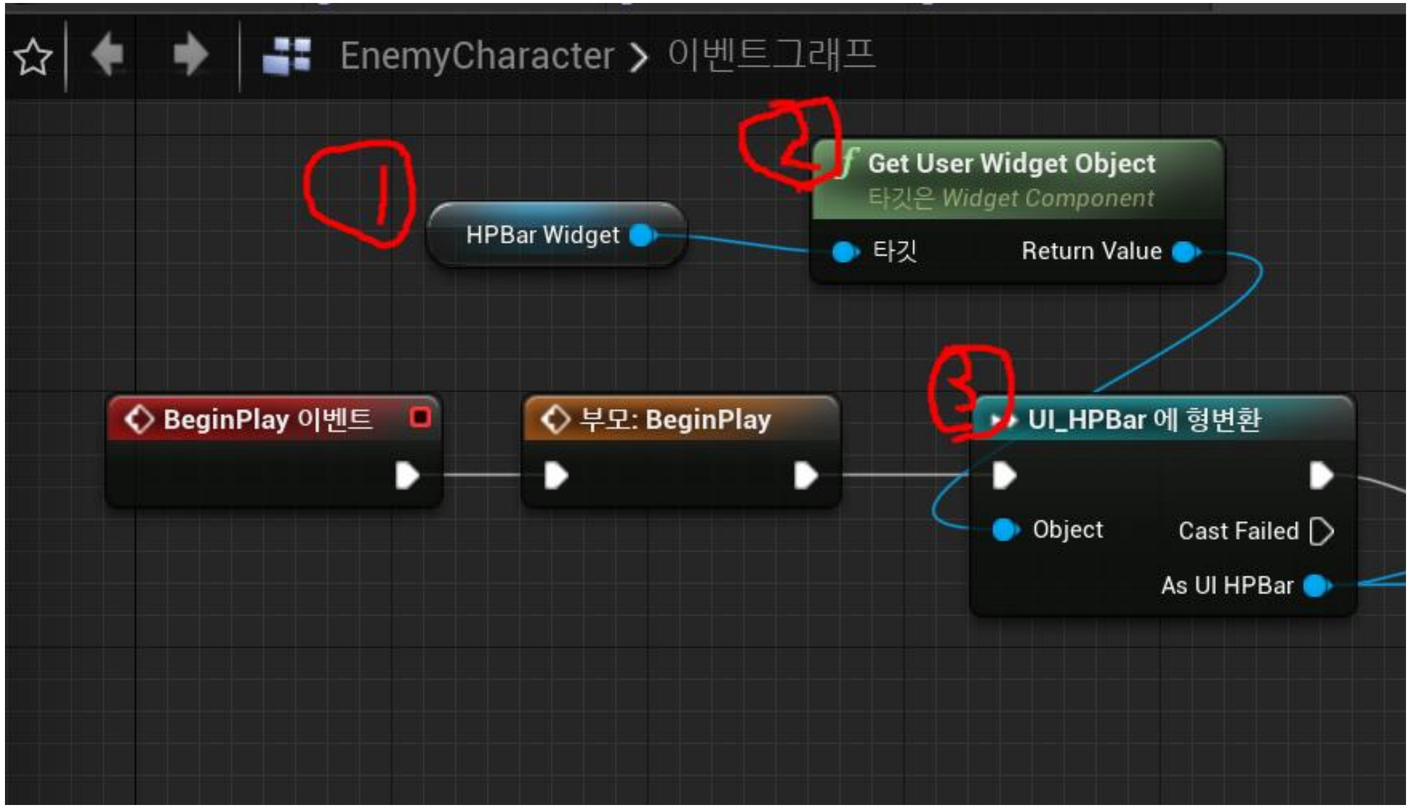
6.2 BindCharacterStat 함수에 NewCharacterStat 입력을 넣는다.



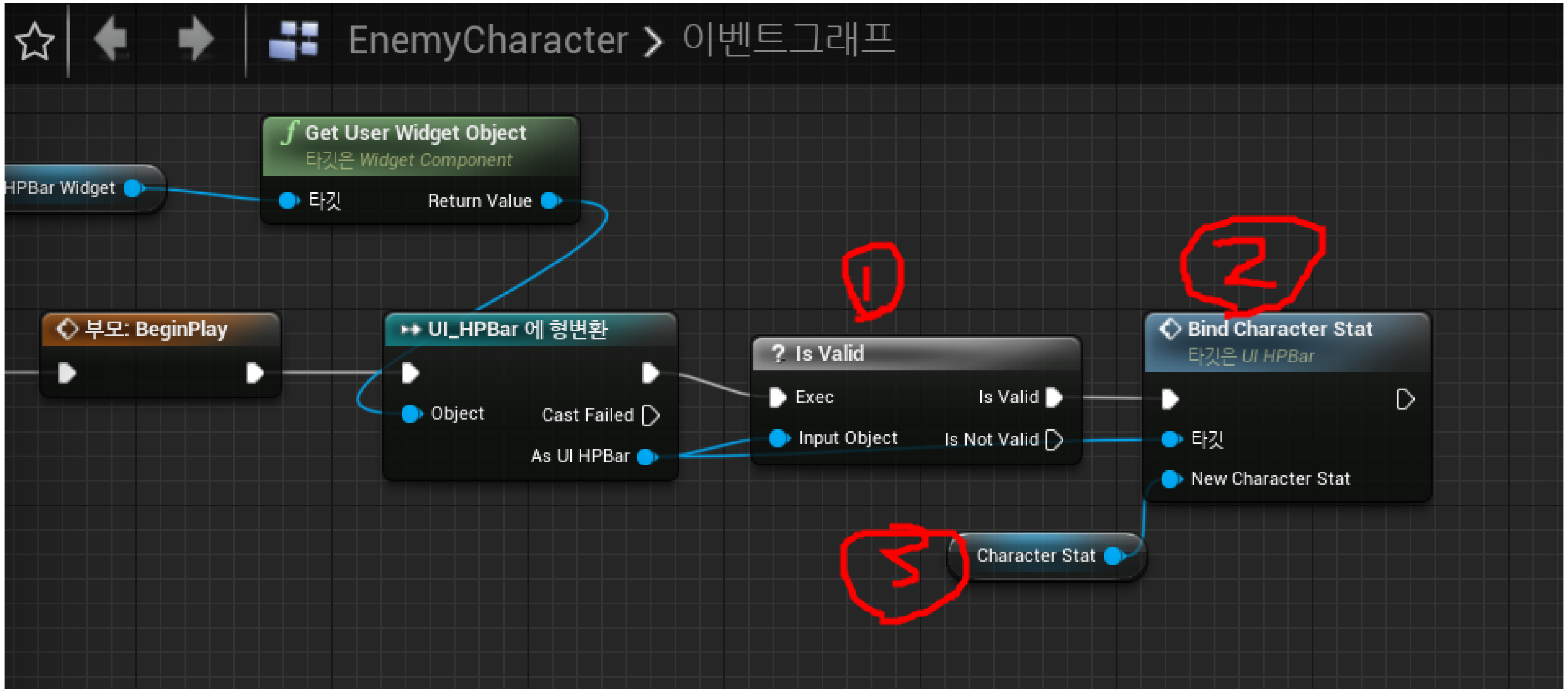
6.3 CurrentCharacterStat에 NewCharacterStat을 연결한고, OnHPChanged를 바인딩하고, UpdateHPWidget함수를 연결한다.



6.4 UpdateHPWidget 함수에 HP의 Percent를 바꾸는 기능을 구현한다.



7. EnemyCharacter에 들어가서 HPBarWidget을 UI_HPBar로 캐스팅한다.



7.1 UI_HPBar가 유효하면 UI_HPBar의 BindCharacterStat 함수를 호출하고, CharacterStat을 NewCharacterStat에 연결한다.