UE4 Utilities

7th Week, 2021

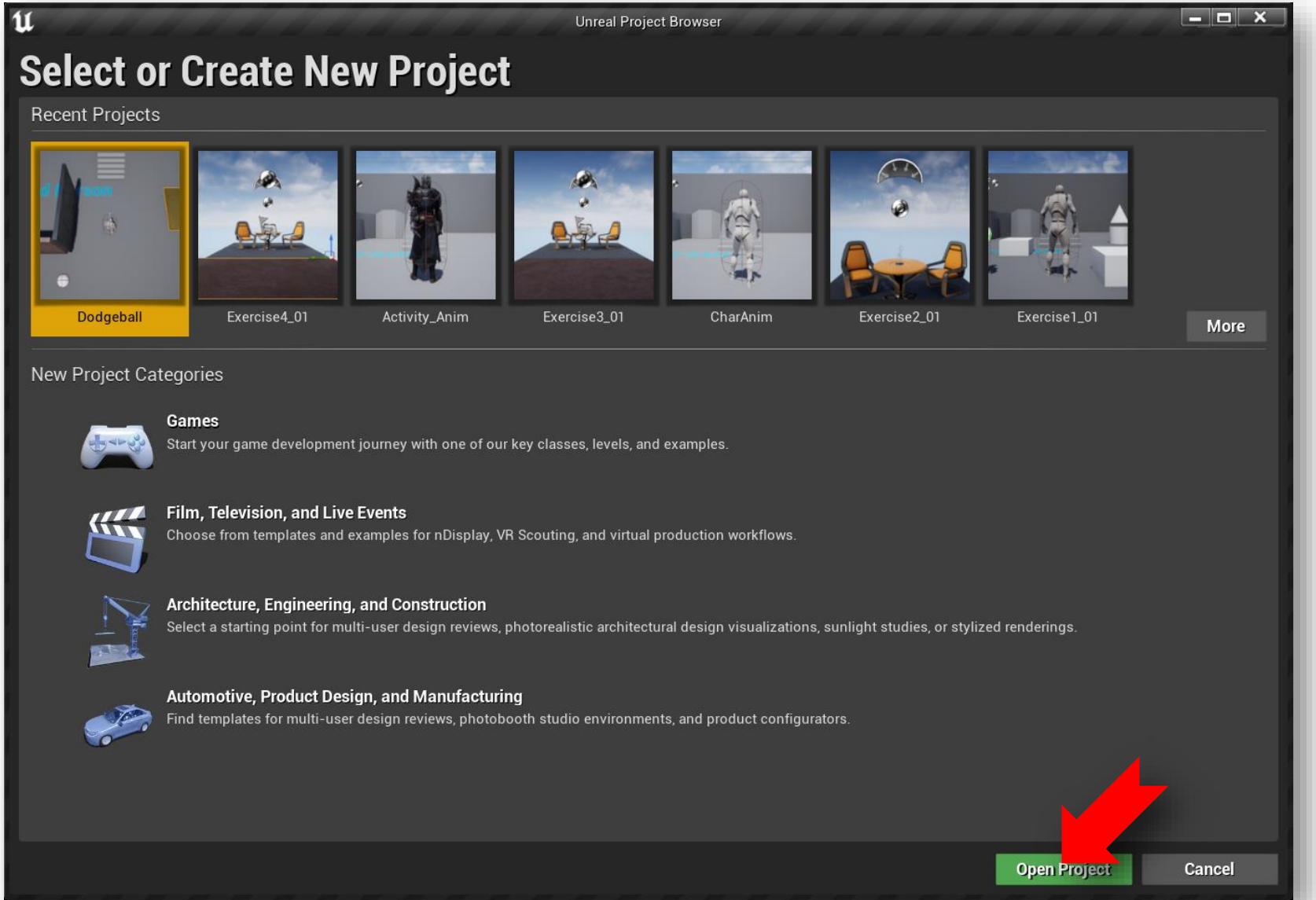
UE4 Utilities

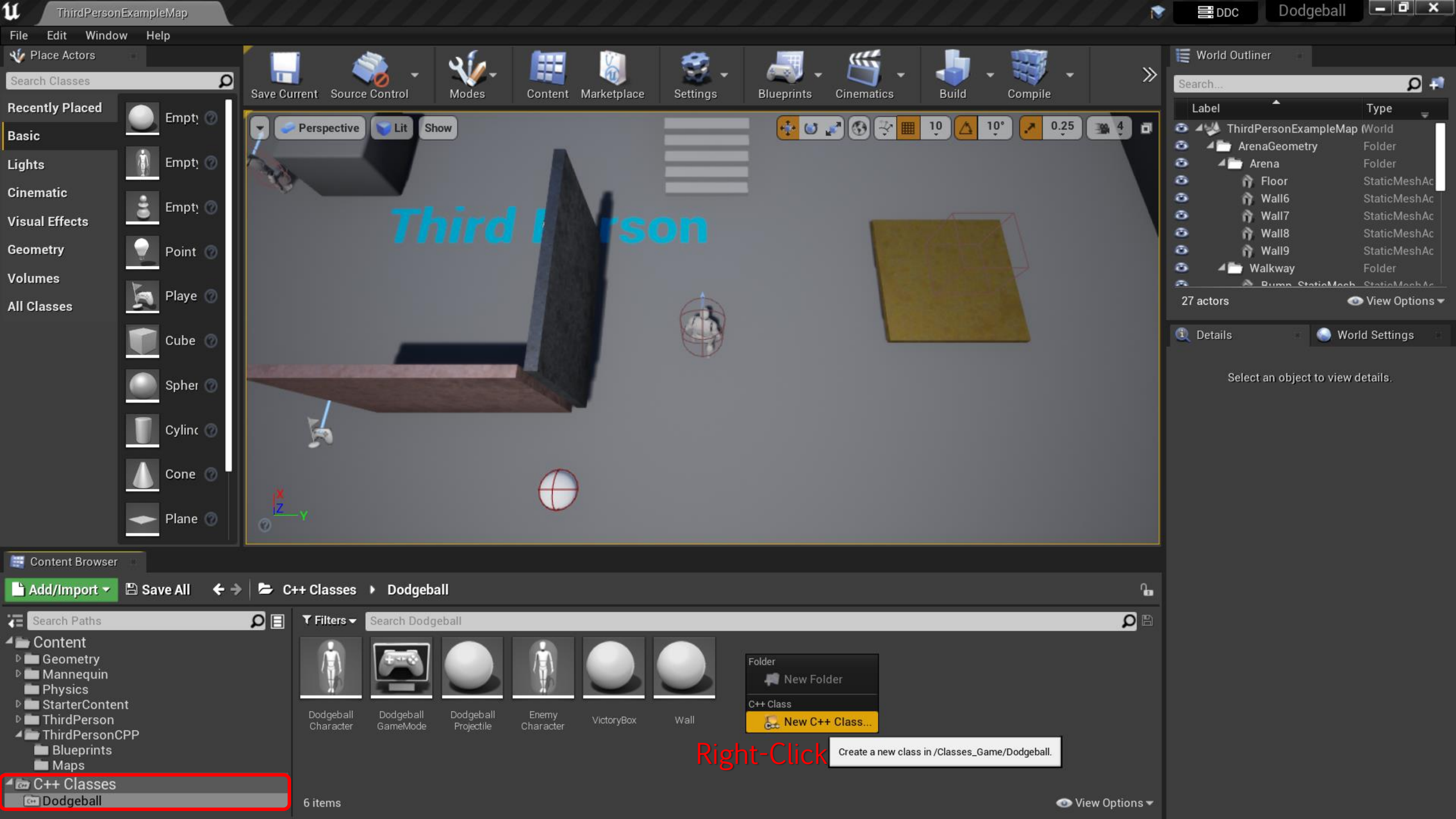
- To easily move logic from one project to another
- To keep your project well structured and organized
- Blueprint Function Libraries
  - To move some generic functions in our project
- Actor Components
  - To move part of some actor classes' source code
  - **Loose Coupling**: a software engineering concept
    - You can easily remove and add things you need.
- Interfaces
  - To make our project better structured and organized

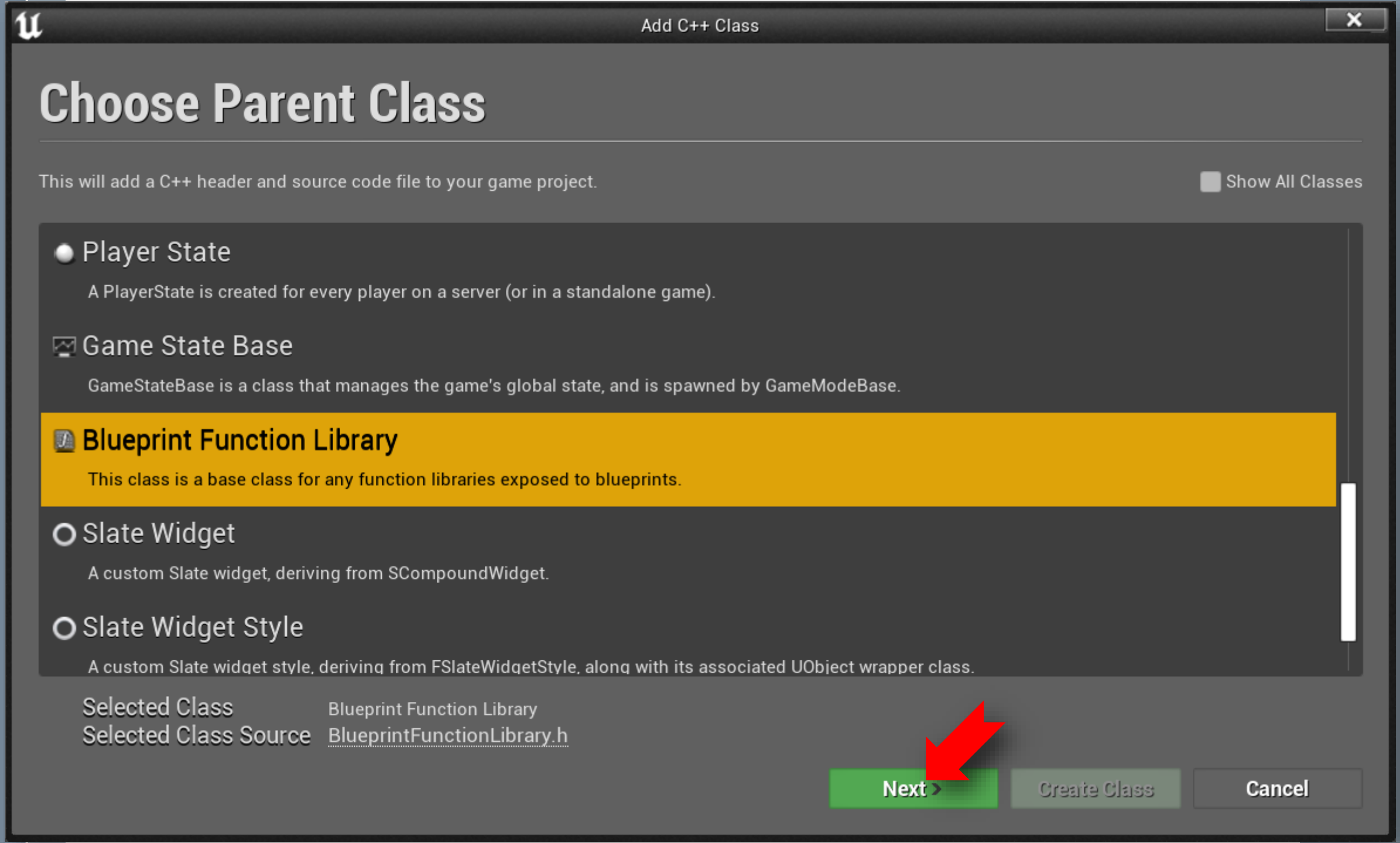
Blueprint Function Libraries

- **BlueprintFunctionLibrary**
  - To contain a collected of static functions that don't really belong to any specific actor and can be used in multiple parts of your project
  - Ex) **KismetMathLibrary** and **KismetSystemLibrary**
- We can move the **CanSeeActor** function from the **EnemyCharacter** class to the **Blueprint Function** library class.

Exercise 7.01: Moving The CanSeeActor Function to The Blueprint Function Library




ThirdPersonExampleMap
DDC
Dodgeball
File
Edit
Window
Help
Place Actors
Search Classes
Recently Placed
Basic
Lights
Cinematic
Visual Effects
Geometry
Volumes
All Classes
Empty
Empty
Empty
Point
Playe
Cube
Spher
Cylinc
Cone
Plane
Save Current
Source Control
Modes
Content
Marketplace
Settings
Blueprints
Cinematics
Build
Compile
Perspective
Lit
Show
10
10°
0.25
4
Third Person
World Outliner
Search...
Label
Type
ThirdPersonExampleMap (World
ArenaGeometry
Folder
Arena
Folder
Floor
StaticMeshAc
Wall6
StaticMeshAc
Wall7
StaticMeshAc
Wall8
StaticMeshAc
Wall9
StaticMeshAc
Walkway
Folder
27 actors
View Options
Details
World Settings
Select an object to view details.
Content Browser
Add/Import
Save All
C++ Classes
Dodgeball
Search Paths
Content
Geometry
Mannequin
Physics
StarterContent
ThirdPerson
ThirdPersonCPP
Blueprints
Maps
C++ Classes
Dodgeball
Filters
Search Dodgeball
Dodgeball Character
Dodgeball GameMode
Dodgeball Projectile
Enemy Character
VictoryBox
Wall
Folder
New Folder
C++ Class
New C++ Class...
Right-Click
Create a new class in /Classes_Game/Dodgeball.
6 items
View Options


Add C++ Class
Choose Parent Class
This will add a C++ header and source code file to your game project.
Show All Classes
Player State
A PlayerState is created for every player on a server (or in a standalone game).
Game State Base
GameStateBase is a class that manages the game's global state, and is spawned by GameModeBase.
Blueprint Function Library
This class is a base class for any function libraries exposed to blueprints.
Slate Widget
A custom Slate widget, deriving from SCompoundWidget.
Slate Widget Style
A custom Slate widget style, deriving from FSlateWidgetStyle, along with its associated UObject wrapper class.
Selected Class
Blueprint Function Library
Selected Class Source
BlueprintFunctionLibrary.h
Next >
Create Class
Cancel

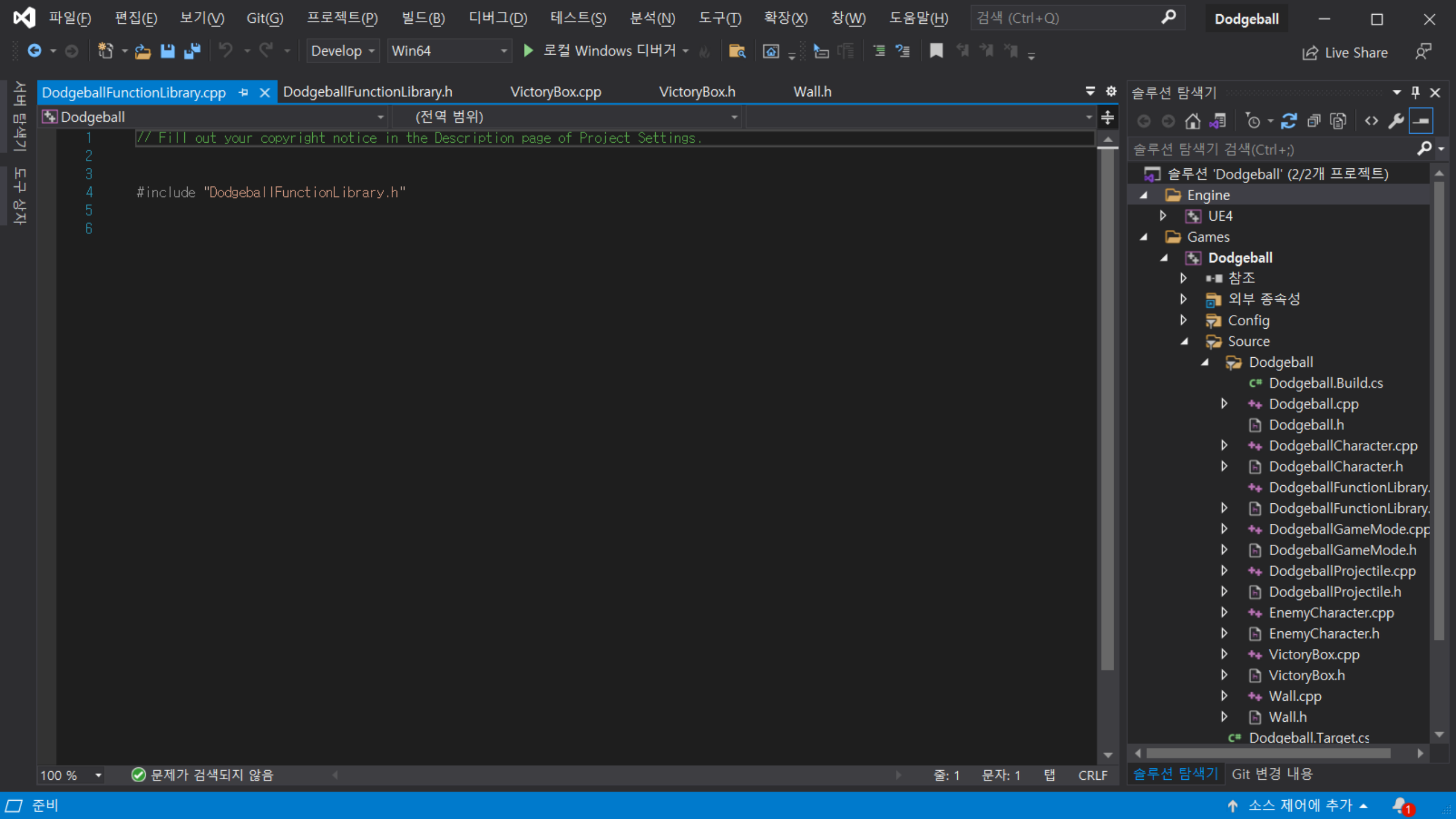
Add C++ Class

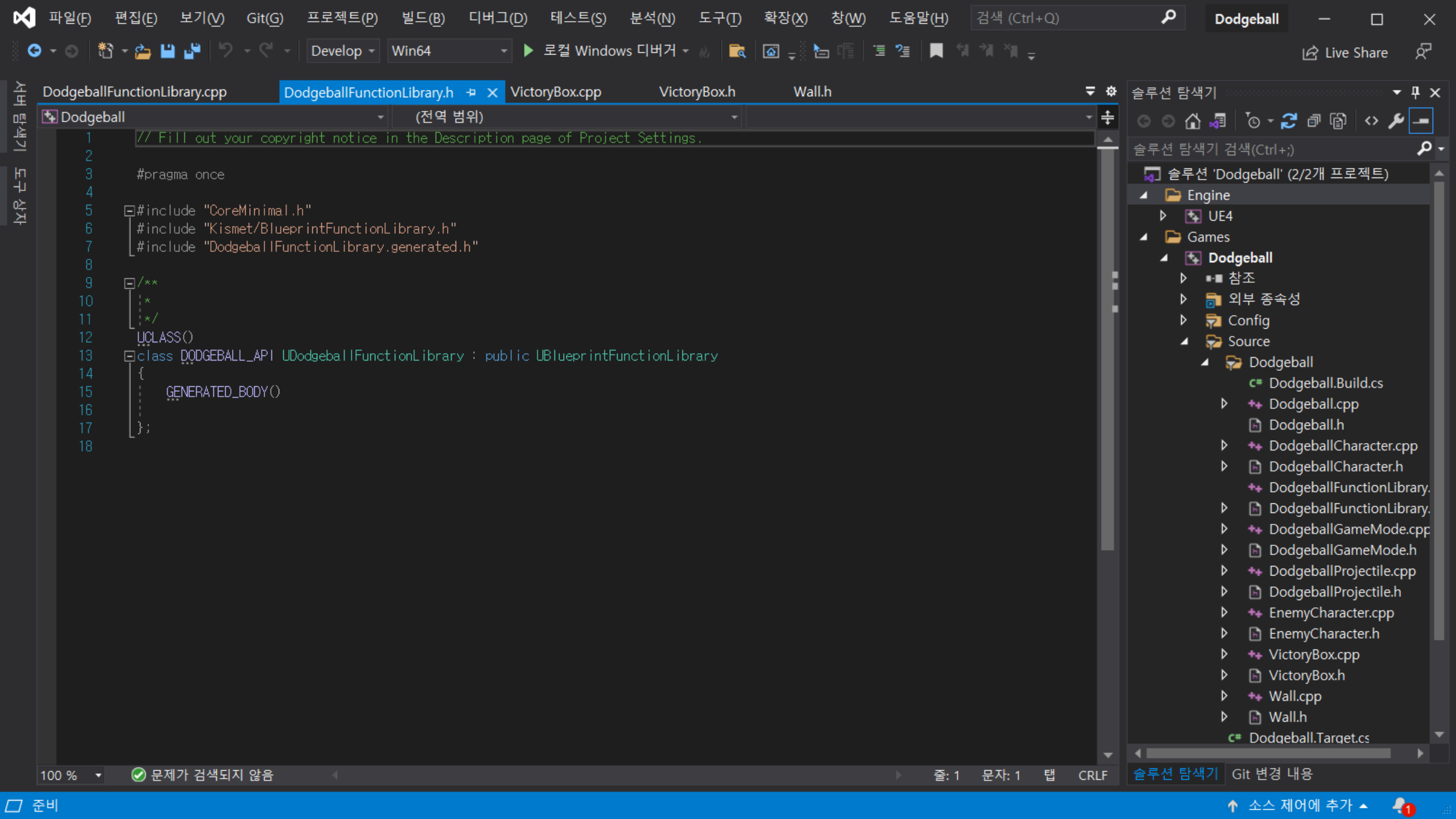
Name Your New Blueprint Function Library

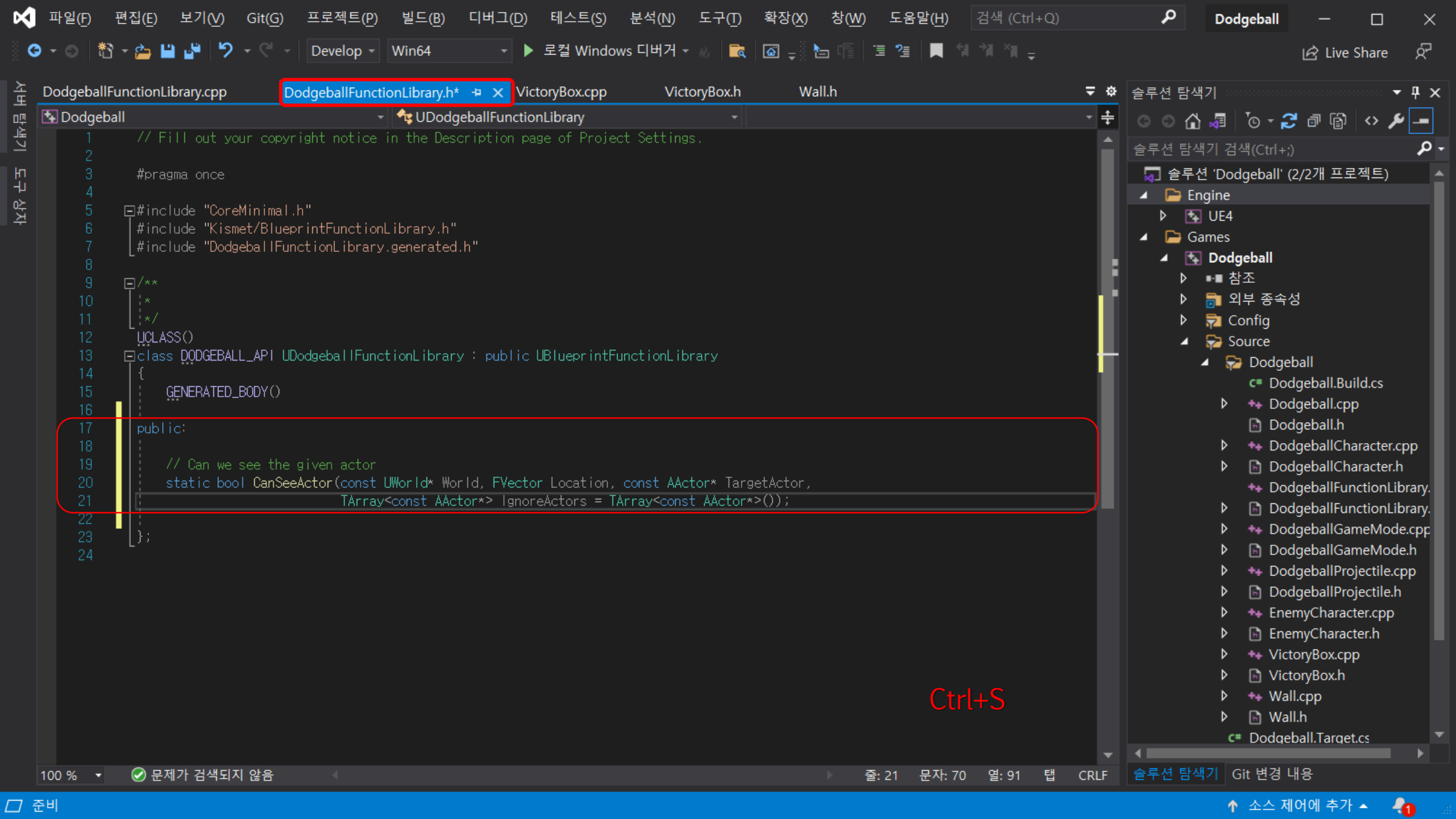
Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.
When you click the "Create" button below, a header (.h) file and a source (.cpp) file will be made using this name.

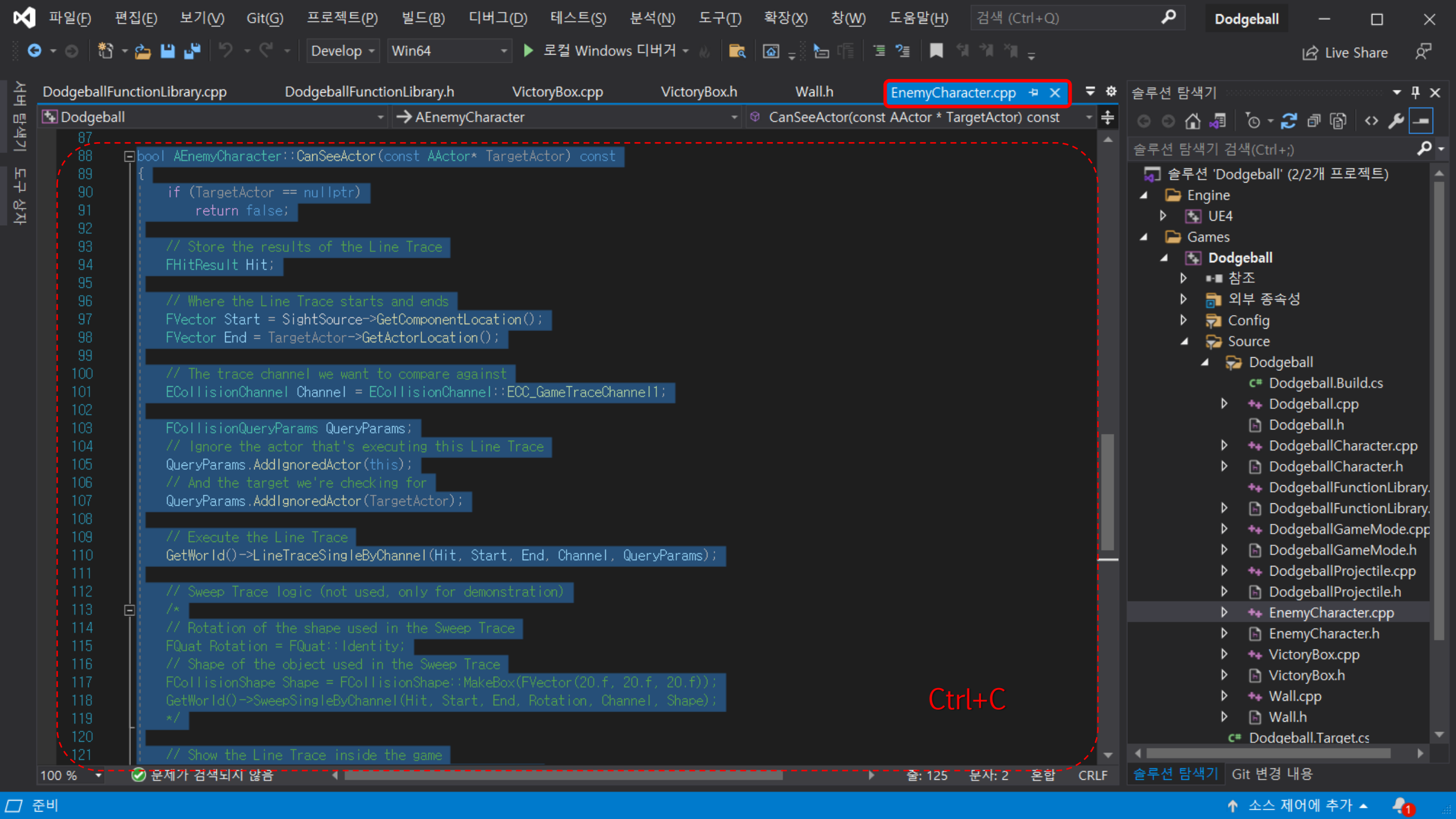
Name	DodgeballFunctionLibrary	Dodgeball (Runtime)	Public / Private
Path	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/		Choose Folder
Header File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/DodgeballFunctionLibrary.h		
Source File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/DodgeballFunctionLibrary.cpp		

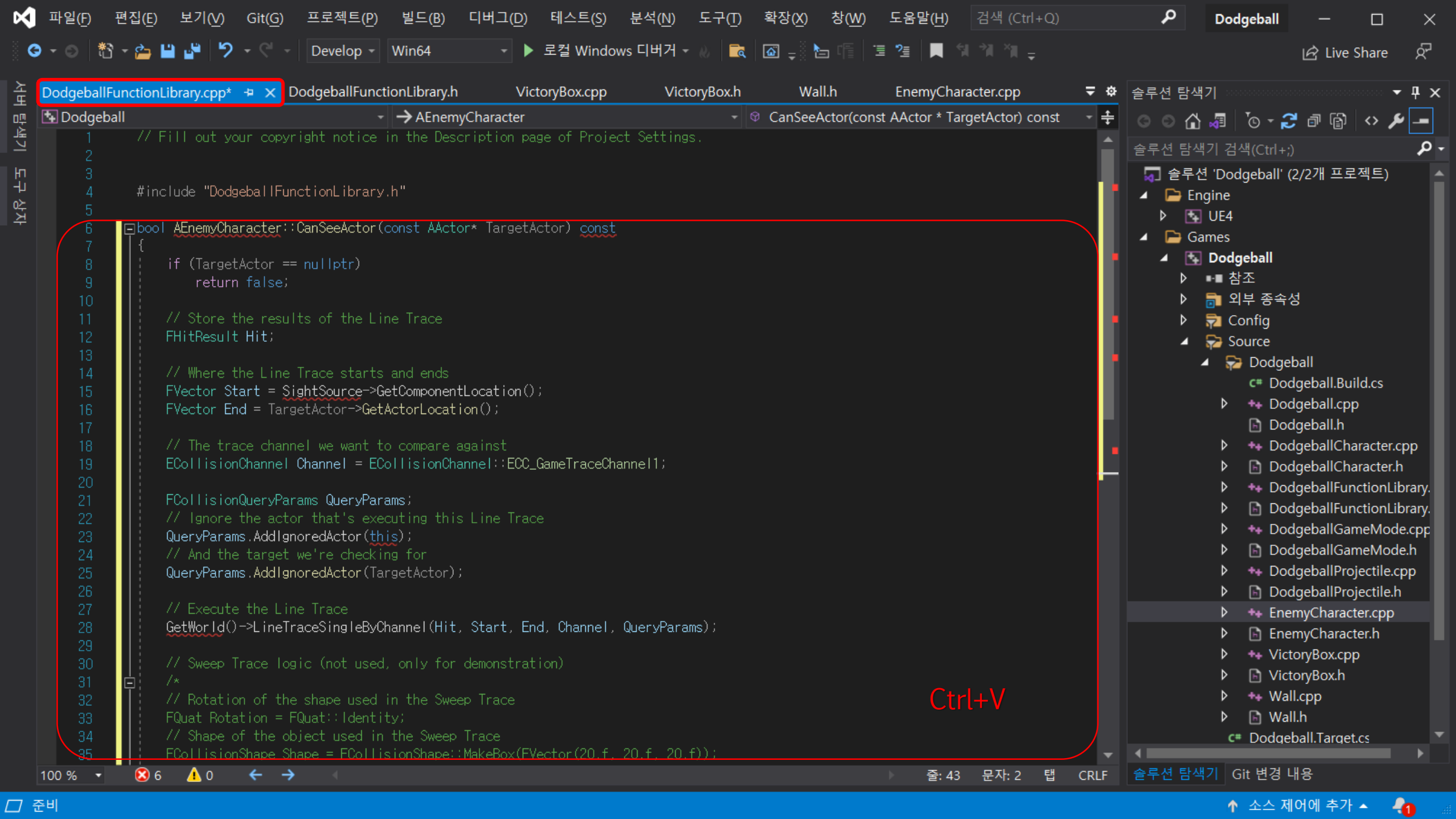
Back | Create Class | Cancel

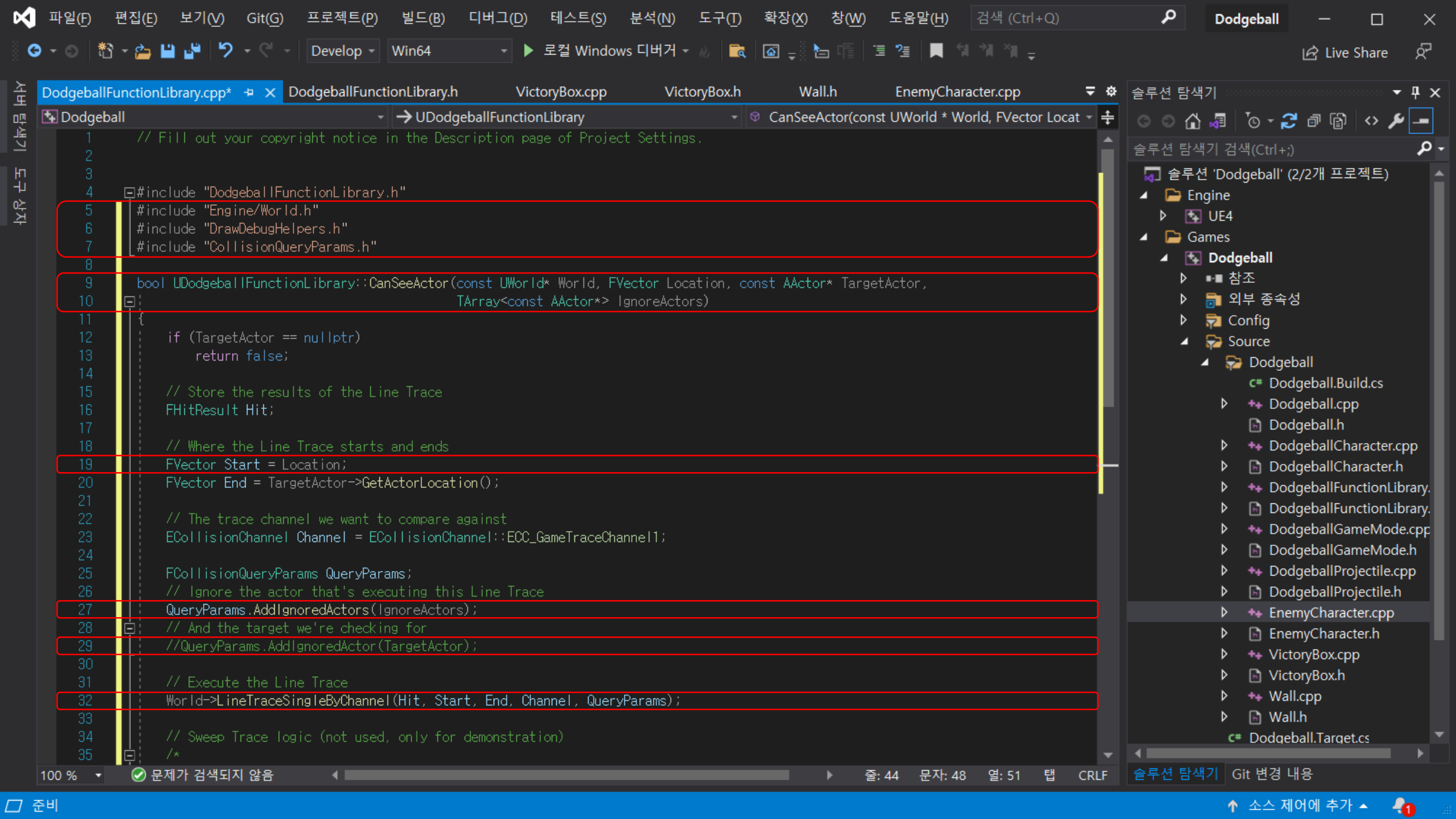

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
서버 탐색기
도구 상자
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
VictoryBox.cpp
VictoryBox.h
Wall.h
Dodgeball
(전역 범위)
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "DodgeballFunctionLibrary.h"
5
6
100 %
문제가 검색되지 않음
줄: 1
문자: 1
탭
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

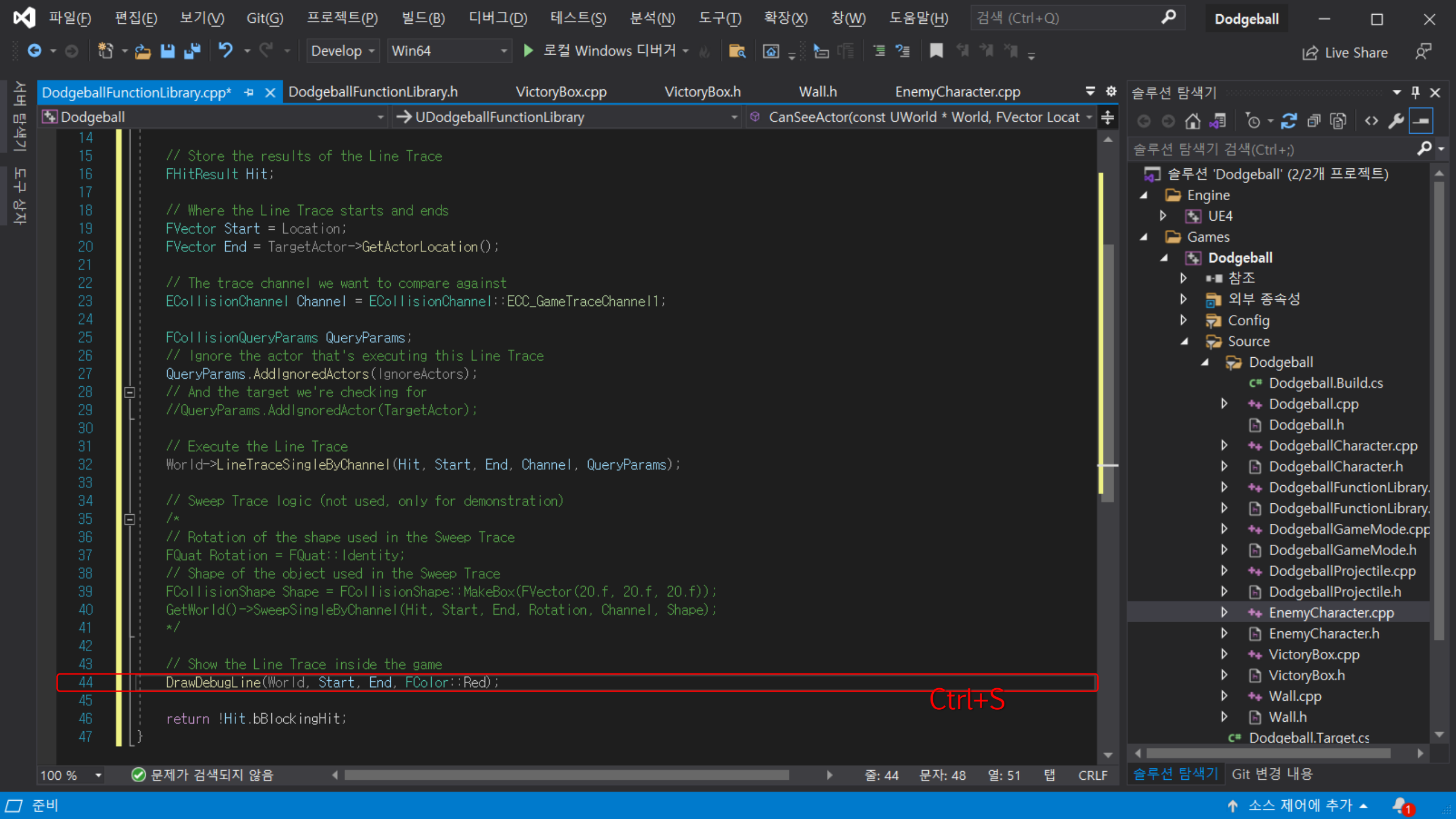

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
서버 탐색기
도구 상자
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
VictoryBox.cpp
VictoryBox.h
Wall.h
Dodgeball
(전역 범위)
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "Kismet/BlueprintFunctionLibrary.h"
7 #include "DodgeballFunctionLibrary.generated.h"
8
9 /**
10 *
11 */
12 UCLASS()
13 class DODGEBALL_API UDodgeballFunctionLibrary : public UBlueprintFunctionLibrary
14 {
15 GENERATED_BODY()
16
17 };
18
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
100 %
문제가 검색되지 않음
줄: 1
문자: 1
탭
CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

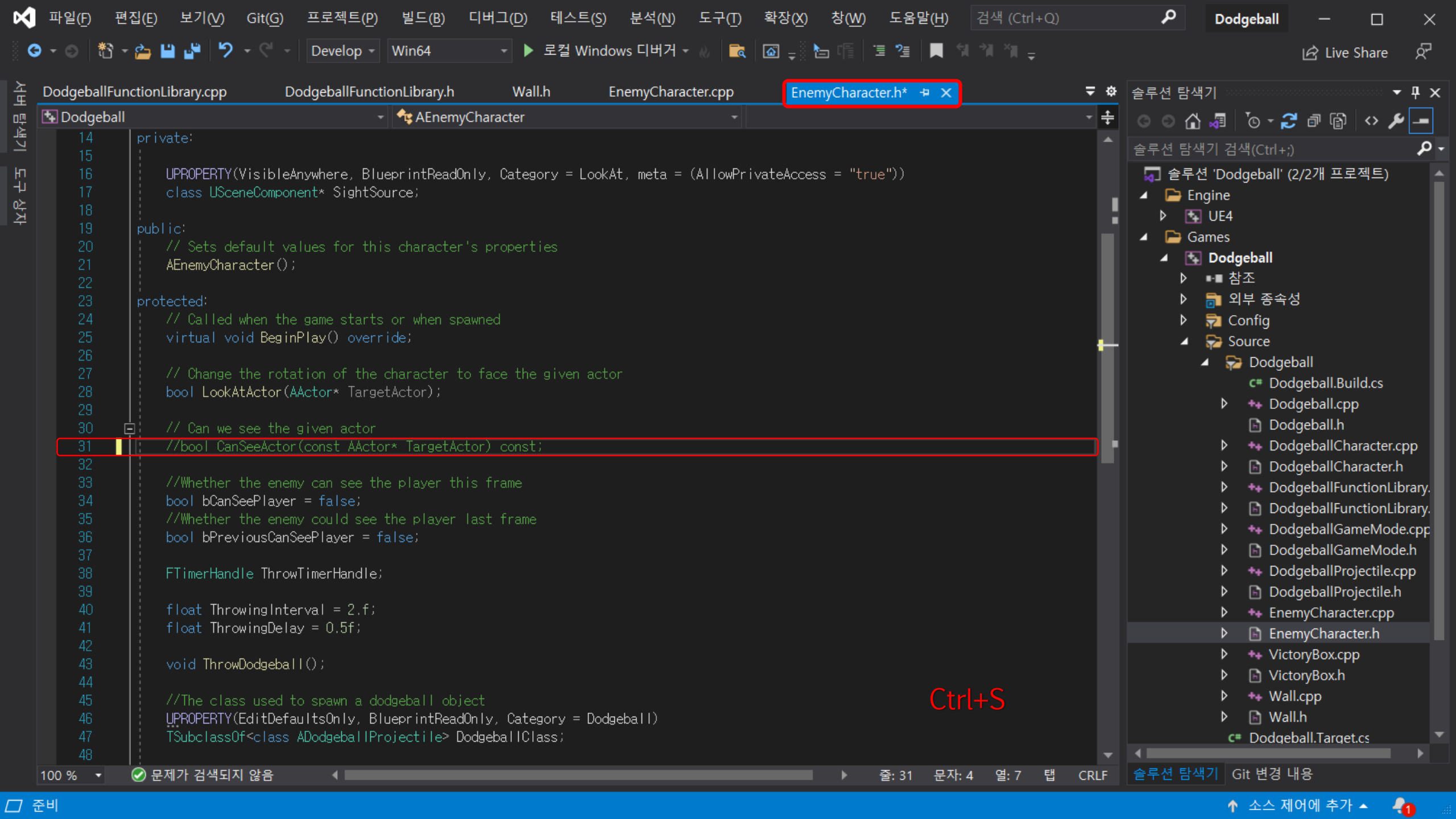

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h*
VictoryBox.cpp
VictoryBox.h
Wall.h
Dodgeball
UDodgeballFunctionLibrary
// Fill out your copyright notice in the Description page of Project Settings.
#pragma once
#include "CoreMinimal.h"
#include "Kismet/BlueprintFunctionLibrary.h"
#include "DodgeballFunctionLibrary.generated.h"
/**
*
*/
UCLASS()
class DODGEBALL_API UDodgeballFunctionLibrary : public UBlueprintFunctionLibrary
{
GENERATED_BODY()
public:
// Can we see the given actor
static bool CanSeeActor(const UWorld* World, FVector Location, const AActor* TargetActor,
TArray<const AActor*> IgnoreActors = TArray<const AActor*>());
};
Ctrl+S
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
100 %
문제가 검색되지 않음
줄: 21
문자: 70
열: 91
탭
CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

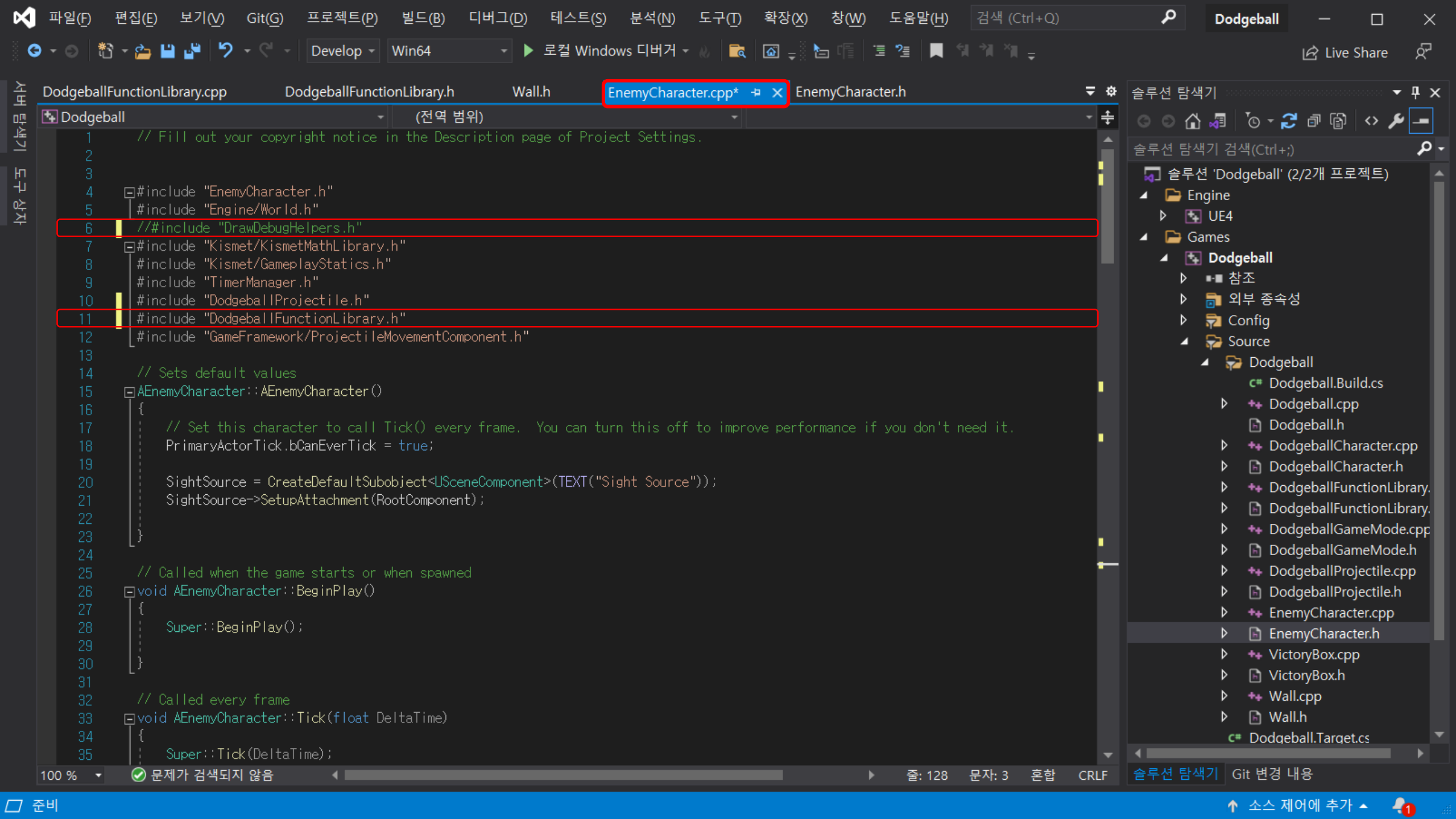

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
VictoryBox.cpp
VictoryBox.h
Wall.h
EnemyCharacter.cpp
Dodgeball
AEnemyCharacter
CanSeeActor(const AActor * TargetActor) const
bool AEnemyCharacter::CanSeeActor(const AActor* TargetActor) const
{
if (TargetActor == nullptr)
return false;
// Store the results of the Line Trace
FHitResult Hit;
// Where the Line Trace starts and ends
FVector Start = SightSource->GetComponentLocation();
FVector End = TargetActor->GetActorLocation();
// The trace channel we want to compare against
ECollisionChannel Channel = ECollisionChannel::ECC_GameTraceChannel1;
FCollisionQueryParams QueryParams;
// Ignore the actor that's executing this Line Trace
QueryParams.AddIgnoredActor(this);
// And the target we're checking for
QueryParams.AddIgnoredActor(TargetActor);
// Execute the Line Trace
GetWorld()->LineTraceSingleByChannel(Hit, Start, End, Channel, QueryParams);
// Sweep Trace logic (not used, only for demonstration)
/*
// Rotation of the shape used in the Sweep Trace
FQuat Rotation = FQuat::Identity;
// Shape of the object used in the Sweep Trace
FCollisionShape Shape = FCollisionShape::MakeBox(FVector(20.f, 20.f, 20.f));
GetWorld()->SweepSingleByChannel(Hit, Start, End, Rotation, Channel, Shape);
*/
// Show the Line Trace inside the game
Ctrl+C
100 %
문제가 검색되지 않음
줄: 125 문자: 2 혼합 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가

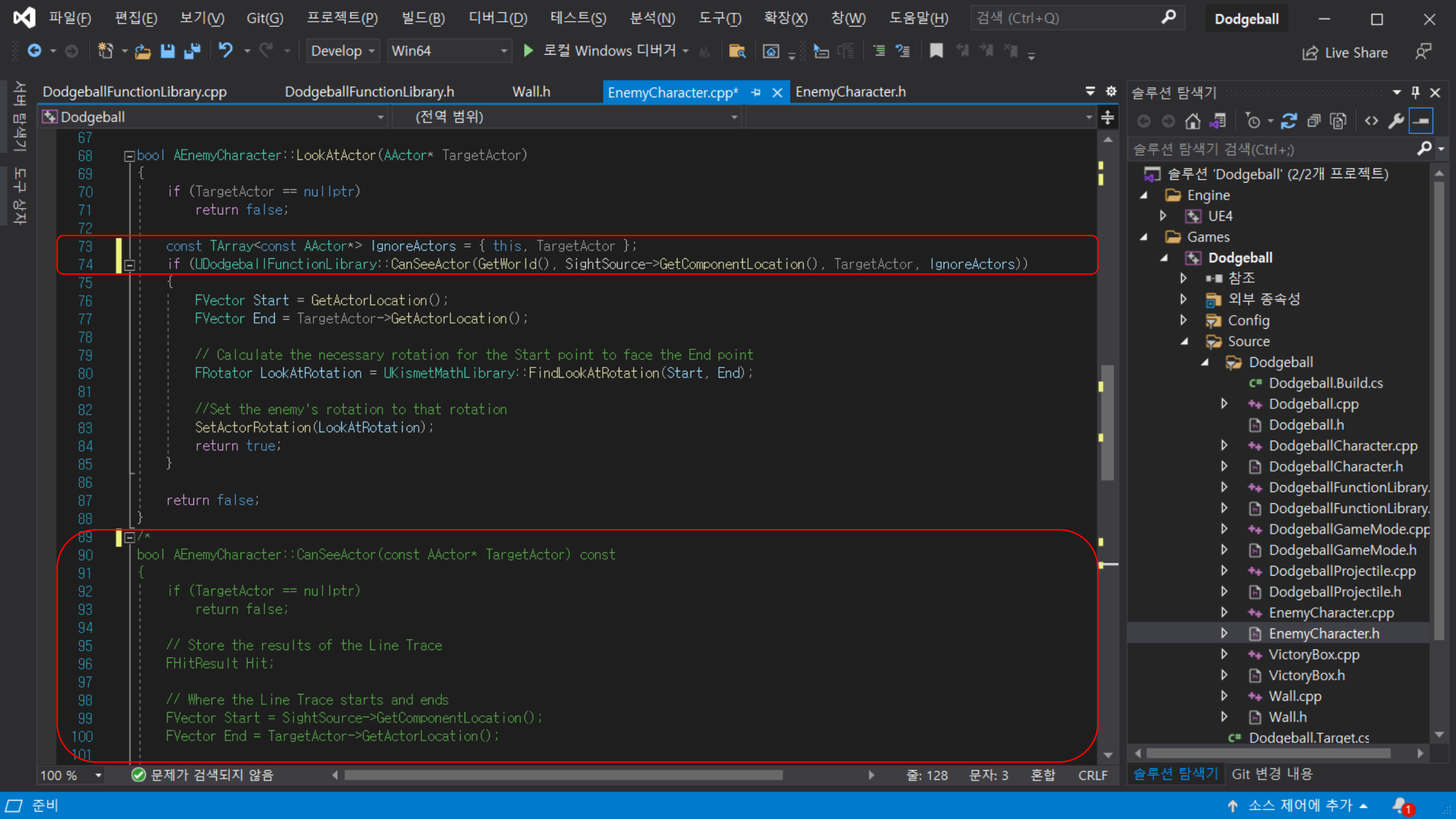

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
DodgeballFunctionLibrary.cpp*
DodgeballFunctionLibrary.h
VictoryBox.cpp
VictoryBox.h
Wall.h
EnemyCharacter.cpp
Dodgeball
AEnemyCharacter
CanSeeActor(const AActor * TargetActor) const
// Fill out your copyright notice in the Description page of Project Settings.
#include "DodgeballFunctionLibrary.h"
bool AEnemyCharacter::CanSeeActor(const AActor* TargetActor) const
{
if (TargetActor == nullptr)
return false;
// Store the results of the Line Trace
FHitResult Hit;
// Where the Line Trace starts and ends
FVector Start = SightSource->GetComponentLocation();
FVector End = TargetActor->GetActorLocation();
// The trace channel we want to compare against
ECollisionChannel Channel = ECollisionChannel::ECC_GameTraceChannel1;
FCollisionQueryParams QueryParams;
// Ignore the actor that's executing this Line Trace
QueryParams.AddIgnoredActor(this);
// And the target we're checking for
QueryParams.AddIgnoredActor(TargetActor);
// Execute the Line Trace
GetWorld()->LineTraceSingleByChannel(Hit, Start, End, Channel, QueryParams);
// Sweep Trace logic (not used, only for demonstration)
/*
// Rotation of the shape used in the Sweep Trace
FQuat Rotation = FQuat::Identity;
// Shape of the object used in the Sweep Trace
FCollisionShape Shape = FCollisionShape::MakeBox(FVector(20.f, 20.f, 20.f));
Ctrl+V
100 %
6
0
줄: 43
문자: 2
탭
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

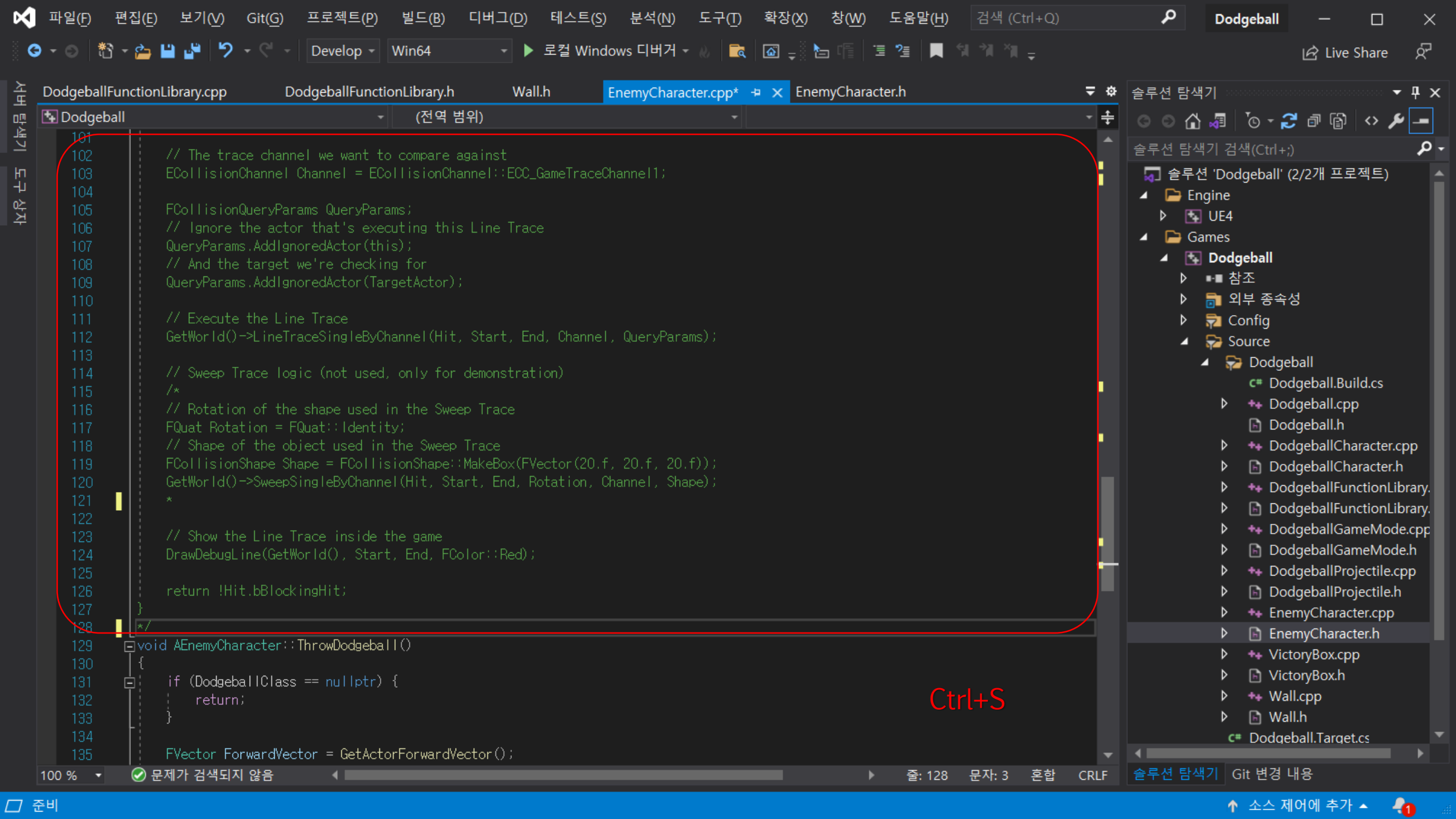

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
서버 탐색기
도구 상자
DodgeballFunctionLibrary.cpp*
DodgeballFunctionLibrary.h
VictoryBox.cpp
VictoryBox.h
Wall.h
EnemyCharacter.cpp
Dodgeball
UDodgeballFunctionLibrary
CanSeeActor(const UWorld * World, FVector Locat
// Fill out your copyright notice in the Description page of Project Settings.
#include "DodgeballFunctionLibrary.h"
#include "Engine/World.h"
#include "DrawDebugHelpers.h"
#include "CollisionQueryParams.h"
bool UDodgeballFunctionLibrary::CanSeeActor(const UWorld* World, FVector Location, const AActor* TargetActor,
TArray<const AActor*> IgnoreActors)
{
if (TargetActor == nullptr)
return false;
// Store the results of the Line Trace
FHitResult Hit;
// Where the Line Trace starts and ends
FVector Start = Location;
FVector End = TargetActor->GetActorLocation();
// The trace channel we want to compare against
ECollisionChannel Channel = ECollisionChannel::ECC_GameTraceChannel1;
FCollisionQueryParams QueryParams;
// Ignore the actor that's executing this Line Trace
QueryParams.AddIgnoredActors(IgnoreActors);
// And the target we're checking for
//QueryParams.AddIgnoredActor(TargetActor);
// Execute the Line Trace
World->LineTraceSingleByChannel(Hit, Start, End, Channel, QueryParams);
// Sweep Trace logic (not used, only for demonstration)
/*
100 %
문제가 검색되지 않음
줄: 44 문자: 48 열: 51 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

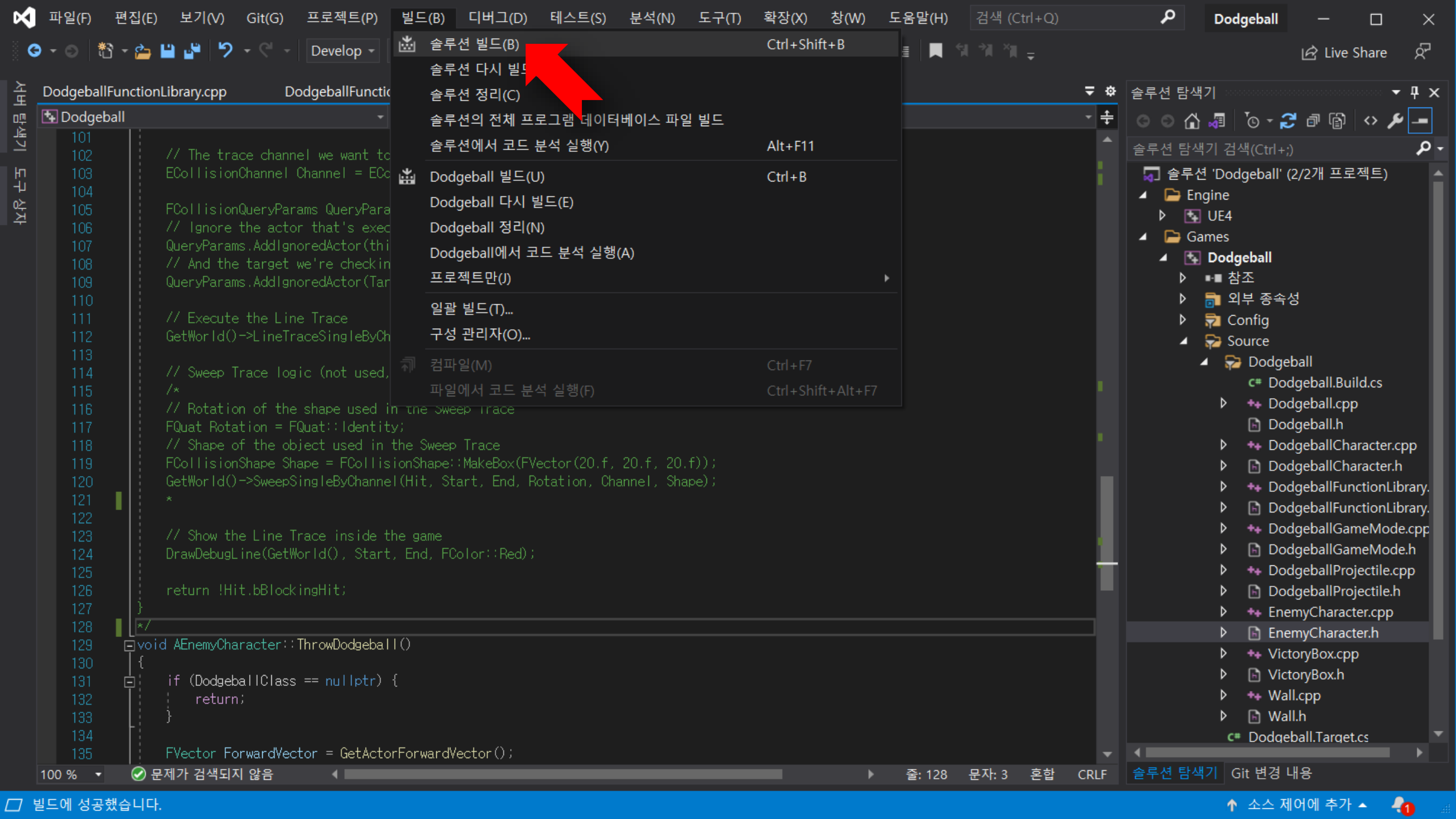

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
DodgeballFunctionLibrary.cpp*
DodgeballFunctionLibrary.h
VictoryBox.cpp
VictoryBox.h
Wall.h
EnemyCharacter.cpp
Dodgeball
UDodgeballFunctionLibrary
CanSeeActor(const UWorld * World, FVector Locat
// Store the results of the Line Trace
FHitResult Hit;
// Where the Line Trace starts and ends
FVector Start = Location;
FVector End = TargetActor->GetActorLocation();
// The trace channel we want to compare against
ECollisionChannel Channel = ECollisionChannel::ECC_GameTraceChannel1;
FCollisionQueryParams QueryParams;
// Ignore the actor that's executing this Line Trace
QueryParams.AddIgnoredActors(IgnoreActors);
// And the target we're checking for
//QueryParams.AddIgnoredActor(TargetActor);
// Execute the Line Trace
World->LineTraceSingleByChannel(Hit, Start, End, Channel, QueryParams);
// Sweep Trace logic (not used, only for demonstration)
/*
// Rotation of the shape used in the Sweep Trace
FQuat Rotation = FQuat::Identity;
// Shape of the object used in the Sweep Trace
FCollisionShape Shape = FCollisionShape::MakeBox(FVector(20.f, 20.f, 20.f));
GetWorld()->SweepSingleByChannel(Hit, Start, End, Rotation, Channel, Shape);
*/
// Show the Line Trace inside the game
DrawDebugLine(World, Start, End, FColor::Red);
return !Hit.bBlockingHit;
}
Ctrl+S
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
100 %
문제가 검색되지 않음
줄: 44 문자: 48 열: 51 탭 CRLF
솔루션 탐색기 Git 변경 내용
준비
소스 제어에 추가

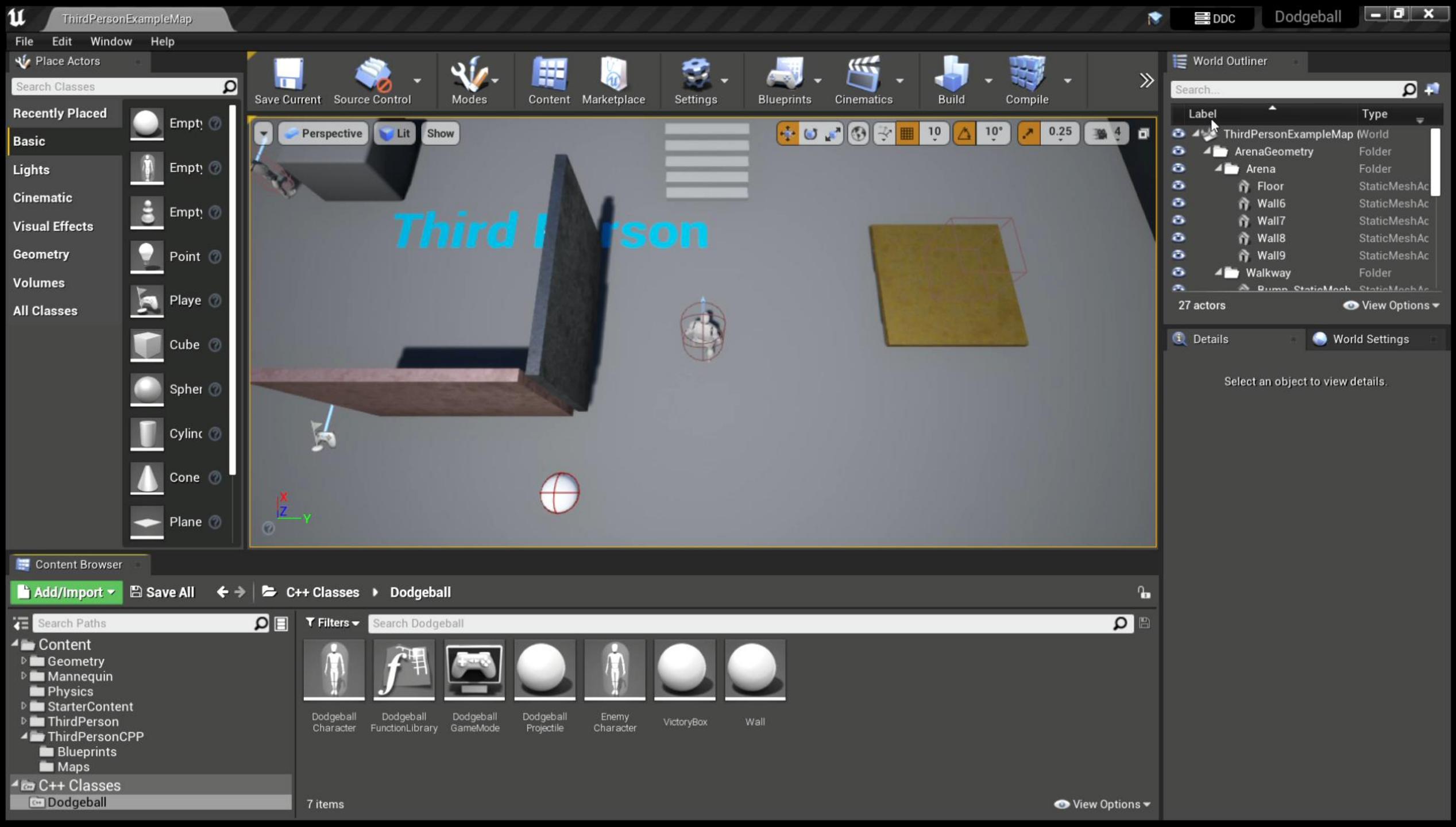

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
Wall.h
EnemyCharacter.cpp
EnemyCharacter.h*
Dodgeball
AEnemyCharacter
14 private:
15
16 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = LookAt, meta = (AllowPrivateAccess = "true"))
17 class USceneComponent* SightSource;
18
19 public:
20 // Sets default values for this character's properties
21 AEnemyCharacter();
22
23 protected:
24 // Called when the game starts or when spawned
25 virtual void BeginPlay() override;
26
27 // Change the rotation of the character to face the given actor
28 bool LookAtActor(AActor* TargetActor);
29
30 // Can we see the given actor
31 //bool CanSeeActor(const AActor* TargetActor) const;
32
33 //Whether the enemy can see the player this frame
34 bool bCanSeePlayer = false;
35 //Whether the enemy could see the player last frame
36 bool bPreviousCanSeePlayer = false;
37
38 FTimerHandle ThrowTimerHandle;
39
40 float ThrowingInterval = 2.f;
41 float ThrowingDelay = 0.5f;
42
43 void ThrowDodgeball();
44
45 //The class used to spawn a dodgeball object
46 UPROPERTY(EditDefaultsOnly, BlueprintReadOnly, Category = Dodgeball)
47 TSubclassOf<class ADodgeballProjectile> DodgeballClass;
48
Ctrl+S
100 %
문제가 검색되지 않음
줄: 31 문자: 4 열: 7 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
서버 탐색기
도구 상자
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
Wall.h
EnemyCharacter.cpp*
EnemyCharacter.h
Dodgeball
(전역 범위)
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "EnemyCharacter.h"
5 #include "Engine/World.h"
6 //#include "DrawDebugHelpers.h"
7 #include "Kismet/KismetMathLibrary.h"
8 #include "Kismet/GameplayStatics.h"
9 #include "TimerManager.h"
10 #include "DodgeballProjectile.h"
11 #include "DodgeballFunctionLibrary.h"
12 #include "GameFramework/ProjectileMovementComponent.h"
13
14 // Sets default values
15 AEnemyCharacter::AEnemyCharacter()
16 {
17 // Set this character to call Tick() every frame. You can turn this off to improve performance if you don't need it.
18 PrimaryActorTick.bCanEverTick = true;
19
20 SightSource = CreateDefaultSubobject<USceneComponent>(TEXT("Sight Source"));
21 SightSource->SetupAttachment(RootComponent);
22
23 }
24
25 // Called when the game starts or when spawned
26 void AEnemyCharacter::BeginPlay()
27 {
28 Super::BeginPlay();
29
30 }
31
32 // Called every frame
33 void AEnemyCharacter::Tick(float DeltaTime)
34 {
35 Super::Tick(DeltaTime);
100 %
문제가 검색되지 않음
줄: 128
문자: 3
혼합
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
Wall.h
EnemyCharacter.cpp*
EnemyCharacter.h
Dodgeball
(전역 범위)
67
68 bool AEnemyCharacter::LookAtActor(AActor* TargetActor)
69 {
70 if (TargetActor == nullptr)
71 return false;
72
73 const TArray<const AActor*> IgnoreActors = { this, TargetActor };
74 if (UDodgeballFunctionLibrary::CanSeeActor(GetWorld(), SightSource->GetComponentLocation(), TargetActor, IgnoreActors))
75 {
76 FVector Start = GetActorLocation();
77 FVector End = TargetActor->GetActorLocation();
78
79 // Calculate the necessary rotation for the Start point to face the End point
80 FRotator LookAtRotation = UKismetMathLibrary::FindLookAtRotation(Start, End);
81
82 //Set the enemy's rotation to that rotation
83 SetActorRotation(LookAtRotation);
84 return true;
85 }
86
87 return false;
88 }
89 /*
90 bool AEnemyCharacter::CanSeeActor(const AActor* TargetActor) const
91 {
92 if (TargetActor == nullptr)
93 return false;
94
95 // Store the results of the Line Trace
96 FHitResult Hit;
97
98 // Where the Line Trace starts and ends
99 FVector Start = SightSource->GetComponentLocation();
100 FVector End = TargetActor->GetActorLocation();
101
100 %
문제가 검색되지 않음
줄: 128 문자: 3 혼합 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
Wall.h
EnemyCharacter.cpp*
EnemyCharacter.h
Dodgeball
(전역 범위)
// The trace channel we want to compare against
ECollisionChannel Channel = ECollisionChannel::ECC_GameTraceChannel1;
FCollisionQueryParams QueryParams;
// Ignore the actor that's executing this Line Trace
QueryParams.AddIgnoredActor(this);
// And the target we're checking for
QueryParams.AddIgnoredActor(TargetActor);
// Execute the Line Trace
GetWorld()->LineTraceSingleByChannel(Hit, Start, End, Channel, QueryParams);
// Sweep Trace logic (not used, only for demonstration)
/*
// Rotation of the shape used in the Sweep Trace
FQuat Rotation = FQuat::Identity;
// Shape of the object used in the Sweep Trace
FCollisionShape Shape = FCollisionShape::MakeBox(FVector(20.f, 20.f, 20.f));
GetWorld()->SweepSingleByChannel(Hit, Start, End, Rotation, Channel, Shape);
*
// Show the Line Trace inside the game
DrawDebugLine(GetWorld(), Start, End, FColor::Red);
return !Hit.bBlockingHit;
}
*/
void AEnemyCharacter::ThrowDodgeball()
{
if (DodgeballClass == nullptr) {
return;
}
FVector ForwardVector = GetActorForwardVector();
Ctrl+S
100 %
문제가 검색되지 않음
줄: 128
문자: 3
혼합
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Live Share
솔루션 빌드(B) Ctrl+Shift+B
솔루션 다시 빌드
솔루션 정리(C)
솔루션의 전체 프로그램 데이터베이스 파일 빌드
솔루션에서 코드 분석 실행(Y) Alt+F11
Dodgeball 빌드(U) Ctrl+B
Dodgeball 다시 빌드(E)
Dodgeball 정리(N)
Dodgeball에서 코드 분석 실행(A)
프로젝트만(J)
일괄 빌드(T)...
구성 관리자(O)...
컴파일(M) Ctrl+F7
파일에서 코드 분석 실행(F) Ctrl+Shift+Alt+F7
DodgeballFunctionLibrary.cpp
Dodgeball
// The trace channel we want to
ECollisionChannel Channel = ECc
FCollisionQueryParams QueryPara
// Ignore the actor that's exec
QueryParams.AddIgnoredActor(thi
// And the target we're checkin
QueryParams.AddIgnoredActor(Tar
// Execute the Line Trace
GetWorld()->LineTraceSingleByCh
// Sweep Trace logic (not used,
/*
// Rotation of the shape used in the Sweep Trace
FQuat Rotation = FQuat::Identity;
// Shape of the object used in the Sweep Trace
FCollisionShape Shape = FCollisionShape::MakeBox(FVector(20.f, 20.f, 20.f));
GetWorld()->SweepSingleByChannel(Hit, Start, End, Rotation, Channel, Shape);
*
// Show the Line Trace inside the game
DrawDebugLine(GetWorld(), Start, End, FColor::Red);
return !Hit.bBlockingHit;
}
*/
void AEnemyCharacter::ThrowDodgeball()
{
if (DodgeballClass == nullptr) {
return;
}
FVector ForwardVector = GetActorForwardVector();
100 %
문제가 검색되지 않음
줄: 128 문자: 3 혼합 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
Wall.h
Dodgeball.Target.cs
솔루션 탐색기
Git 변경 내용
빌드에 성공했습니다.
소스 제어에 추가


ThirdPersonExampleMap
DDC
Dodgeball
File
Edit
Window
Help
Place Actors
Search Classes
Recently Placed
Basic
Lights
Cinematic
Visual Effects
Geometry
Volumes
All Classes
Empty
Empty
Empty
Point
Playe
Cube
Spher
Cylinc
Cone
Plane
Save Current
Source Control
Modes
Content
Marketplace
Settings
Blueprints
Cinematics
Build
Compile
Perspective
Lit
Show
10
10°
0.25
4
Third Person
World Outliner
Search...
Label
Type
ThirdPersonExampleMap (World
ArenaGeometry
Folder
Arena
Folder
Floor
StaticMeshAc
Wall6
StaticMeshAc
Wall7
StaticMeshAc
Wall8
StaticMeshAc
Wall9
StaticMeshAc
Walkway
Folder
27 actors
View Options
Details
World Settings
Select an object to view details.
Content Browser
Add/Import
Save All
C++ Classes
Dodgeball
Search Paths
Content
Geometry
Mannequin
Physics
StarterContent
ThirdPerson
ThirdPersonCPP
Blueprints
Maps
C++ Classes
Dodgeball
Filters
Search Dodgeball
Dodgeball Character
Dodgeball FunctionLibrary
Dodgeball GameMode
Dodgeball Projectile
Enemy Character
VictoryBox
Wall
7 items
View Options

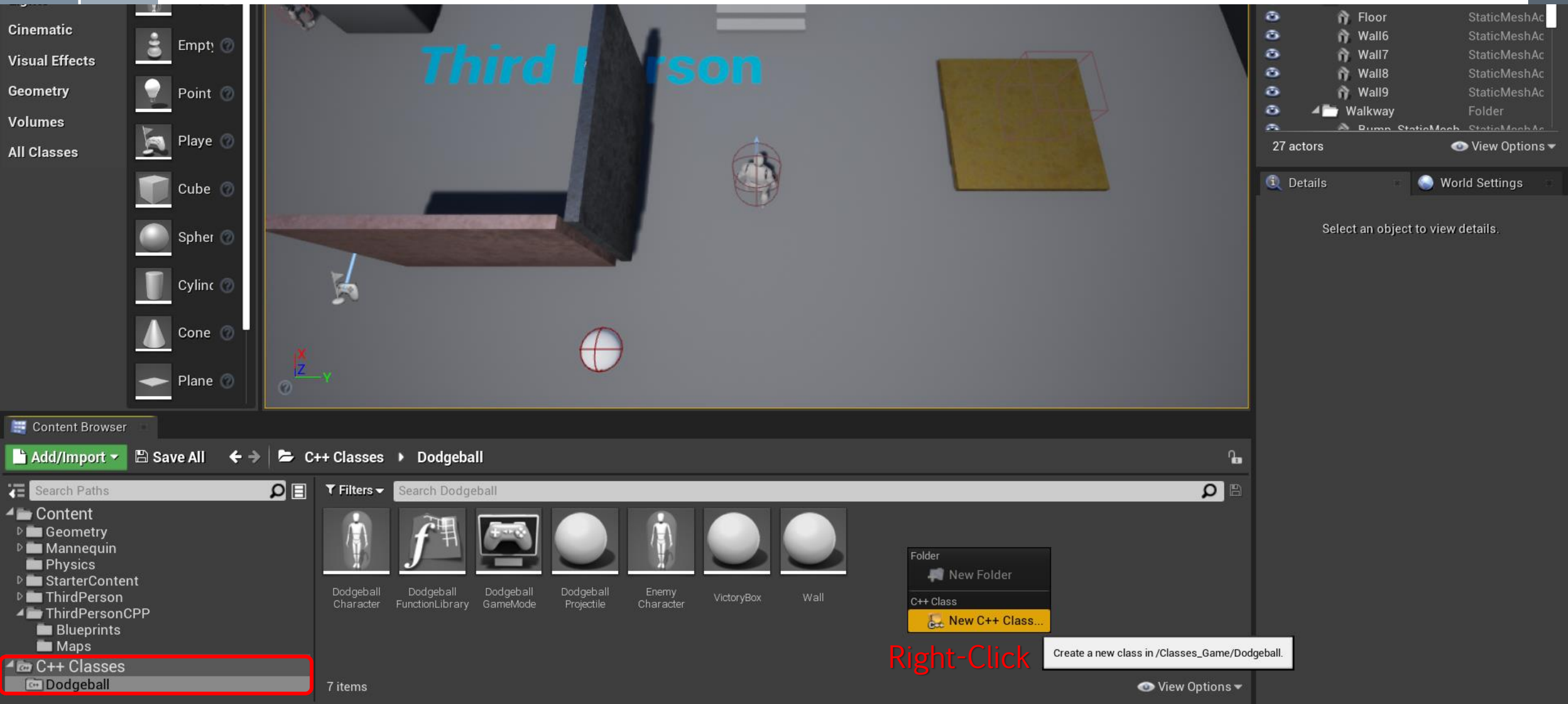
Actor Components (1)

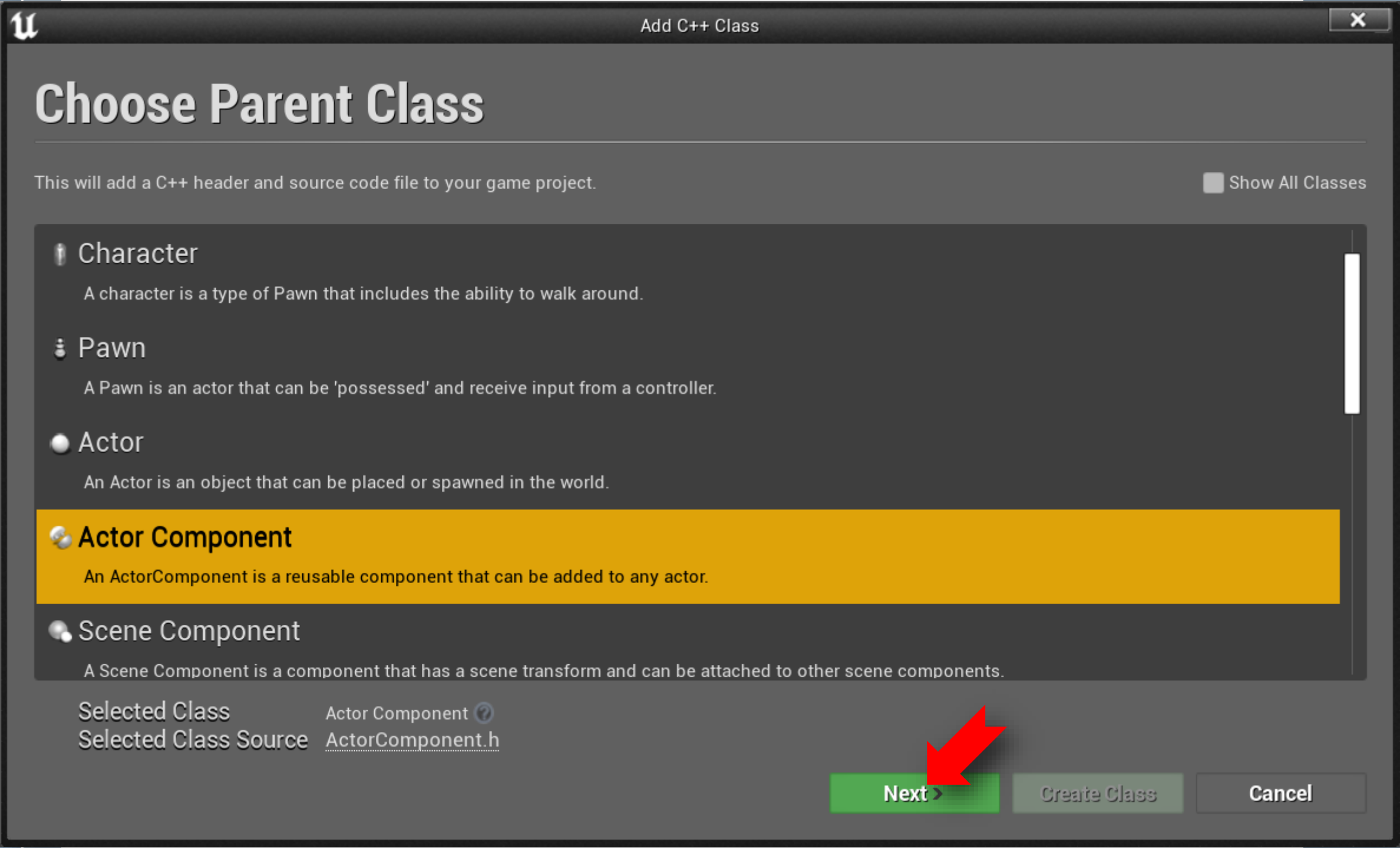
- Objects that can be added to an Actor and can have multiple types of functionality
  - Ex) being responsible for a character's inventory or making a character fly
- Must always belong to and live inside an Actor, which is referred to as their **Owner**
- Several different types of existing Actor Components
  - Code-only Actor Components
  - Mesh Components
  - Collision Components
  - Camera Components

Actor Components (2)

- Two main way to add logic to our Actors:
  - Directly in the **Actor** class or **Actor Components**
- Loose Coupling
  - To use Actor Components instead of placing logic directly inside an Actor
  - When adding logic to your project can be encapsulated in a separate component
- Ex) Two options to implement the health logic for both of the player character and enemy character
  - You implement the health logic in a base character class
  - You implement the health logic in an Actor Components

Exercise 7.02: Creating The HealthComponent Actor Component




Add C++ Class
Choose Parent Class
This will add a C++ header and source code file to your game project.
Show All Classes
Character
A character is a type of Pawn that includes the ability to walk around.
Pawn
A Pawn is an actor that can be 'possessed' and receive input from a controller.
Actor
An Actor is an object that can be placed or spawned in the world.
Actor Component
An ActorComponent is a reusable component that can be added to any actor.
Scene Component
A Scene Component is a component that has a scene transform and can be attached to other scene components.
Selected Class
Actor Component
Selected Class Source
ActorComponent.h
Next >
Create Class
Cancel

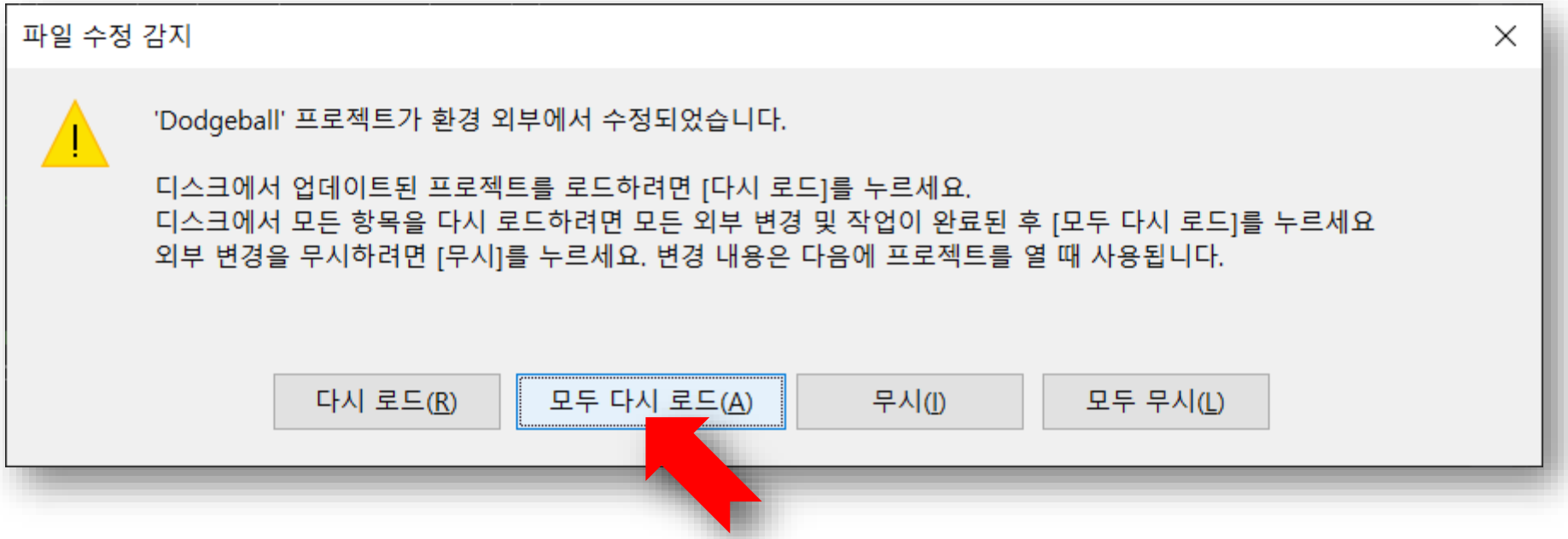
Add C++ Class

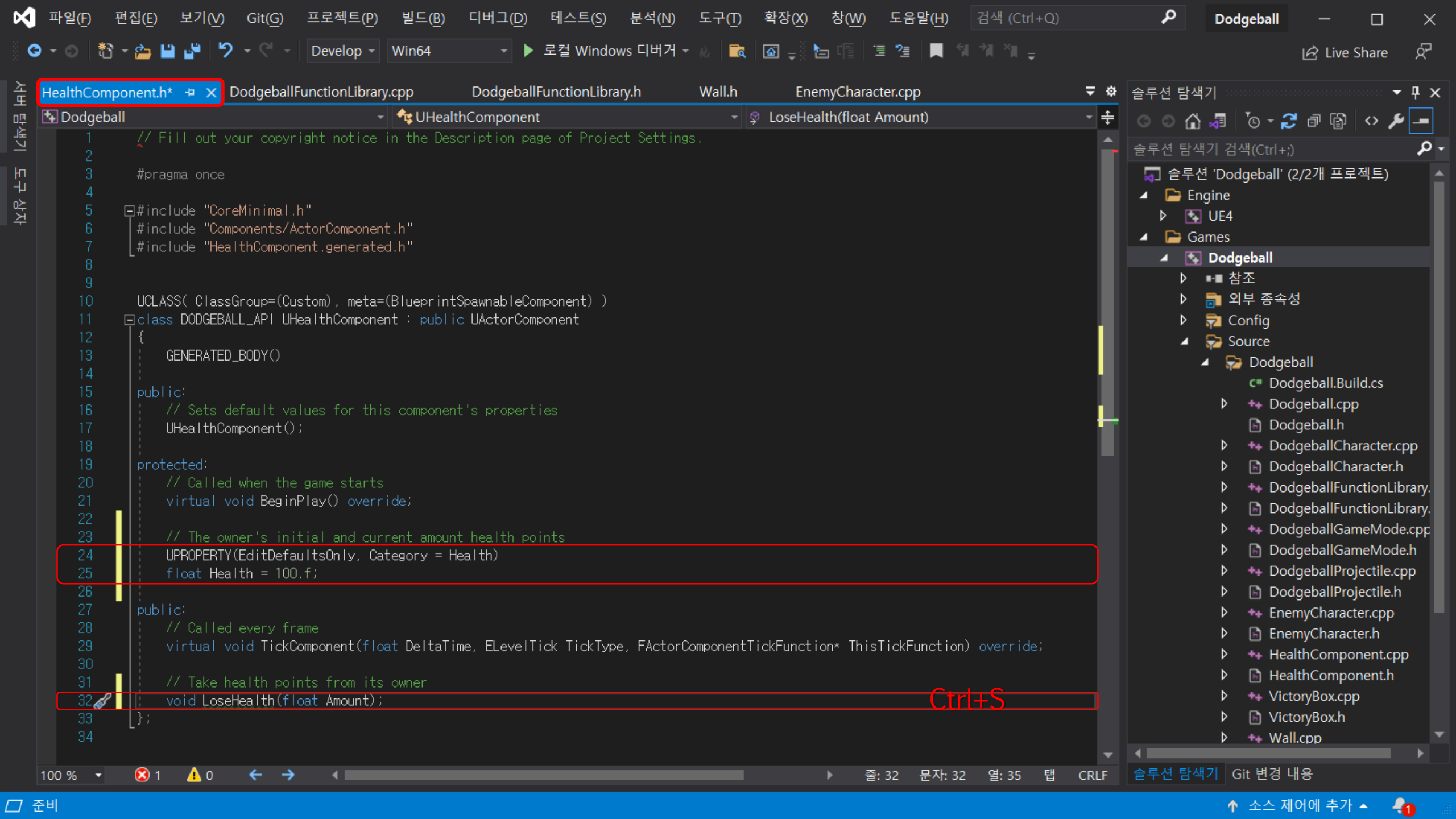
Name Your New Actor Component

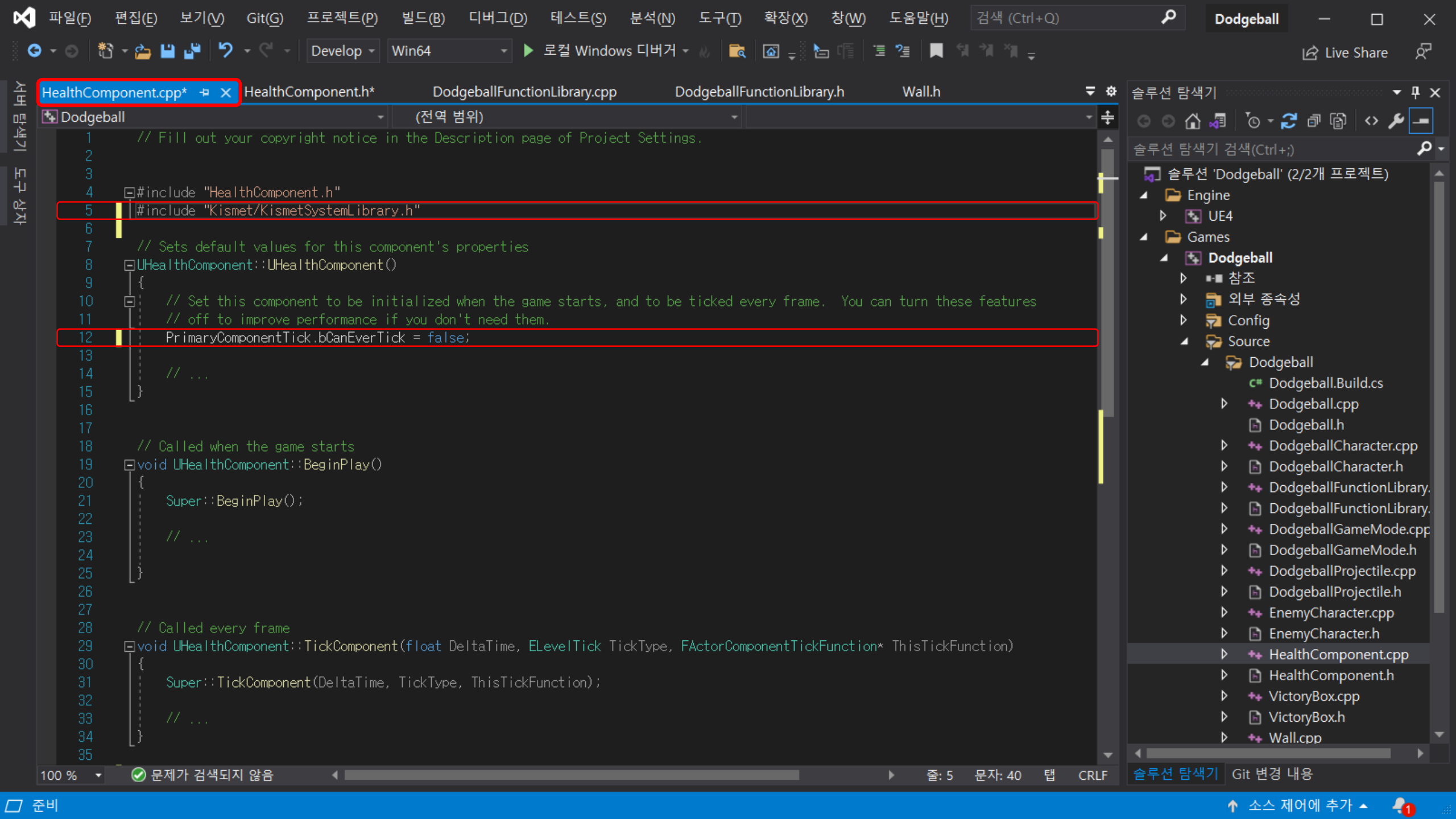
Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.
When you click the "Create" button below, a header (.h) file and a source (.cpp) file will be made using this name.

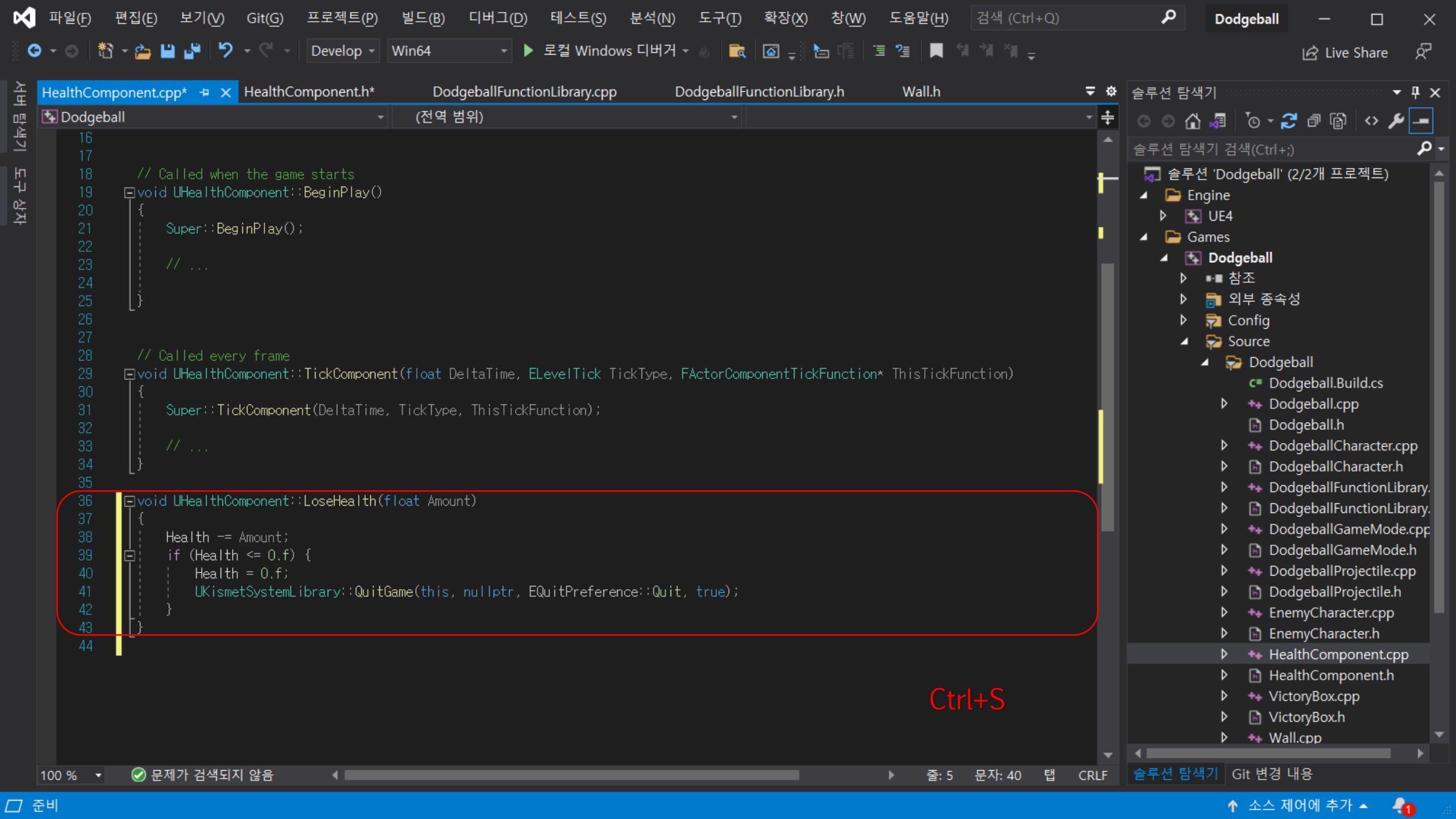
Name	HealthComponent	Dodgeball (Runtime)	Public / Private
Path	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/		Choose Folder
Header File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/HealthComponent.h		
Source File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/HealthComponent.cpp		

Back | Create Class | Cancel

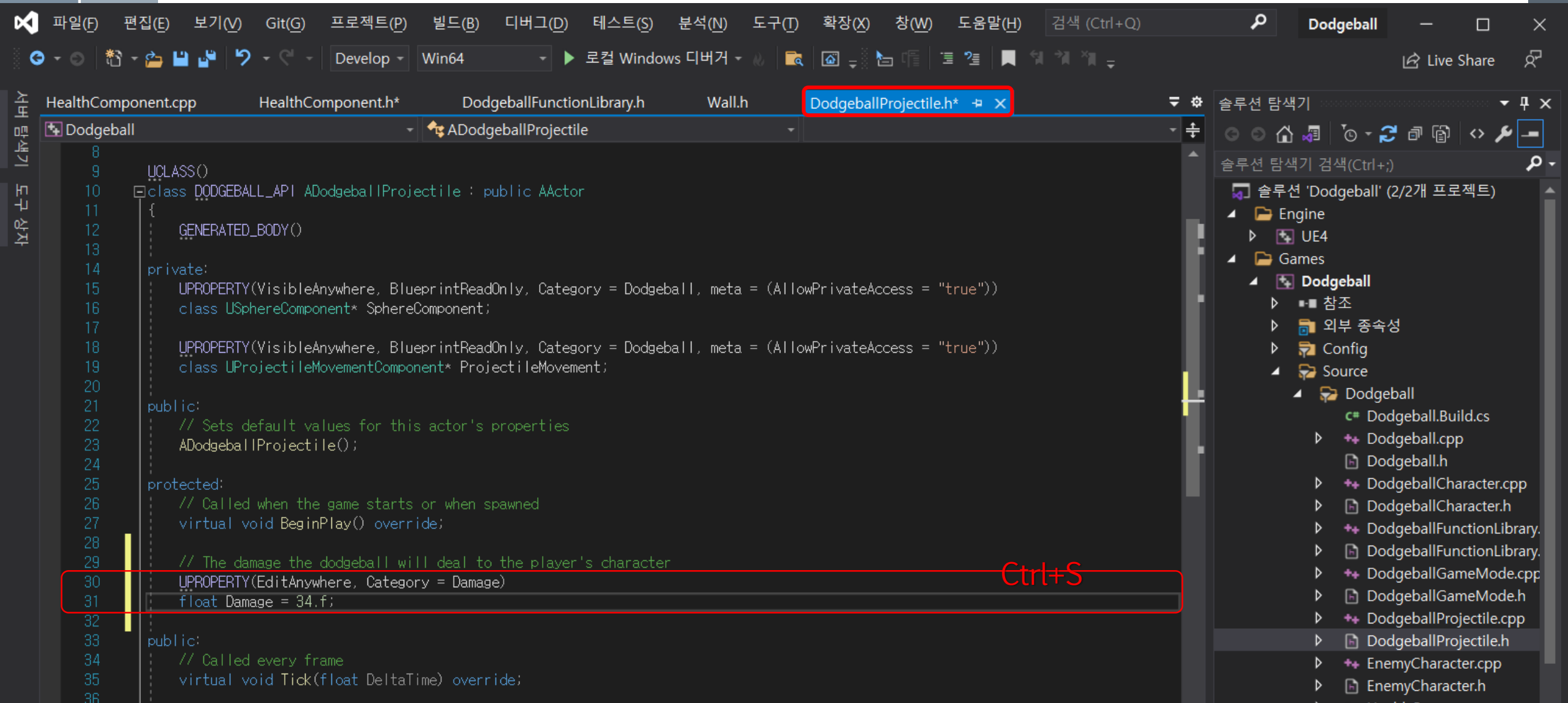

파일 수정 감지
'Dodgeball' 프로젝트가 환경 외부에서 수정되었습니다.
디스크에서 업데이트된 프로젝트를 로드하려면 [다시 로드]를 누르세요.
디스크에서 모든 항목을 다시 로드하려면 모든 외부 변경 및 작업이 완료된 후 [모두 다시 로드]를 누르세요
외부 변경을 무시하려면 [무시]를 누르세요. 변경 내용은 다음에 프로젝트를 열 때 사용됩니다.
다시 로드(R)
모두 다시 로드(A)
무시(I)
모두 무시(L)

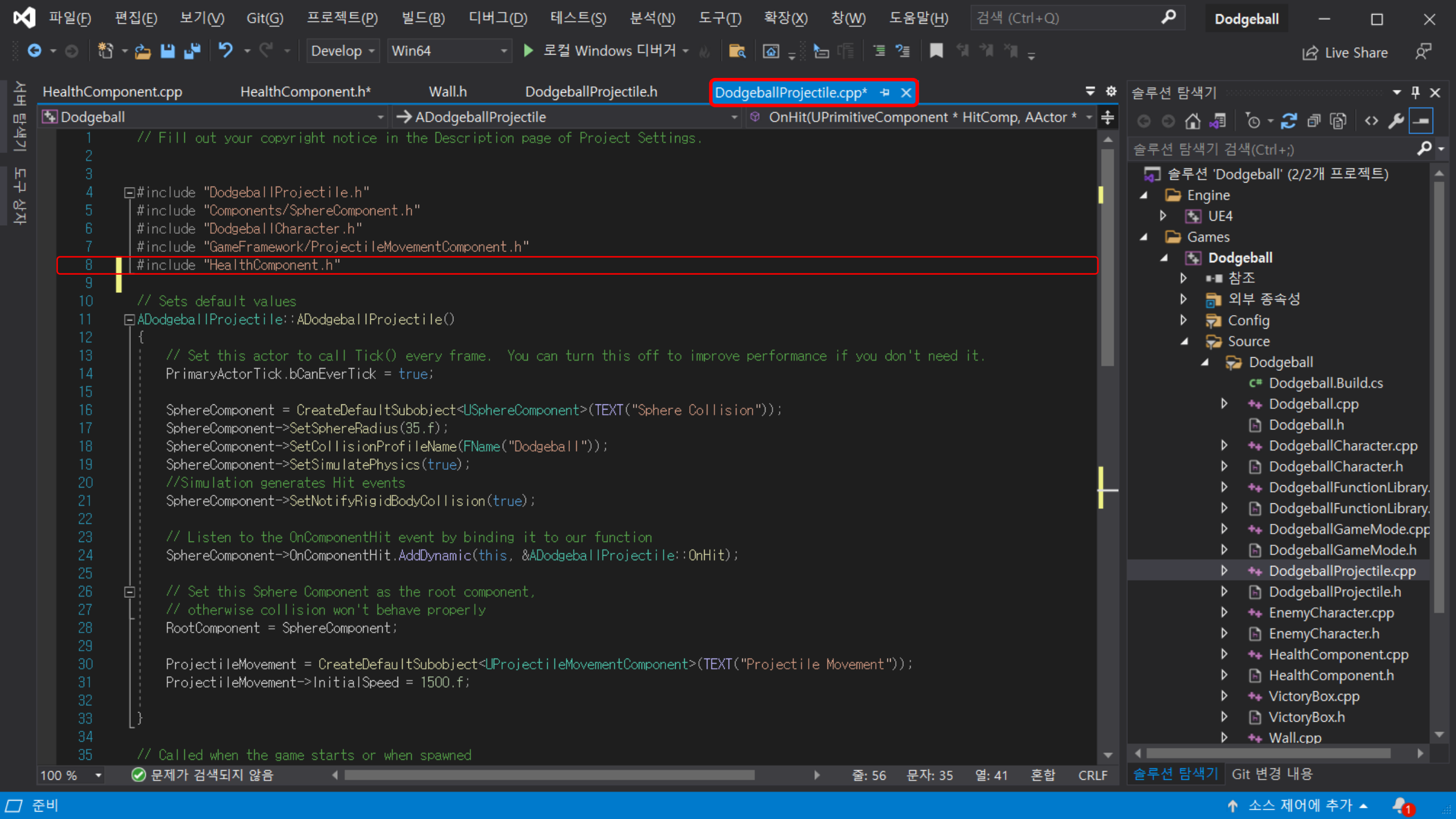

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthComponent.h*
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
Wall.h
EnemyCharacter.cpp
Dodgeball
UHealthComponent
LoseHealth(float Amount)
// Fill out your copyright notice in the Description page of Project Settings.
#pragma once
#include "CoreMinimal.h"
#include "Components/ActorComponent.h"
#include "HealthComponent.generated.h"
UCLASS(ClassGroup=(Custom), meta=(BlueprintSpawnableComponent))
class DODGEBALL_API UHealthComponent : public UActorComponent
{
GENERATED_BODY()
public:
// Sets default values for this component's properties
UHealthComponent();
protected:
// Called when the game starts
virtual void BeginPlay() override;
// The owner's initial and current amount health points
UPROPERTY(EditDefaultsOnly, Category = Health)
float Health = 100.f;
public:
// Called every frame
virtual void TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction) override;
// Take health points from its owner
void LoseHealth(float Amount);
};
Ctrl+S
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
100 %
1
0
줄: 32
문자: 32
열: 35
탭
CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

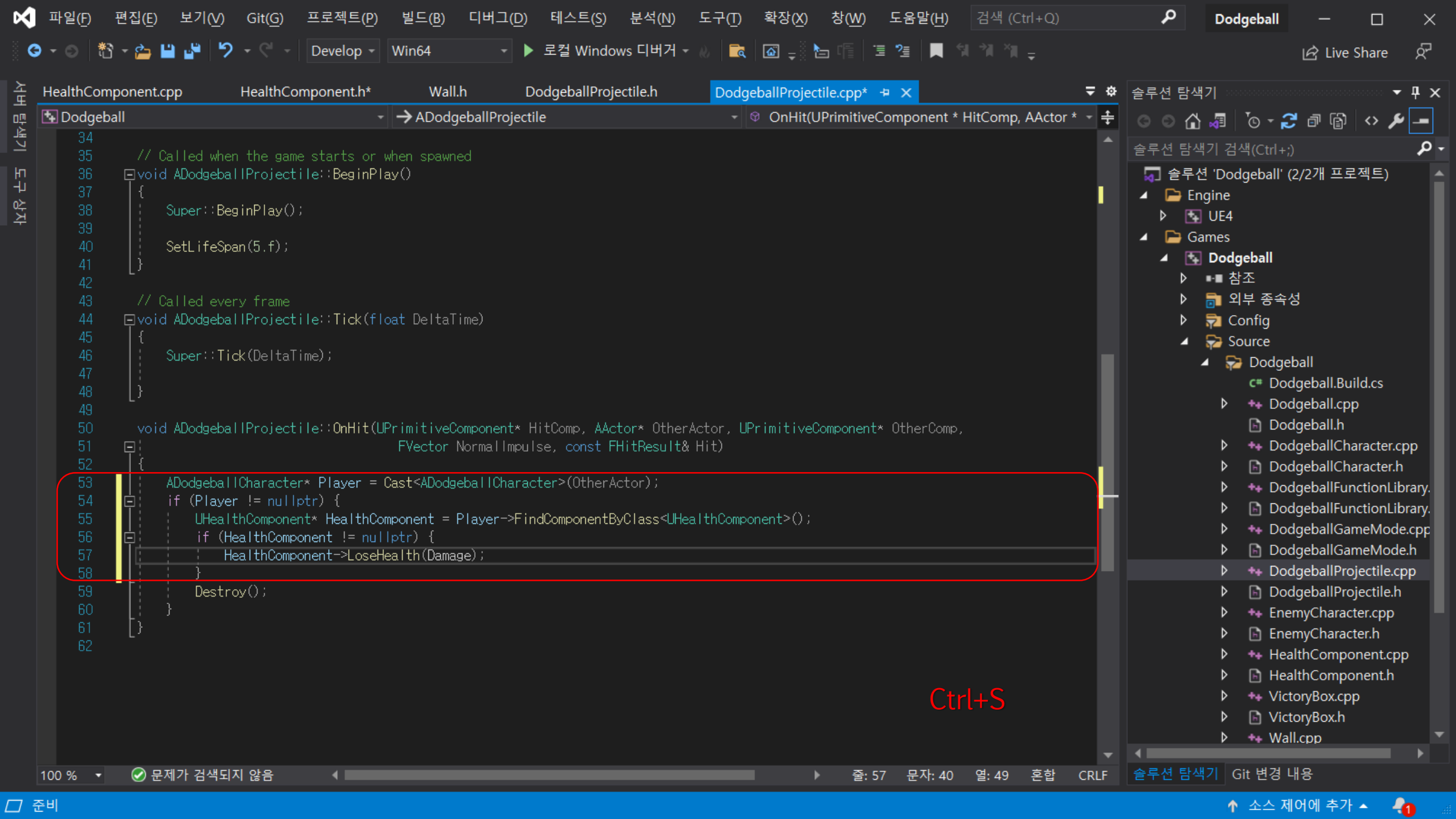

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthComponent.cpp*
HealthComponent.h*
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
Wall.h
Dodgeball
(전역 범위)
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "HealthComponent.h"
5 #include "Kismet/KismetSystemLibrary.h"
6
7 // Sets default values for this component's properties
8 UHealthComponent::UHealthComponent()
9 {
10 // Set this component to be initialized when the game starts, and to be ticked every frame. You can turn these features
11 // off to improve performance if you don't need them.
12 PrimaryComponentTick.bCanEverTick = false;
13
14 // ...
15 }
16
17
18 // Called when the game starts
19 void UHealthComponent::BeginPlay()
20 {
21 Super::BeginPlay();
22
23 // ...
24
25 }
26
27
28 // Called every frame
29 void UHealthComponent::TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction)
30 {
31 Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
32
33 // ...
34 }
35
100 %
문제가 검색되지 않음
줄: 5
문자: 40
탭
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가

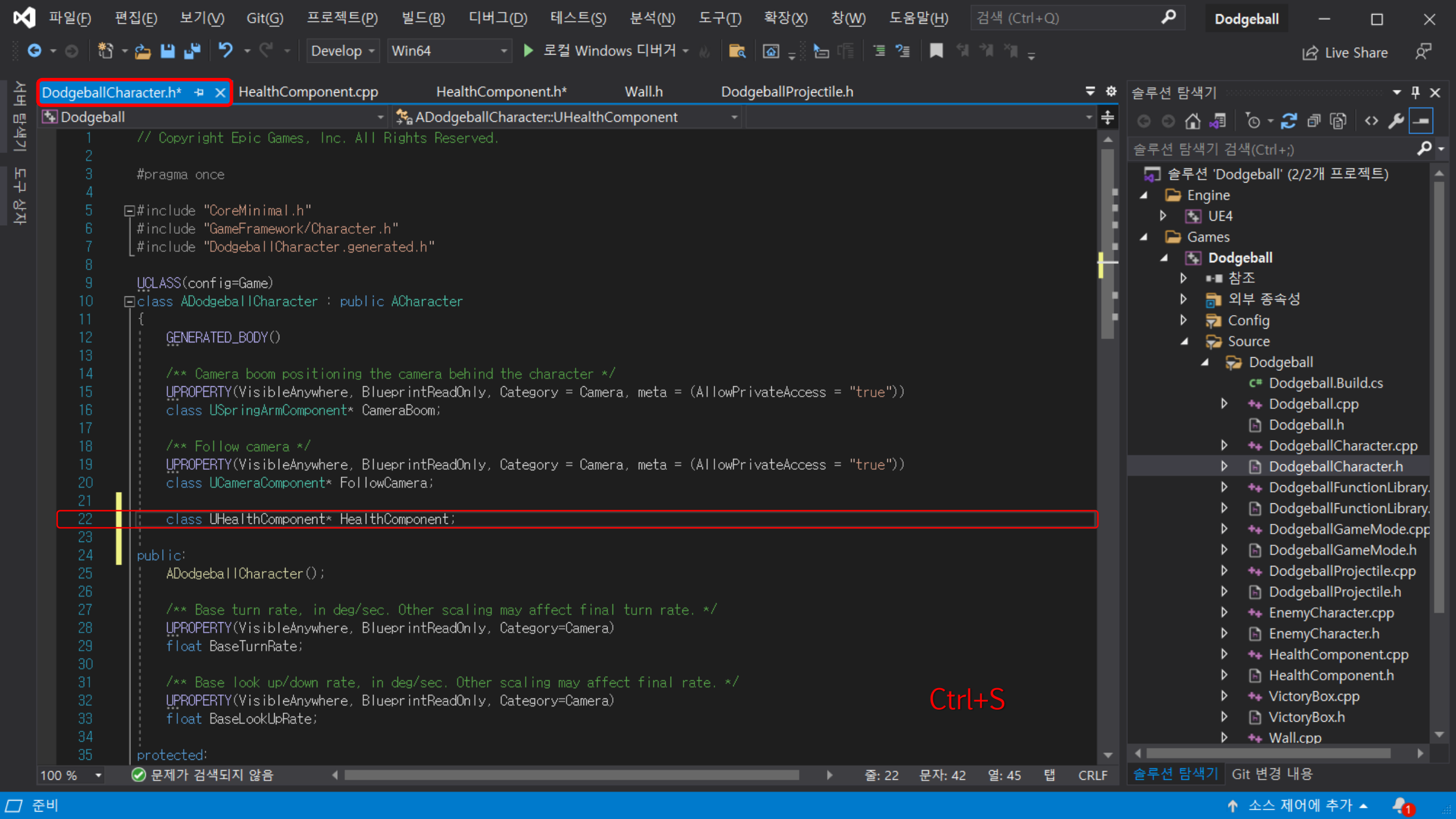

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthComponent.cpp*
HealthComponent.h*
DodgeballFunctionLibrary.cpp
DodgeballFunctionLibrary.h
Wall.h
Dodgeball
(전역 범위)
16
17
18 // Called when the game starts
19 void UHealthComponent::BeginPlay()
20 {
21 Super::BeginPlay();
22
23 // ...
24
25 }
26
27
28 // Called every frame
29 void UHealthComponent::TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction)
30 {
31 Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
32
33 // ...
34 }
35
36 void UHealthComponent::LoseHealth(float Amount)
37 {
38 Health -= Amount;
39 if (Health <= 0.f) {
40 Health = 0.f;
41 UKismetSystemLibrary::QuitGame(this, nullptr, EQuitPreference::Quit, true);
42 }
43 }
44
Ctrl+S
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
100 %
문제가 검색되지 않음
줄: 5
문자: 40
탭
CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

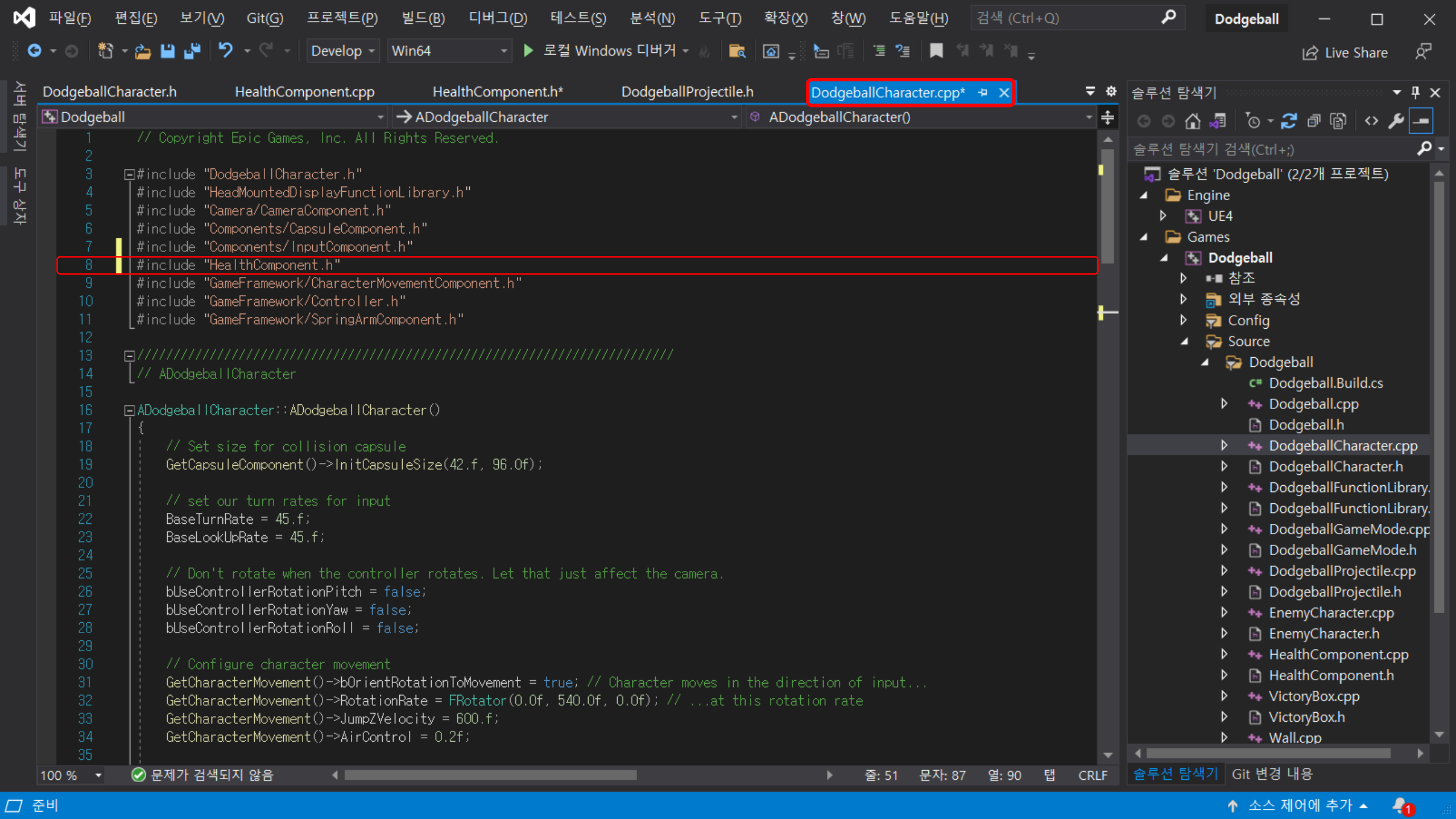
Exercise 7.03: Integrating The HealthComponent Actor Component

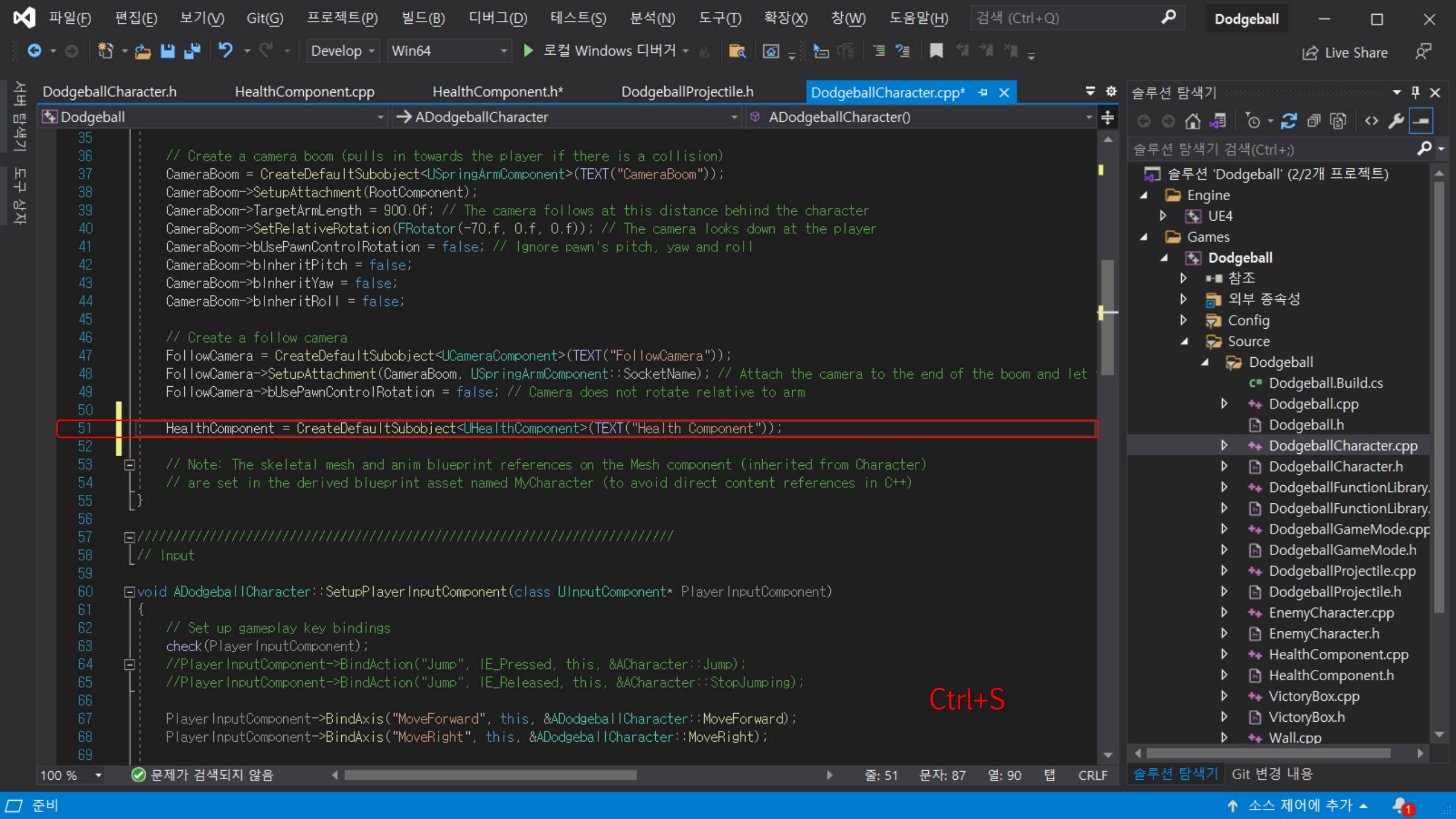


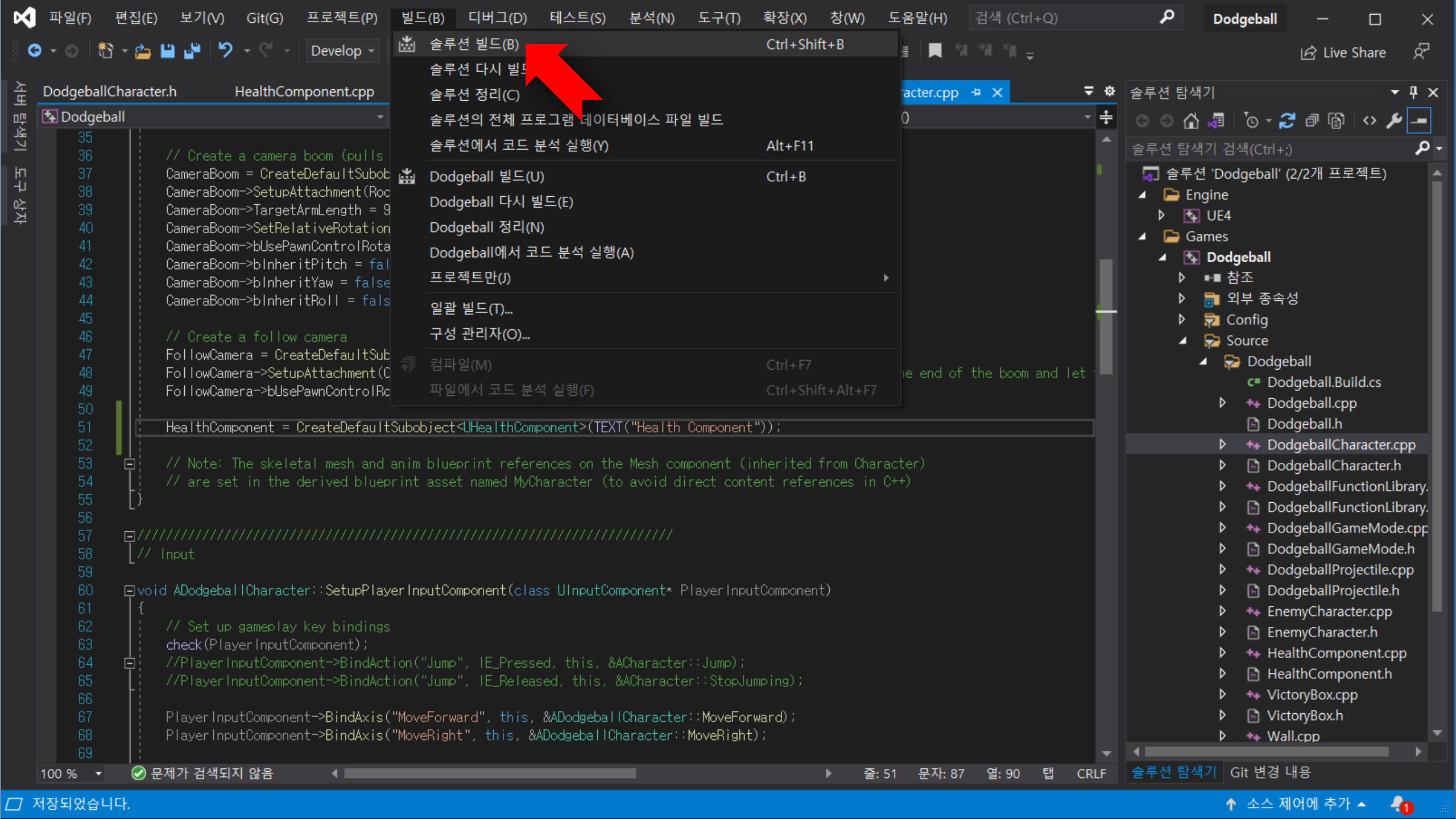

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthComponent.cpp
HealthComponent.h*
Wall.h
DodgeballProjectile.h
DodgeballProjectile.cpp*
Dodgeball
ADodgeballProjectile
OnHit(UPrimitiveComponent * HitComp, AActor *
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "DodgeballProjectile.h"
5 #include "Components/SphereComponent.h"
6 #include "DodgeballCharacter.h"
7 #include "GameFramework/ProjectileMovementComponent.h"
8 #include "HealthComponent.h"
9
10 // Sets default values
11 ADodgeballProjectile::ADodgeballProjectile()
12 {
13 // Set this actor to call Tick() every frame. You can turn this off to improve performance if you don't need it.
14 PrimaryActorTick.bCanEverTick = true;
15
16 SphereComponent = CreateDefaultSubobject<USphereComponent>(TEXT("Sphere Collision"));
17 SphereComponent->SetSphereRadius(35.f);
18 SphereComponent->SetCollisionProfileName(FName("Dodgeball"));
19 SphereComponent->SetSimulatePhysics(true);
20 //Simulation generates Hit events
21 SphereComponent->SetNotifyRigidBodyCollision(true);
22
23 // Listen to the OnComponentHit event by binding it to our function
24 SphereComponent->OnComponentHit.AddDynamic(this, &ADodgeballProjectile::OnHit);
25
26 // Set this Sphere Component as the root component,
27 // otherwise collision won't behave properly
28 RootComponent = SphereComponent;
29
30 ProjectileMovement = CreateDefaultSubobject<UProjectileMovementComponent>(TEXT("Projectile Movement"));
31 ProjectileMovement->InitialSpeed = 1500.f;
32
33 }
34
35 // Called when the game starts or when spawned
100 %
문제가 검색되지 않음
줄: 56 문자: 35 열: 41 혼합 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가

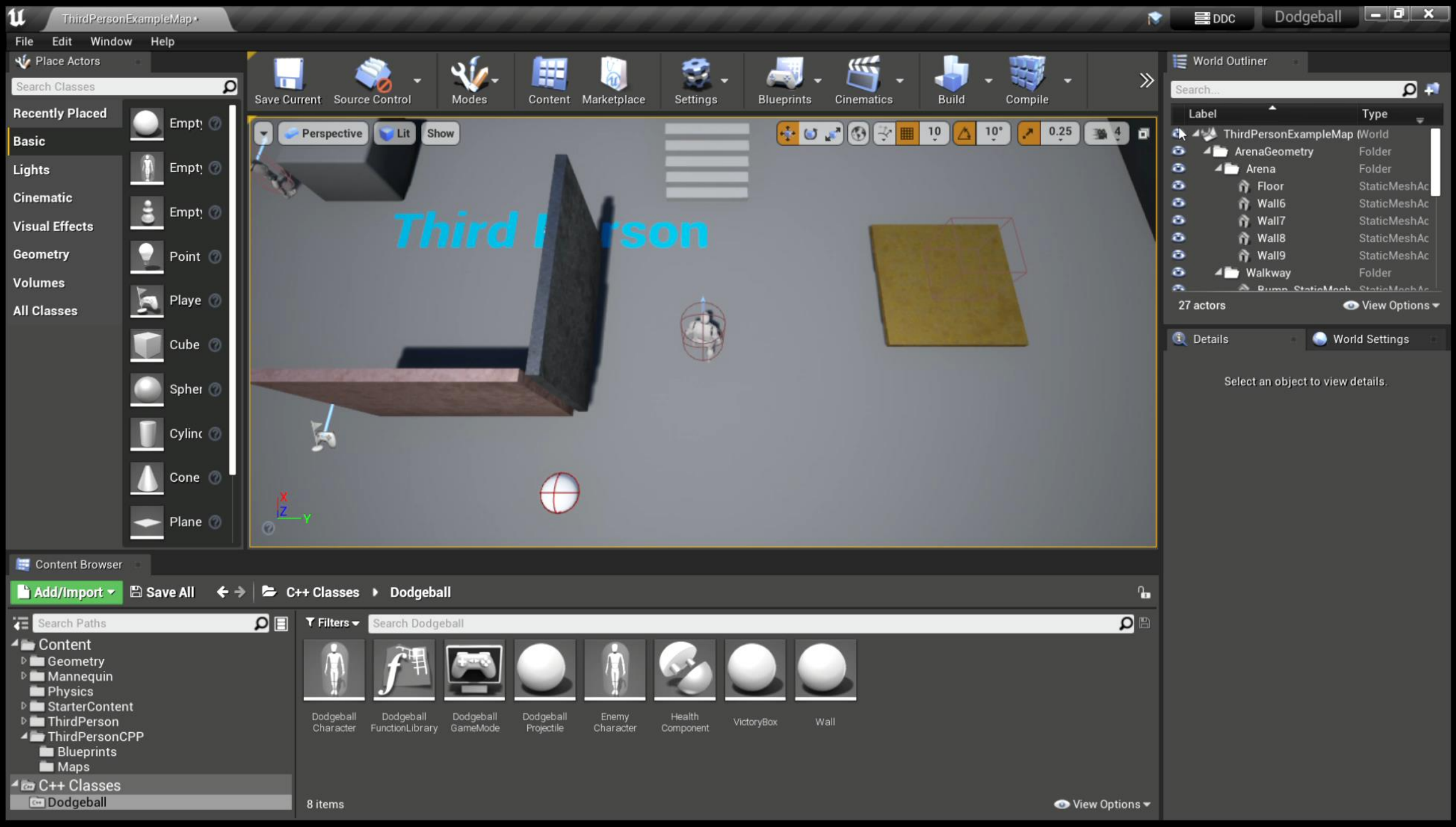

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthComponent.cpp
HealthComponent.h*
Wall.h
DodgeballProjectile.h
DodgeballProjectile.cpp*
Dodgeball
ADodgeballProjectile
OnHit(UPrimitiveComponent * HitComp, AActor *
34
35 // Called when the game starts or when spawned
36 void ADodgeballProjectile::BeginPlay()
37 {
38 Super::BeginPlay();
39
40 SetLifeSpan(5.f);
41 }
42
43 // Called every frame
44 void ADodgeballProjectile::Tick(float DeltaTime)
45 {
46 Super::Tick(DeltaTime);
47
48 }
49
50 void ADodgeballProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp,
51 FVector NormalImpulse, const FHitResult& Hit)
52 {
53 ADodgeballCharacter* Player = Cast<ADodgeballCharacter>(OtherActor);
54 if (Player != nullptr) {
55 UHealthComponent* HealthComponent = Player->FindComponentByClass<UHealthComponent>();
56 if (HealthComponent != nullptr) {
57 HealthComponent->LoseHealth(Damage);
58 }
59 Destroy();
60 }
61 }
62
Ctrl+S
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
100 %
문제가 검색되지 않음
줄: 57 문자: 40 열: 49 혼합 CRLF
솔루션 탐색기 Git 변경 내용
서버 탐색기 도구 상자
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
DodgeballCharacter.h*
HealthComponent.cpp
HealthComponent.h*
Wall.h
DodgeballProjectile.h
Dodgeball
ADodgeballCharacter::UHealthComponent
1 // Copyright Epic Games, Inc. All Rights Reserved.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/Character.h"
7 #include "DodgeballCharacter.generated.h"
8
9 UCLASS(config=Game)
10 class ADodgeballCharacter : public ACharacter
11 {
12 GENERATED_BODY()
13
14 /** Camera boom positioning the camera behind the character */
15 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Camera, meta = (AllowPrivateAccess = "true"))
16 class USpringArmComponent* CameraBoom;
17
18 /** Follow camera */
19 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Camera, meta = (AllowPrivateAccess = "true"))
20 class UCameraComponent* FollowCamera;
21
22 class UHealthComponent* HealthComponent;
23
24 public:
25 ADodgeballCharacter();
26
27 /** Base turn rate, in deg/sec. Other scaling may affect final turn rate. */
28 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category=Camera)
29 float BaseTurnRate;
30
31 /** Base look up/down rate, in deg/sec. Other scaling may affect final rate. */
32 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category=Camera)
33 float BaseLookUpRate;
34
35 protected:
Ctrl+S
100 %
문제가 검색되지 않음
줄: 22 문자: 42 열: 45 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
서버 탐색기
도구 상자
DodgeballCharacter.h
HealthComponent.cpp
HealthComponent.h*
DodgeballProjectile.h
DodgeballCharacter.cpp*
Dodgeball
ADodgeballCharacter
ADodgeballCharacter()
// Copyright Epic Games, Inc. All Rights Reserved.
#include "DodgeballCharacter.h"
#include "HeadMountedDisplayFunctionLibrary.h"
#include "Camera/CameraComponent.h"
#include "Components/CapsuleComponent.h"
#include "Components/InputComponent.h"
#include "HealthComponent.h"
#include "GameFramework/CharacterMovementComponent.h"
#include "GameFramework/Controller.h"
#include "GameFramework/SpringArmComponent.h"
//
// ADodgeballCharacter
ADodgeballCharacter::ADodgeballCharacter()
{
// Set size for collision capsule
GetCapsuleComponent()->InitCapsuleSize(42.f, 96.0f);
// set our turn rates for input
BaseTurnRate = 45.f;
BaseLookUpRate = 45.f;
// Don't rotate when the controller rotates. Let that just affect the camera.
bUseControllerRotationPitch = false;
bUseControllerRotationYaw = false;
bUseControllerRotationRoll = false;
// Configure character movement
GetCharacterMovement()->bOrientRotationToMovement = true; // Character moves in the direction of input...
GetCharacterMovement()->RotationRate = FRotator(0.0f, 540.0f, 0.0f); // ...at this rotation rate
GetCharacterMovement()->JumpZVelocity = 600.f;
GetCharacterMovement()->AirControl = 0.2f;
100 %
문제가 검색되지 않음
줄: 51 문자: 87 열: 90 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
DodgeballCharacter.h
HealthComponent.cpp
HealthComponent.h*
DodgeballProjectile.h
DodgeballCharacter.cpp*
Dodgeball
ADodgeballCharacter
ADodgeballCharacter()
35
36 // Create a camera boom (pulls in towards the player if there is a collision)
37 CameraBoom = CreateDefaultSubobject<USpringArmComponent>(TEXT("CameraBoom"));
38 CameraBoom->SetupAttachment(RootComponent);
39 CameraBoom->TargetArmLength = 900.0f; // The camera follows at this distance behind the character
40 CameraBoom->SetRelativeRotation(FRotator(-70.f, 0.f, 0.f)); // The camera looks down at the player
41 CameraBoom->bUsePawnControlRotation = false; // Ignore pawn's pitch, yaw and roll
42 CameraBoom->bInheritPitch = false;
43 CameraBoom->bInheritYaw = false;
44 CameraBoom->bInheritRoll = false;
45
46 // Create a follow camera
47 FollowCamera = CreateDefaultSubobject<UCameraComponent>(TEXT("FollowCamera"));
48 FollowCamera->SetupAttachment(CameraBoom, USpringArmComponent::SocketName); // Attach the camera to the end of the boom and let
49 FollowCamera->bUsePawnControlRotation = false; // Camera does not rotate relative to arm
50
51 HealthComponent = CreateDefaultSubobject<UHealthComponent>(TEXT("Health Component"));
52
53 // Note: The skeletal mesh and anim blueprint references on the Mesh component (inherited from Character)
54 // are set in the derived blueprint asset named MyCharacter (to avoid direct content references in C++)
55 }
56
57 //
58 // Input
59
60 void ADodgeballCharacter::SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent)
61 {
62 // Set up gameplay key bindings
63 check(PlayerInputComponent);
64 //PlayerInputComponent->BindAction("Jump", IE_Pressed, this, &ACharacter::Jump);
65 //PlayerInputComponent->BindAction("Jump", IE_Released, this, &ACharacter::StopJumping);
66
67 PlayerInputComponent->BindAxis("MoveForward", this, &ADodgeballCharacter::MoveForward);
68 PlayerInputComponent->BindAxis("MoveRight", this, &ADodgeballCharacter::MoveRight);
69
Ctrl+S
100 %
문제가 검색되지 않음
줄: 51 문자: 87 열: 90 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Live Share
Develop
솔루션 빌드(B) Ctrl+Shift+B
솔루션 다시 빌드
솔루션 정리(C)
솔루션의 전체 프로그램 데이터베이스 파일 빌드
솔루션에서 코드 분석 실행(Y) Alt+F11
Dodgeball 빌드(U) Ctrl+B
Dodgeball 다시 빌드(E)
Dodgeball 정리(N)
Dodgeball에서 코드 분석 실행(A)
프로젝트만(J)
일괄 빌드(T)...
구성 관리자(O)...
컴파일(M) Ctrl+F7
파일에서 코드 분석 실행(F) Ctrl+Shift+Alt+F7
DodgeballCharacter.h
HealthComponent.cpp
Dodgeball
// Create a camera boom (pulls
CameraBoom = CreateDefaultSubob
CameraBoom->SetupAttachment(Roc
CameraBoom->TargetArmLength = 9
CameraBoom->SetRelativeRotation
CameraBoom->bUsePawnControlRota
CameraBoom->bInheritPitch = fal
CameraBoom->bInheritYaw = false
CameraBoom->bInheritRoll = fals
// Create a follow camera
FollowCamera = CreateDefaultSub
FollowCamera->SetupAttachment(C
FollowCamera->bUsePawnControlRo
HealthComponent = CreateDefaultSubobject<UHealthComponent>(TEXT("Health Component"));
// Note: The skeletal mesh and anim blueprint references on the Mesh component (inherited from Character)
// are set in the derived blueprint asset named MyCharacter (to avoid direct content references in C++)
// Input
void ADodgeballCharacter::SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent)
// Set up gameplay key bindings
check(PlayerInputComponent);
//PlayerInputComponent->BindAction("Jump", IE_Pressed, this, &ACharacter::Jump);
//PlayerInputComponent->BindAction("Jump", IE_Released, this, &ACharacter::StopJumping);
PlayerInputComponent->BindAxis("MoveForward", this, &ADodgeballCharacter::MoveForward);
PlayerInputComponent->BindAxis("MoveRight", this, &ADodgeballCharacter::MoveRight);
100 %
문제가 검색되지 않음
줄: 51 문자: 87 열: 90 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
VictoryBox.cpp
VictoryBox.h
Wall.cpp
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
저장되었습니다.
소스 제어에 추가


ThirdPersonExampleMap
DDC
Dodgeball
File
Edit
Window
Help
Place Actors
Search Classes
Recently Placed
Basic
Lights
Cinematic
Visual Effects
Geometry
Volumes
All Classes
Empty
Empty
Empty
Point
Playe
Cube
Spher
Cylinc
Cone
Plane
Save Current
Source Control
Modes
Content
Marketplace
Settings
Blueprints
Cinematics
Build
Compile
Perspective
Lit
Show
10
10°
0.25
4
Third Person
X
Z
Y
World Outliner
Search...
Label
Type
ThirdPersonExampleMap (World
ArenaGeometry
Folder
Arena
Folder
Floor
StaticMeshAc
Wall6
StaticMeshAc
Wall7
StaticMeshAc
Wall8
StaticMeshAc
Wall9
StaticMeshAc
Walkway
Folder
27 actors
View Options
Details
World Settings
Select an object to view details.
Content Browser
Add/Import
Save All
C++ Classes
Dodgeball
Search Paths
Content
Geometry
Mannequin
Physics
StarterContent
ThirdPerson
ThirdPersonCPP
Blueprints
Maps
C++ Classes
Dodgeball
Filters
Search Dodgeball
Dodgeball Character
Dodgeball FunctionLibrary
Dodgeball GameMode
Dodgeball Projectile
Enemy Character
Health Component
VictoryBox
Wall
8 items
View Options

Interfaces

- Classes that contain a collection of functions that an object must have if it implements that Interface
- Essentially works as a contract that the object signs, saying that it will implement all the functions presents on that Interface
- Owner implement that Interface and then call that function from the Interface.
  - This will make it easy for us to specify how each actor behaves.
    - Ex) when running out of health points: some actors might simply destroyed, others might trigger an in-game event, and others might simply end the game

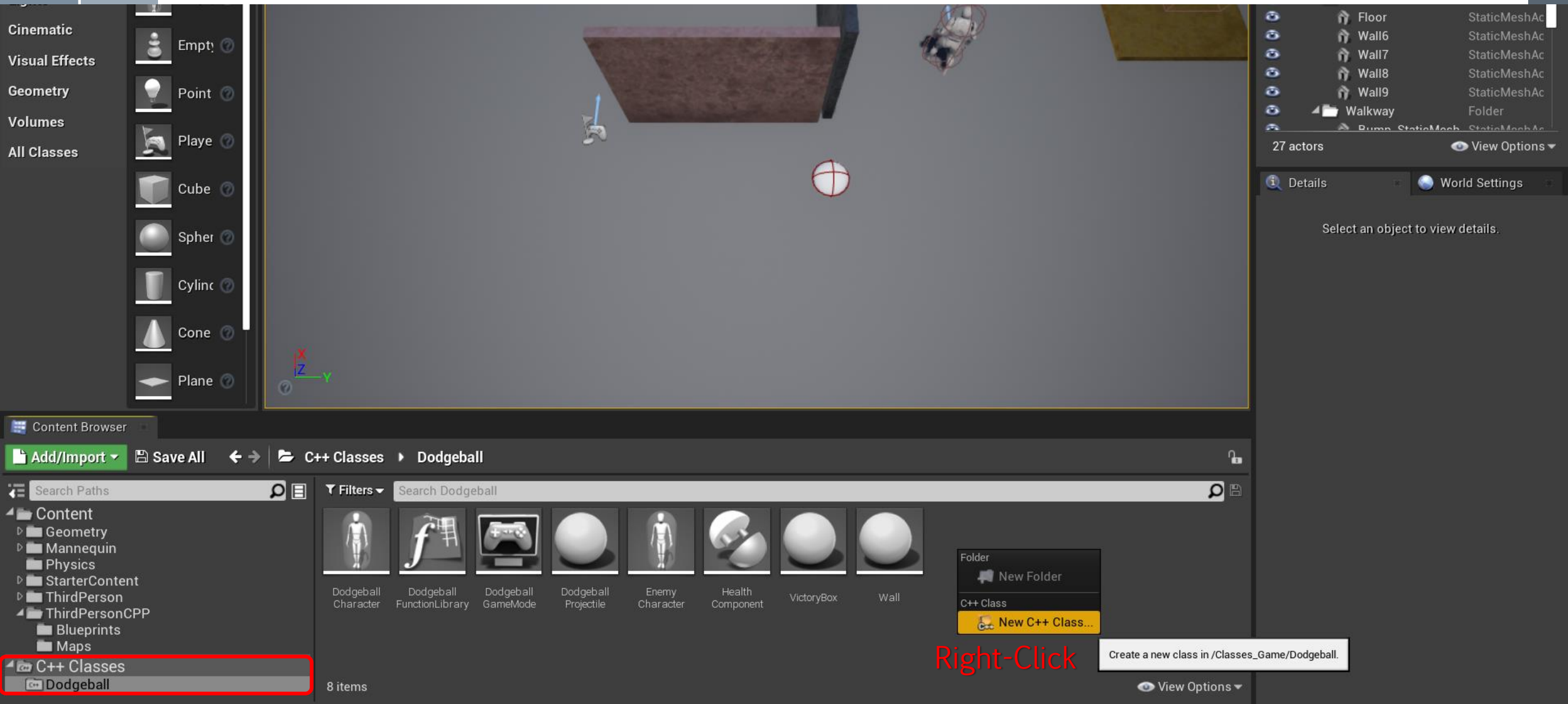
Blueprint Native Event

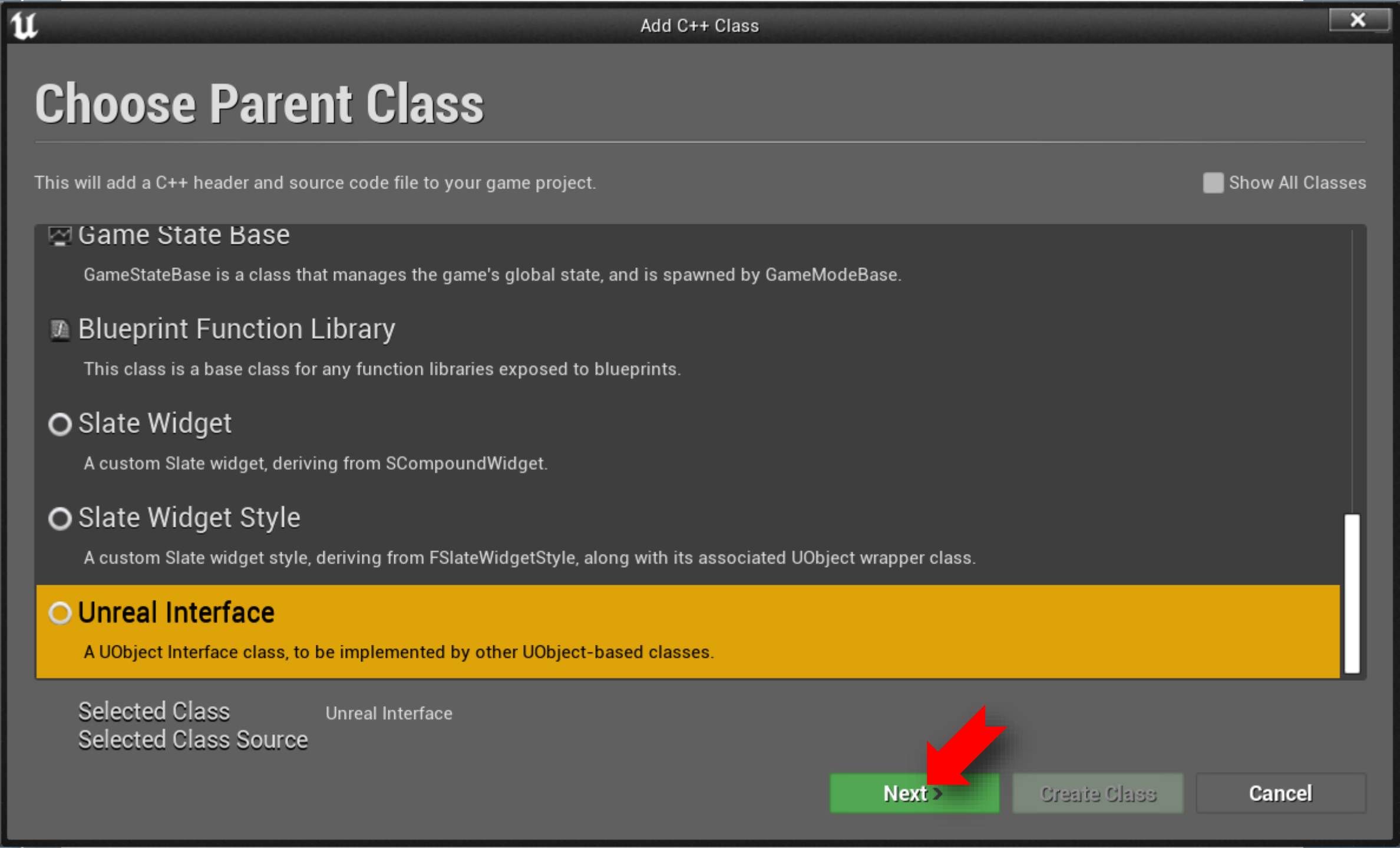
- An event that is declared in C++ that can have a default behavior but that can be overridden in Blueprint
- When using the **UFUNCTION** macro in C++, simply adding the **BlueprintNativeEvent** tag to that macro

```
UFUNCTION(BlueprintNativeEvent)
void MyEvent();
virtual void MyEvent_Implementation();
```

  - Blueprint signature: to override the event in Blueprint
  - C++ signature: to override the event in C++
    - Simply the name of event followed by **_Implementation**, and should always be a **virtual** function
    - In order to implement its default behavior, you have to implement

Exercise 7.04 Creating The HealthInteface Class




Add C++ Class
Choose Parent Class
This will add a C++ header and source code file to your game project.
Show All Classes
Game State Base
GameStateBase is a class that manages the game's global state, and is spawned by GameModeBase.
Blueprint Function Library
This class is a base class for any function libraries exposed to blueprints.
Slate Widget
A custom Slate widget, deriving from SCompoundWidget.
Slate Widget Style
A custom Slate widget style, deriving from FSlateWidgetStyle, along with its associated UObject wrapper class.
Unreal Interface
A UObject Interface class, to be implemented by other UObject-based classes.
Selected Class
Unreal Interface
Selected Class Source
Next >
Create Class
Cancel

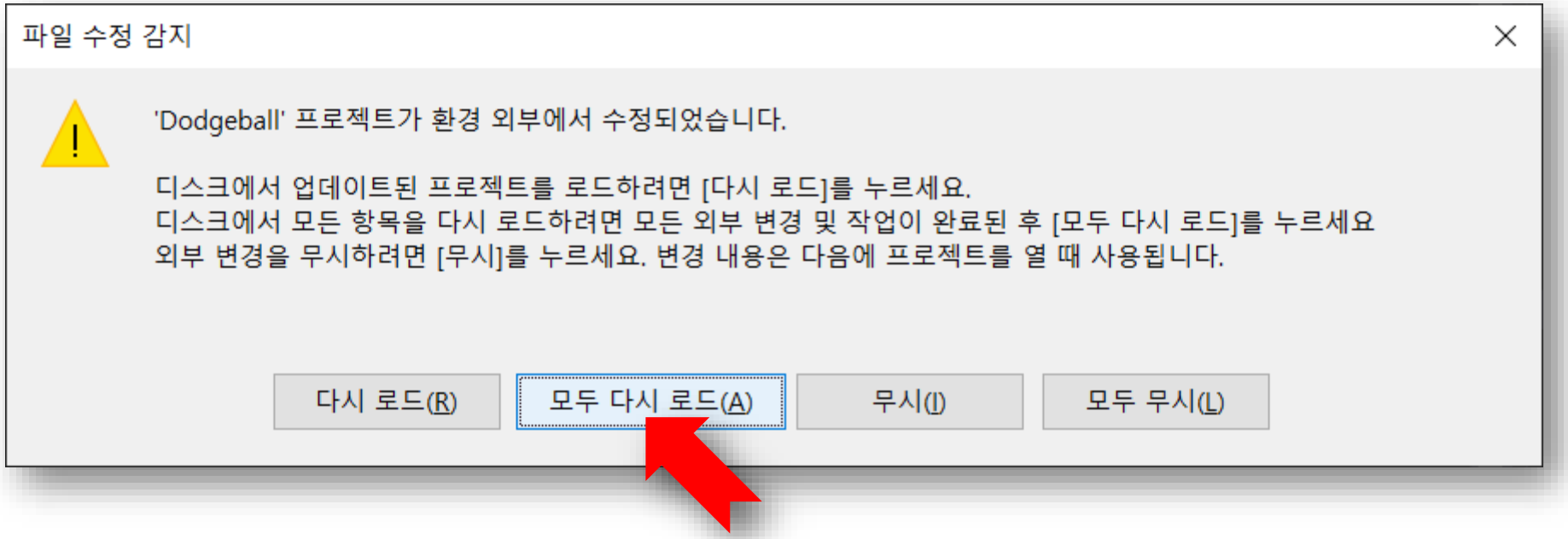
Add C++ Class

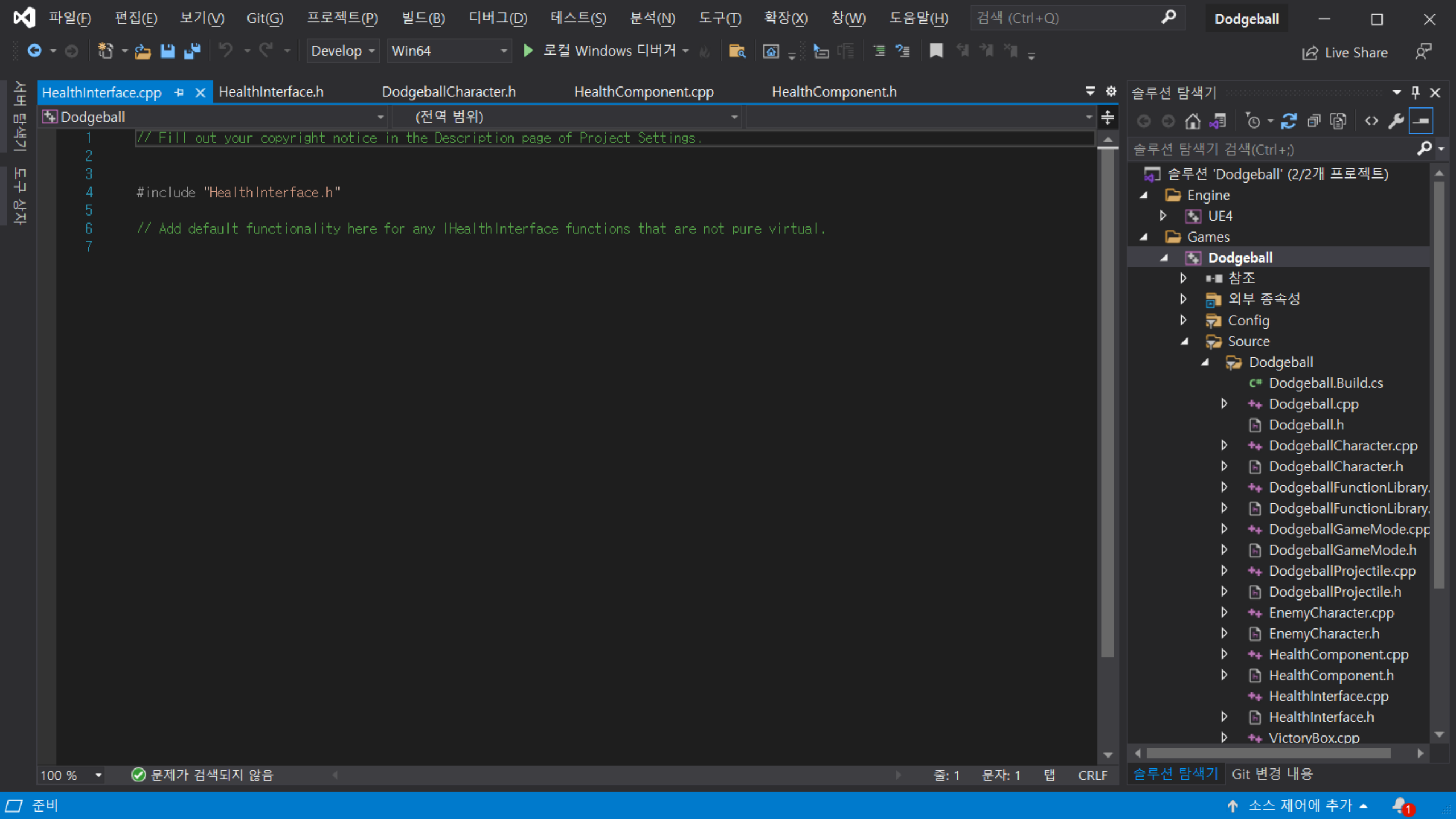
Name Your New Unreal Interface

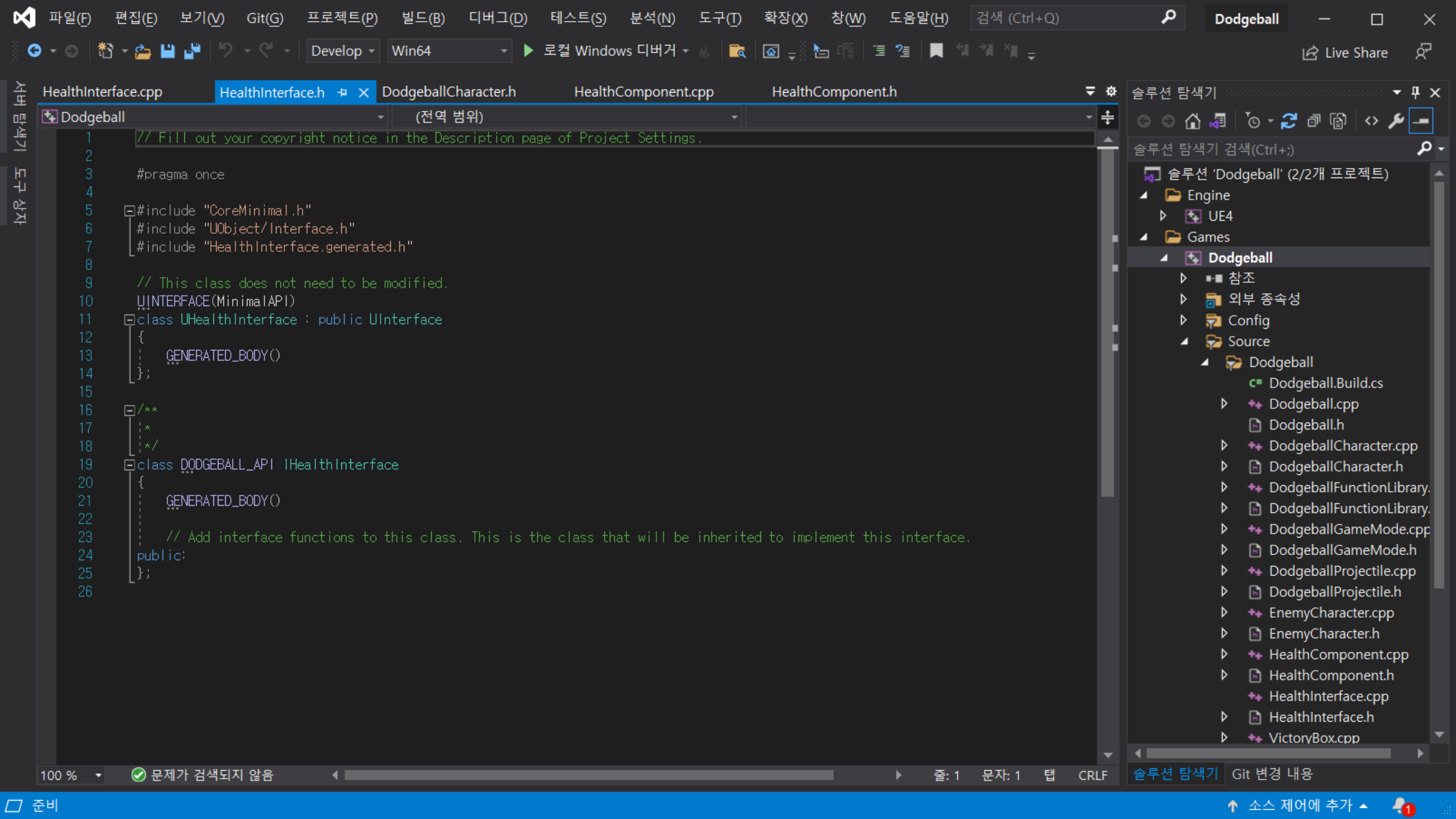
Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.
When you click the "Create" button below, a header (.h) file and a source (.cpp) file will be made using this name.

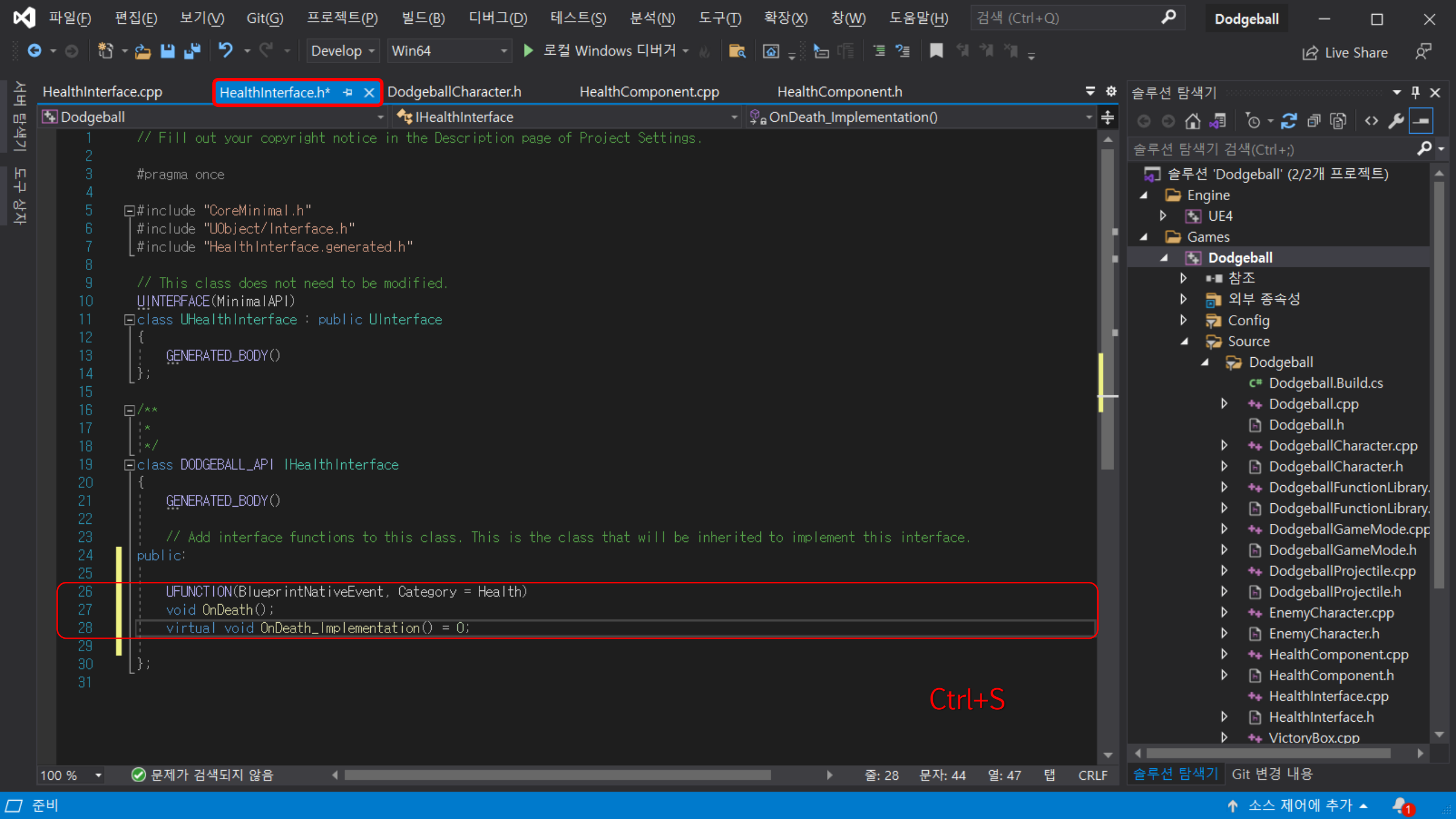
Name	HealthInterface	Dodgeball (Runtime)	Public / Private
Path	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/		Choose Folder
Header File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/HealthInterface.h		
Source File	C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/HealthInterface.cpp		

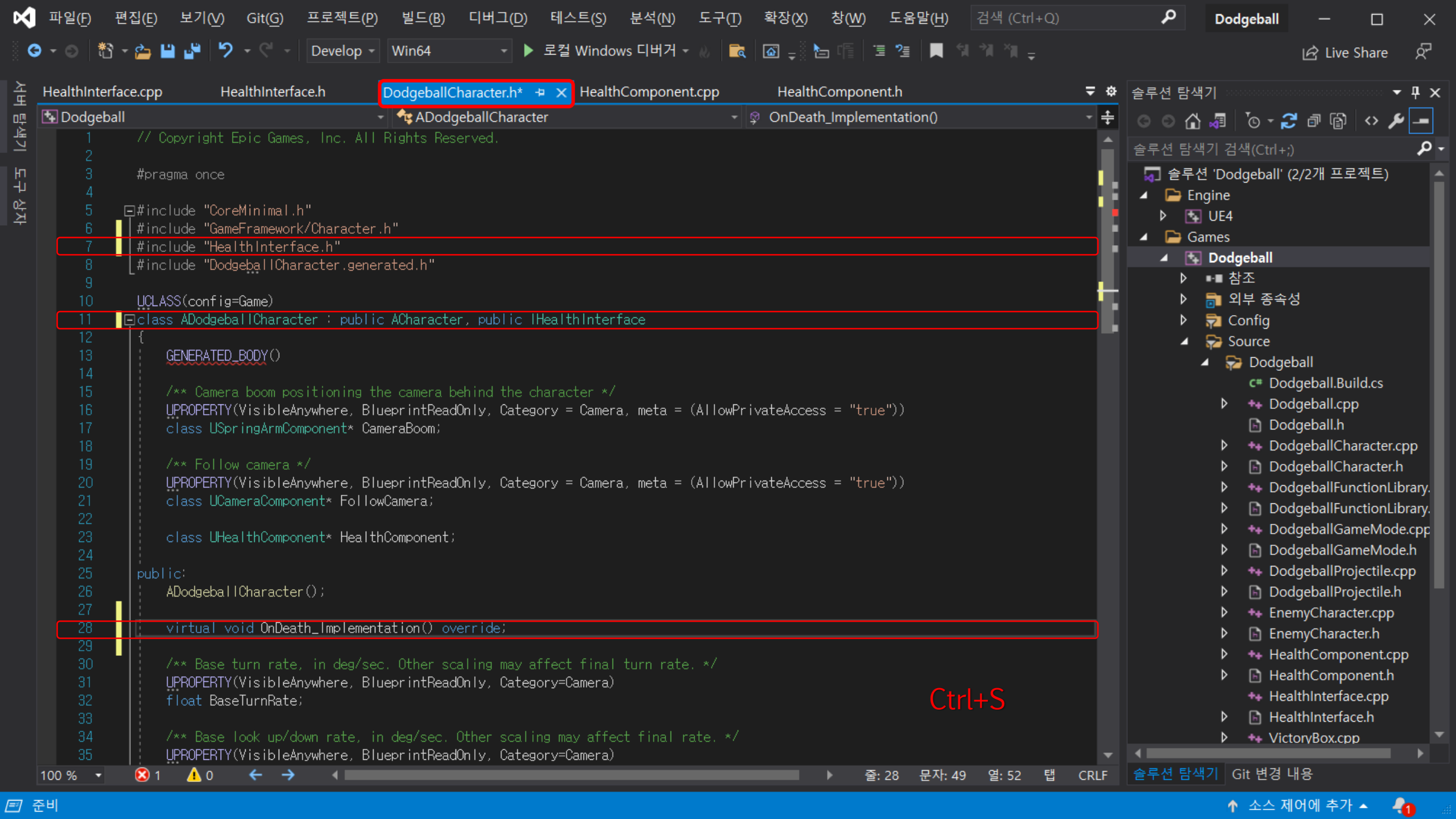
Back | Create Class | Cancel

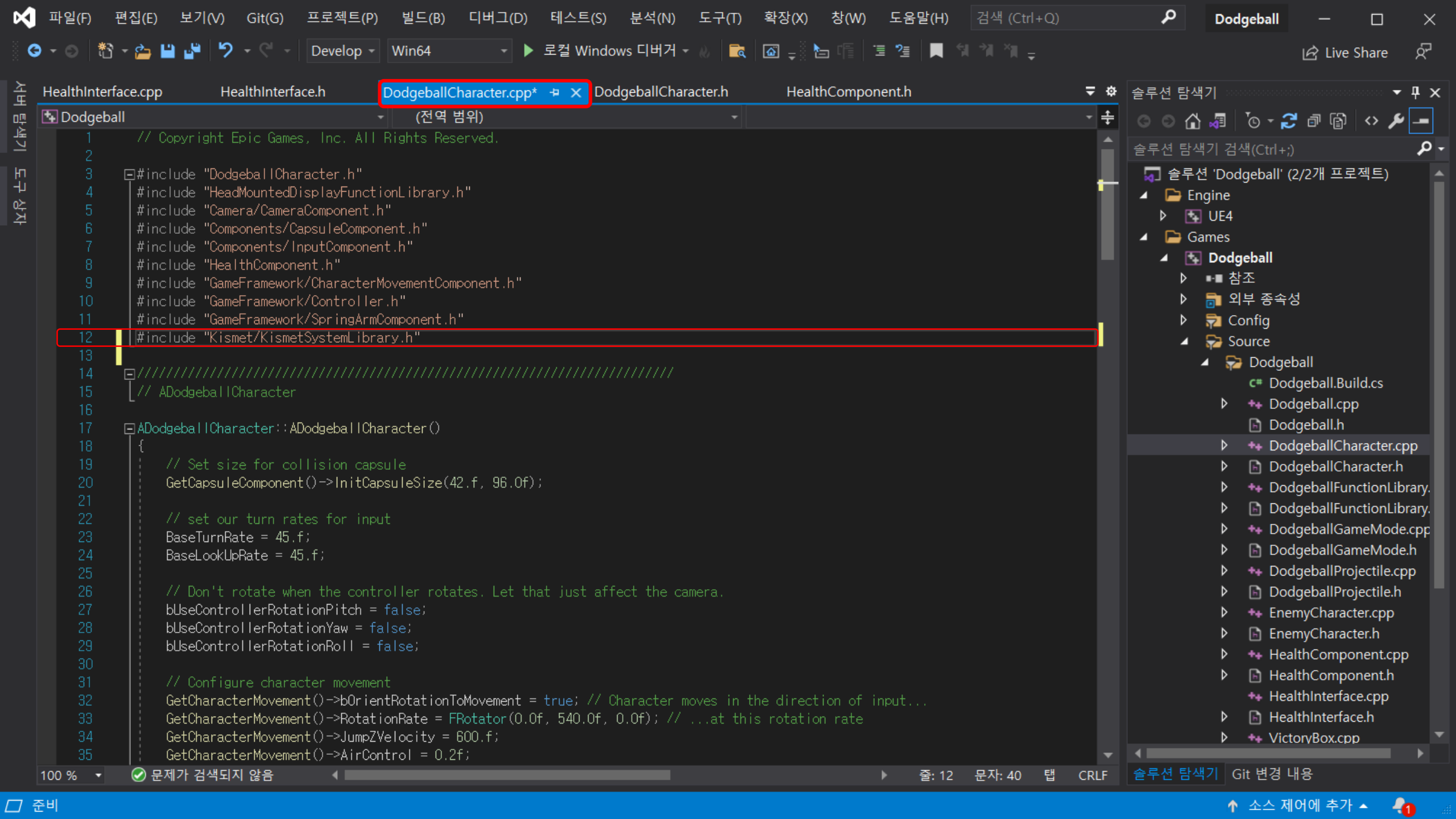

파일 수정 감지
'Dodgeball' 프로젝트가 환경 외부에서 수정되었습니다.
디스크에서 업데이트된 프로젝트를 로드하려면 [다시 로드]를 누르세요.
디스크에서 모든 항목을 다시 로드하려면 모든 외부 변경 및 작업이 완료된 후 [모두 다시 로드]를 누르세요
외부 변경을 무시하려면 [무시]를 누르세요. 변경 내용은 다음에 프로젝트를 열 때 사용됩니다.
다시 로드(R)
모두 다시 로드(A)
무시(I)
모두 무시(L)

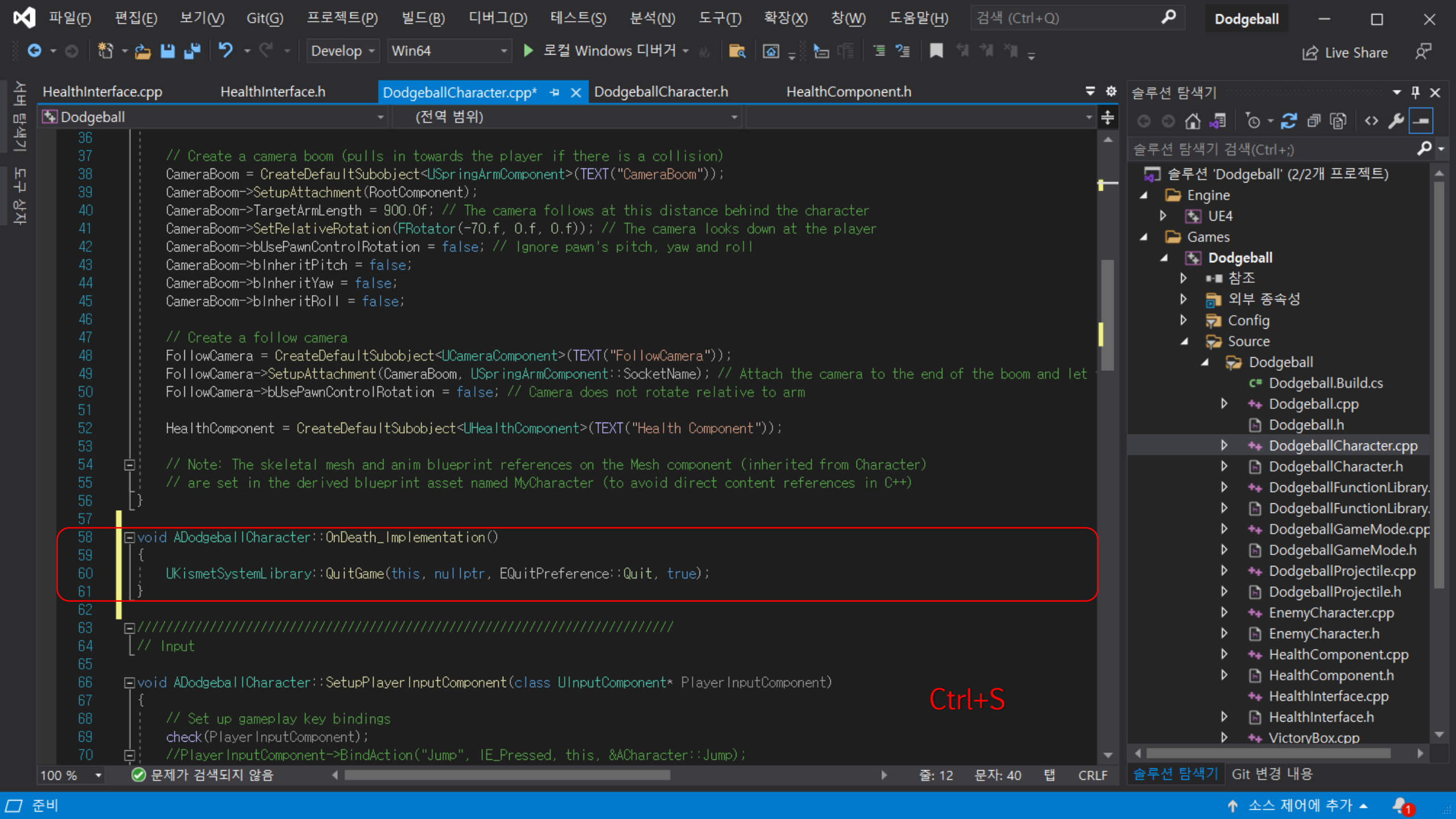

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
서버 탐색기
도구 상자
HealthInterface.cpp
HealthInterface.h
DodgeballCharacter.h
HealthComponent.cpp
HealthComponent.h
Dodgeball
(전역 범위)
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "HealthInterface.h"
5
6 // Add default functionality here for any IHealthInterface functions that are not pure virtual.
7
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
VictoryBox.cpp
100 %
문제가 검색되지 않음
줄: 1
문자: 1
탭
CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

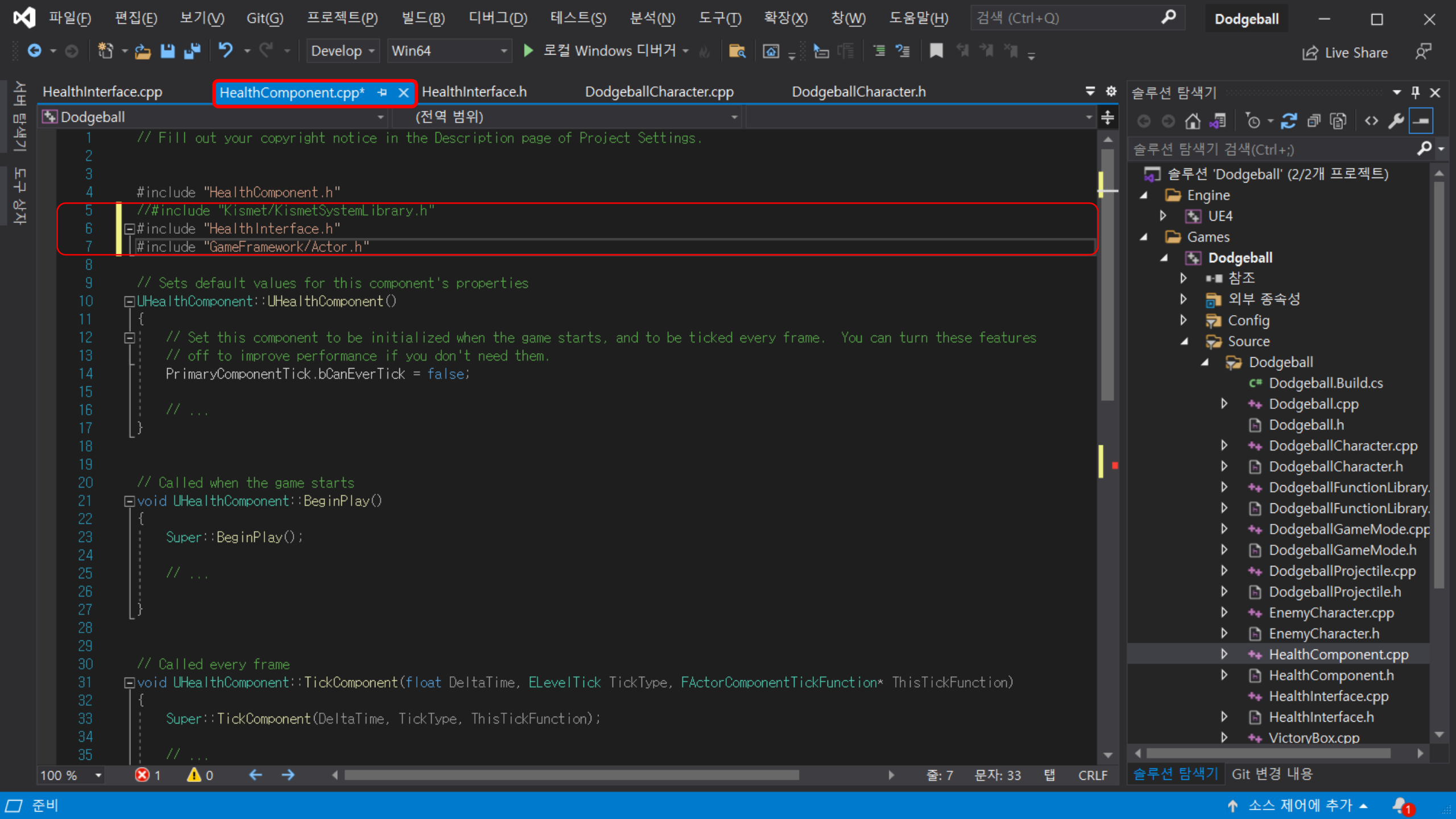

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
서버 탐색기
도구 상자
HealthInterface.cpp
HealthInterface.h
DodgeballCharacter.h
HealthComponent.cpp
HealthComponent.h
Dodgeball
(전역 범위)
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "UObject/Interface.h"
7 #include "HealthInterface.generated.h"
8
9 // This class does not need to be modified.
10 UINTERFACE(MinimalAPI)
11 class UHealthInterface : public UInterface
12 {
13 GENERATED_BODY()
14 };
15
16 /**
17 *
18 */
19 class DODGEBALL_API IHealthInterface
20 {
21 GENERATED_BODY()
22
23 // Add interface functions to this class. This is the class that will be inherited to implement this interface.
24 public:
25 };
26
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
VictoryBox.cpp
100 %
문제가 검색되지 않음
줄: 1
문자: 1
탭
CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

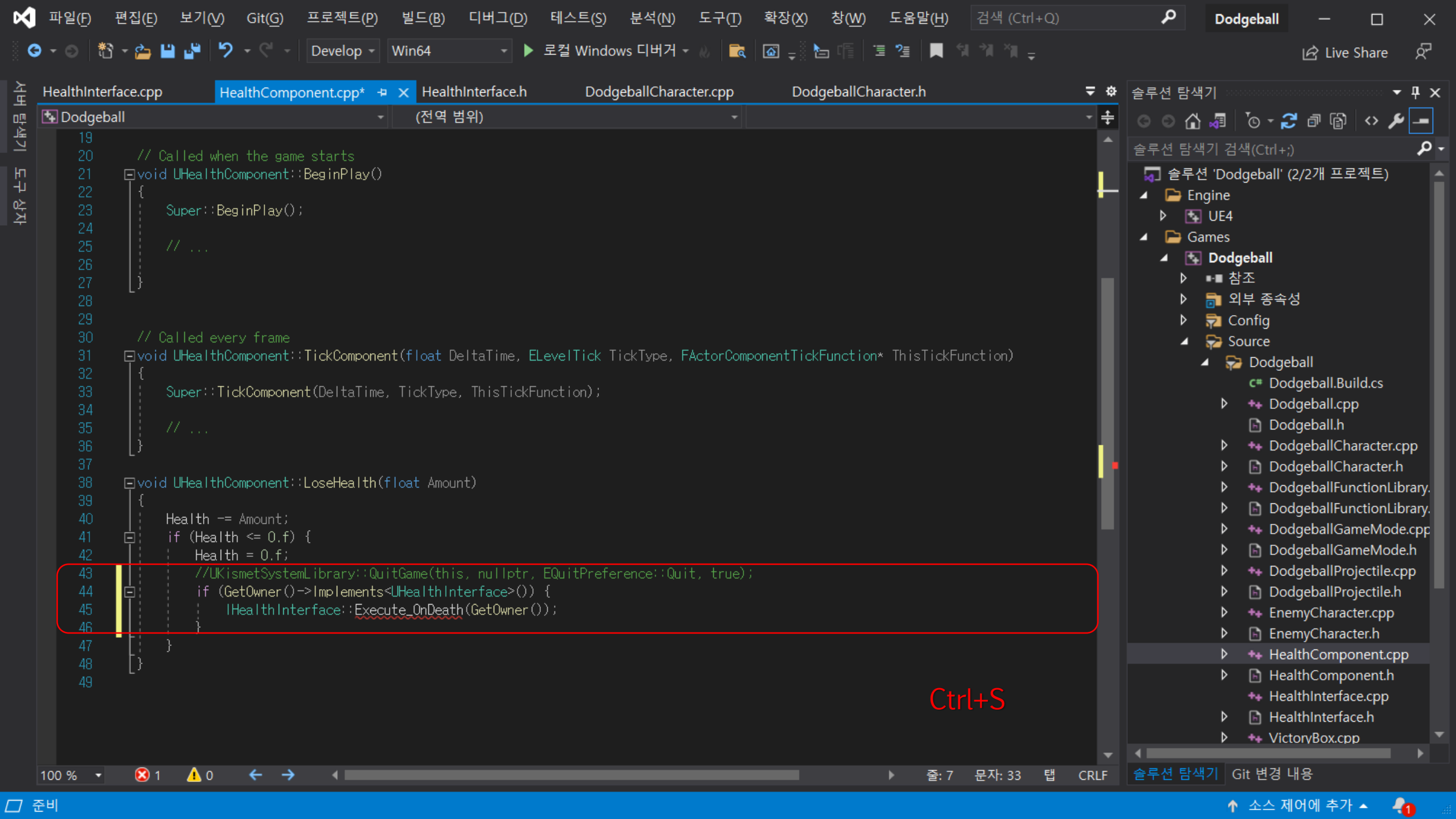

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthInterface.cpp
HealthInterface.h*
DodgeballCharacter.h
HealthComponent.cpp
HealthComponent.h
Dodgeball
IHealthInterface
OnDeath_Implementation()
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "UObject/Interface.h"
7 #include "HealthInterface.generated.h"
8
9 // This class does not need to be modified.
10 UINTERFACE(MinimalAPI)
11 class UHealthInterface : public UInterface
12 {
13 GENERATED_BODY()
14 };
15
16 /**
17 *
18 */
19 class DODGEBALL_API IHealthInterface
20 {
21 GENERATED_BODY()
22
23 // Add interface functions to this class. This is the class that will be inherited to implement this interface.
24 public:
25
26 UFUNCTION(BlueprintNativeEvent, Category = Health)
27 void OnDeath();
28 virtual void OnDeath_Implementation() = 0;
29
30 };
31
Ctrl+S
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
VictoryBox.cpp
100 %
문제가 검색되지 않음
줄: 28 문자: 44 열: 47 탭 CRLF
솔루션 탐색기 Git 변경 내용
준비
소스 제어에 추가

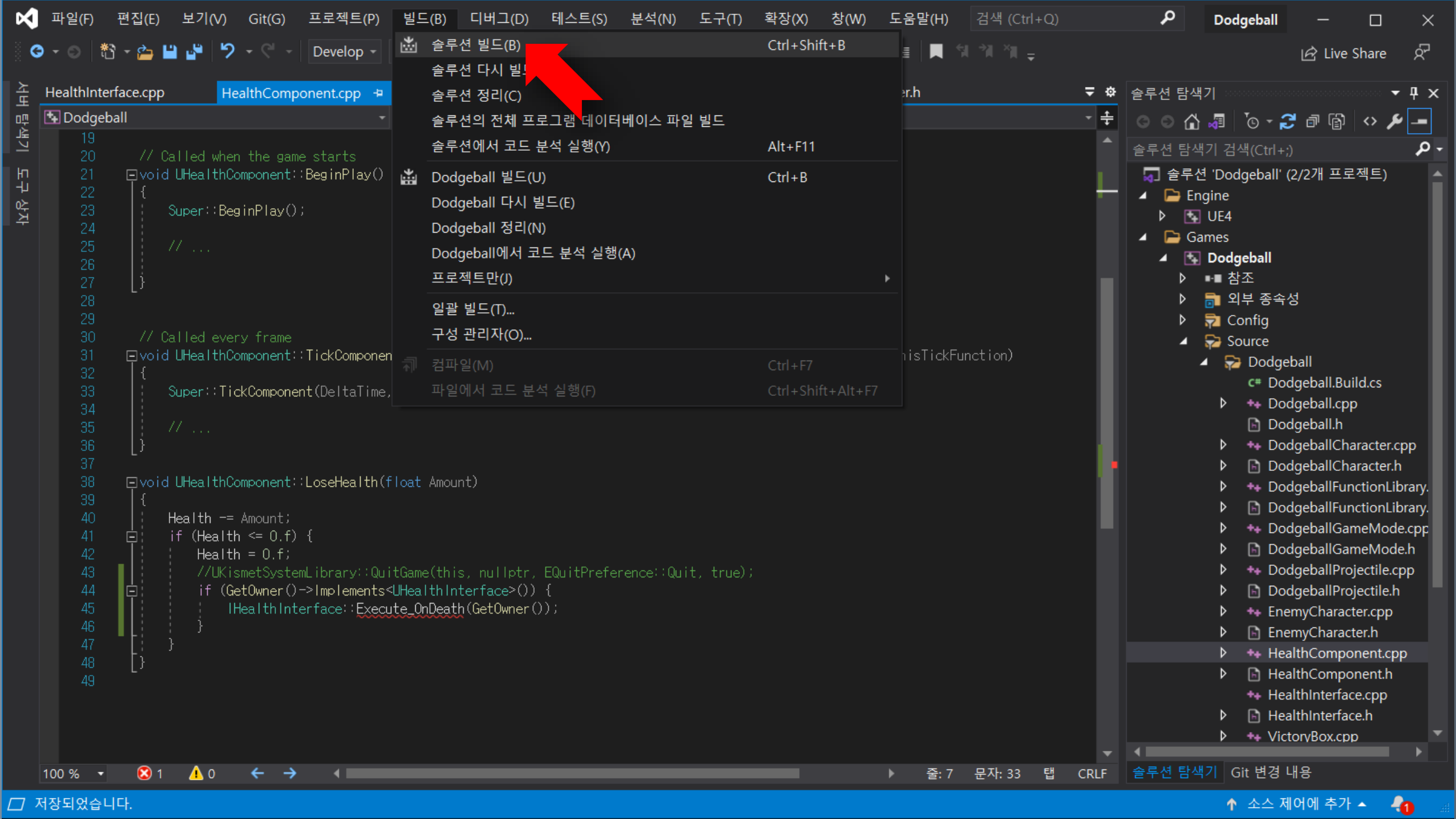

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthInterface.cpp
HealthInterface.h
DodgeballCharacter.h*
HealthComponent.cpp
HealthComponent.h
Dodgeball
ADodgeballCharacter
OnDeath_Implementation()
1 // Copyright Epic Games, Inc. All Rights Reserved.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/Character.h"
7 #include "HealthInterface.h"
8 #include "DodgeballCharacter.generated.h"
9
10 UCLASS(config=Game)
11 class ADodgeballCharacter : public ACharacter, public IHealthInterface
12 {
13 GENERATED_BODY()
14
15 /** Camera boom positioning the camera behind the character */
16 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Camera, meta = (AllowPrivateAccess = "true"))
17 class USpringArmComponent* CameraBoom;
18
19 /** Follow camera */
20 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Camera, meta = (AllowPrivateAccess = "true"))
21 class UCameraComponent* FollowCamera;
22
23 class UHealthComponent* HealthComponent;
24
25 public:
26 ADodgeballCharacter();
27
28 virtual void OnDeath_Implementation() override;
29
30 /** Base turn rate, in deg/sec. Other scaling may affect final turn rate. */
31 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category=Camera)
32 float BaseTurnRate;
33
34 /** Base look up/down rate, in deg/sec. Other scaling may affect final rate. */
35 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category=Camera)
Ctrl+S
100 %
1
0
줄: 28 문자: 49 열: 52 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
VictoryBox.cpp
솔루션 탐색기 Git 변경 내용
서버 탐색기 도구 상자
준비
소스 제어에 추가

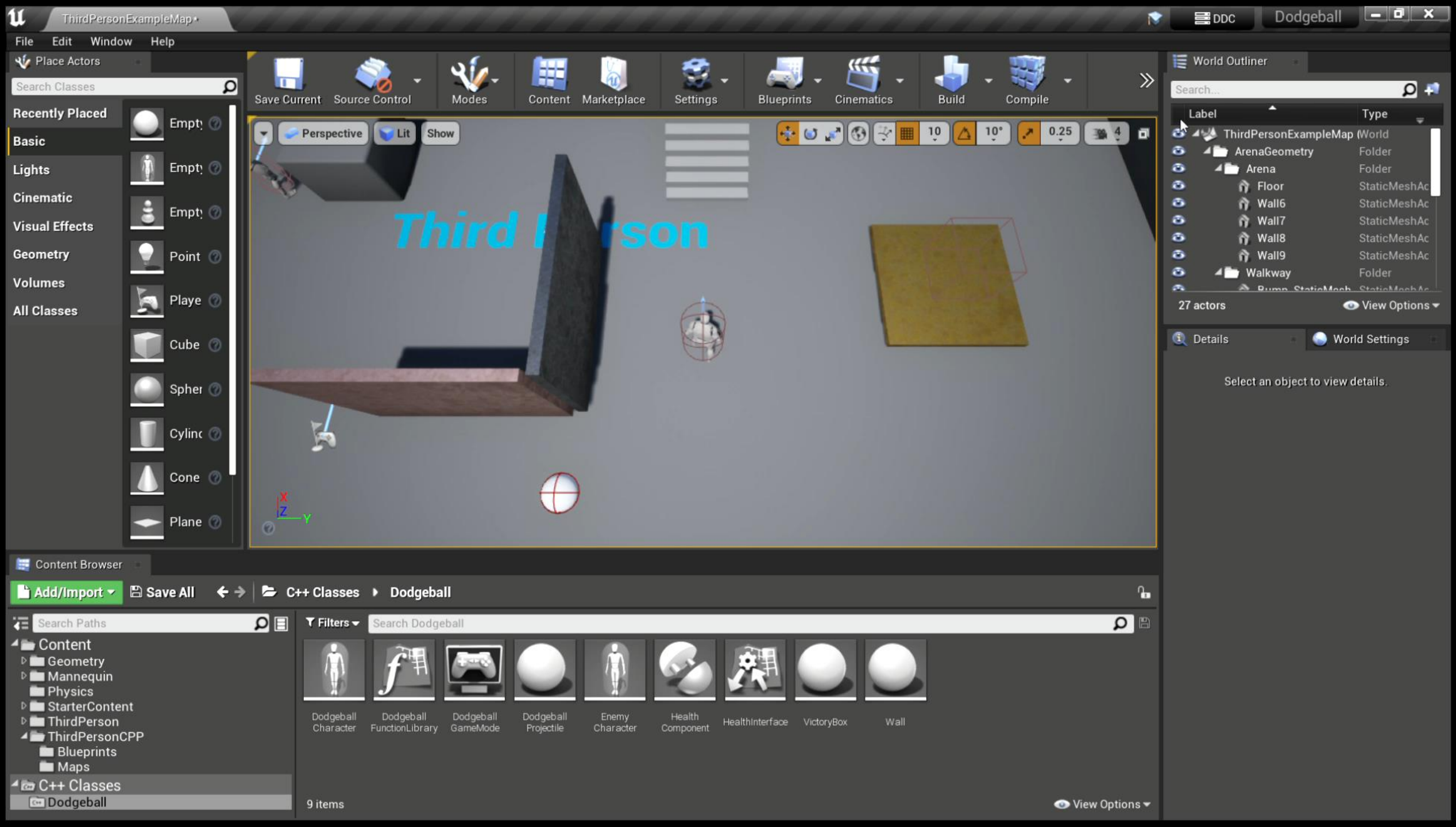

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthInterface.cpp
HealthInterface.h
DodgeballCharacter.cpp*
DodgeballCharacter.h
HealthComponent.h
Dodgeball
(전역 범위)
1 // Copyright Epic Games, Inc. All Rights Reserved.
2
3 #include "DodgeballCharacter.h"
4 #include "HeadMountedDisplayFunctionLibrary.h"
5 #include "Camera/CameraComponent.h"
6 #include "Components/CapsuleComponent.h"
7 #include "Components/InputComponent.h"
8 #include "HealthComponent.h"
9 #include "GameFramework/CharacterMovementComponent.h"
10 #include "GameFramework/Controller.h"
11 #include "GameFramework/SpringArmComponent.h"
12 #include "Kismet/KismetSystemLibrary.h"
13
14 //
15 // ADodgeballCharacter
16
17 ADodgeballCharacter::ADodgeballCharacter()
18 {
19 // Set size for collision capsule
20 GetCapsuleComponent()->InitCapsuleSize(42.f, 96.0f);
21
22 // set our turn rates for input
23 BaseTurnRate = 45.f;
24 BaseLookUpRate = 45.f;
25
26 // Don't rotate when the controller rotates. Let that just affect the camera.
27 bUseControllerRotationPitch = false;
28 bUseControllerRotationYaw = false;
29 bUseControllerRotationRoll = false;
30
31 // Configure character movement
32 GetCharacterMovement()->bOrientRotationToMovement = true; // Character moves in the direction of input...
33 GetCharacterMovement()->RotationRate = FRotator(0.0f, 540.0f, 0.0f); // ...at this rotation rate
34 GetCharacterMovement()->JumpZVelocity = 600.f;
35 GetCharacterMovement()->AirControl = 0.2f;
100 %
문제가 검색되지 않음
줄: 12
문자: 40
탭
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
VictoryBox.cpp
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthInterface.cpp
HealthInterface.h
DodgeballCharacter.cpp*
DodgeballCharacter.h
HealthComponent.h
Dodgeball
(전역 범위)
36
37 // Create a camera boom (pulls in towards the player if there is a collision)
38 CameraBoom = CreateDefaultSubobject<USpringArmComponent>(TEXT("CameraBoom"));
39 CameraBoom->SetupAttachment(RootComponent);
40 CameraBoom->TargetArmLength = 900.0f; // The camera follows at this distance behind the character
41 CameraBoom->SetRelativeRotation(FRotator(-70.f, 0.f, 0.f)); // The camera looks down at the player
42 CameraBoom->bUsePawnControlRotation = false; // Ignore pawn's pitch, yaw and roll
43 CameraBoom->bInheritPitch = false;
44 CameraBoom->bInheritYaw = false;
45 CameraBoom->bInheritRoll = false;
46
47 // Create a follow camera
48 FollowCamera = CreateDefaultSubobject<UCameraComponent>(TEXT("FollowCamera"));
49 FollowCamera->SetupAttachment(CameraBoom, USpringArmComponent::SocketName); // Attach the camera to the end of the boom and let
50 FollowCamera->bUsePawnControlRotation = false; // Camera does not rotate relative to arm
51
52 HealthComponent = CreateDefaultSubobject<UHealthComponent>(TEXT("Health Component"));
53
54 // Note: The skeletal mesh and anim blueprint references on the Mesh component (inherited from Character)
55 // are set in the derived blueprint asset named MyCharacter (to avoid direct content references in C++)
56 }
57
58 void ADodgeballCharacter::OnDeath_Implementation()
59 {
60 UKismetSystemLibrary::QuitGame(this, nullptr, EQuitPreference::Quit, true);
61 }
62
63 //
64 // Input
65
66 void ADodgeballCharacter::SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent)
67 {
68 // Set up gameplay key bindings
69 check(PlayerInputComponent);
70 //PlayerInputComponent->BindAction("Jump", IE_Pressed, this, &ACharacter::Jump);
Ctrl+S
100 %
문제가 검색되지 않음
줄: 12
문자: 40
탭
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
VictoryBox.cpp
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가

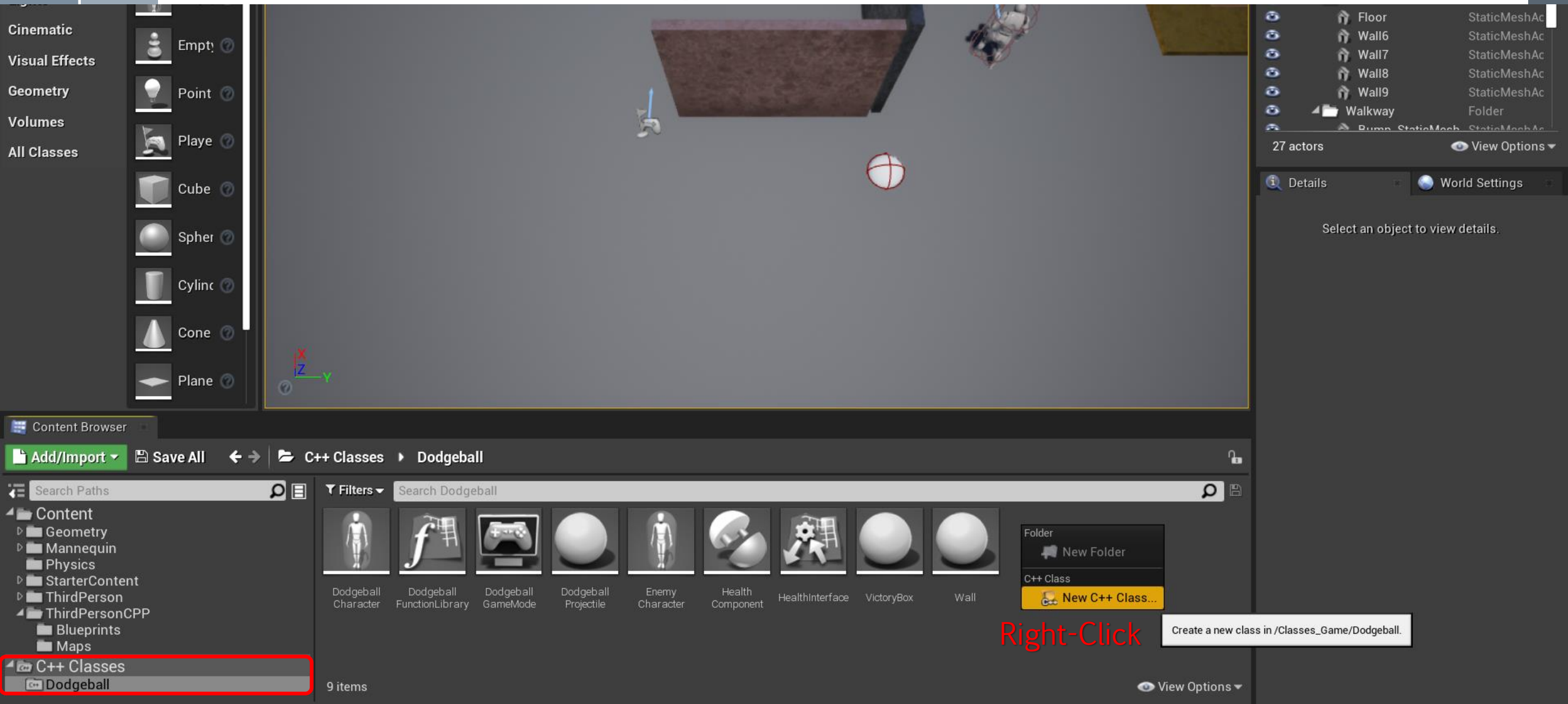

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthInterface.cpp
HealthComponent.cpp*
HealthInterface.h
DodgeballCharacter.cpp
DodgeballCharacter.h
Dodgeball
(전역 범위)
// Fill out your copyright notice in the Description page of Project Settings.
#include "HealthComponent.h"
//#include "Kismet/KismetSystemLibrary.h"
#include "HealthInterface.h"
#include "GameFramework/Actor.h"
// Sets default values for this component's properties
UHealthComponent::UHealthComponent()
{
// 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;
// ...
}
// Called when the game starts
void UHealthComponent::BeginPlay()
{
Super::BeginPlay();
// ...
}
// Called every frame
void UHealthComponent::TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction)
{
Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
// ...
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
VictoryBox.cpp
100 %
1
0
줄: 7
문자: 33
탭
CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

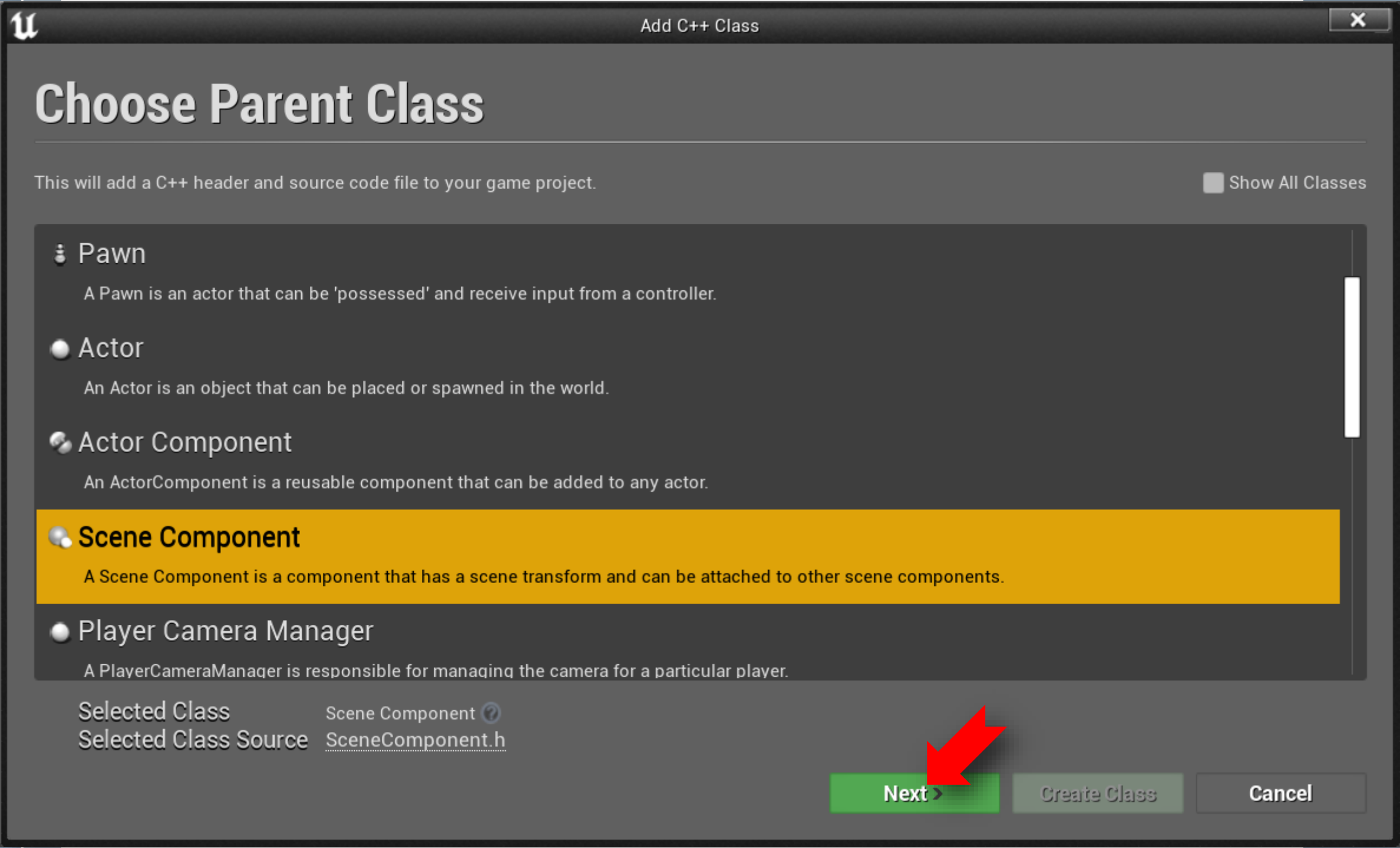

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
HealthInterface.cpp
HealthComponent.cpp*
HealthInterface.h
DodgeballCharacter.cpp
DodgeballCharacter.h
Dodgeball
(전역 범위)
19
20 // Called when the game starts
21 void UHealthComponent::BeginPlay()
22 {
23 Super::BeginPlay();
24
25 // ...
26
27 }
28
29
30 // Called every frame
31 void UHealthComponent::TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction)
32 {
33 Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
34
35 // ...
36 }
37
38 void UHealthComponent::LoseHealth(float Amount)
39 {
40 Health -= Amount;
41 if (Health <= 0.f) {
42 Health = 0.f;
43 //UKismetSystemLibrary::QuitGame(this, nullptr, EQuitPreference::Quit, true);
44 if (GetOwner()->Implements<UHealthInterface>()) {
45 IHealthInterface::Execute_OnDeath(GetOwner());
46 }
47 }
48 }
49
Ctrl+S
100 %
1
0
줄: 7
문자: 33
탭
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
VictoryBox.cpp
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가

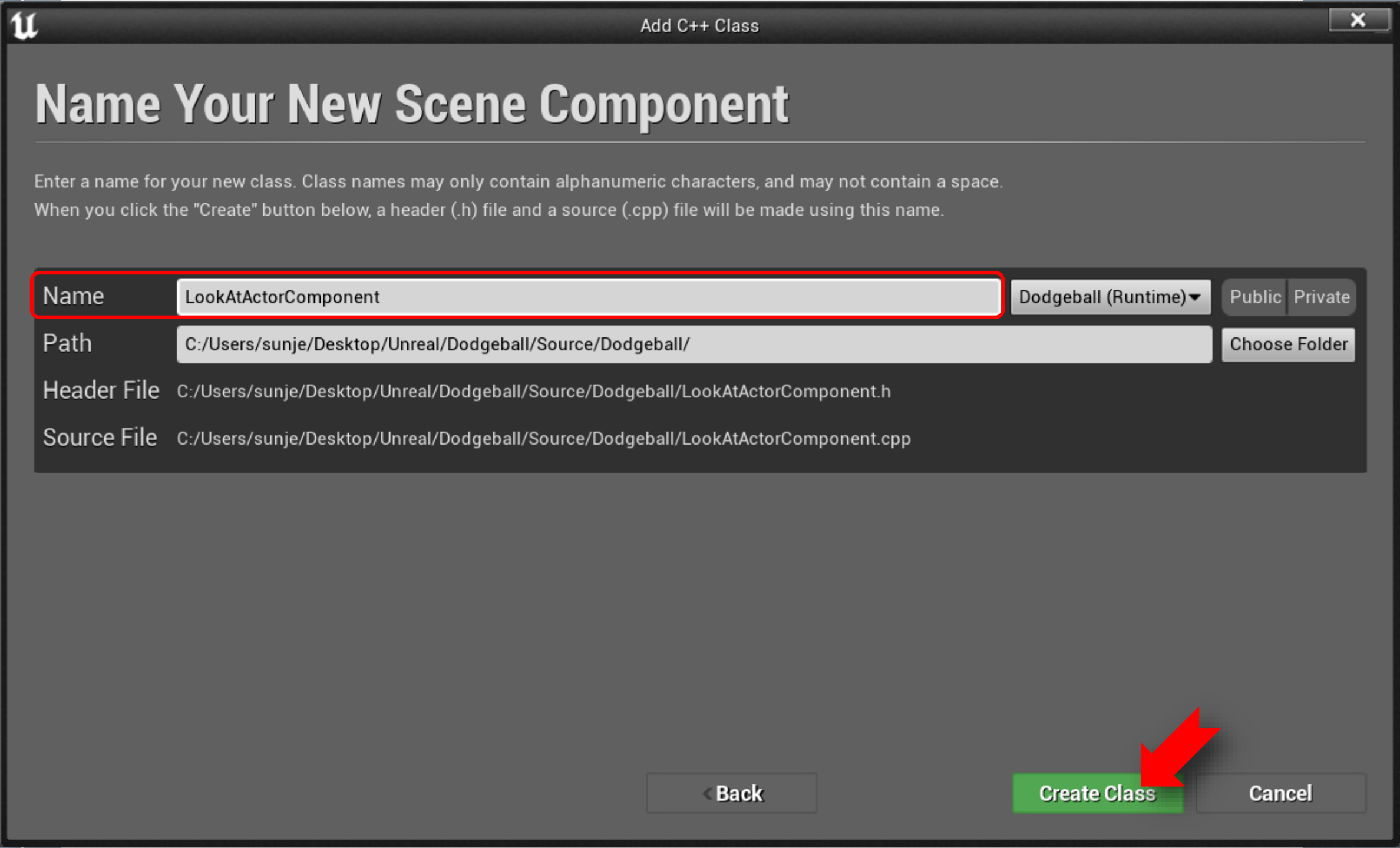

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Live Share
솔루션 빌드(B) Ctrl+Shift+B
솔루션 다시 빌드
솔루션 정리(C)
솔루션의 전체 프로그램 데이터베이스 파일 빌드
솔루션에서 코드 분석 실행(Y) Alt+F11
Dodgeball 빌드(U) Ctrl+B
Dodgeball 다시 빌드(E)
Dodgeball 정리(N)
Dodgeball에서 코드 분석 실행(A)
프로젝트만(J)
일괄 빌드(T)...
구성 관리자(O)...
컴파일(M) Ctrl+F7
파일에서 코드 분석 실행(F) Ctrl+Shift+Alt+F7
HealthInterface.cpp
HealthComponent.cpp
Dodgeball
// Called when the game starts
void UHealthComponent::BeginPlay()
{
Super::BeginPlay();
// ...
}
// Called every frame
void UHealthComponent::TickComponen
{
Super::TickComponent(DeltaTime,
// ...
}
void UHealthComponent::LoseHealth(float Amount)
{
Health -= Amount;
if (Health <= 0.f) {
Health = 0.f;
//UKismetSystemLibrary::QuitGame(this, nullptr, EQuitPreference::Quit, true);
if (GetOwner()->Implements<UHealthInterface>()) {
IHealthInterface::Execute_OnDeath(GetOwner());
}
}
}
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
VictoryBox.cpp
100 %
1
0
줄: 7
문자: 33
탭
CRLF
솔루션 탐색기
Git 변경 내용
저장되었습니다.
소스 제어에 추가

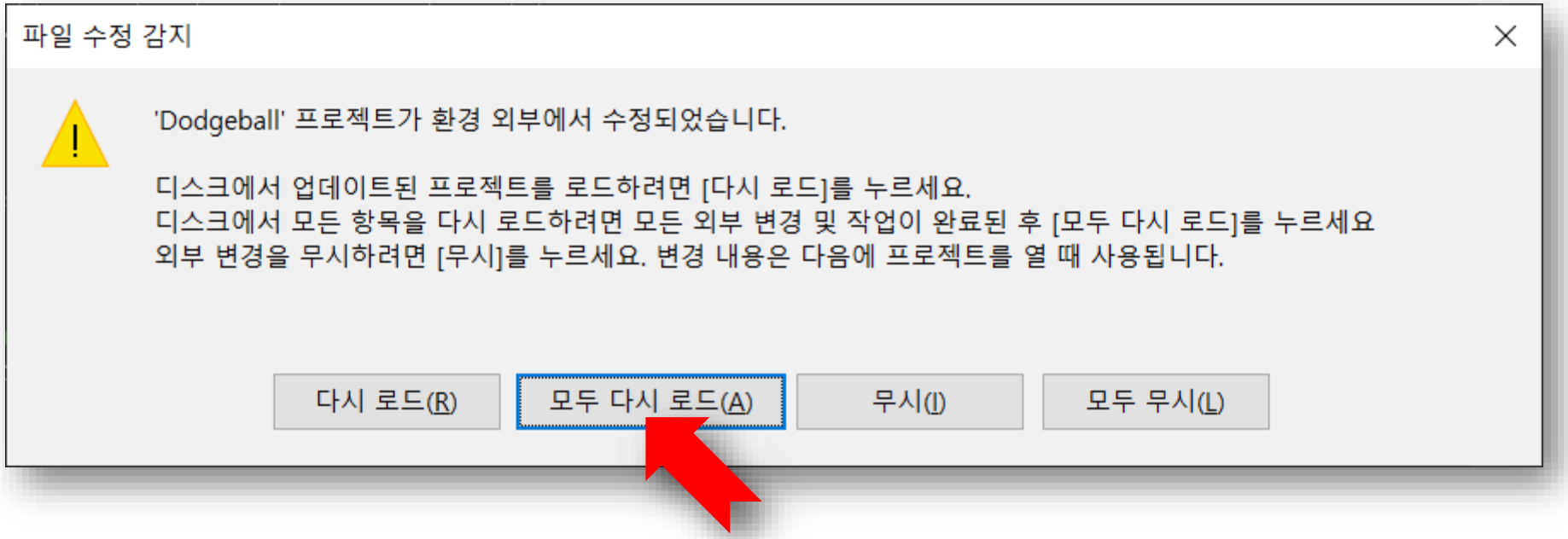

ThirdPersonExampleMap*
DDC
Dodgeball
File Edit Window Help
Place Actors
Search Classes
Recently Placed
Basic
Lights
Cinematic
Visual Effects
Geometry
Volumes
All Classes
Empty
Empty
Empty
Point
Playe
Cube
Spher
Cylinc
Cone
Plane
Save Current
Source Control
Modes
Content
Marketplace
Settings
Blueprints
Cinematics
Build
Compile
Perspective
Lit
Show
Third Person
World Outliner
Search...
Label
Type
ThirdPersonExampleMap (World
ArenaGeometry Folder
Arena Folder
Floor StaticMeshAc
Wall6 StaticMeshAc
Wall7 StaticMeshAc
Wall8 StaticMeshAc
Wall9 StaticMeshAc
Walkway Folder
27 actors
View Options
Details
World Settings
Select an object to view details.
Content Browser
Add/Import
Save All
C++ Classes
Dodgeball
Search Paths
Filters
Search Dodgeball
Content
Geometry
Mannequin
Physics
StarterContent
ThirdPerson
ThirdPersonCPP
Blueprints
Maps
C++ Classes
Dodgeball
Dodgeball Character
Dodgeball FunctionLibrary
Dodgeball GameMode
Dodgeball Projectile
Enemy Character
Health Component
HealthInterface
VictoryBox
Wall
9 items
View Options

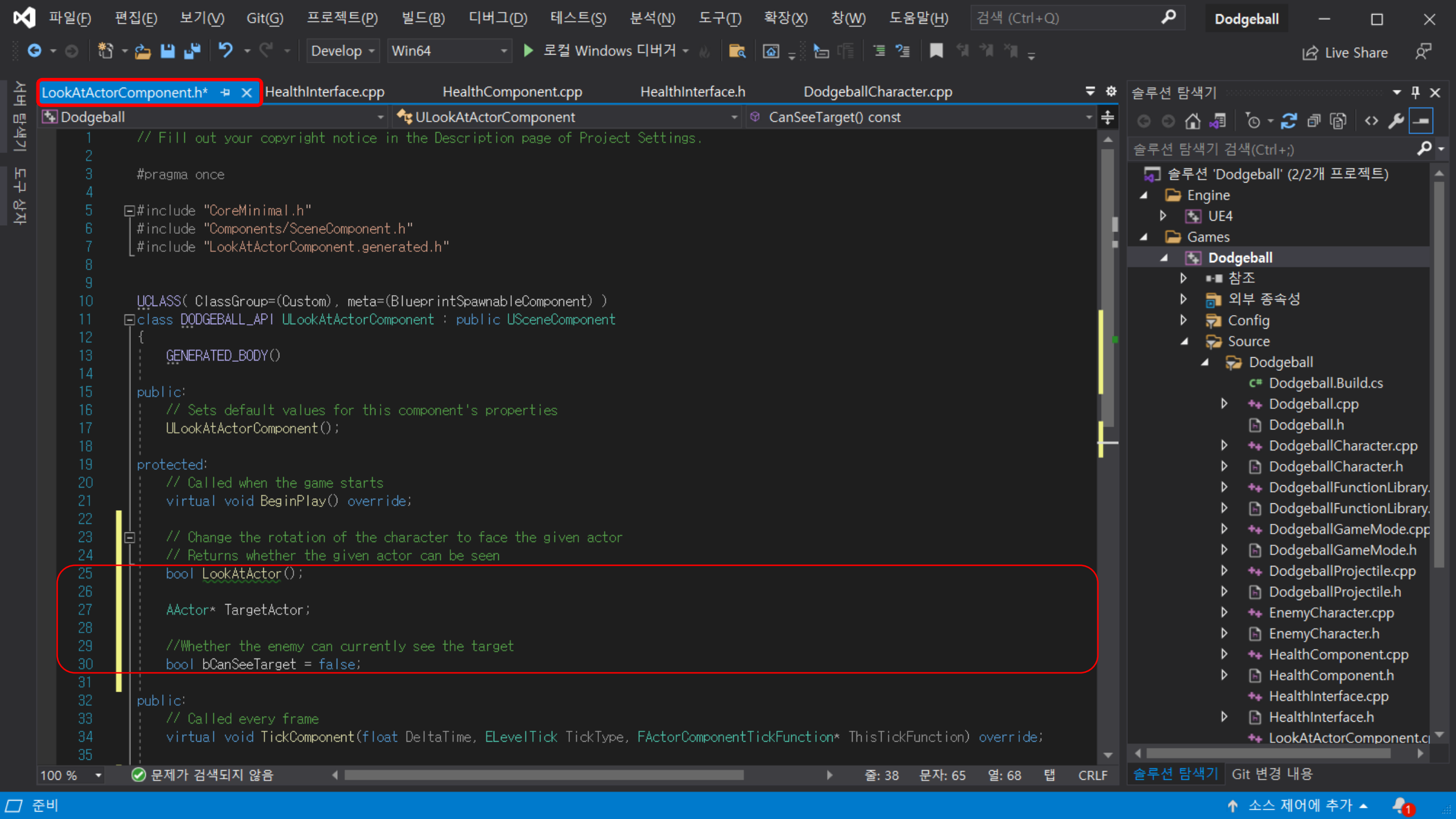
Activity 7.01: Moving The LookAtActor Logic To An Actor Component

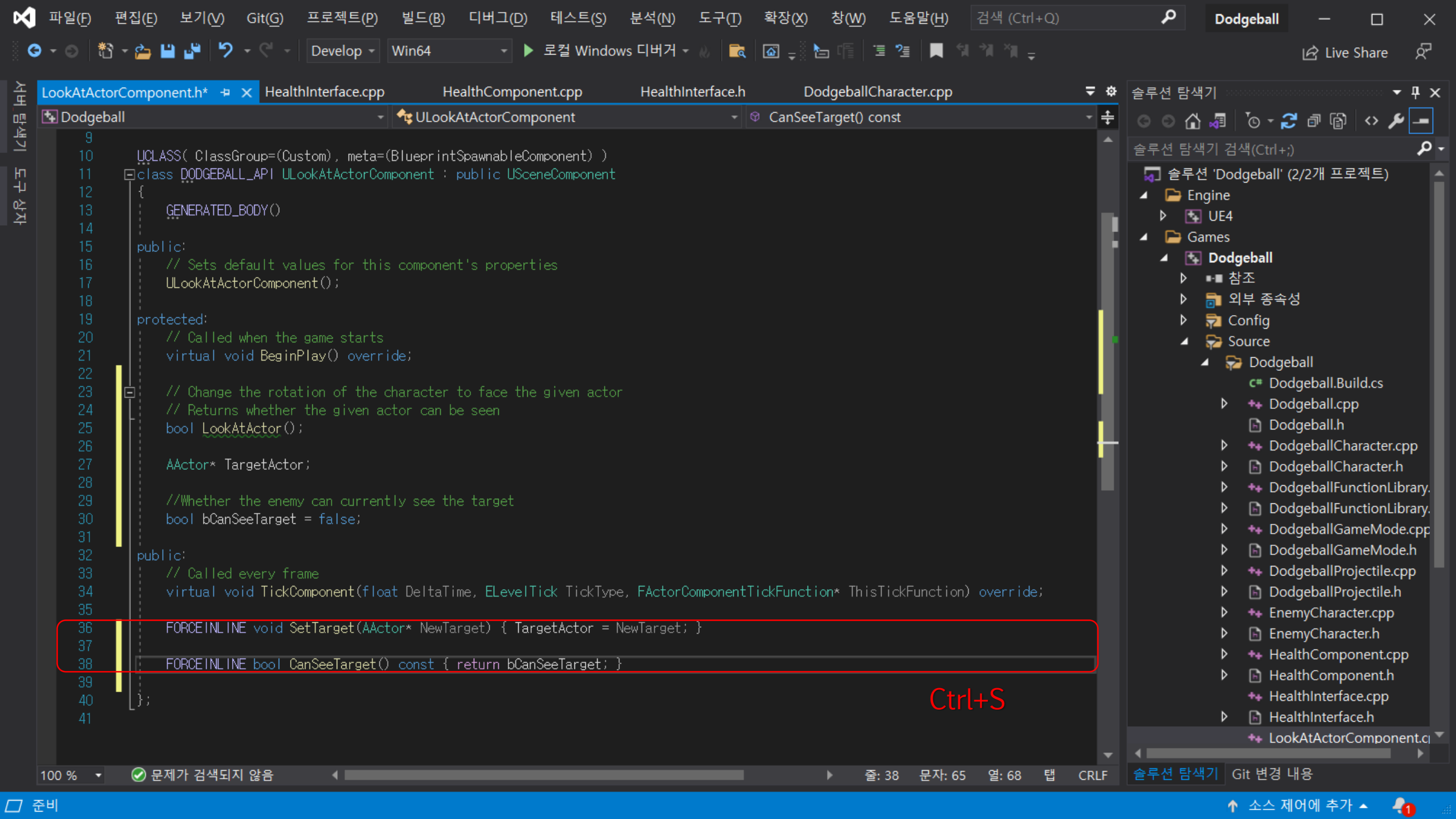


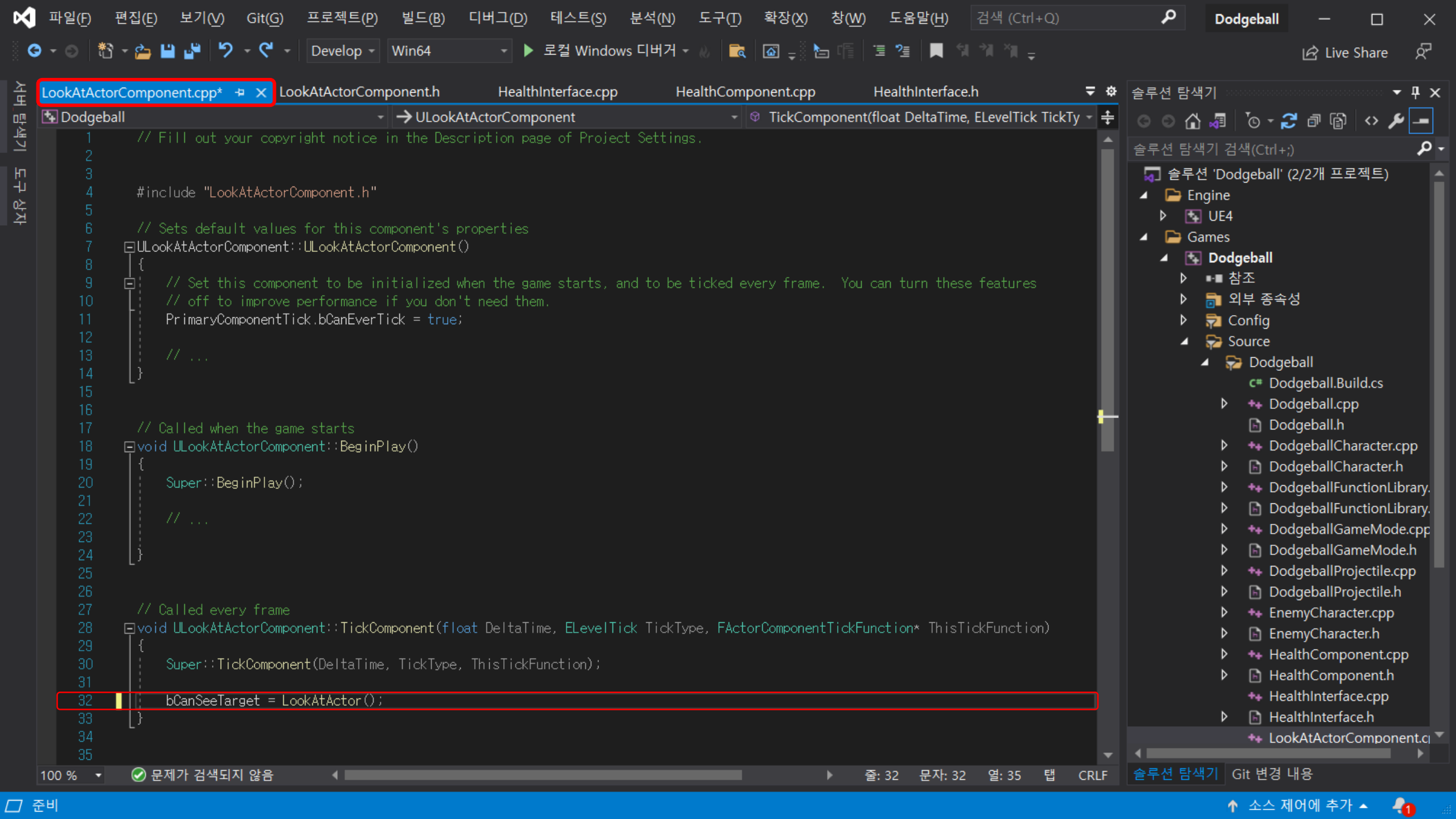

Add C++ Class
Choose Parent Class
This will add a C++ header and source code file to your game project.
Show All Classes
Pawn
A Pawn is an actor that can be 'possessed' and receive input from a controller.
Actor
An Actor is an object that can be placed or spawned in the world.
Actor Component
An ActorComponent is a reusable component that can be added to any actor.
Scene Component
A Scene Component is a component that has a scene transform and can be attached to other scene components.
Player Camera Manager
A PlayerCameraManager is responsible for managing the camera for a particular player.
Selected Class
Scene Component
Selected Class Source
SceneComponent.h
Next >
Create Class
Cancel

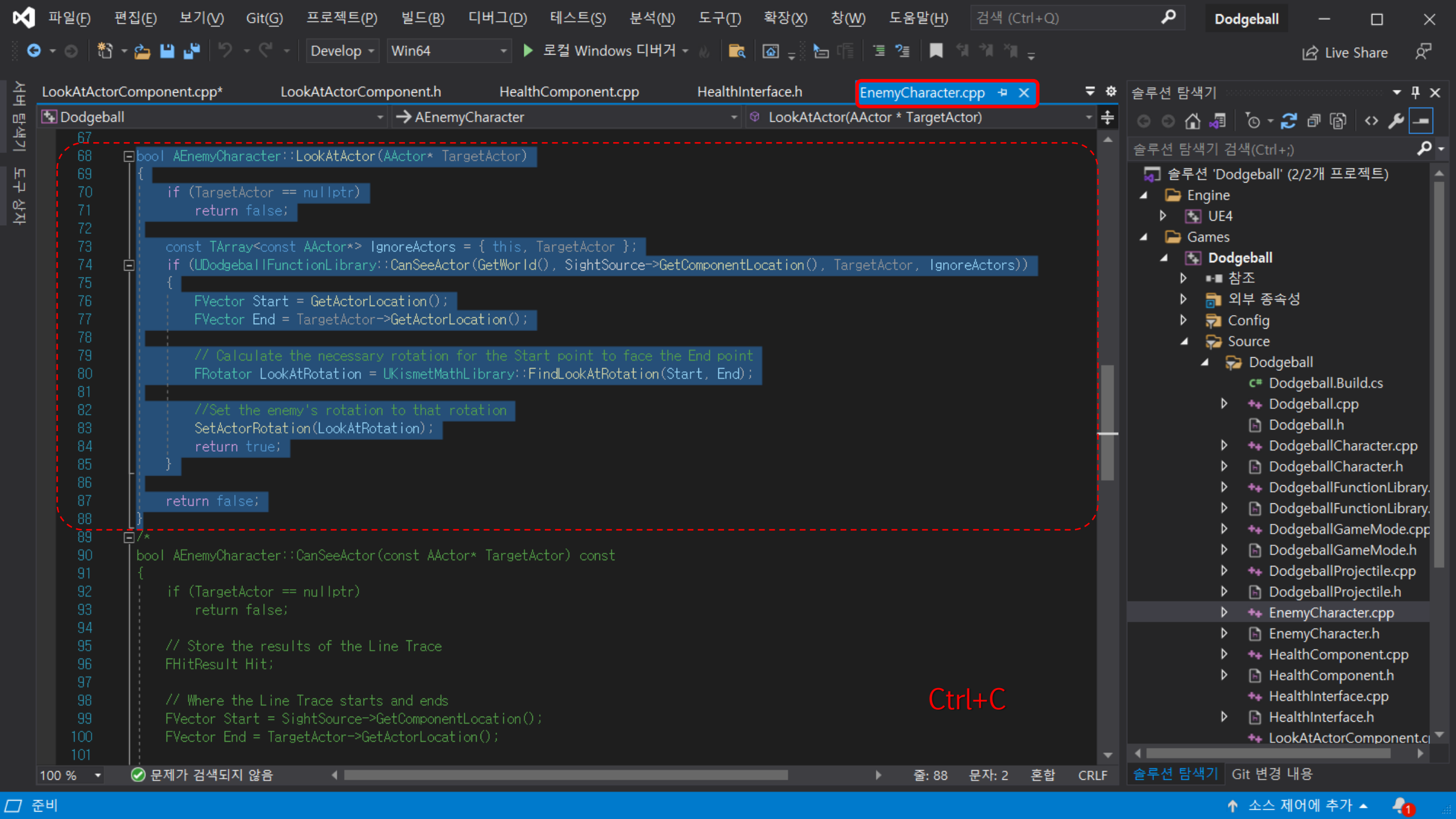

Add C++ Class
Name Your New Scene Component
Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.
When you click the "Create" button below, a header (.h) file and a source (.cpp) file will be made using this name.
Name
LookAtActorComponent
Dodgeball (Runtime)
Public
Private
Path
C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/
Choose Folder
Header File
C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/LookAtActorComponent.h
Source File
C:/Users/sunje/Desktop/Unreal/Dodgeball/Source/Dodgeball/LookAtActorComponent.cpp
Back
Create Class
Cancel

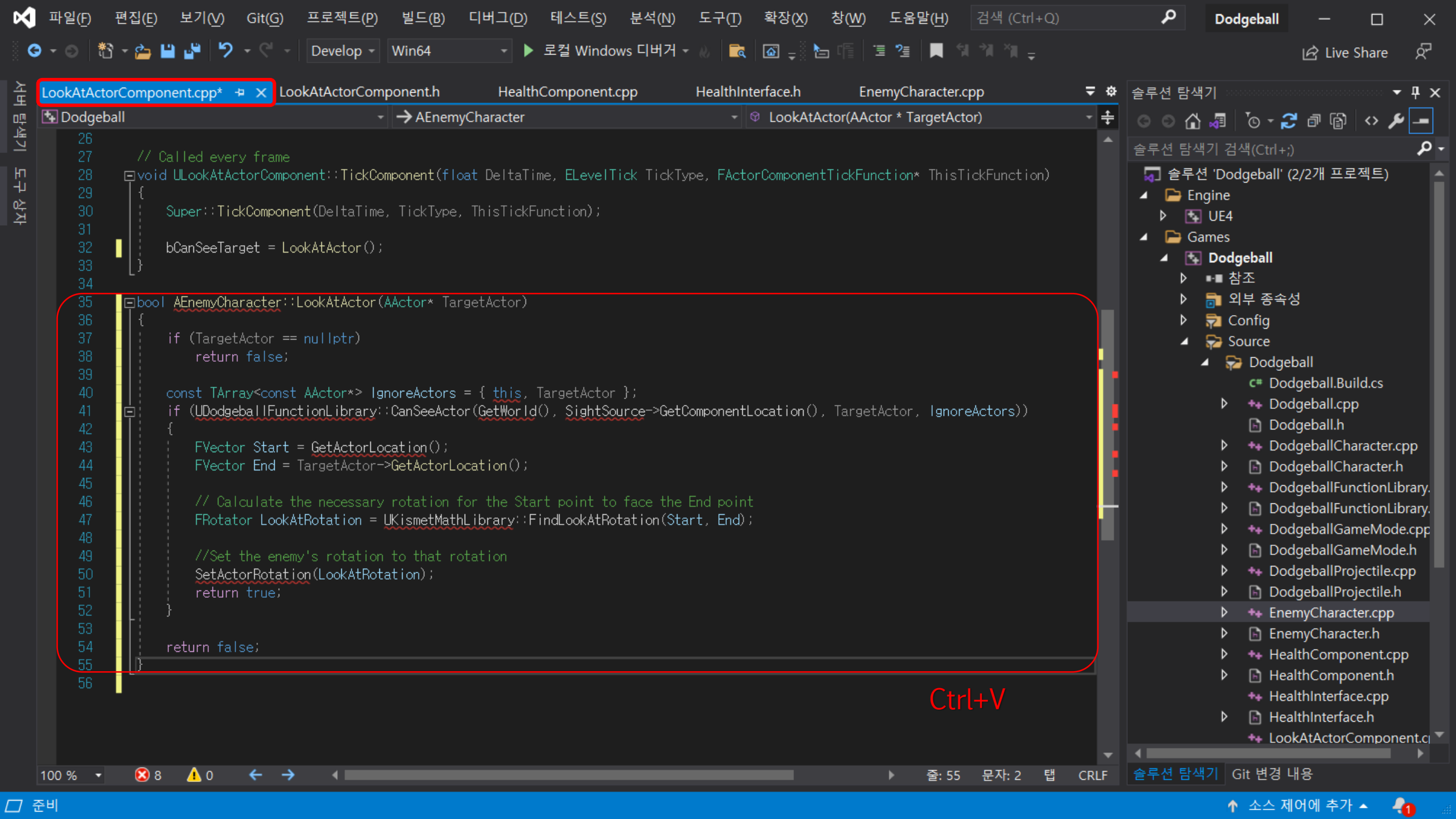

파일 수정 감지
'Dodgeball' 프로젝트가 환경 외부에서 수정되었습니다.
디스크에서 업데이트된 프로젝트를 로드하려면 [다시 로드]를 누르세요.
디스크에서 모든 항목을 다시 로드하려면 모든 외부 변경 및 작업이 완료된 후 [모두 다시 로드]를 누르세요
외부 변경을 무시하려면 [무시]를 누르세요. 변경 내용은 다음에 프로젝트를 열 때 사용됩니다.
다시 로드(R)
모두 다시 로드(A)
무시(I)
모두 무시(L)

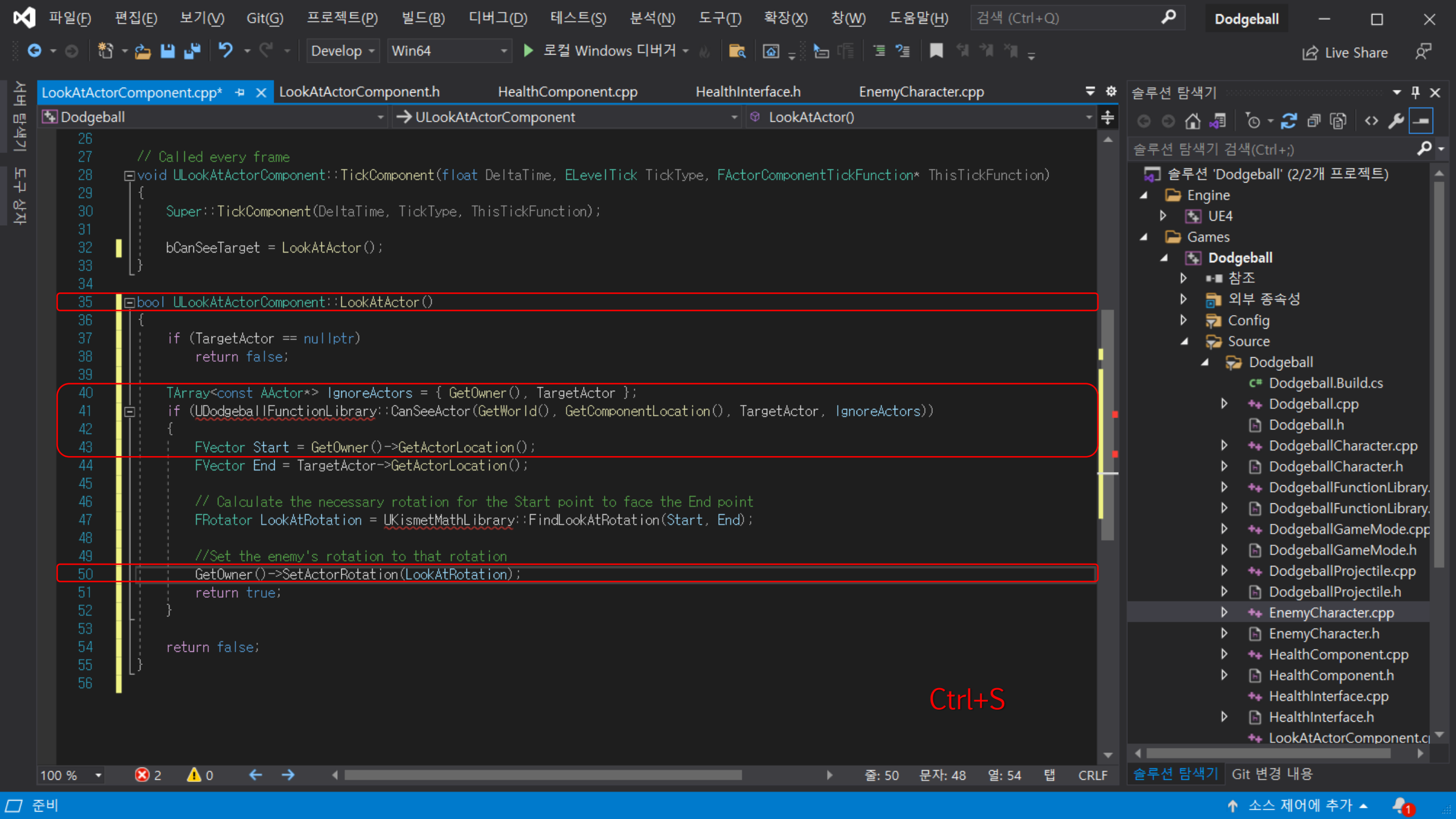

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.h*
HealthInterface.cpp
HealthComponent.cpp
HealthInterface.h
DodgeballCharacter.cpp
Dodgeball
ULookAtActorComponent
CanSeeTarget() const
// Fill out your copyright notice in the Description page of Project Settings.
#pragma once
#include "CoreMinimal.h"
#include "Components/SceneComponent.h"
#include "LookAtActorComponent.generated.h"
UCLASS(ClassGroup=(Custom), meta=(BlueprintSpawnableComponent))
class DODGEBALL_API ULookAtActorComponent : public USceneComponent
{
GENERATED_BODY()
public:
// Sets default values for this component's properties
ULookAtActorComponent();
protected:
// Called when the game starts
virtual void BeginPlay() override;
// Change the rotation of the character to face the given actor
// Returns whether the given actor can be seen
bool LookAtActor();
AActor* TargetActor;
//Whether the enemy can currently see the target
bool bCanSeeTarget = false;
public:
// Called every frame
virtual void TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction) override;
100 %
문제가 검색되지 않음
줄: 38 문자: 65 열: 68 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.cp
솔루션 탐색기
Git 변경 내용
서버 탐색기
도구 상자
준비
소스 제어에 추가

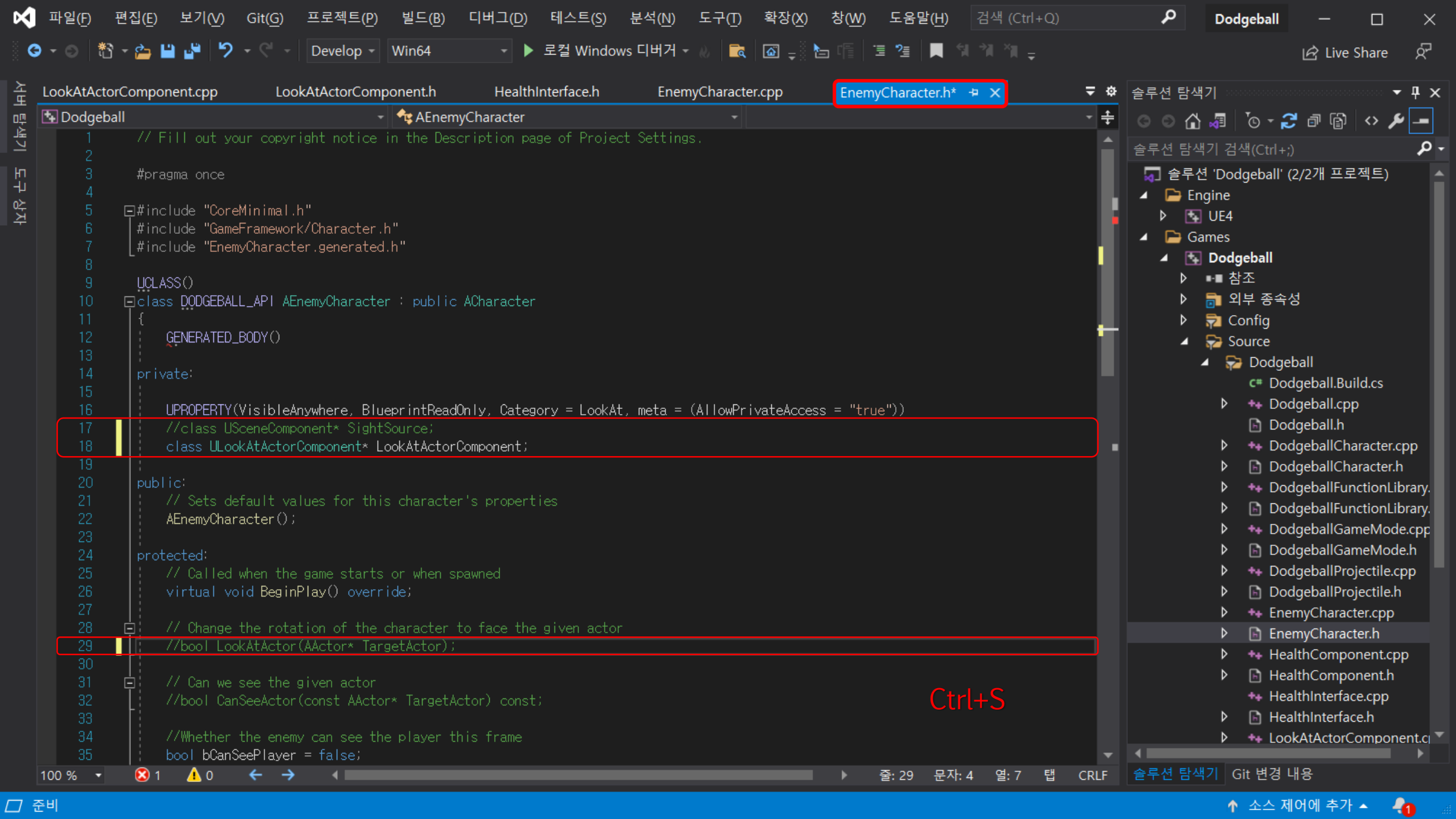

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.h*
HealthInterface.cpp
HealthComponent.cpp
HealthInterface.h
DodgeballCharacter.cpp
Dodgeball
ULookAtActorComponent
CanSeeTarget() const
UCLASS(ClassGroup=(Custom), meta=(BlueprintSpawnableComponent))
class DODGEBALL_API ULookAtActorComponent : public USceneComponent
{
GENERATED_BODY()
public:
// Sets default values for this component's properties
ULookAtActorComponent();
protected:
// Called when the game starts
virtual void BeginPlay() override;
// Change the rotation of the character to face the given actor
// Returns whether the given actor can be seen
bool LookAtActor();
AActor* TargetActor;
//Whether the enemy can currently see the target
bool bCanSeeTarget = false;
public:
// Called every frame
virtual void TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction) override;
FORCEINLINE void SetTarget(AActor* NewTarget) { TargetActor = NewTarget; }
FORCEINLINE bool CanSeeTarget() const { return bCanSeeTarget; }
};
Ctrl+S
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.c
100 %
문제가 검색되지 않음
줄: 38 문자: 65 열: 68 탭 CRLF
솔루션 탐색기 Git 변경 내용
준비
소스 제어에 추가

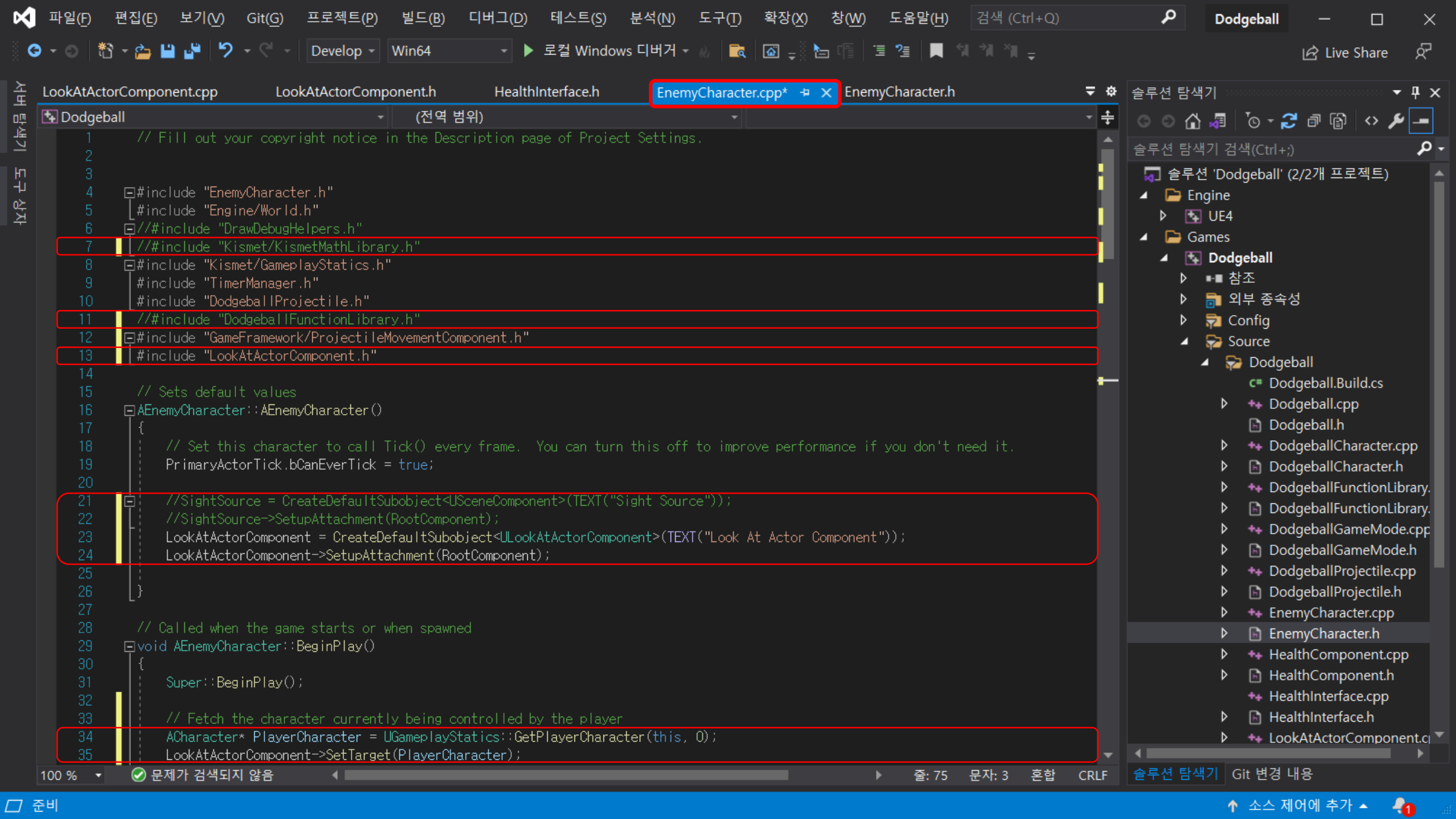

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.cpp*
LookAtActorComponent.h
HealthInterface.cpp
HealthComponent.cpp
HealthInterface.h
Dodgeball
ULookAtActorComponent
TickComponent(float DeltaTime, ELevelTick TickTy
// Fill out your copyright notice in the Description page of Project Settings.
#include "LookAtActorComponent.h"
// Sets default values for this component's properties
ULookAtActorComponent::ULookAtActorComponent()
{
// 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 = true;
// ...
}
// Called when the game starts
void ULookAtActorComponent::BeginPlay()
{
Super::BeginPlay();
// ...
}
// Called every frame
void ULookAtActorComponent::TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction)
{
Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
bCanSeeTarget = LookAtActor();
}
100 %
문제가 검색되지 않음
줄: 32 문자: 32 열: 35 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.cp
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

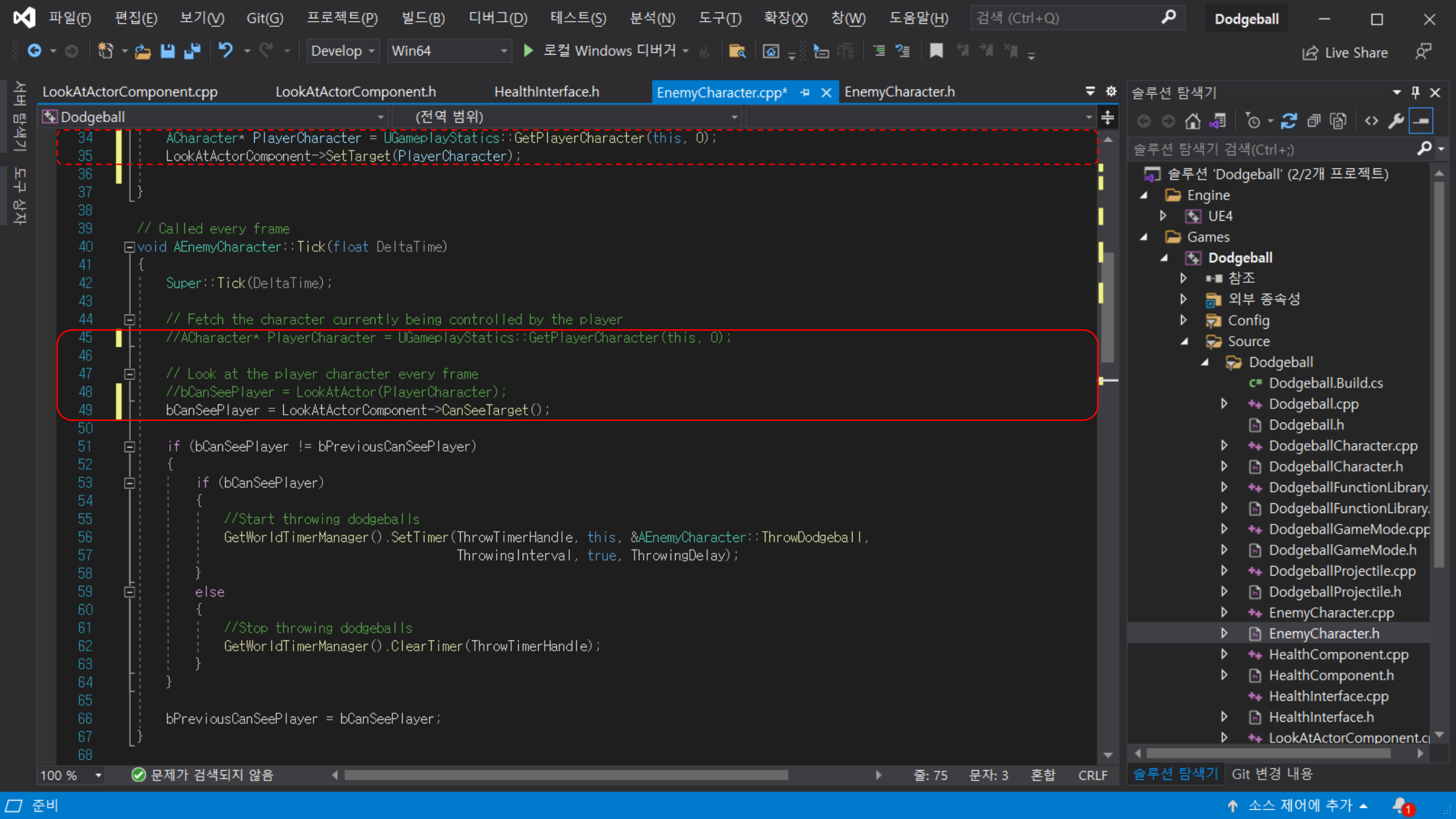

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.cpp*
LookAtActorComponent.h
HealthComponent.cpp
HealthInterface.h
EnemyCharacter.cpp
Dodgeball
AEnemyCharacter
LookAtActor(AActor * TargetActor)
bool AEnemyCharacter::LookAtActor(AActor* TargetActor)
{
if (TargetActor == nullptr)
return false;
const TArray<const AActor*> IgnoreActors = { this, TargetActor };
if (UDodgeballFunctionLibrary::CanSeeActor(GetWorld(), SightSource->GetComponentLocation(), TargetActor, IgnoreActors))
{
FVector Start = GetActorLocation();
FVector End = TargetActor->GetActorLocation();
// Calculate the necessary rotation for the Start point to face the End point
FRotator LookAtRotation = UKismetMathLibrary::FindLookAtRotation(Start, End);
//Set the enemy's rotation to that rotation
SetActorRotation(LookAtRotation);
return true;
}
return false;
}
/*
bool AEnemyCharacter::CanSeeActor(const AActor* TargetActor) const
{
if (TargetActor == nullptr)
return false;
// Store the results of the Line Trace
FHitResult Hit;
// Where the Line Trace starts and ends
FVector Start = SightSource->GetComponentLocation();
FVector End = TargetActor->GetActorLocation();
Ctrl+C
100 %
문제가 검색되지 않음
줄: 88
문자: 2
혼합
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.c
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

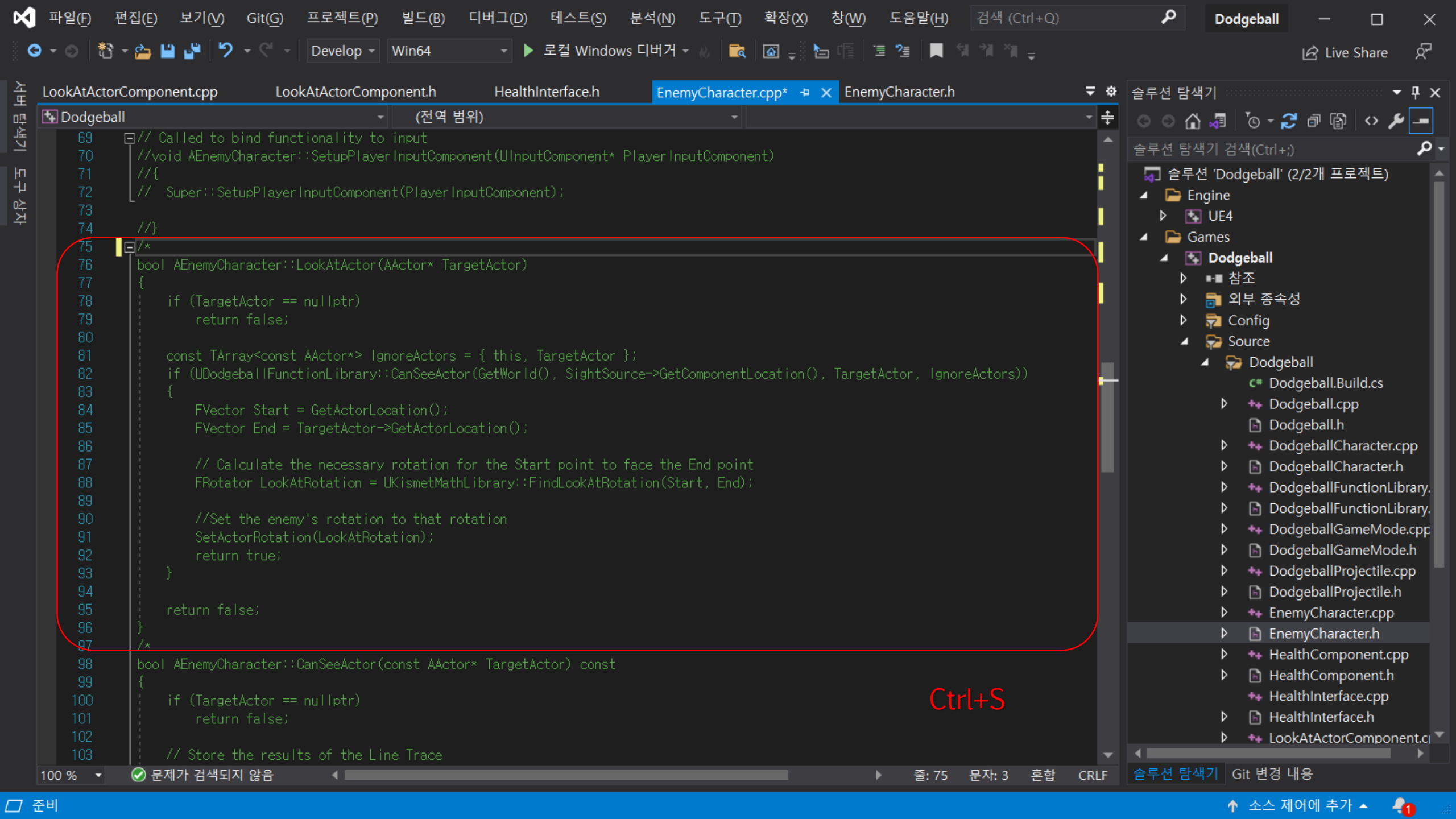

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.cpp*
LookAtActorComponent.h
HealthComponent.cpp
HealthInterface.h
EnemyCharacter.cpp
Dodgeball
AEnemyCharacter
LookAtActor(AActor * TargetActor)
26
27 // Called every frame
28 void ULookAtActorComponent::TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction)
29 {
30 Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
31
32 bCanSeeTarget = LookAtActor();
33 }
34
35 bool AEnemyCharacter::LookAtActor(AActor* TargetActor)
36 {
37 if (TargetActor == nullptr)
38 return false;
39
40 const TArray<const AActor*> IgnoreActors = { this, TargetActor };
41 if (UDodgeballFunctionLibrary::CanSeeActor(GetWorld(), SightSource->GetComponentLocation(), TargetActor, IgnoreActors))
42 {
43 FVector Start = GetActorLocation();
44 FVector End = TargetActor->GetActorLocation();
45
46 // Calculate the necessary rotation for the Start point to face the End point
47 FRotator LookAtRotation = UKismetMathLibrary::FindLookAtRotation(Start, End);
48
49 //Set the enemy's rotation to that rotation
50 SetActorRotation(LookAtRotation);
51 return true;
52 }
53
54 return false;
55 }
56
Ctrl+V
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.c
100 %
8
0
줄: 55
문자: 2
탭
CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가
서버 탐색기
도구 상자

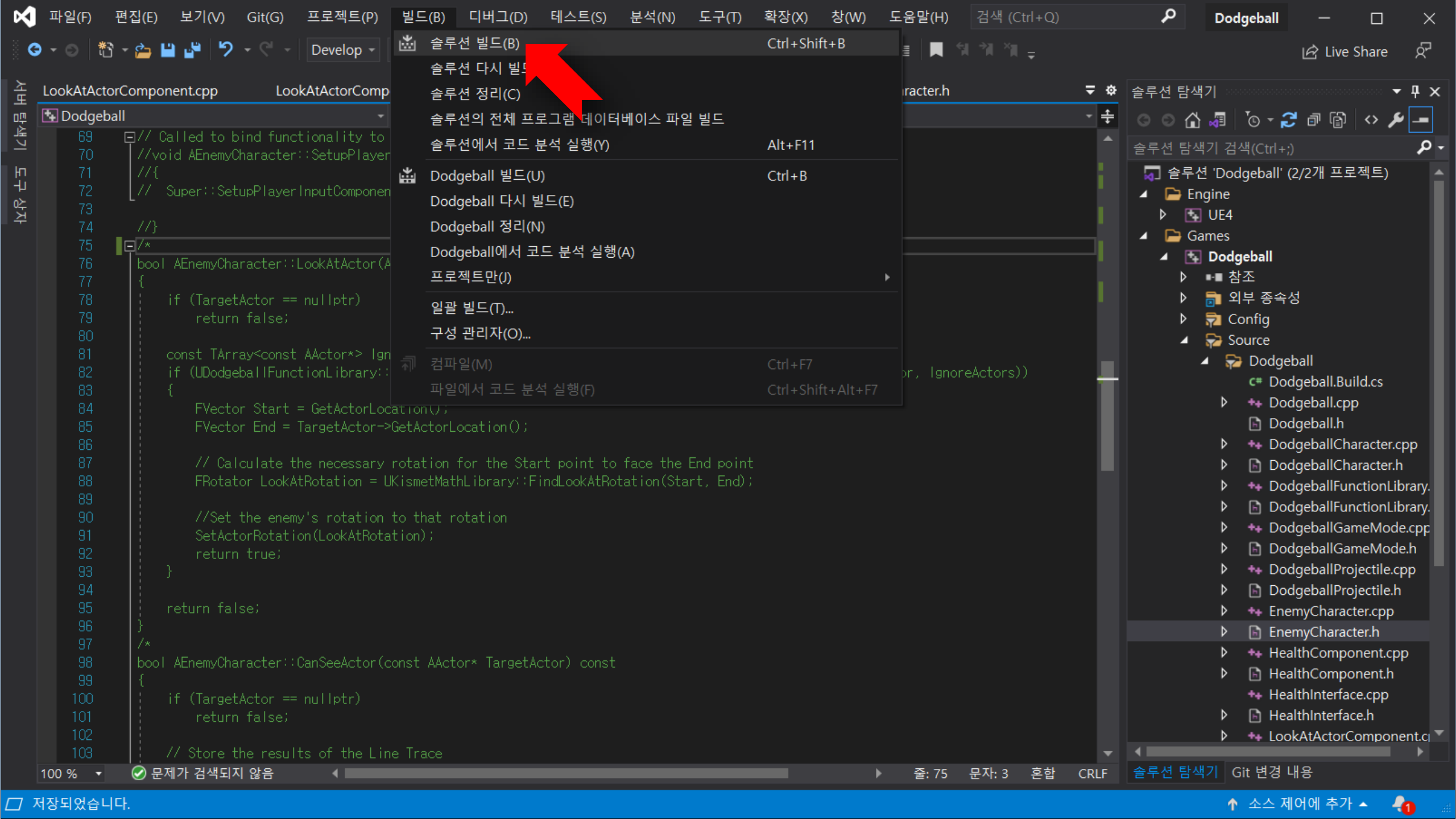

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.cpp*
LookAtActorComponent.h
HealthComponent.cpp
HealthInterface.h
EnemyCharacter.cpp
Dodgeball
ULookAtActorComponent
LookAtActor()
// Called every frame
void ULookAtActorComponent::TickComponent(float DeltaTime, ELevelTick TickType, FActorComponentTickFunction* ThisTickFunction)
{
Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
bCanSeeTarget = LookAtActor();
}
bool ULookAtActorComponent::LookAtActor()
{
if (TargetActor == nullptr)
return false;
TArray<const AActor*> IgnoreActors = { GetOwner(), TargetActor };
if (UDodgeballFunctionLibrary::CanSeeActor(GetWorld(), GetComponentLocation(), TargetActor, IgnoreActors))
{
FVector Start = GetOwner()->GetActorLocation();
FVector End = TargetActor->GetActorLocation();
// Calculate the necessary rotation for the Start point to face the End point
FRotator LookAtRotation = UKismetMathLibrary::FindLookAtRotation(Start, End);
//Set the enemy's rotation to that rotation
GetOwner()->SetActorRotation(LookAtRotation);
return true;
}
return false;
}
Ctrl+S
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.cp
100 %
2
0
줄: 50
문자: 48
열: 54
탭
CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가

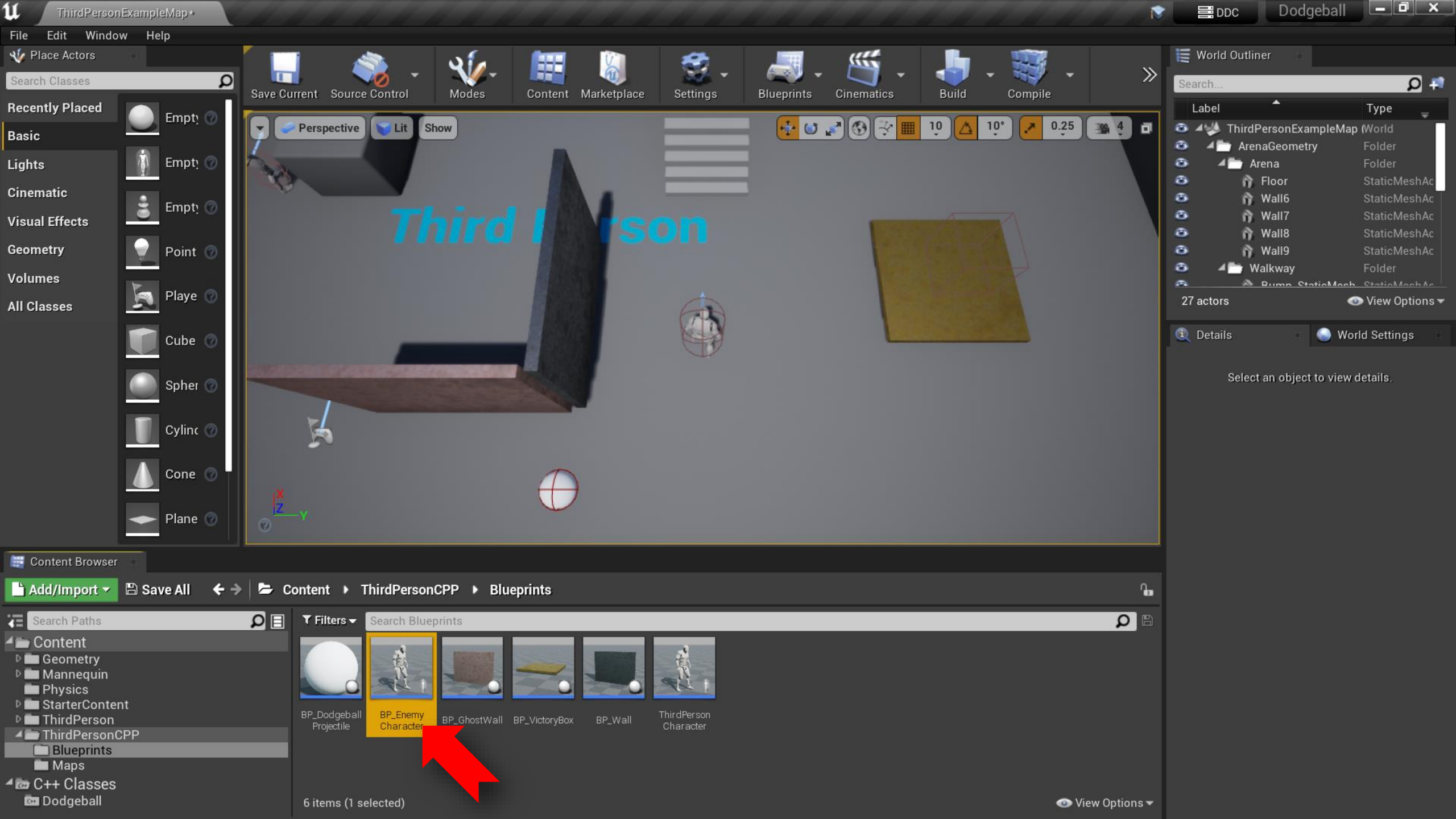

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.cpp
LookAtActorComponent.h
HealthInterface.h
EnemyCharacter.cpp
EnemyCharacter.h*
Dodgeball
AEnemyCharacter
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/Character.h"
7 #include "EnemyCharacter.generated.h"
8
9 UCLASS()
10 class DODGEBALL_API AEnemyCharacter : public ACharacter
11 {
12 GENERATED_BODY()
13
14 private:
15
16 UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = LookAt, meta = (AllowPrivateAccess = "true"))
17 //class USceneComponent* SightSource;
18 class ULookAtActorComponent* LookAtActorComponent;
19
20 public:
21 // Sets default values for this character's properties
22 AEnemyCharacter();
23
24 protected:
25 // Called when the game starts or when spawned
26 virtual void BeginPlay() override;
27
28 // Change the rotation of the character to face the given actor
29 //bool LookAtActor(AActor* TargetActor);
30
31 // Can we see the given actor
32 //bool CanSeeActor(const AActor* TargetActor) const;
33
34 //Whether the enemy can see the player this frame
35 bool bCanSeePlayer = false;
Ctrl+S
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.cp
100 %
1
0
줄: 29 문자: 4 열: 7 탭 CRLF
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가
서버 탐색기
도구 상자


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.cpp
LookAtActorComponent.h
HealthInterface.h
EnemyCharacter.cpp*
EnemyCharacter.h
Dodgeball
(전역 범위)
// Fill out your copyright notice in the Description page of Project Settings.
#include "EnemyCharacter.h"
#include "Engine/World.h"
//#include "DrawDebugHelpers.h"
//#include "Kismet/KismetMathLibrary.h"
#include "Kismet/GameplayStatics.h"
#include "TimerManager.h"
#include "DodgeballProjectile.h"
//#include "DodgeballFunctionLibrary.h"
#include "GameFramework/ProjectileMovementComponent.h"
#include "LookAtActorComponent.h"
// Sets default values
AEnemyCharacter::AEnemyCharacter()
{
// Set this character to call Tick() every frame. You can turn this off to improve performance if you don't need it.
PrimaryActorTick.bCanEverTick = true;
//SightSource = CreateDefaultSubobject<USceneComponent>(TEXT("Sight Source"));
//SightSource->SetupAttachment(RootComponent);
LookAtActorComponent = CreateDefaultSubobject<ULookAtActorComponent>(TEXT("Look At Actor Component"));
LookAtActorComponent->SetupAttachment(RootComponent);
}
// Called when the game starts or when spawned
void AEnemyCharacter::BeginPlay()
{
Super::BeginPlay();
// Fetch the character currently being controlled by the player
ACharacter* PlayerCharacter = UGameplayStatics::GetPlayerCharacter(this, 0);
LookAtActorComponent->SetTarget(PlayerCharacter);
100 %
문제가 검색되지 않음
줄: 75
문자: 3
혼합
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.c
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.cpp
LookAtActorComponent.h
HealthInterface.h
EnemyCharacter.cpp*
EnemyCharacter.h
Dodgeball
(전역 범위)
34 ACharacter* PlayerCharacter = UGameplayStatics::GetPlayerCharacter(this, 0);
35 LookAtActorComponent->SetTarget(PlayerCharacter);
36
37 }
38
39 // Called every frame
40 void AEnemyCharacter::Tick(float DeltaTime)
41 {
42 Super::Tick(DeltaTime);
43
44 // Fetch the character currently being controlled by the player
45 //ACharacter* PlayerCharacter = UGameplayStatics::GetPlayerCharacter(this, 0);
46
47 // Look at the player character every frame
48 //bCanSeePlayer = LookAtActor(PlayerCharacter);
49 bCanSeePlayer = LookAtActorComponent->CanSeeTarget();
50
51 if (bCanSeePlayer != bPreviousCanSeePlayer)
52 {
53 if (bCanSeePlayer)
54 {
55 //Start throwing dodgeballs
56 GetWorldTimerManager().SetTimer(ThrowTimerHandle, this, &AEnemyCharacter::ThrowDodgeball,
57 ThrowingInterval, true, ThrowingDelay);
58 }
59 else
60 {
61 //Stop throwing dodgeballs
62 GetWorldTimerManager().ClearTimer(ThrowTimerHandle);
63 }
64 }
65
66 bPreviousCanSeePlayer = bCanSeePlayer;
67 }
68
100 %
문제가 검색되지 않음
줄: 75 문자: 3 혼합 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.c
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Win64
로컬 Windows 디버거
Live Share
LookAtActorComponent.cpp
LookAtActorComponent.h
HealthInterface.h
EnemyCharacter.cpp*
EnemyCharacter.h
Dodgeball
(전역 범위)
// Called to bind functionality to input
//void AEnemyCharacter::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
//{
// Super::SetupPlayerInputComponent(PlayerInputComponent);
//}
/*
bool AEnemyCharacter::LookAtActor(AActor* TargetActor)
{
if (TargetActor == nullptr)
return false;
const TArray<const AActor*> IgnoreActors = { this, TargetActor };
if (UDodgeballFunctionLibrary::CanSeeActor(GetWorld(), SightSource->GetComponentLocation(), TargetActor, IgnoreActors))
{
FVector Start = GetActorLocation();
FVector End = TargetActor->GetActorLocation();
// Calculate the necessary rotation for the Start point to face the End point
FRotator LookAtRotation = UKismetMathLibrary::FindLookAtRotation(Start, End);
//Set the enemy's rotation to that rotation
SetActorRotation(LookAtRotation);
return true;
}
return false;
}
/*
bool AEnemyCharacter::CanSeeActor(const AActor* TargetActor) const
{
if (TargetActor == nullptr)
return false;
// Store the results of the Line Trace
Ctrl+S
100 %
문제가 검색되지 않음
줄: 75
문자: 3
혼합
CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.c
솔루션 탐색기
Git 변경 내용
준비
소스 제어에 추가


파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H)
검색 (Ctrl+Q)
Dodgeball
Develop
Live Share
솔루션 빌드(B) Ctrl+Shift+B
솔루션 다시 빌드
솔루션 정리(C)
솔루션의 전체 프로그램 데이터베이스 파일 빌드
솔루션에서 코드 분석 실행(Y) Alt+F11
Dodgeball 빌드(U) Ctrl+B
Dodgeball 다시 빌드(E)
Dodgeball 정리(N)
Dodgeball에서 코드 분석 실행(A)
프로젝트만(J)
일괄 빌드(T)...
구성 관리자(O)...
컴파일(M) Ctrl+F7
파일에서 코드 분석 실행(F) Ctrl+Shift+Alt+F7
LookAtActorComponent.cpp
Dodgeball
// Called to bind functionality to
//void AEnemyCharacter::SetupPlayer
//{
// Super::SetupPlayerInputComponen
//}
/*
bool AEnemyCharacter::LookAtActor(A
{
if (TargetActor == nullptr)
return false;
const TArray<const AActor*> Ign
if (UDodgeballFunctionLibrary::
{
FVector Start = GetActorLocation();
FVector End = TargetActor->GetActorLocation();
// Calculate the necessary rotation for the Start point to face the End point
FRotator LookAtRotation = UKismetMathLibrary::FindLookAtRotation(Start, End);
//Set the enemy's rotation to that rotation
SetActorRotation(LookAtRotation);
return true;
}
return false;
}
/*
bool AEnemyCharacter::CanSeeActor(const AActor* TargetActor) const
{
if (TargetActor == nullptr)
return false;
// Store the results of the Line Trace
100 %
문제가 검색되지 않음
줄: 75 문자: 3 혼합 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Dodgeball' (2/2개 프로젝트)
Engine
UE4
Games
Dodgeball
참조
외부 종속성
Config
Source
Dodgeball
Dodgeball.Build.cs
Dodgeball.cpp
Dodgeball.h
DodgeballCharacter.cpp
DodgeballCharacter.h
DodgeballFunctionLibrary.
DodgeballFunctionLibrary.
DodgeballGameMode.cpp
DodgeballGameMode.h
DodgeballProjectile.cpp
DodgeballProjectile.h
EnemyCharacter.cpp
EnemyCharacter.h
HealthComponent.cpp
HealthComponent.h
HealthInterface.cpp
HealthInterface.h
LookAtActorComponent.c
솔루션 탐색기
Git 변경 내용
저장되었습니다.
소스 제어에 추가


ThirdPersonExampleMap*
DDC
Dodgeball
File
Edit
Window
Help
Place Actors
Search Classes
Recently Placed
Basic
Lights
Cinematic
Visual Effects
Geometry
Volumes
All Classes
Empty
Empty
Empty
Point
Playe
Cube
Spher
Cylinc
Cone
Plane
Save Current
Source Control
Modes
Content
Marketplace
Settings
Blueprints
Cinematics
Build
Compile
Perspective
Lit
Show
10
10°
0.25
4
Third Person
World Outliner
Search...
Label
Type
ThirdPersonExampleMap (World
ArenaGeometry
Folder
Arena
Folder
Floor
StaticMeshAc
Wall6
StaticMeshAc
Wall7
StaticMeshAc
Wall8
StaticMeshAc
Wall9
StaticMeshAc
Walkway
Folder
27 actors
View Options
Details
World Settings
Select an object to view details.
Content Browser
Add/Import
Save All
Content
ThirdPersonCPP
Blueprints
Search Paths
Content
Geometry
Mannequin
Physics
StarterContent
ThirdPerson
ThirdPersonCPP
Blueprints
Maps
C++ Classes
Dodgeball
Filters
Search Blueprints
BP_Dodgeball Projectile
BP_Enemy Character
BP_GhostWall
BP_VictoryBox
BP_Wall
ThirdPerson Character
6 items (1 selected)
View Options

BP_EnemyCharacter

File Edit Asset View Debug Window Help

Parent class: Enemy Character

Components

+ Add Component | Search

BP_EnemyCharacter(self)

- Capsule Component (CollisionCy
- Arrow Component (Arrow) (Inhe
- Mesh (CharacterMesh0) (Inherit
- Look at Actor Component (Look
- Character Movement (CharMoveC

My Blueprint

+ Add New | Search

Graphs

- EventGraph
  - Event BeginPlay
  - Event ActorBeginOverlap
  - Event Tick

Functions (29 Overridable)

- ConstructionScript

Macros

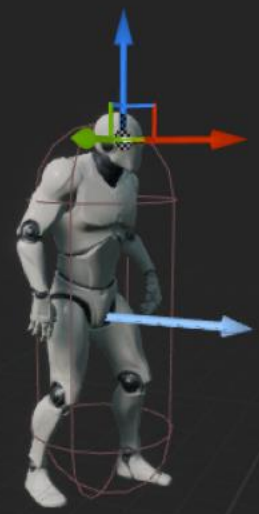
Variables

Event Dispatchers

Compile | Save | Browse | Find | Hide Unrelated | Class Settings | Class Defaults | Simulation | Play

Viewport | Construction Scrip | Event Graph

Perspective | Lit | 10 | 10° | 0.25 | 4



Details

Search Details

Variable

Variable Name	LookAtActorComponent
Tooltip	
Category	LookAt
Editable when Inherited	☑

Transform

Location	X 10.0	Y 0.0	Z 80.0
Rotation	X 0.0°	Y 0.0°	Z 0.0°
Scale	X 1.0	Y 1.0	Z 1.0
Mobility	Static	Stationa	Movabl

Sockets

Parent Socket

Rendering

Visible	☑
Hidden in Game	☐

Component Tick

Start with Tick Enabled	☑
Tick Interval (secs)	0.0

Tags

Component Tags: 0 Array elements

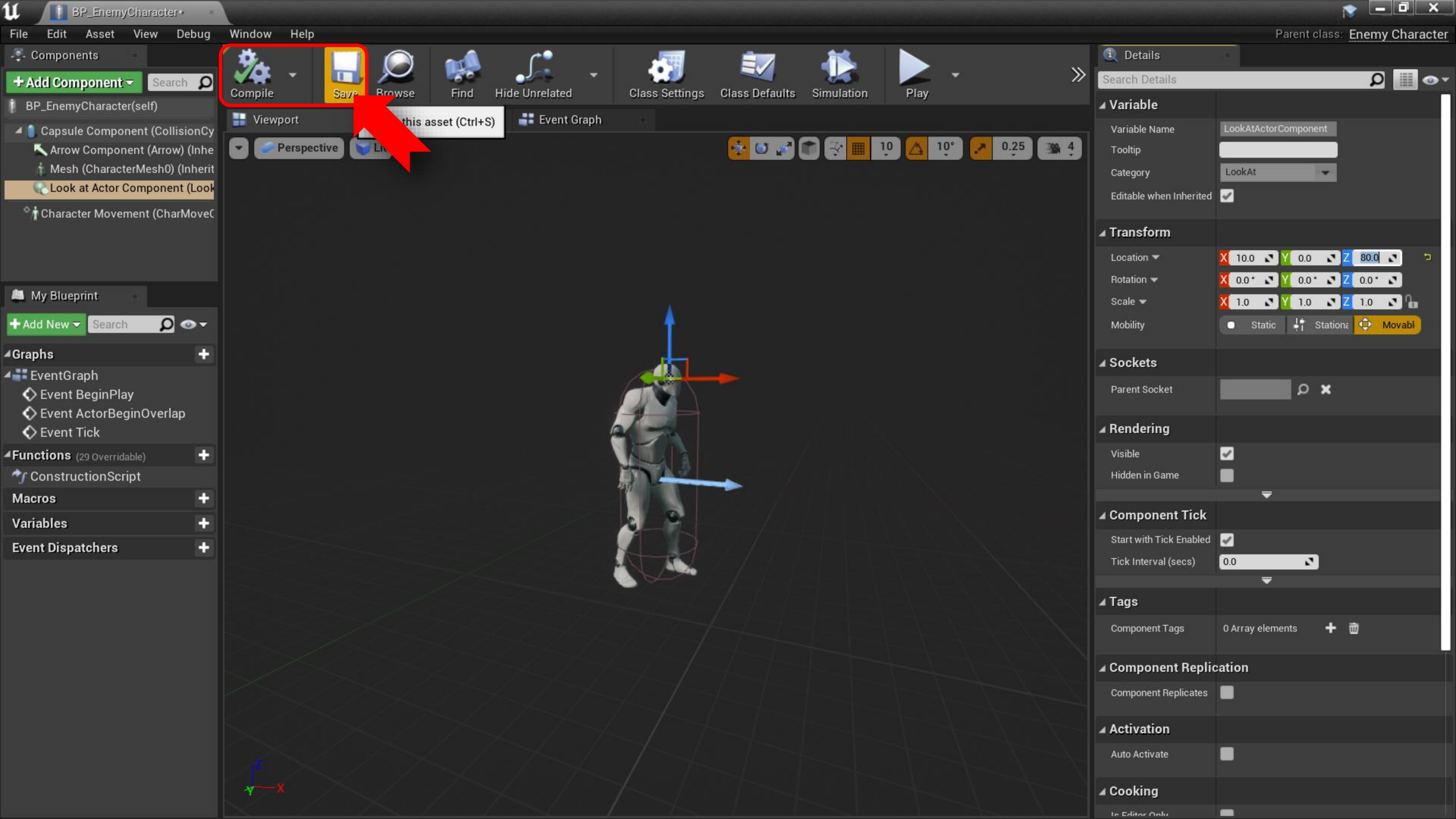
Component Replication

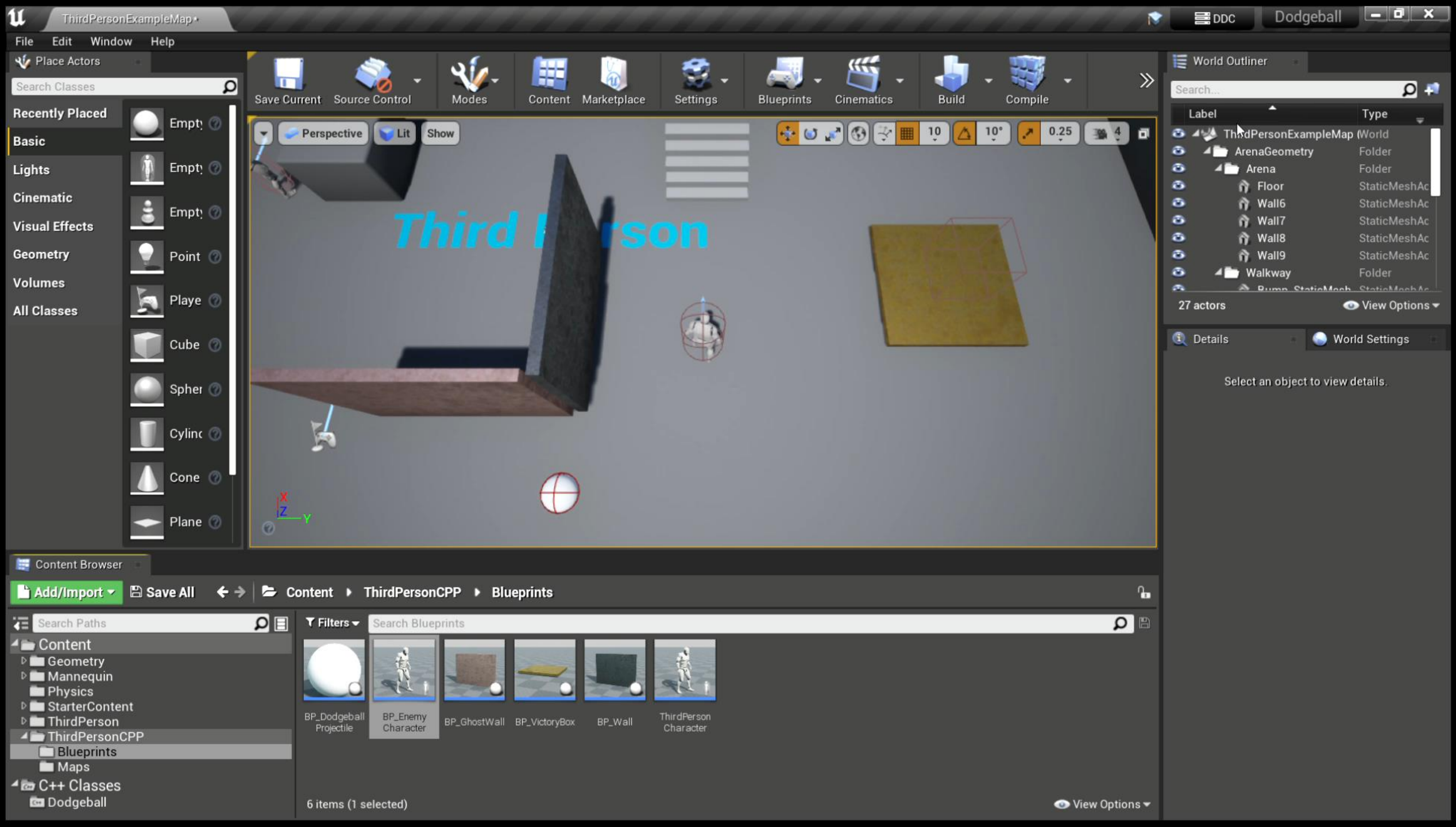
Component Replicates ☐

Activation

Auto Activate ☐

Cooking


BP_EnemyCharacter
File Edit Asset View Debug Window Help
Parent class: Enemy Character
Components
+ Add Component
Search
BP_EnemyCharacter(self)
Capsule Component (CollisionCy
Arrow Component (Arrow) (Inhe
Mesh (CharacterMesh0) (Inherit
Look at Actor Component (Look
Character Movement (CharMoveC
Compile
Save
Browse
Find
Hide Unrelated
Class Settings
Class Defaults
Simulation
Play
this asset (Ctrl+S)
Viewport
Event Graph
Perspective
My Blueprint
+ Add New
Search
Graphs
EventGraph
Event BeginPlay
Event ActorBeginOverlap
Event Tick
Functions (29 Overridable)
ConstructionScript
Macros
Variables
Event Dispatchers
Details
Search Details
Variable
Variable Name
LookAtActorComponent
Tooltip
Category
LookAt
Editable when Inherited
Transform
Location
Rotation
Scale
Mobility
Static
Stationa
Movabl
Sockets
Parent Socket
Rendering
Visible
Hidden in Game
Component Tick
Start with Tick Enabled
Tick Interval (secs)
0.0
Tags
Component Tags
0 Array elements
Component Replication
Component Replicates
Activation
Auto Activate
Cooking


ThirdPersonExampleMap*
DDC
Dodgeball
File
Edit
Window
Help
Place Actors
Search Classes
Recently Placed
Basic
Lights
Cinematic
Visual Effects
Geometry
Volumes
All Classes
Empty
Empty
Empty
Point
Playe
Cube
Spher
Cylinc
Cone
Plane
Save Current
Source Control
Modes
Content
Marketplace
Settings
Blueprints
Cinematics
Build
Compile
Perspective
Lit
Show
10
10°
0.25
4
Third Person
X
Z
Y
World Outliner
Search...
Label
Type
ThirdPersonExampleMap (World
ArenaGeometry
Folder
Arena
Folder
Floor
StaticMeshAc
Wall6
StaticMeshAc
Wall7
StaticMeshAc
Wall8
StaticMeshAc
Wall9
StaticMeshAc
Walkway
Folder
27 actors
View Options
Details
World Settings
Select an object to view details.
Content Browser
Add/Import
Save All
Content
ThirdPersonCPP
Blueprints
Search Paths
Content
Geometry
Mannequin
Physics
StarterContent
ThirdPerson
ThirdPersonCPP
Blueprints
Maps
C++ Classes
Dodgeball
Filters
Search Blueprints
BP_Dodgeball Projectile
BP_Enemy Character
BP_GhostWall
BP_VictoryBox
BP_Wall
ThirdPerson Character
6 items (1 selected)
View Options