

# Spawning The Player Projectile

14<sup>th</sup> Week, 2021



UNREAL  
ENGINE



# Select or Create New Project

## Recent Projects



More

## New Project Categories



### Games

Start your game development journey with one of our key classes, levels, and examples.



### Film, Television, and Live Events

Choose from templates and examples for nDisplay, VR Scouting, and virtual production workflows.

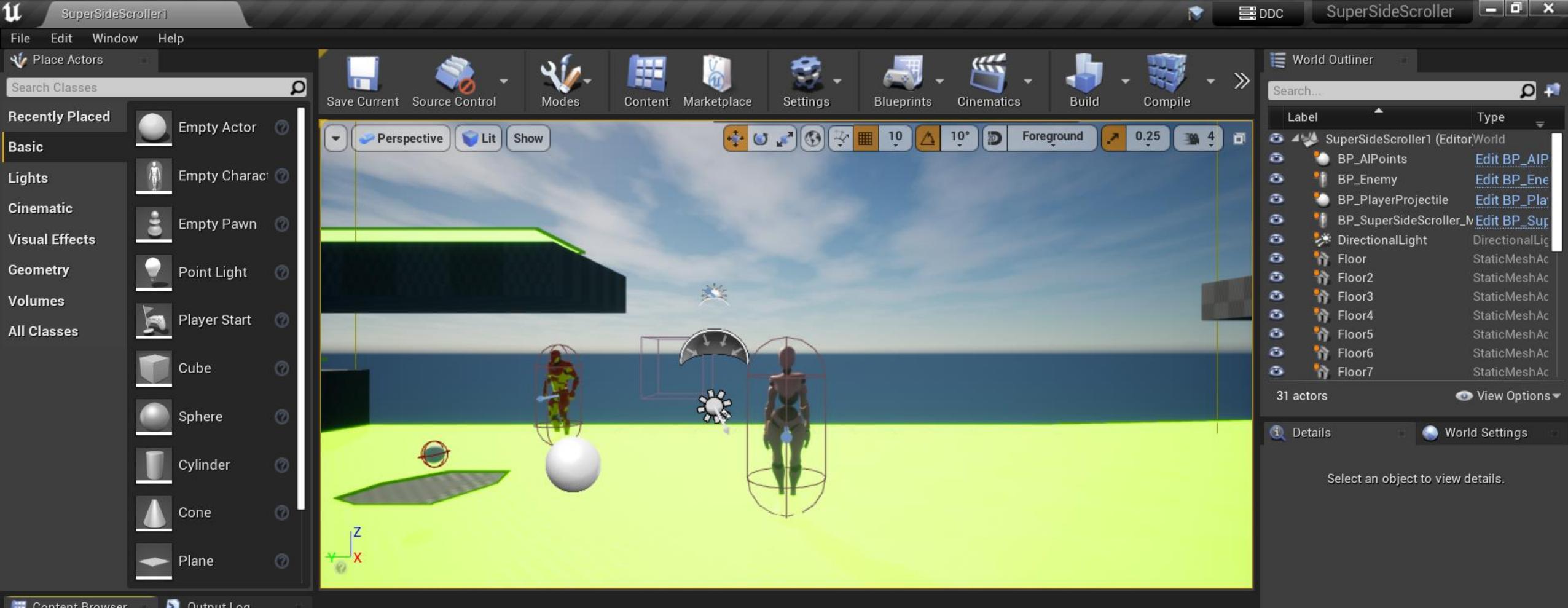


### Architecture, Engineering, and Construction

Select a starting point for multi-user design reviews, photorealistic architectural design visualizations, sunlight studies, or stylized renderings.

Open Project

Cancel



The screenshot shows the "Content Browser" panel from the Unreal Engine 4 Editor. The top bar has tabs for "Add/Import", "Save All", and "C++ Classes". The current project is "SuperSideScroller".

The left sidebar shows a tree view of content assets under the "Content" category. The "Enemy" folder contains sub-folders for AI, Blueprints, and Materials. The "Geometry" and "MainCharacter" folders are collapsed. The "StarterContent" folder is also listed. The "C++ Classes" tab is selected, showing a list of C++ classes: SuperSideScroller1, BP\_AIPoints, BP\_Enemy, BP\_PlayerProjectile, BP\_SuperSideScroller\_M, DirectionalLight, Floor, Floor2, Floor3, Floor4, Floor5, Floor6, and Floor7. One item, "SuperSideScrollerGameMode", is selected and highlighted with a yellow border and a red arrow pointing to it.

The bottom left corner shows the number "5 items (1 selected)".

Live Share

SuperSideScrollerGameMode.cpp\*

```
SuperSideScrollerGameMode.cpp*  X
SuperSideScroller
ASuperSideScrollerGameMode
ASuperSideScrollerGameMode0

1 // Copyright Epic Games, Inc. All Rights Reserved.
2
3 #include "SuperSideScrollerGameMode.h"
4 #include "SuperSideScrollerCharacter.h"
5 #include "UObject/ConstructorHelpers.h"
6
7 ASuperSideScrollerGameMode::ASuperSideScrollerGameMode()
8 {
9     /*
10     // set default pawn class to our Blueprinted character
11     static ConstructorHelpers::FClassFinder<APawn> PlayerPawnBPClass(TEXT("/Game/SideScrollerCPP/Blueprints/SideScrollerCharacter"));
12     if (PlayerPawnBPClass.Class != nullptr)
13     {
14         DefaultPawnClass = PlayerPawnBPClass.Class;
15     }
16 */
17 }
```

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+.)

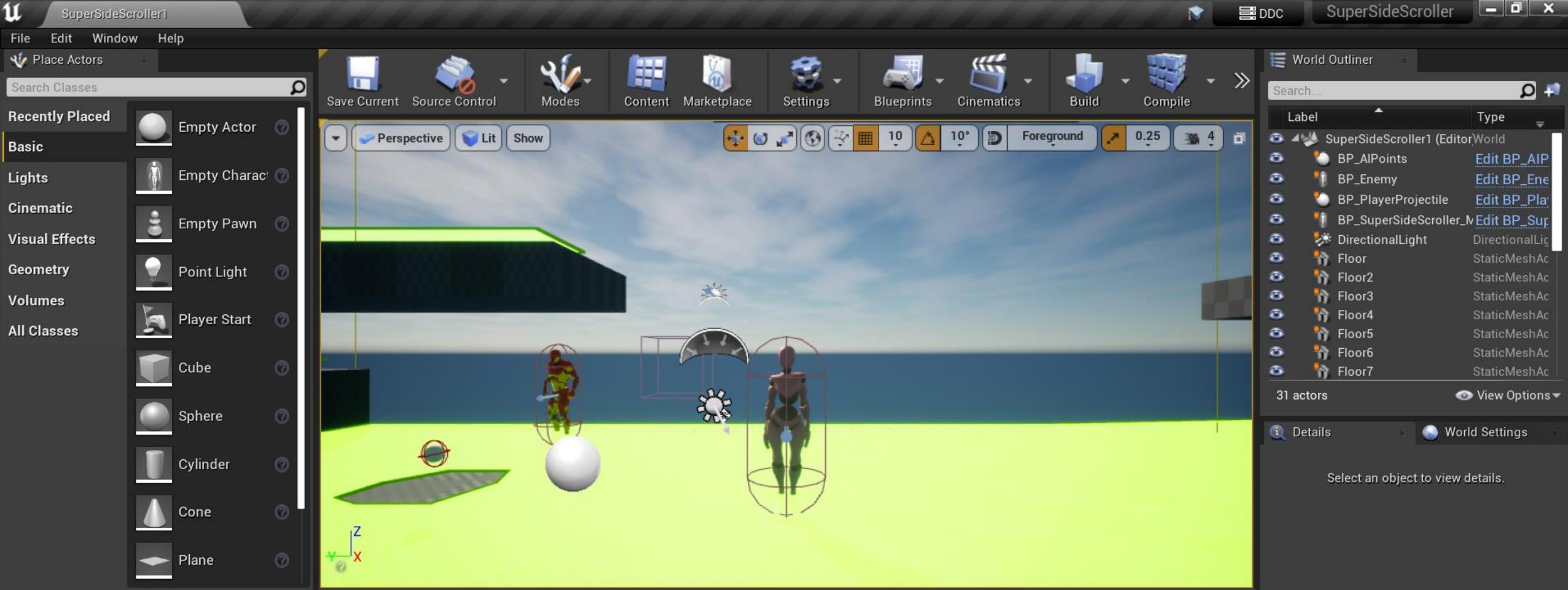
- 솔루션 'SuperSideScroller' (2/2개 프로젝트)
  - Engine
    - UE4
  - Games
    - SuperSideScroller
      - 참조
      - 외부 종속성
      - Config
      - Source
        - SuperSideScroller
          - EnemyBase.cpp
          - EnemyBase.h
          - PlayerProjectile.cpp
          - PlayerProjectile.h
          - SuperSideScroller.Build.cs
          - SuperSideScroller.cpp
          - SuperSideScroller.h
          - SuperSideScroller\_Player.cpp
          - SuperSideScroller\_Player.h
          - SuperSideScrollerCharacter
          - SuperSideScrollerCharacter
          - SuperSideScrollerGameMode
          - SuperSideScrollerGameMode
          - SuperSideScroller.Target.cs
          - SuperSideScrollerEditor.Target.cs
  - Visualizers

'18 전OO

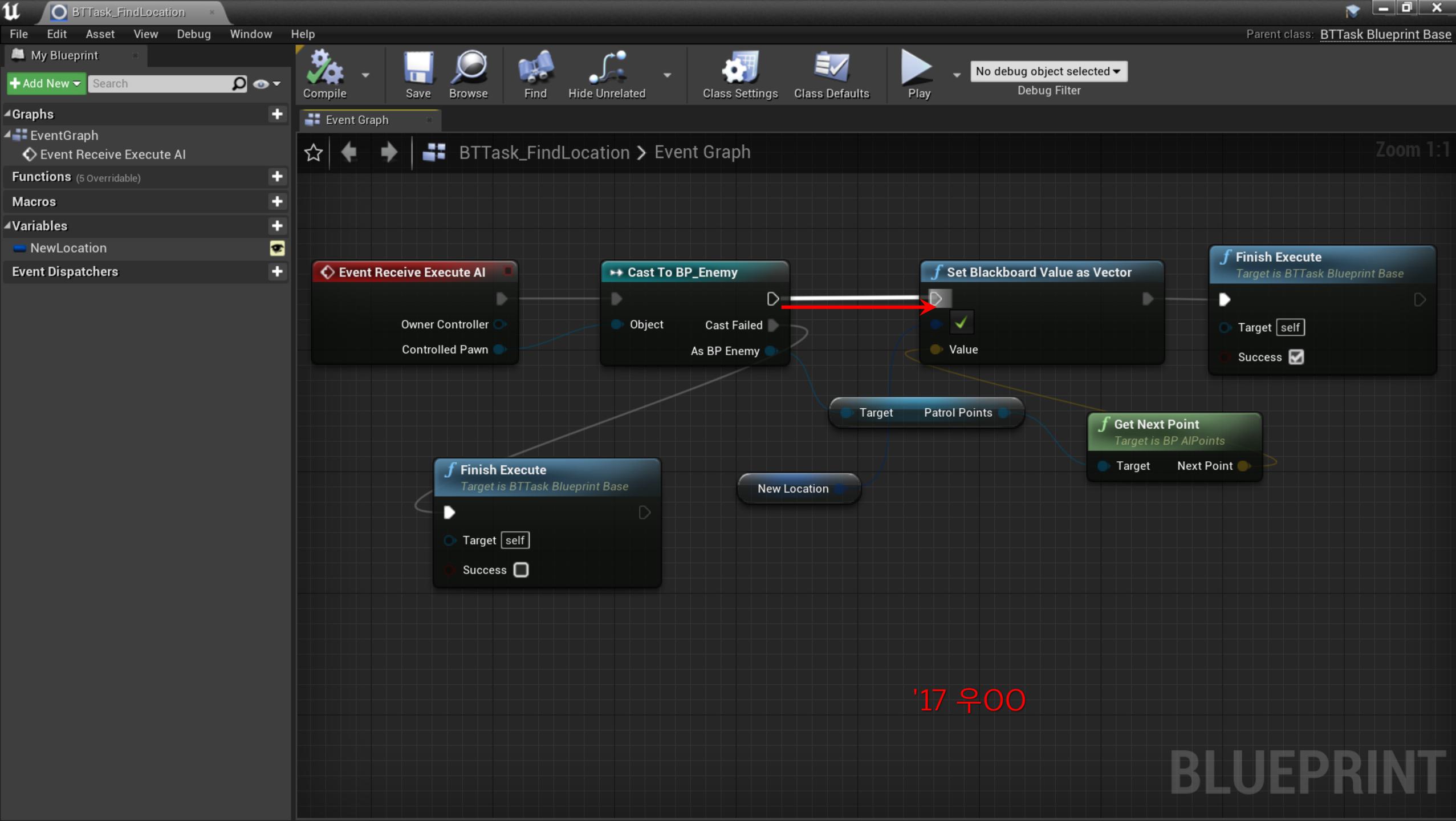
100 % 문제가 검색되지 않음 줄: 16 문자: 4 열: 7 템 CRLF

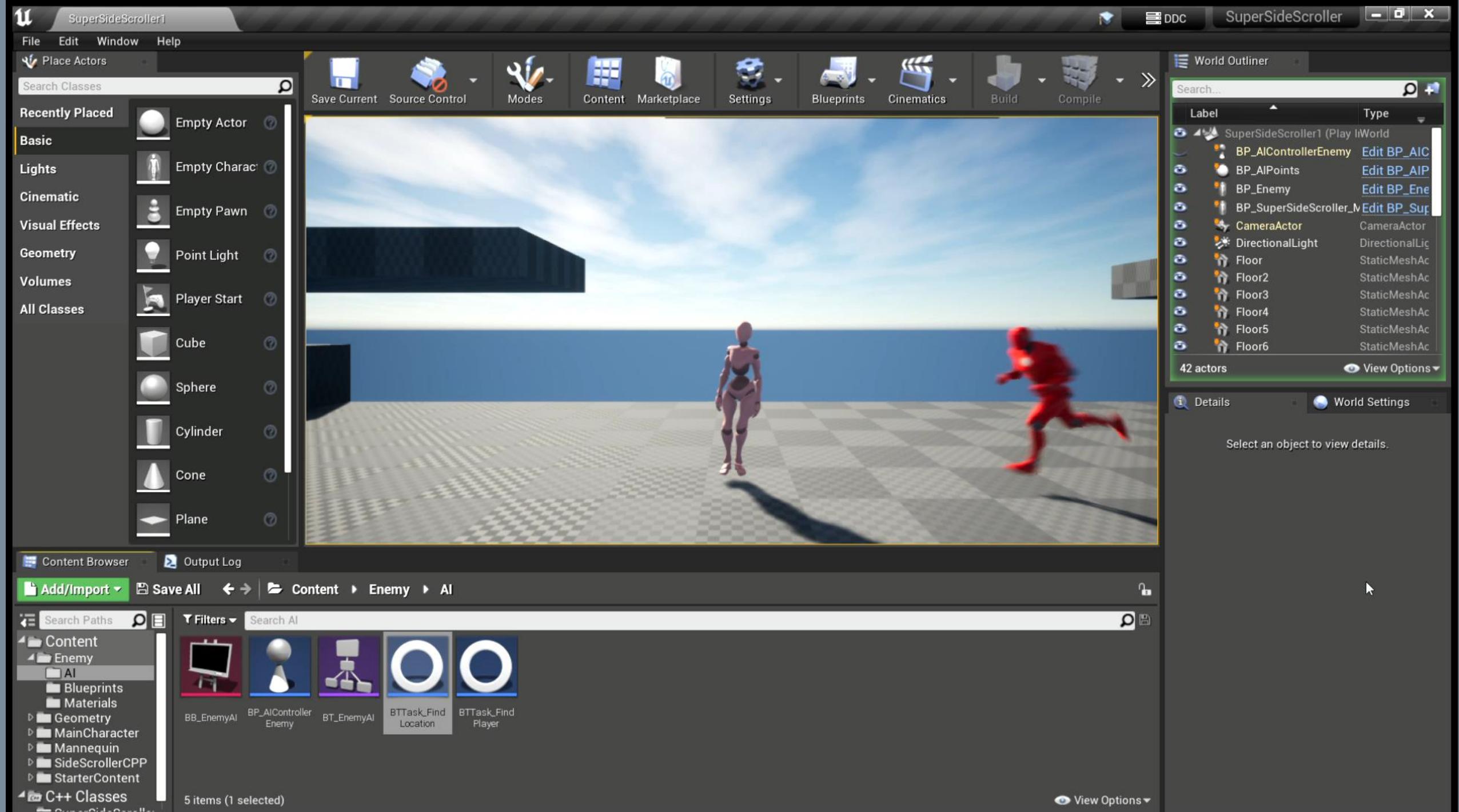
슬루션 탐색기 Git 변경 내용

준비 ↑ 소스 제어에 추가 ▲



The screenshot shows the Content Browser. The top bar has buttons for "Add/Import", "Save All", and navigation. The left sidebar shows a file tree: Content > Enemy > AI. The main area displays icons for various assets: a monitor icon for BB\_EnemyAI, a pawn icon for BP\_AIController\_Enemy, a network icon for BT\_EnemyAI, a yellow highlighted circle icon for BTTask\_Find Location (with a red arrow pointing to it), and a blue highlighted circle icon for BTTask\_Find Player. The bottom left shows "5 items (1 selected)". The bottom right has a "View Options" button.





Behavior Tree Node Reference: [docs.unrealengine.com/4.27/en-US/InteractiveExperiences/ArtificialIntelligence/BehaviorTree/Nodes/Selector.htm](#)

## > Selector



**Selector** Nodes execute their children from left to right. They stop executing when one of their children succeeds. If a Selector's child succeeds, the Selector succeeds. If all the Selector's children fail, the Selector fails.

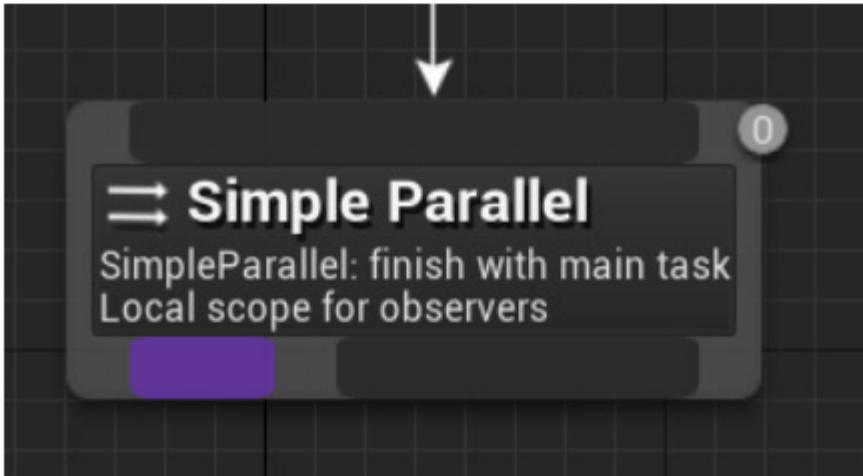
Behavior Tree Node Reference: [docs.unrealengine.com/4.27/en-US/InteractiveExperiences/ArtificialIntelligence/BehaviorTree/Nodes/Sequence.htm](#)

## > Sequence

**Sequence** nodes execute their children from left to right. They stop executing when one of their children fails. If a child fails, then the Sequence fails. If all the Sequence's children succeed, then the Sequence succeeds.

Behavior Tree Node Reference: [docs.unrealengine.com/4.27/en-US/InteractiveExperiences/ArtificialIntelligence/BehaviorTree/Nodes/Parallel/SimpleParallel.htm](#)

## Simple Parallel



The image shows the Simple Parallel node in the Behavior Tree Editor. The node is a dark gray rectangle with rounded corners. At the top center is a white icon consisting of two horizontal bars. To the right of the icon, the text "Simple Parallel" is written in a large, bold, white sans-serif font. Below this, in a smaller white font, are the words "SimpleParallel: finish with main task" and "Local scope for observers". A small purple rectangular button is located at the bottom left of the node. A white arrow points downwards from the top center of the node towards the explanatory text below.

The **Simple Parallel** node allows a single main Task node to be executed alongside of a full tree. When the main Task finishes, the setting in **Finish Mode** dictates if the node should finish immediately, aborting the secondary tree, or if it should delay for the secondary tree to finish.



# Introduction

- › You will learn
  - How to use the **UAnimNotify** class to spawn the player projectile at a specific frame of the **Throw** Animation Montage
  - How to add this new notify to the Montage itself
  - How to add a new **Socket** to the main character skeleton from the projectile will spawn
  - How to use **Particle Systems** and **SoundCues** to add a layer of visual and audio polish to the game



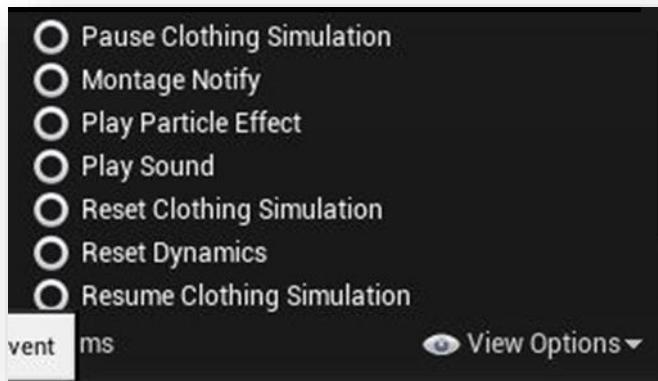
# Anim Notifies and Anim Notify States (1)

- › To use **Anim Notifies** and **Anim Notify States**
  - To add custom events within animation that will allow for additional effects, layers, and functionality to occur
- › The main difference between **Anim Notifies** and **Anim Notify States**
  - **Anim Notify State** possesses three distinct events that **Anim Notify** does not.
    - › **Notify Begin**, **Notify End**, and **Notify Tick**
- › **Anim Notify** is a much more simplified version that uses just a single function, **Notify()**
  - To allow programmers to add functionality to the notify itself



## Anim Notifies and Anim Notify States (2)

- › A full list of default **Anim Notifies** states can be seen in the following screenshot:

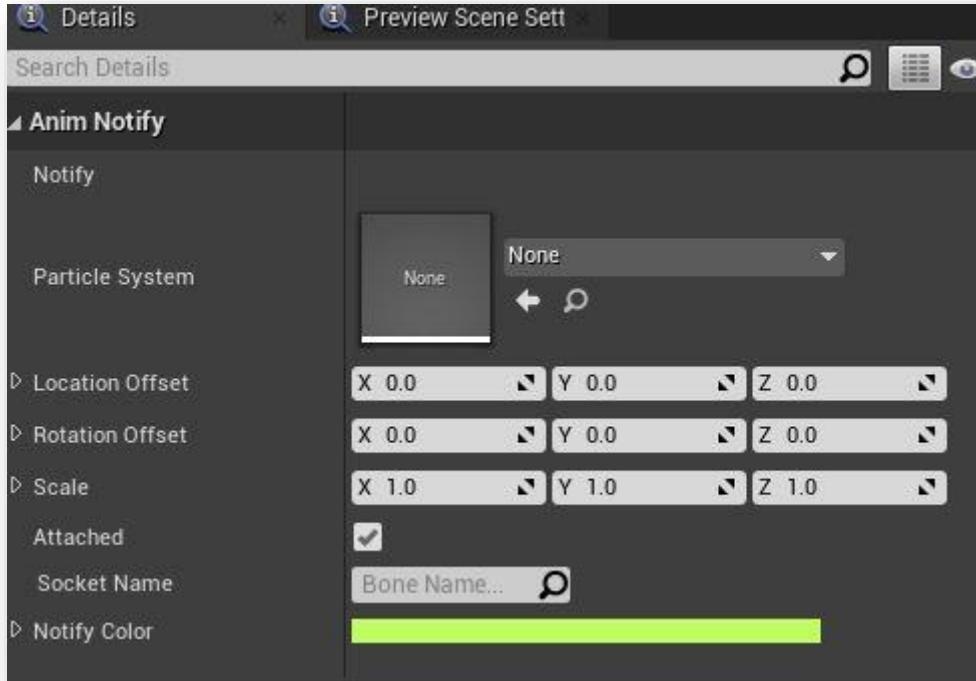


- **Play Particle Effect** and **Play Sound** will be used later.

# Anim Notifies and Anim Notify States (3)

## › Play Particle Effect

- To spawn and play a particle system at a certain frame of your animation

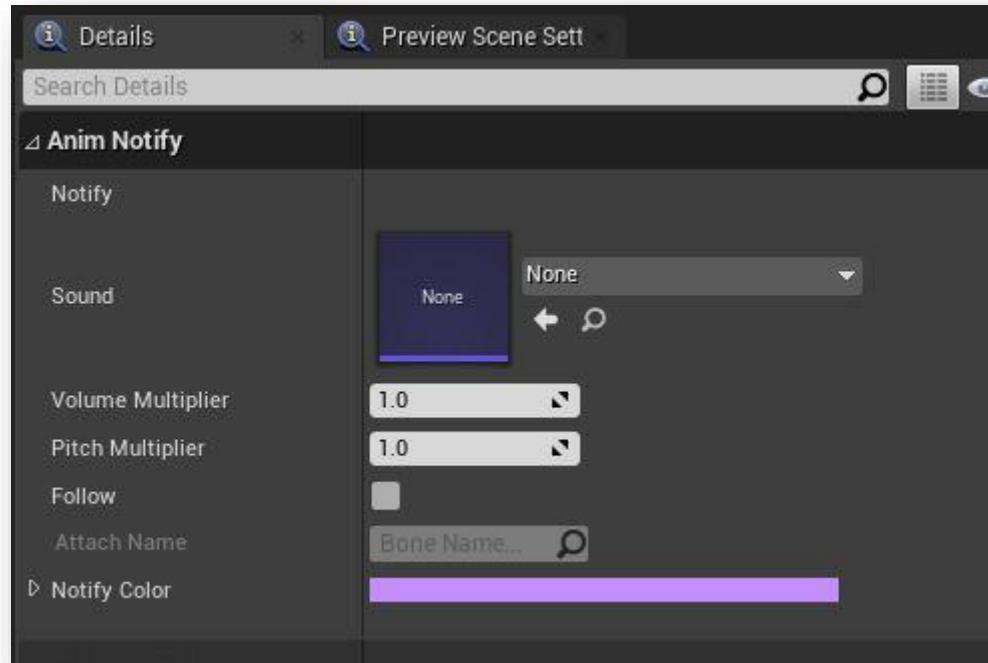


The Details panel of the Play Particle Effect notify,  
which allows you to customize the particle

# Anim Notifies and Anim Notify States (4)

## › Play Sound

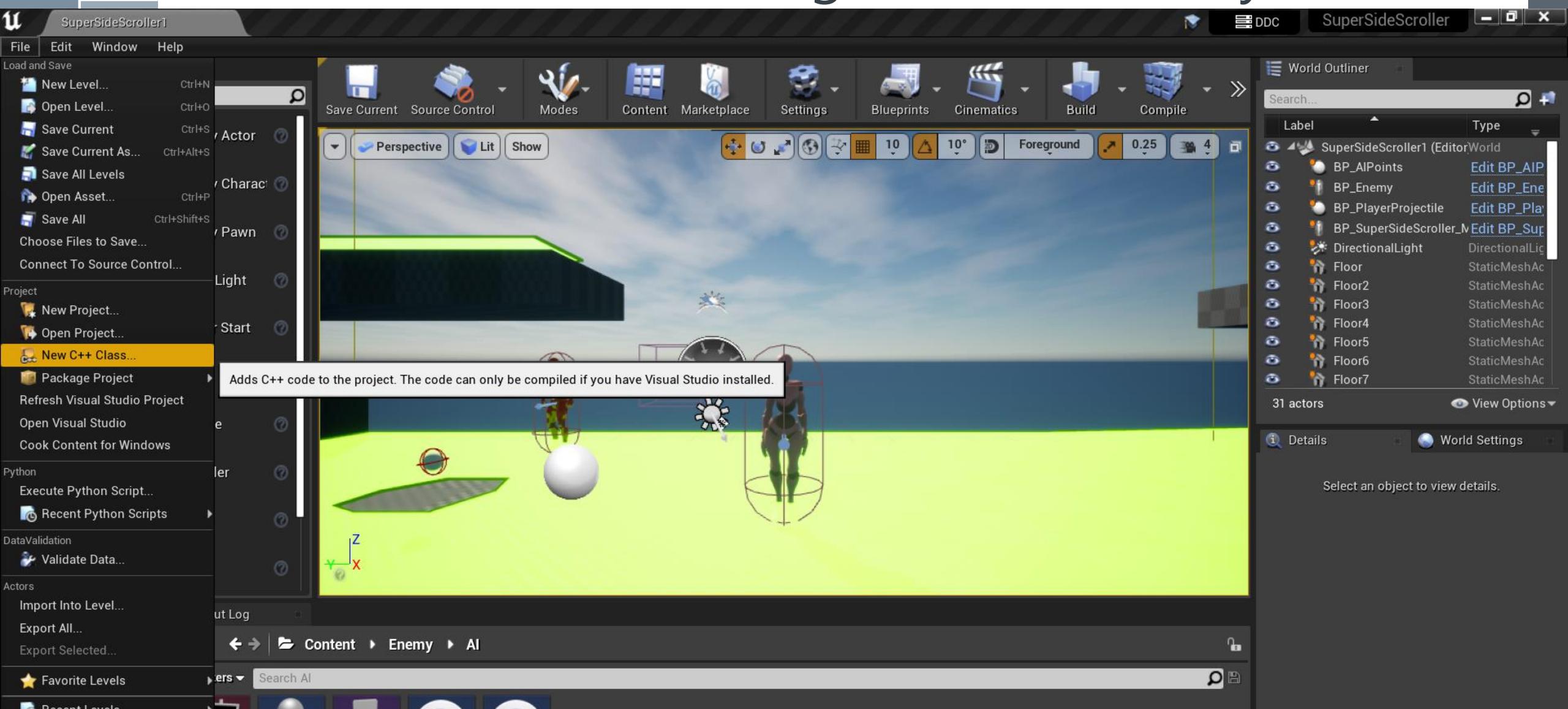
- To play a **Soundcue** or **Soundwave** at a certain frame of your animation



The Details panel of the Play Sound notify,  
which allows you to customize the sound



# Exercise 14.01: Creating A UAnimNotify Class





# Choose Parent Class

This will add a C++ header and source code file to your game project.

Show All Classes

anim noti

Object

AnimNotify

- AnimNotify\_PauseClothingSimulation
- AnimNotify\_PlayMontageNotify
- AnimNotify\_PlayNiagaraEffect
- AnimNotify\_PlayParticleEffect
- AnimNotify\_PlaySound
- AnimNotify\_ResetClothingSimulation
- AnimNotify\_ResetDynamics
- AnimNotify\_ResumeClothingSimulation

AnimNotifyState

- AnimNotify\_PlayMontageNotifyWindow
- AnimNotifyState\_DisableRootMotion

17 items (1 selected)

View Options ▾

Selected Class

Anim Notify

Selected Class Source

AnimNotify.h

Next >

Create Class

Cancel



# Name Your New Anim Notify

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	<input type="text" value="Anim_ProjectileNotify"/> <span style="border: 2px solid red; padding: 2px;">Anim_ProjectileNotify</span>	SuperSideScroller (Runtime) ▾	Public	Private
Path	<input type="text" value="C:/Users/sunje/Desktop/Unreal/SuperSideScroller/Source/SuperSideScroller/"/> <span style="border: 1px solid #ccc; padding: 2px;">Choose Folder</span>			
Header File	C:/Users/sunje/Desktop/Unreal/SuperSideScroller/Source/SuperSideScroller/Anim_ProjectileNotify.h			
Source File	C:/Users/sunje/Desktop/Unreal/SuperSideScroller/Source/SuperSideScroller/Anim_ProjectileNotify.cpp			

< Back

Create Class ↗

Cancel

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes: 파일(F), 편집(E), 보기(V), Git(G), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(I), 확장(X), 창(W), 도움말(H), 검색 (Ctrl+Q). The status bar at the bottom shows: 100 %, 문제가 검색되지 않음 (No problems found), 줄: 19, 문자: 95, 열: 98, 탭, CRLF.

The main window displays the code editor with three tabs: Anim\_ProjectileNotify.cpp (selected), Anim\_ProjectileNotify.h\*, and SuperSideScrollerGameMode.cpp. The code editor shows the following C++ code:

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "Animation/AnimNotifies/AnimNotify.h"
7 #include "Anim_ProjectileNotify.generated.h"
8
9 /**
10  * 
11 */
12 UCLASS()
13 class SUPERSIDESCRROLLER_API UAnim_ProjectileNotify : public UAnimNotify
14 {
15     GENERATED_BODY()
16
17     public:
18         virtual void Notify(USkeletalMeshComponent* MeshComp, UAnimSequenceBase* Animation) override;
19     };
20
21 }
22
```

A red rectangular box highlights the line of code: `virtual void Notify(USkeletalMeshComponent* MeshComp, UAnimSequenceBase* Animation) override;`. A red circle with the text "Ctrl+S" is overlaid on the right side of the code editor area.

The Solution Explorer on the right shows the project structure:

- 솔루션 탐색기
- 솔루션 탐색기 검색(Ctrl+Shift+F)
- 솔루션 탐색기 'SuperSideScroller' (2/2개 프로젝트)
  - Engine
  - Games
  - SuperSideScroller
    - 참조
    - 외부 종속성
    - Config
    - Source
      - SuperSideScroller
        - Anim\_ProjectileNotify.cpp
        - Anim\_ProjectileNotify.h
        - EnemyBase.cpp
        - EnemyBase.h
        - PlayerProjectile.cpp
        - PlayerProjectile.h
        - SuperSideScroller.Build.cs
        - SuperSideScroller.cpp
        - SuperSideScroller.h
        - SuperSideScroller\_Player.cpp
        - SuperSideScroller\_Player.h
        - SuperSideScrollerCharacter.cpp
        - SuperSideScrollerCharacter.h
        - SuperSideScrollerGameMode.cpp
        - SuperSideScrollerGameMode.h
        - SuperSideScroller.Target.cs
        - SuperSideScrollerEditor.Target.cs
      - SuperSideScroller.uproject

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Super...ller

Develop Win64 로컬 Windows 디버거 Live Share

Anim\_ProjectileNotify.cpp\* Anim\_ProjectileNotify.h SuperSideScrollerGameMode.cpp

SuperSideScroller

```
1 // Fill out your copyright notice in the Description page of Project Settings.  
2  
3 #include "Anim_ProjectileNotify.h"  
4  
5 void UAnim_ProjectileNotify::Notify(USkeletalMeshComponent* MeshComp, UAnimSequenceBase* Animation)  
6 {  
7     UE_LOG(LogTemp, Warning, TEXT("Throw Notify"));  
8 }  
9  
10
```

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

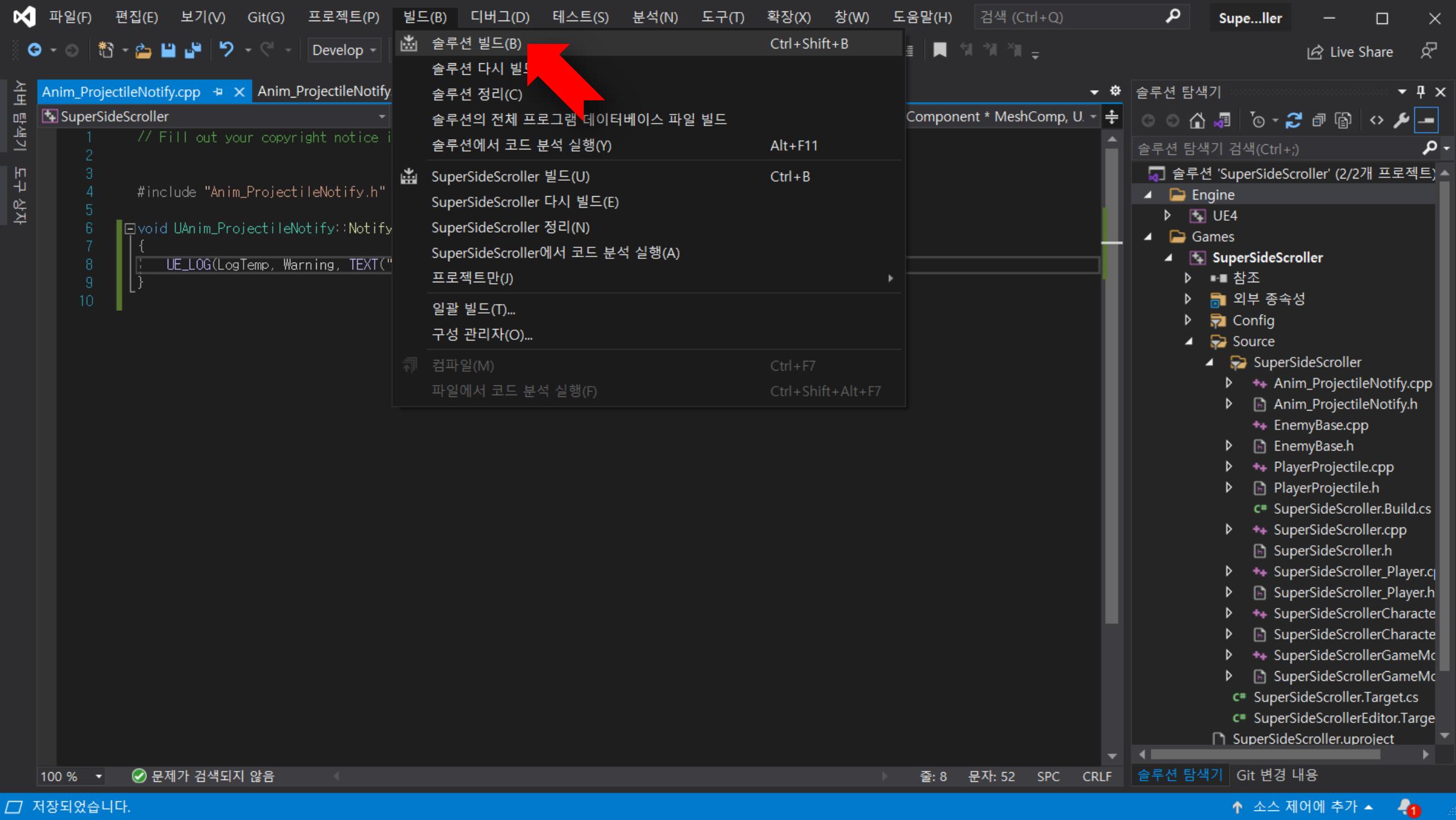
솔루션 탐색기 'SuperSideScroller' (2/2개 프로젝트)

- Engine
- Games
- SuperSideScroller
  - 참조
  - 외부 종속성
  - Config
  - Source
    - SuperSideScroller
      - Anim\_ProjectileNotify.cpp
      - Anim\_ProjectileNotify.h
      - EnemyBase.cpp
      - EnemyBase.h
      - PlayerProjectile.cpp
      - PlayerProjectile.h
      - SuperSideScroller.Build.cs
      - SuperSideScroller.cpp
      - SuperSideScroller.h
      - SuperSideScroller\_Player.cpp
      - SuperSideScroller\_Player.h
      - SuperSideScrollerCharacter.cpp
      - SuperSideScrollerCharacter.h
      - SuperSideScrollerGameMode.cpp
      - SuperSideScrollerGameMode.h
      - SuperSideScroller.Target.cs
      - SuperSideScrollerEditor.Target.cs
    - SuperSideScroller.uproject

Ctrl+S

100 % 문제가 검색되지 않음 줄: 8 문자: 52 SPC CRLF

저장되었습니다. ↑ 소스 제어에 추가 ↗ 1





# Exercise 14.02: Adding the Notify to the Throw Montage

The screenshot shows the Unreal Engine Editor interface. The main view displays a character in a dynamic pose, likely performing a throw, with a white sphere nearby. The Content Browser at the bottom shows a list of animations under the MainCharacter > Animation folder. The 'AM\_Throw' animation is selected and highlighted with a yellow background. A red arrow points to this selection. The top right panel shows a list of actors in the scene, including various floor objects and a directional light.

Content Browser

Output Log

Add/Import Save All

Content > MainCharacter > Animation

Search Paths Filters Search Animation

Content

Enemy

AI

Blueprints

Materials

Geometry

MainCharacter

Animation

Blueprints

Mesh

Projectile

Mannequin

AM\_Throw

Idle JumpEnd JumpingStart JumpLoop Running SideScroller\_IdleRun\_1D Throw Walking

9 items (1 selected)

View Options

BP\_PlayerProjectile Edit BP\_Pl...

BP\_SuperSideScroller\_M Edit BP\_Sup...

DirectionalLight

Floor

Floor2

Floor3

Floor4

Floor5

Floor6

Floor7

StaticMeshAc

StaticMeshAc

StaticMeshAc

StaticMeshAc

StaticMeshAc

StaticMeshAc

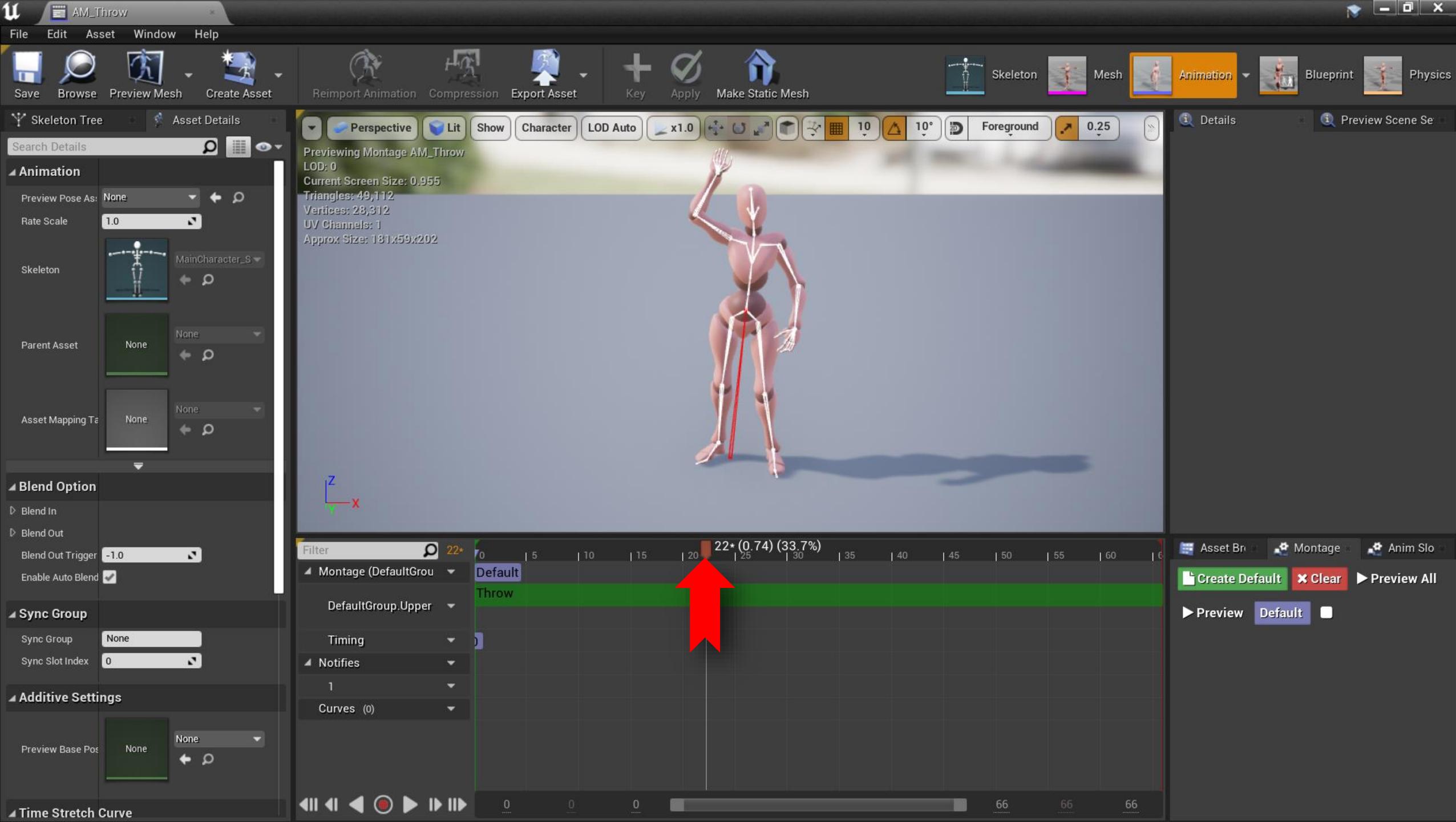
StaticMeshAc

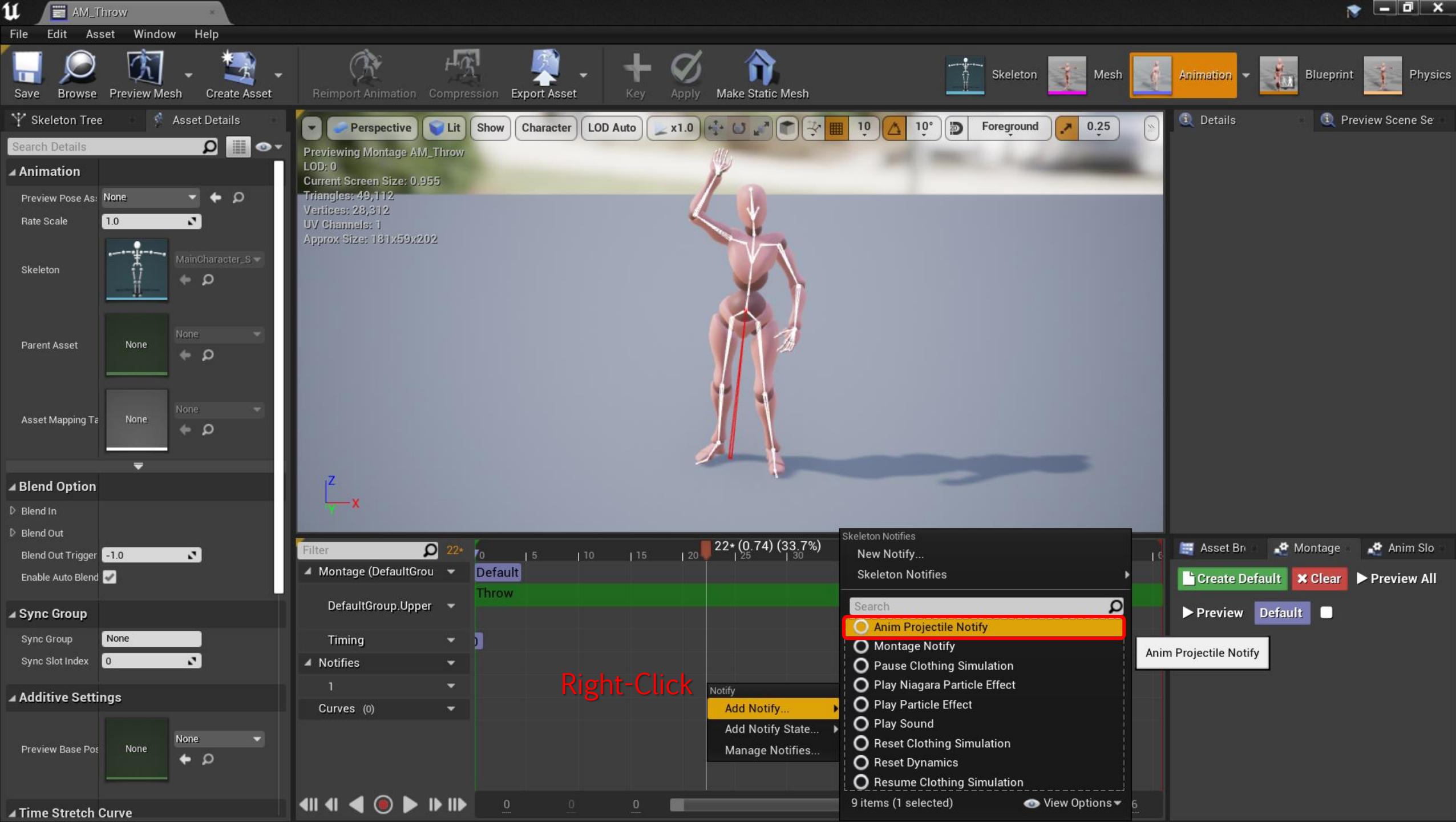
31 actors

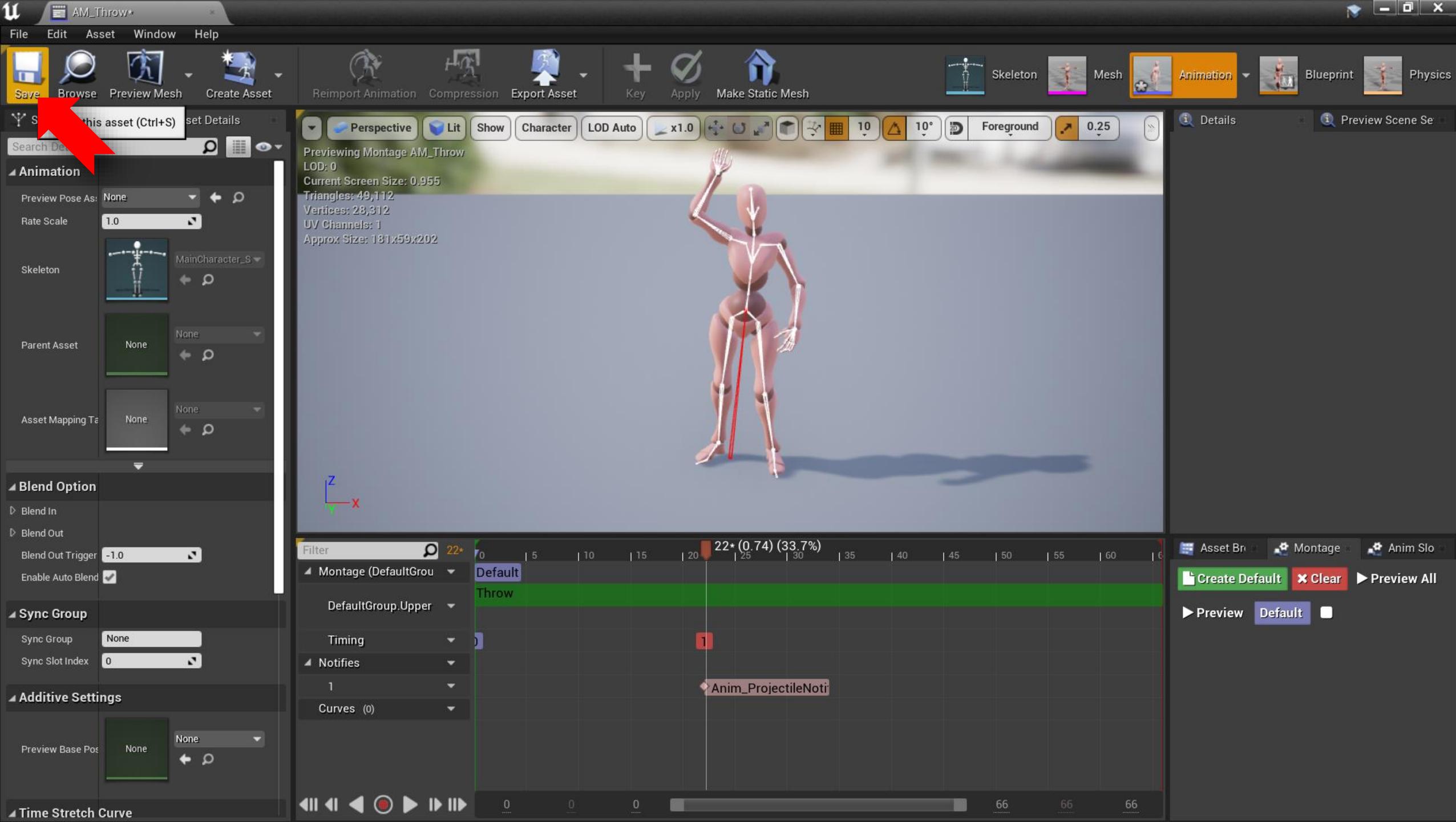
View Options

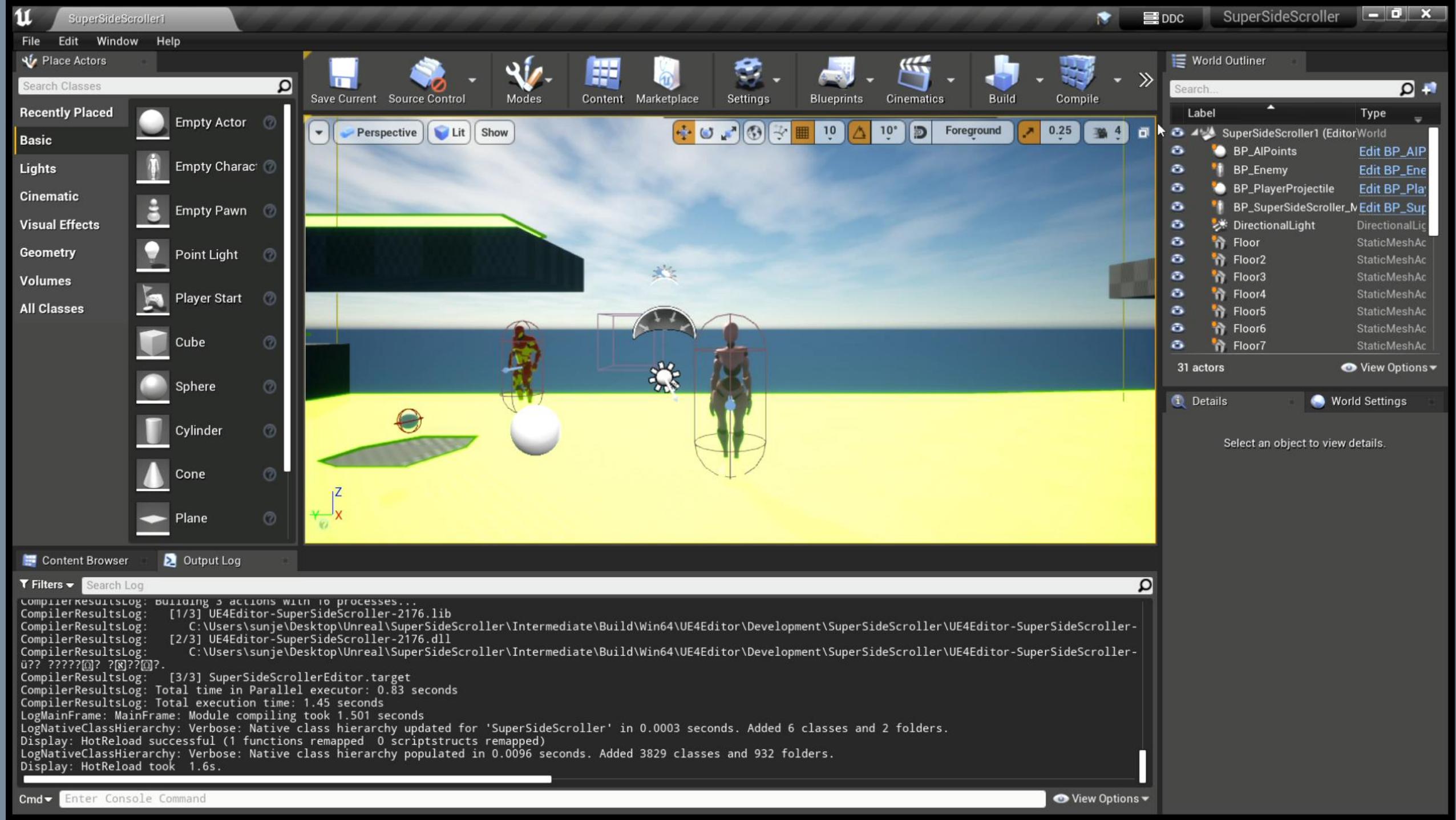
Details World Settings

Select an object to view details.











# Playing Animation Montages

- › To combine individual animation sequences into one complete montage
- › How do we play the Animation Montage on a character?
  - Playing Animation Montages in Blueprints
  - Playing Animation Montages in C++

# Playing Animation Montages in Blueprints

- › In Blueprints, the **Play Montage** function is available



The Play Montage function in Blueprint



# Playing Animation Montages in C++

- › **UAnimInstance::Montage\_Play()** function
  - This function requires the Animation Montage to play
  - A value of **EMontagePlayReturnType** type
    - › **EMontagePlayReturnType::MontageLength** (default)
      - To return the length of the Montage itself, in seconds
    - › **EMontagePlayReturnType::Duration**
      - To return the play duration of the montage, which is equal to the length of montage, divided by the play rate
  - A **float** value for determining the start position to play the montage
  - A **Boolean** value for determining whether playing this Montage should stop or interrupt all montages



# Exercise 14.03: Playing the Throw Animation in C++

The screenshot shows the Unreal Engine Editor interface. The central viewports displays a 3D scene with a character in a dynamic pose, likely performing a throw, and a white sphere nearby. The left sidebar lists basic geometry classes: Empty Pawn, Point Light, Player Start, Cube, Sphere, Cylinder, Cone, and Plane. The bottom-left Content Browser shows the project structure under Content/Enemy/MainCharacter/Blueprints, with two Blueprint files listed: AnimBP\_SuperSideScroller\_MainCharacter and BP\_SuperSideScroller\_MainCharacter. A red arrow points to the second Blueprint file. The top-right details panel shows 31 actors in the scene, including various floor meshes and directional lights.

Cinematic  
Visual Effects  
Geometry  
Volumes  
All Classes

Empty Pawn  
Point Light  
Player Start  
Cube  
Sphere  
Cylinder  
Cone  
Plane

BP\_PlayerProjectile Edit BP\_Pl/  
BP\_SuperSideScroller\_M Edit BP\_Sup/  
DirectionalLight DirectionalLig/  
Floor StaticMeshAc/  
Floor2 StaticMeshAc/  
Floor3 StaticMeshAc/  
Floor4 StaticMeshAc/  
Floor5 StaticMeshAc/  
Floor6 StaticMeshAc/  
Floor7 StaticMeshAc/  
31 actors View Options

Content Browser Output Log

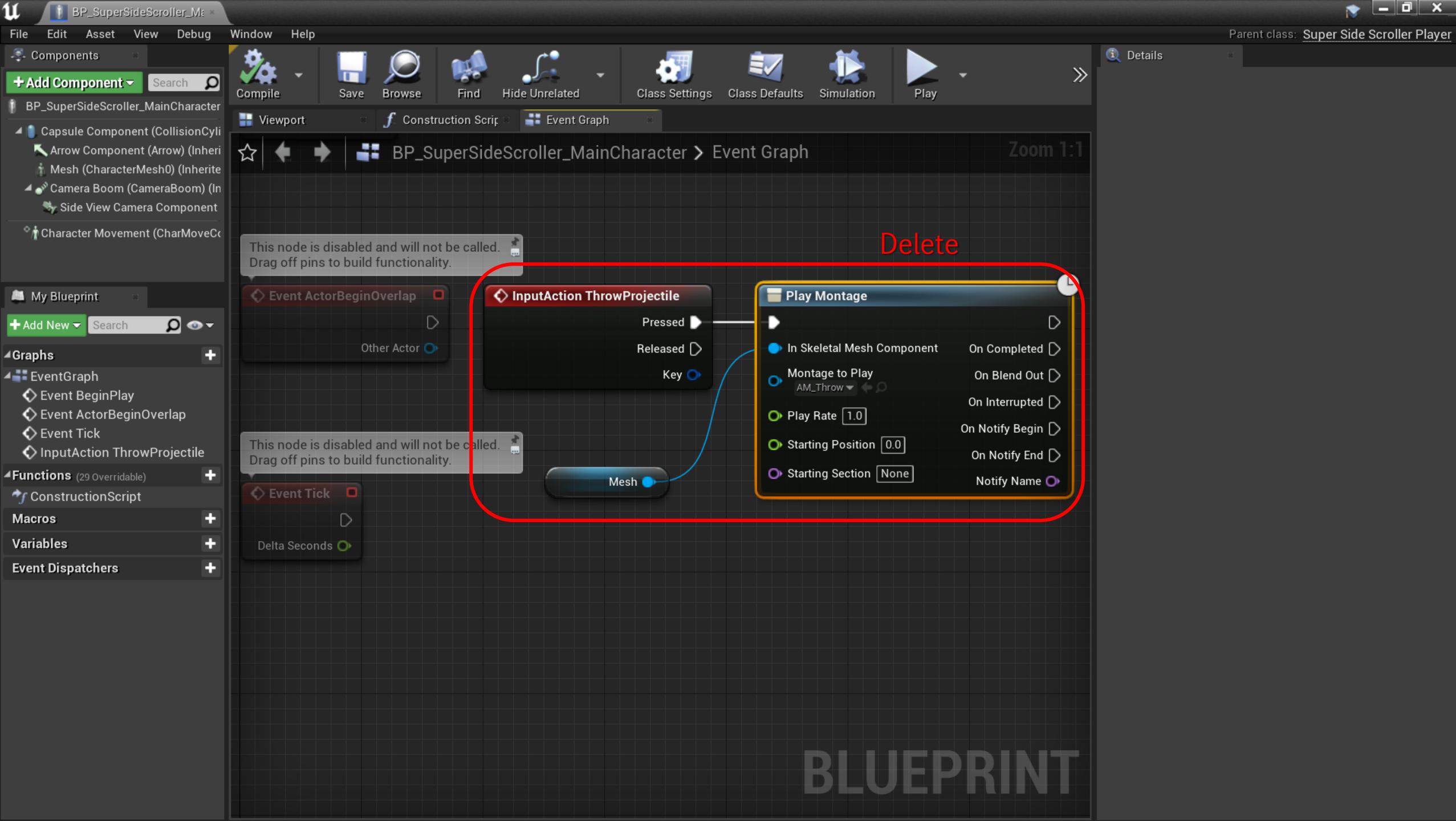
Add/Import Save All Content MainCharacter Blueprints

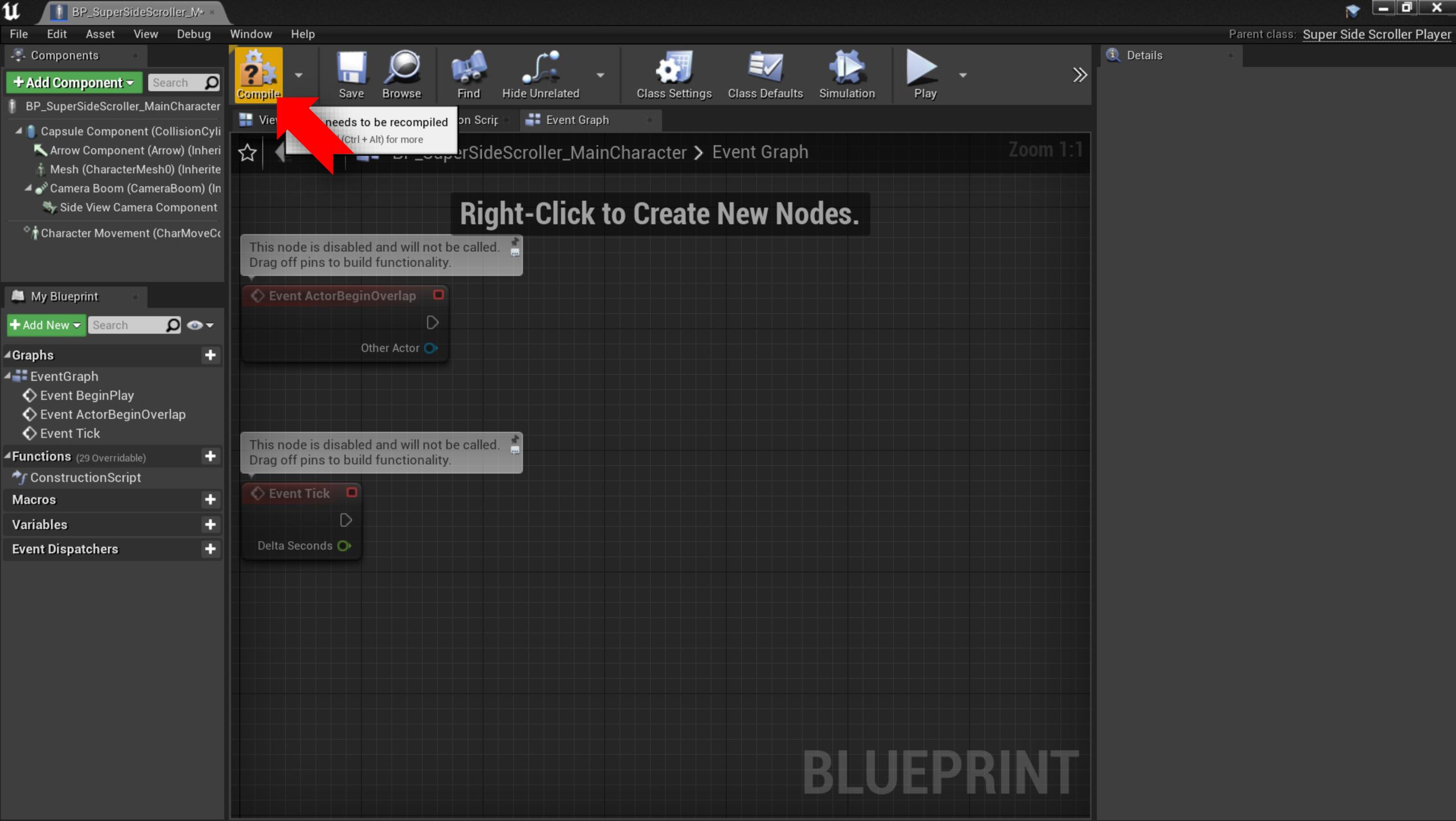
Content Paths Filters Search Blueprints

Content  
  Enemy  
    AI  
    Blueprints  
    Materials  
  Geometry  
  MainCharacter  
    Animation  
    Blueprints  
    Mesh  
    Projectile  
    Mannequin

AnimBP\_SuperSideScroller\_MainCharacter  
BP\_SuperSideScroller\_MainCharacter

2 items (1 selected) View Options





파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Super...ller

Develop Win64 로컬 Windows 디버거

SuperSideScroller\_Player.cpp SuperSideScroller\_Player.h\* Anim\_ProjectileNotify.cpp Anim\_ProjectileNotify.h

```
#include "SuperSideScroller_Player.generated.h"

// Class definition
UCLASS()
class SUPERSIDESCROLLER_API ASuperSideScroller_Player : public ASuperSideScrollerCharacter
{
    GENERATED_BODY()

public:
    // Constructor
    ASuperSideScroller_Player();

protected:
    // Override base character class function to setup our player input component
    virtual void SetupPlayerInputComponent(class UInputComponent* InputComponent) override;

    // Sprinting
    void Sprint();
    // StopSprinting
    void StopSprinting();

    // ThrowProjectile
    void ThrowProjectile();

private:
    // Bool to control if we are sprinting. Failsafe.
    bool bIsSprinting;

    UPROPERTY(EditAnywhere)
    class UAnimMontage* ThrowMontage;
};


```

Ctrl+S

SuperSideScroller 탐색기

SuperSideScroller 탐색기 검색(Ctrl+Shift+F)

SuperSideScroller (2/2개 프로젝트)

- Engine
- UE4
- Games
- SuperSideScroller
  - 참조
  - 외부 종속성
  - Config
  - Source
    - SuperSideScroller
      - Anim\_ProjectileNotify.cpp
      - Anim\_ProjectileNotify.h
      - EnemyBase.cpp
      - EnemyBase.h
      - PlayerProjectile.cpp
      - PlayerProjectile.h
      - SuperSideScroller.Build.cs
      - SuperSideScroller.cpp
      - SuperSideScroller.h
      - SuperSideScroller\_Player.cpp
      - SuperSideScroller\_Player.h
      - SuperSideScrollerCharacter.cpp
      - SuperSideScrollerCharacter.h
      - SuperSideScrollerGameMode.cpp
      - SuperSideScrollerGameMode.h
      - SuperSideScroller.Target.cs
      - SuperSideScrollerEditor.Target.cs
    - SuperSideScroller.uproject

준비 소스 제어에 추가

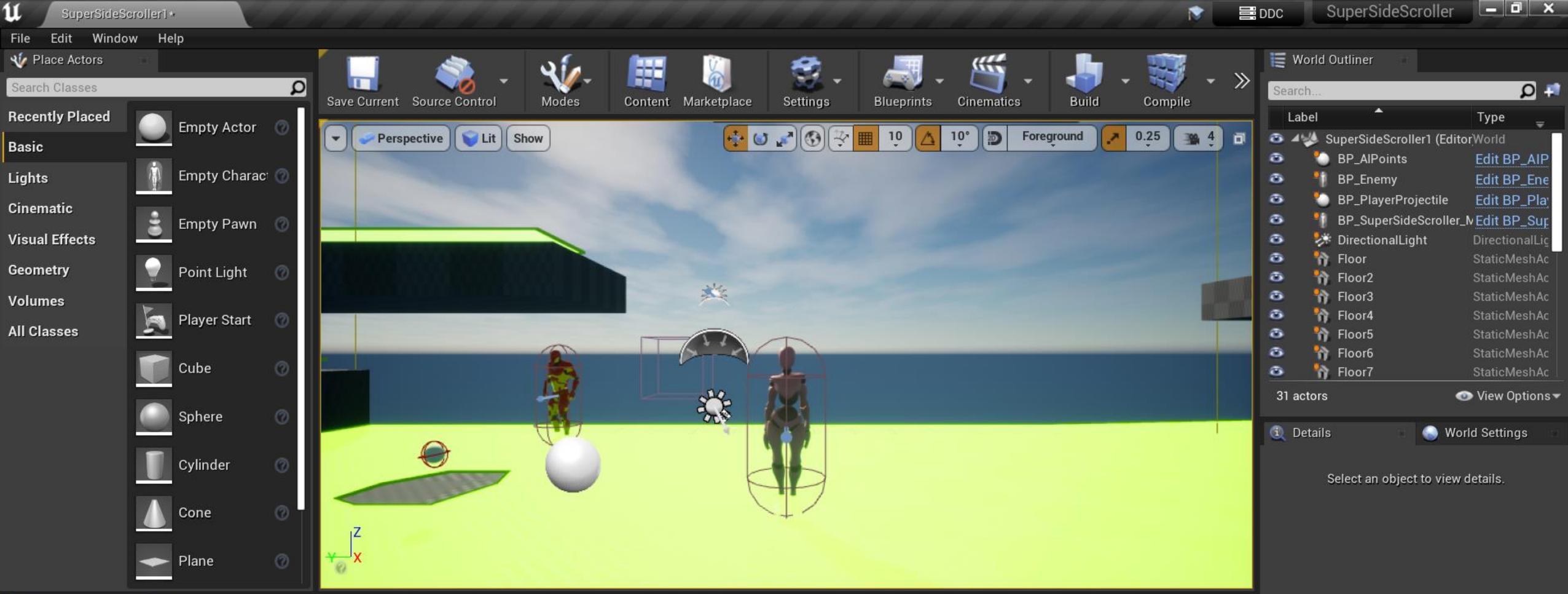
The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Top Bar:** 파일(F), 편집(E), 보기(V), Git(G), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(I), 확장(X), 창(W), 도움말(H), 검색 (Ctrl+Q), Live Share.
- Solution Explorer:** Shows the project structure:
  - SuperSideScroller
  - Engine
  - Games
  - SuperSideScroller
    - 참조
    - 외부 종속성
    - Config
    - Source
      - SuperSideScroller
        - Anim\_ProjectileNotify.cpp
        - Anim\_ProjectileNotify.h
        - EnemyBase.cpp
        - EnemyBase.h
        - PlayerProjectile.cpp
        - PlayerProjectile.h
        - SuperSideScroller.Build.cs
        - SuperSideScroller.cpp
        - SuperSideScroller.h
        - SuperSideScroller\_Player.cpp
        - SuperSideScroller\_Player.h
        - SuperSideScrollerCharacter.h
        - SuperSideScrollerCharacter.h
        - SuperSideScrollerGameMode.h
        - SuperSideScrollerGameMode.h
        - SuperSideScroller.Target.cs
        - SuperSideScrollerEditor.Target.cs
      - SuperSideScroller\_Player.h
      - SuperSideScrollerCharacter.h
      - SuperSideScrollerCharacter.h
      - SuperSideScrollerGameMode.h
      - SuperSideScrollerGameMode.h
      - SuperSideScroller.Target.cs
      - SuperSideScrollerEditor.Target.cs
  - Code Editor:** The file `SuperSideScroller_Player.cpp` is open. A red box highlights the line `#include "Animation/AnimInstance.h"`. The code implements the `ASuperSideScroller_Player` class, handling player input and projectile throwing.
  - Status Bar:** 100 %, 문제가 검색되지 않음 (No problems found), 줄: 7, 문자: 36, 탭, CRLF.
  - Bottom Bar:** 저장되었습니다. (File saved), ↑ 소스 제어에 추가 ↑ (Add to source control ↑), Git 변경 내용 (Git change history).

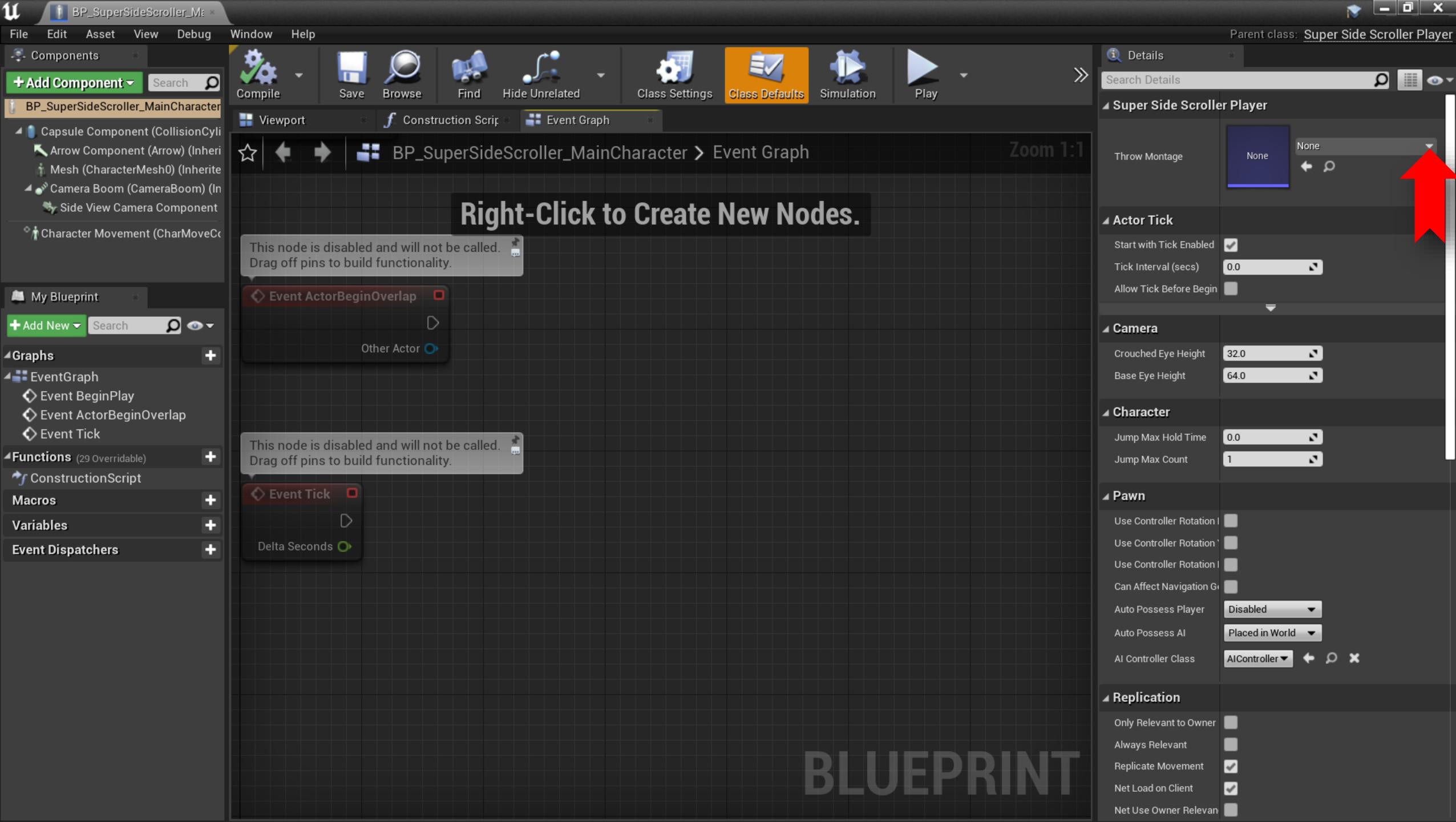


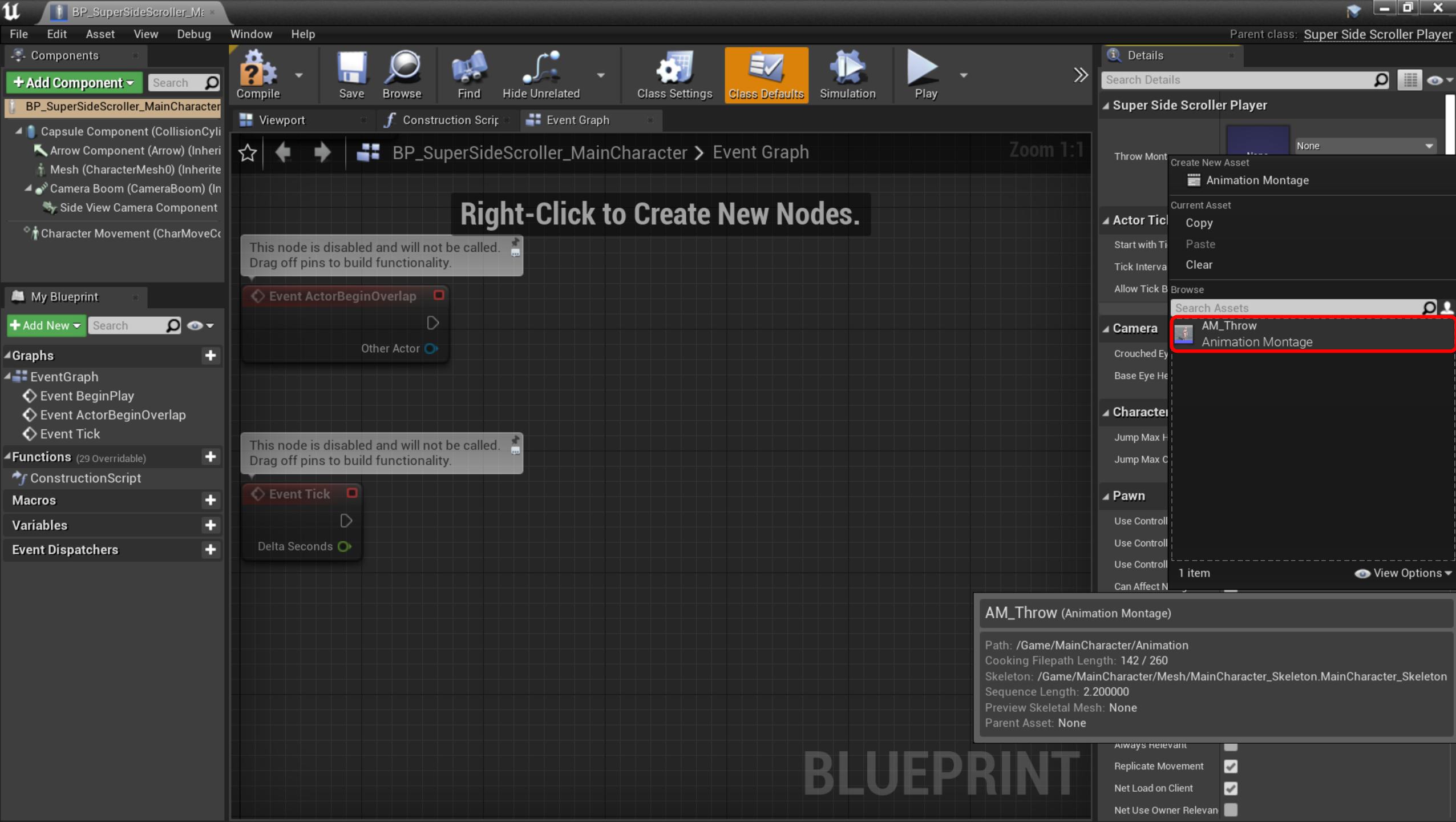
The screenshot shows the Microsoft Visual Studio interface. The top menu bar includes '파일(F)', '편집(E)', '보기(V)', 'Git(G)', '프로젝트(P)', '빌드(B)', '디버그(D)', '테스트(S)', '분석(N)', '도구(T)', '확장(X)', '창(W)', '도움말(H)', and '검색 (Ctrl+Q)'. A red arrow points to the '빌드(B)' menu item. The 'Develop' dropdown is also visible. The main code editor window displays C++ code for 'ASuperSideScroller\_Player' class. To the right, the 'Solution Explorer' shows the project structure for 'SuperSideScroller' under 'SuperSideScroller'. The bottom status bar indicates '100 %' completion, '문제가 검색되지 않음' (No errors found), and file details like '줄: 58' (Line: 58), '문자: 64' (Characters: 64), '열: 73' (Column: 73), '탭' (Tab), 'CRLF', and '슬루션 탐색기' (Solution Explorer). A 'Live Share' icon is in the top right corner.

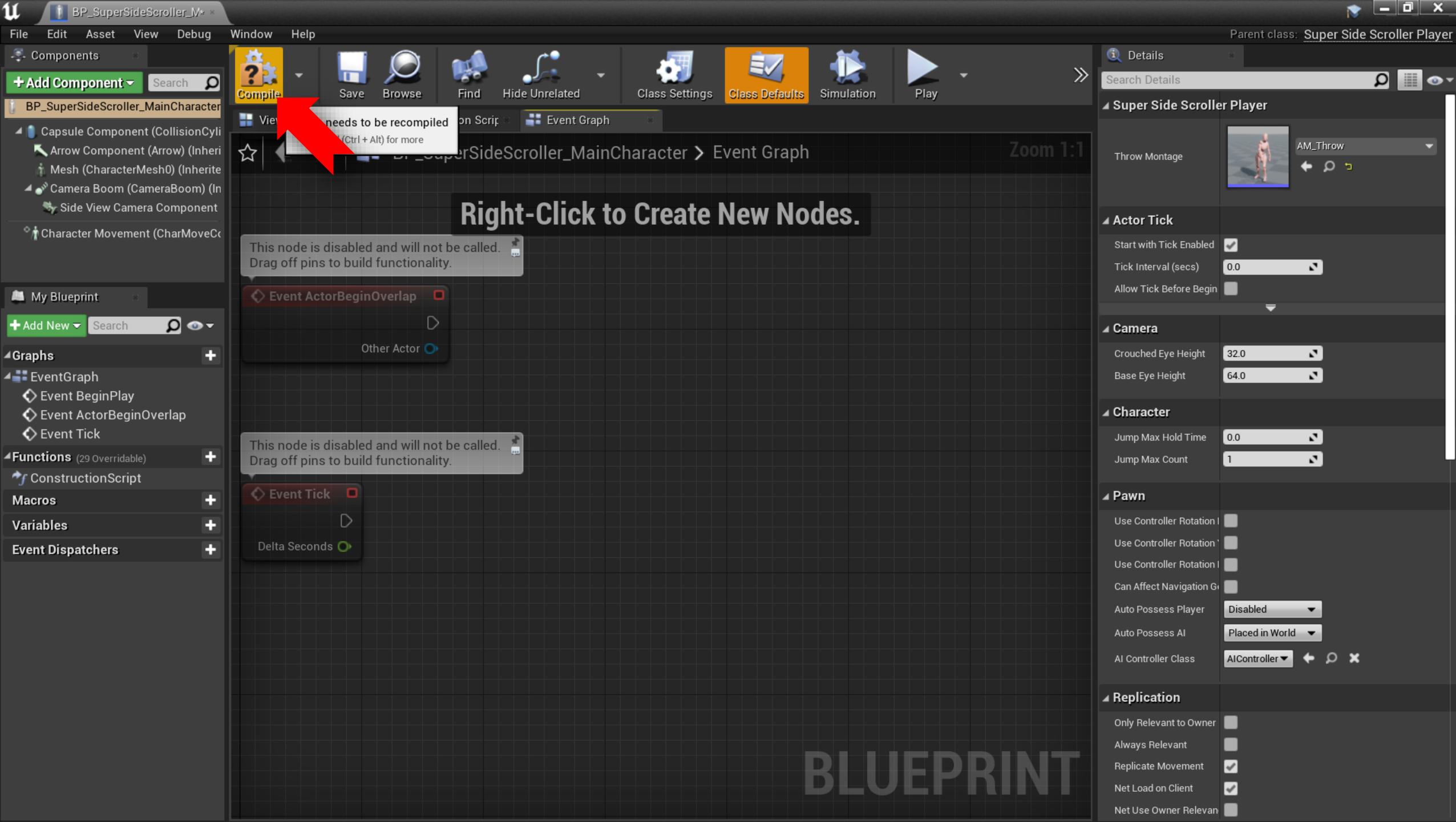
```
33 void ASuperSideScroller_Player::Spr
34 {
35     //First check the blsSprinting
36     if (!blsSprinting) {
37         blsSprinting = true;
38         GetCharacterMovement() ->Max
39     }
40 }
41
42
43 void ASuperSideScroller_Player::Sto
44 {
45     //First check the blsSprinting
46     if (blsSprinting) {
47         blsSprinting = false;
48         GetCharacterMovement() ->MaxWalkSpeed = 300.0f;
49     }
50 }
51
52 void ASuperSideScroller_Player::ThrowProjectile()
53 {
54     //UE_LOG(LogTemp, Warning, TEXT("THROW PROJECTILE!"));
55     if (ThrowMontage) {
56         bool blsMontagePlaying = GetMesh() ->GetAnimInstance() ->Montage_IsPlaying(ThrowMontage);
57         if (!blsMontagePlaying) {
58             GetMesh() ->GetAnimInstance() ->Montage_Play(ThrowMontage, 2.0f);
59         }
60     }
61 }
62
```

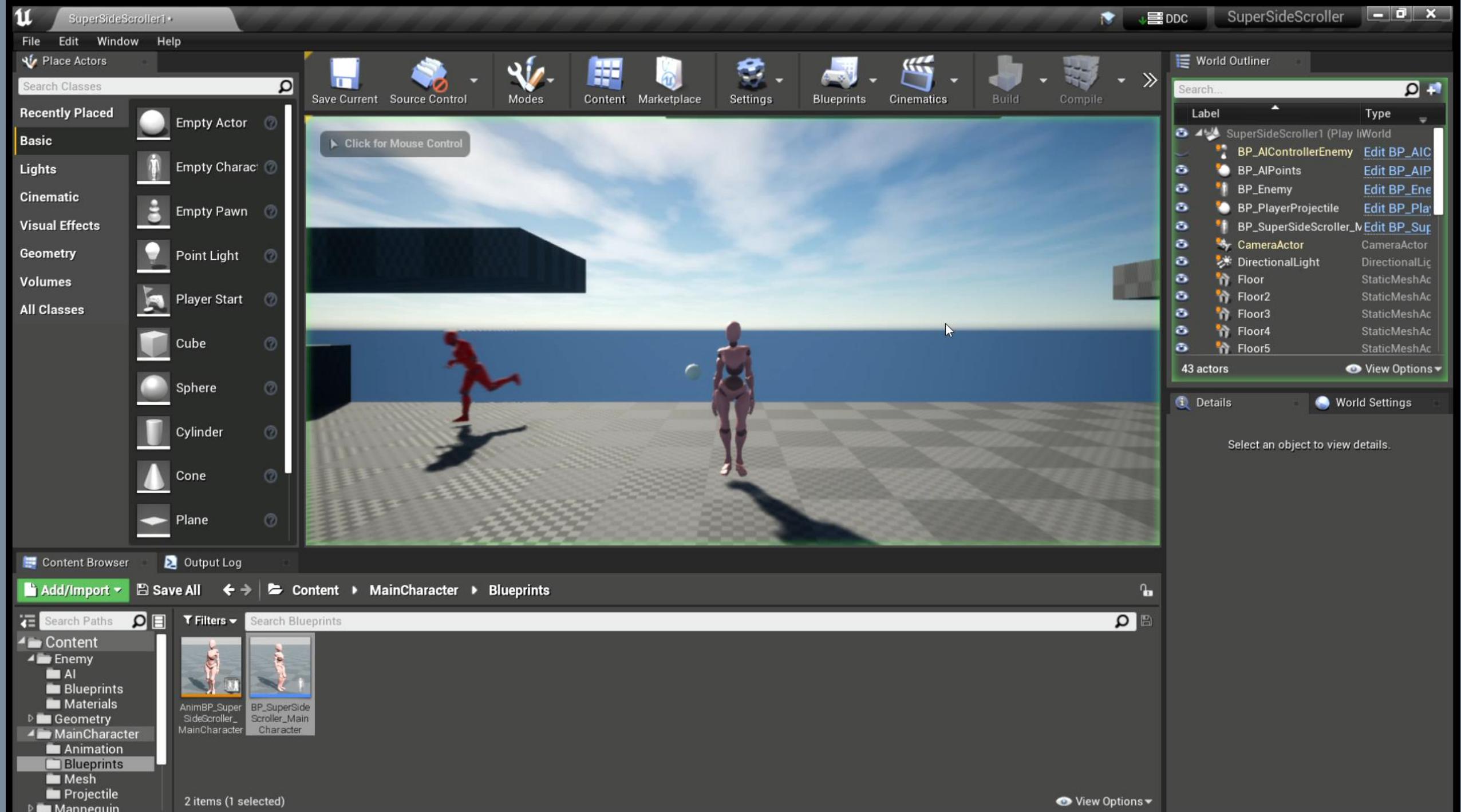


This screenshot shows the Content Browser panel of the Unreal Engine 4 Editor. The top navigation bar includes Add/Import, Save All, Content, MainCharacter, and Blueprints. The left sidebar shows the Content tree with Content, Enemy, Geometry, MainCharacter, and Mannequin. The MainCharacter folder is expanded, showing sub-folders like Animation, Blueprints, Mesh, Projectile, and Mannequin. The Content Browser lists items under the MainCharacter folder: AnimBP\_SuperSideScroller\_MainCharacter and BP\_SuperSideScroller\_MainCharacter. The BP\_SuperSideScroller\_MainCharacter item is highlighted with a yellow selection box and has a red arrow pointing to it from the previous screenshot. The bottom status bar indicates "2 items (1 selected)" and "View Options".







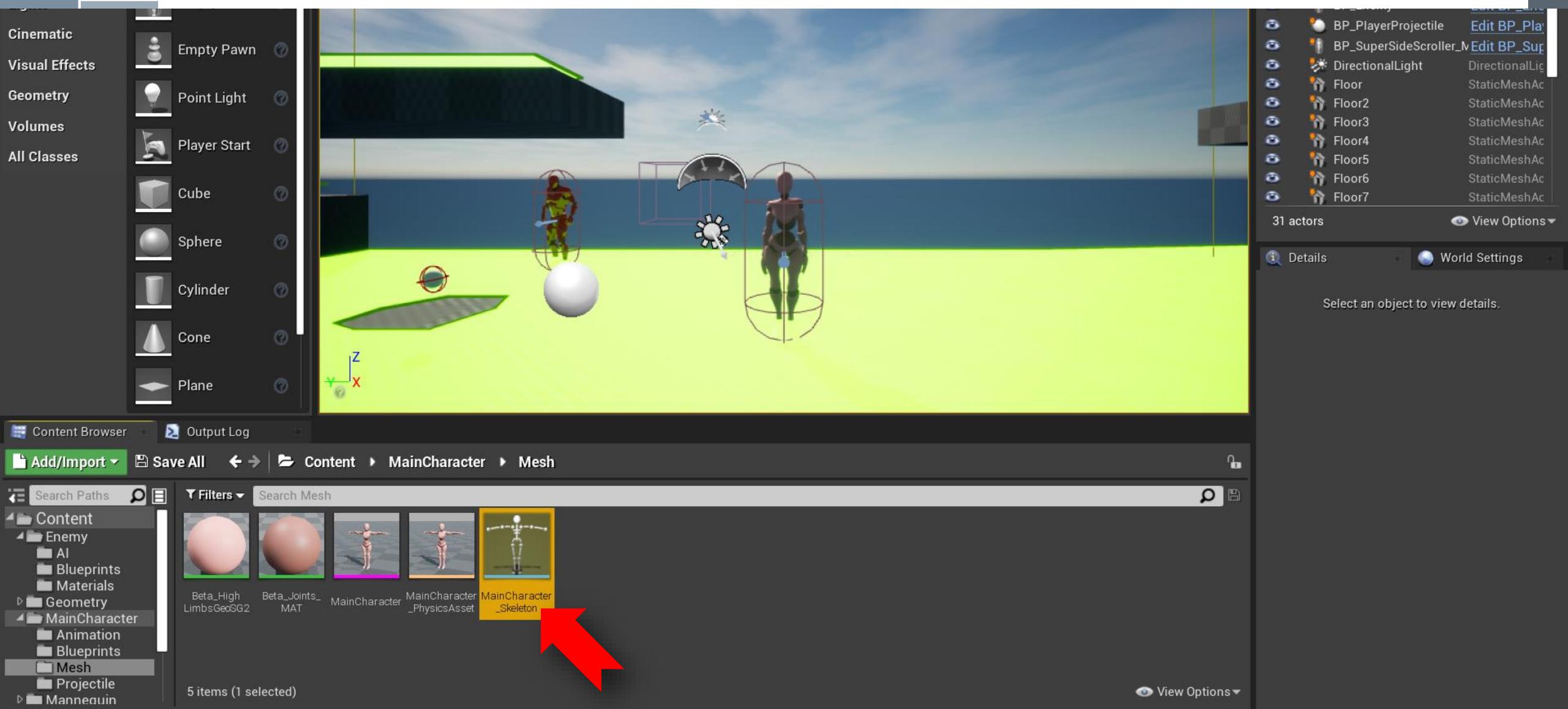


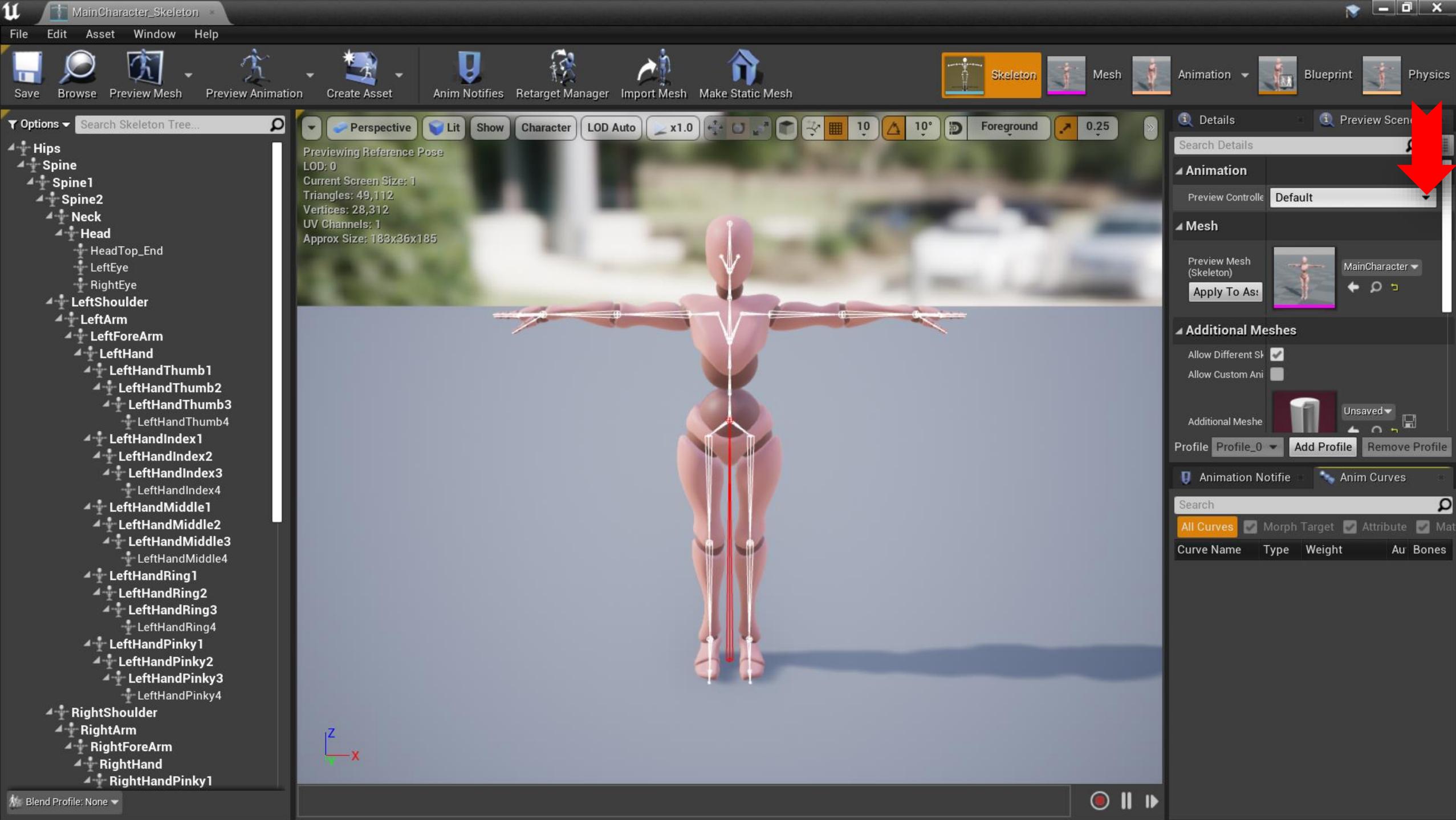


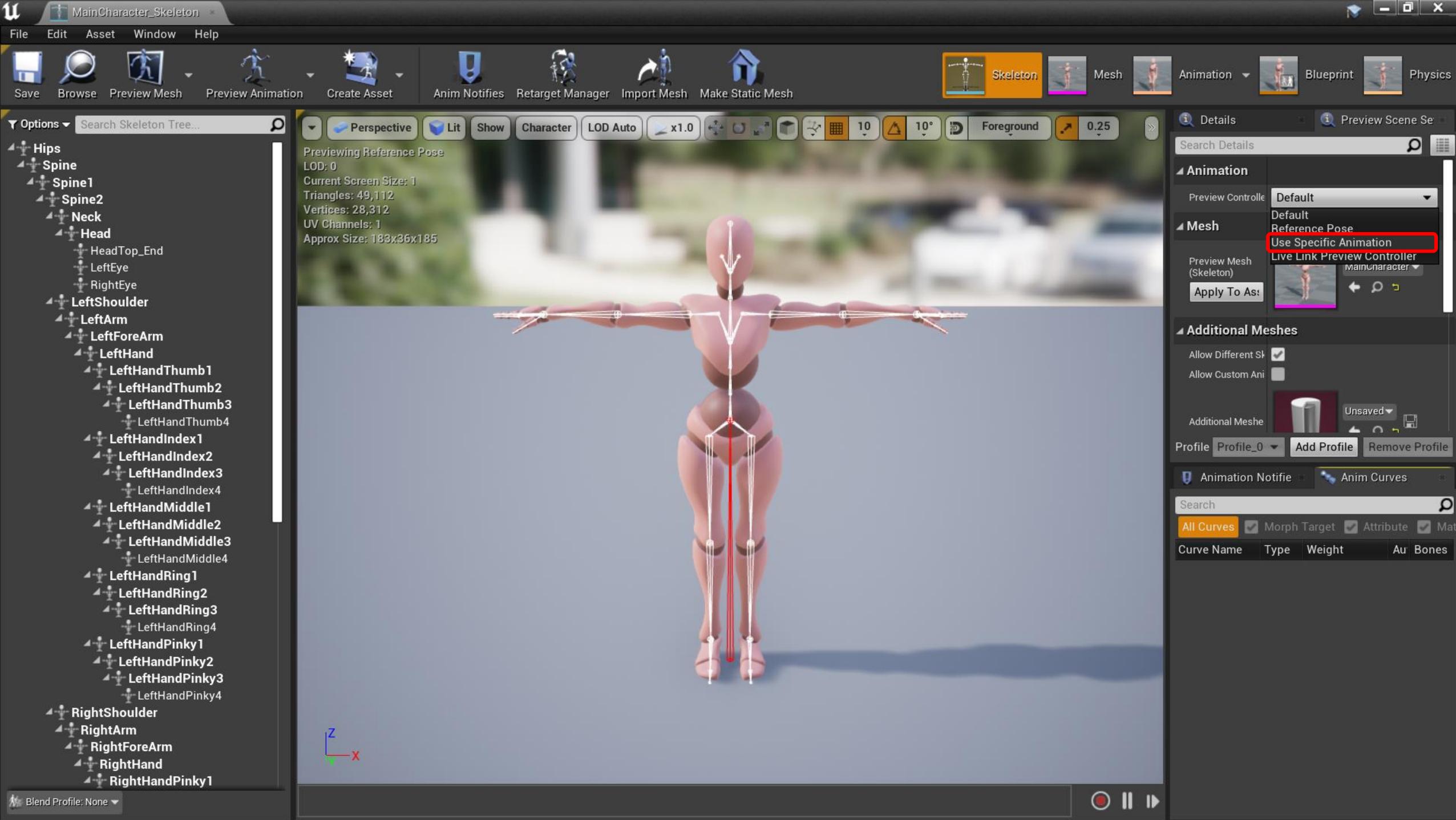
# Game World and Spawning Objects

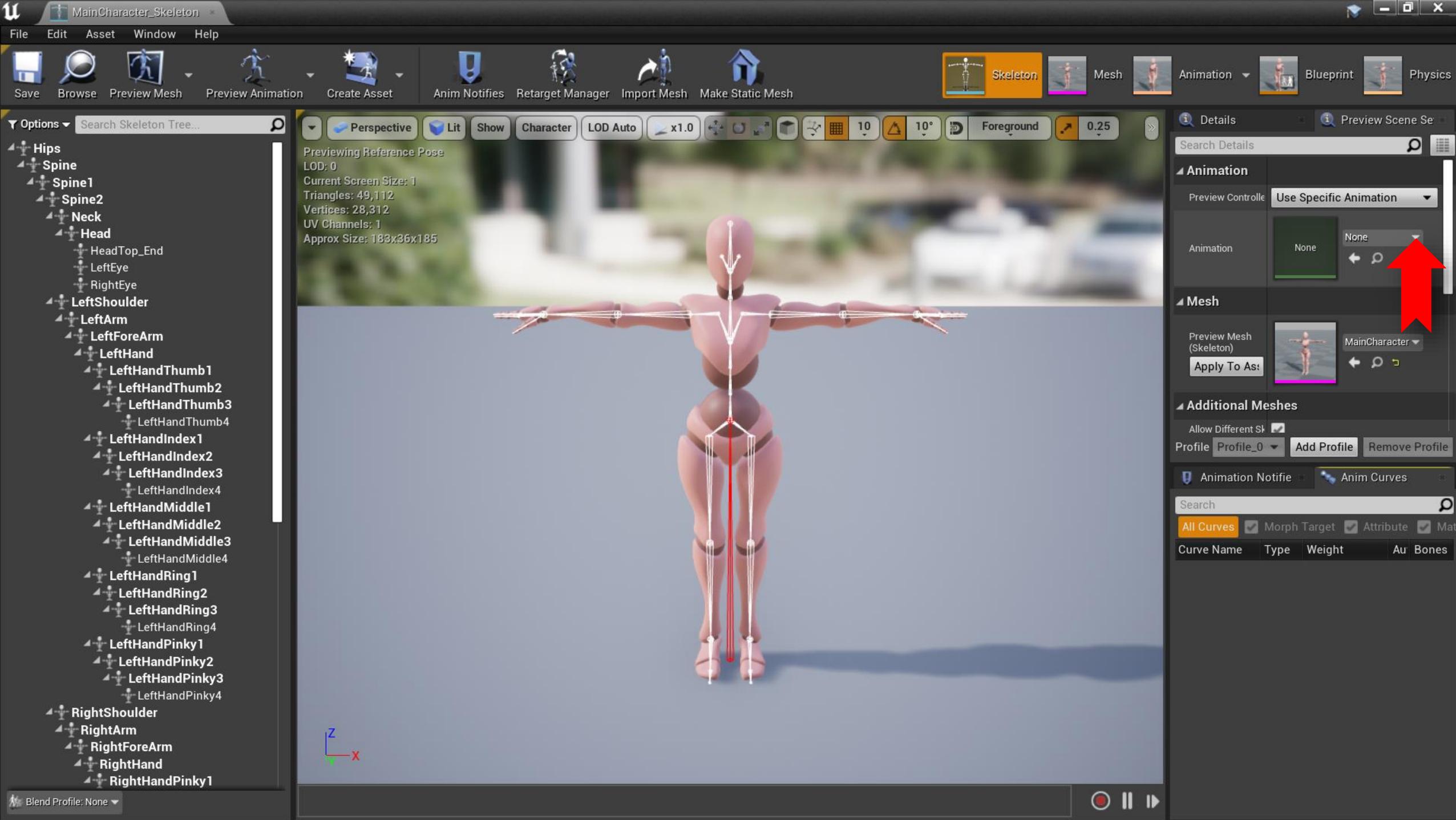
- › The **UWorld** class object is the single, top-level object that represents your level
  - Spawning and removing objects from the world, detecting when levels are being changed or streamed in/out, and even performing line traces to assist with inter-object detection
- › The **UWorld** class has multiple variations of the **SpawnActor()** function, depending on how you want to spawn the object.
  - **UClass**: the class of the object that you want to spawn in
  - **FActorSpawnParameters**: a struct of variables that give the spawned object more context and references to what has spawned it
  - **Transform**: the location, rotation, and scale properties

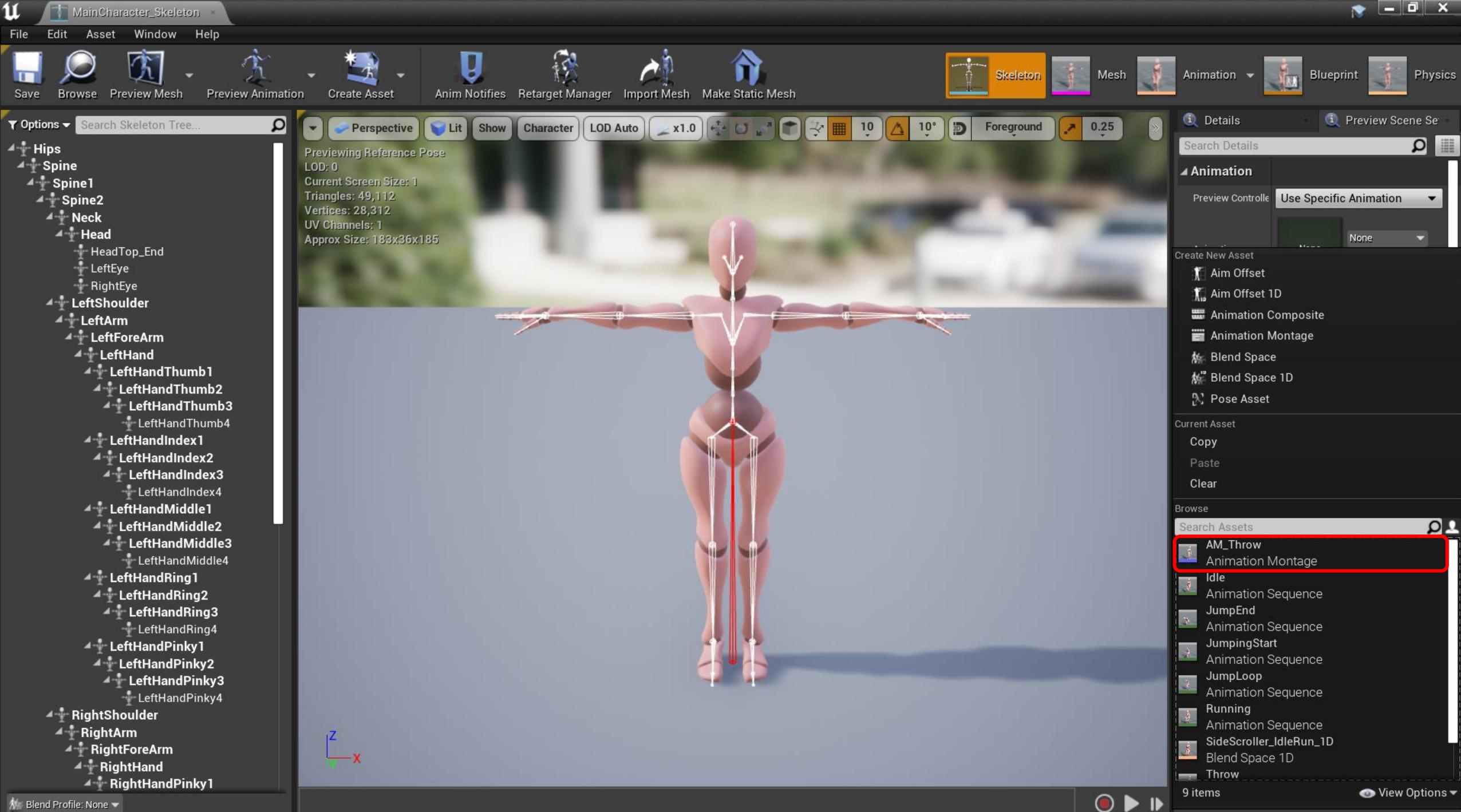
# Exercise 14.04: Creating the Projectile Spawn Socket

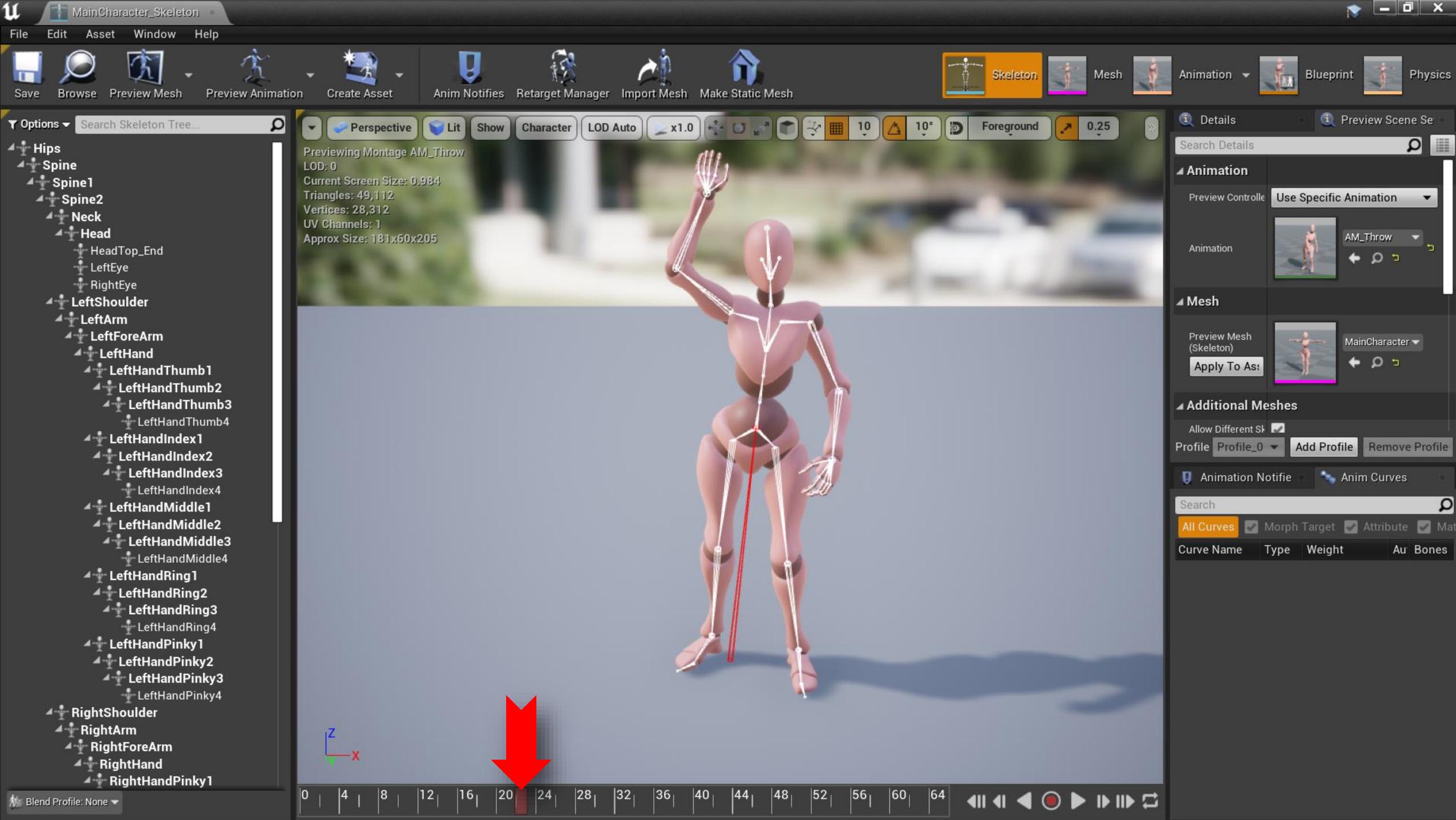
















Right-Click

- Selected Bone Actions
  - Copy Selected Bone Names
  - Reset Selected Bone Transforms
  - Add Socket
  - Paste Sockets Ctrl+V
  - Paste Sockets To Selected Bone Ctrl+Shift+V
  - Add Virtual Bone
  - LOD Bone Reduction
  - Remove Selected...
  - Remove Children...
- Attached Assets Actions
  - Add Preview Asset
  - Remove All Attached Assets

Add a socket to this bone in the skeleton (disabled when socket filter is set to "Mesh Sockets" or "Sockets Hidden" mode)



MainCharacter\_Skeleton.x

File Edit Asset Window Help

Save Browse Preview Mesh Preview Animation Create Asset Anim Notifies Retarget Manager Import Mesh Make Static Mesh Skeleton Mesh Animation Blueprint Physics

Options Search Skeleton Tree... Details Preview Scene

Previewing Montage AM\_Throw  
LOD: 0  
Current Screen Size: 0.984  
Triangles: 49,112  
Vertices: 28,312  
UV Channels: 1  
Approx Size: 181x60x205

ProjectileSocket

Socket Parameters  
Socket Name: ProjectileSocket  
Bone Name: RightHand  
Relative Location: X: 12.9 Y: 25.4 Z: -7.1  
X: 12.961717 Y: 25.44845 Z: -7.120584  
Relative Rotation: X: 0.0 Y: 0.0 Z: 0.0  
Relative Scale: X: 1.0 Y: 1.0 Z: 1.0  
Force Always Animate:

Animation Notify Anim Curves

Search All Curves Morph Target Attribute Material  
Curve Name Type Weight Au Bones

Blend Profile: None

0 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 | 44 | 48 | 52 | 56 | 60 | 64

LeftHandPinky2 LeftHandPinky3 LeftHandPinky4 RightShoulder RightArm RightForeArm RightHand RightHandPinky1 RightHandPinky2 RightHandPinky3 RightHandPinky4 RightHandRing1 RightHandRing2 RightHandRing3 RightHandRing4 RightHandMiddle1 RightHandMiddle2 RightHandMiddle3 RightHandMiddle4 RightHandIndex1 RightHandIndex2 RightHandIndex3 RightHandIndex4 RightHandThumb1 RightHandThumb2 RightHandThumb3 RightHandThumb4

ProjectileSocket

LeftUpLeg LeftLeg LeftFoot LeftToeBase LeftToe\_End RightUpLeg RightLeg RightFoot RightToeBase RightToe\_End

Perspective Lit Show Character LOD Auto x1.0 10 10° Foreground 0.25

The screenshot displays the Skeletal Editor interface in the Unreal Engine 4. The main view shows a pink humanoid skeleton from a three-quarter perspective. A blue line extends from the right hand's center to a red cube labeled "ProjectileSocket". The Skeletal Hierarchy panel on the left lists various bones, with "ProjectileSocket" highlighted in yellow. The Details panel on the right shows the "Socket Parameters" for "ProjectileSocket", including its name, bone it's attached to ("RightHand"), and its relative location (X: 12.961717, Y: 25.44845, Z: -7.120584). The Animation Curves panel at the bottom allows for curve editing. The Timeline at the bottom shows frame numbers from 0 to 64.





# Exercise 14.05: Preparing the SpawnProjectile() Function

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 참(W) 도움말(H) 검색 (Ctrl+Q) Super...ller - Live Share X

SuperSideScroller\_Player.cpp SuperSideScroller\_Player.h\* Anim\_ProjectileNotify.cpp Anim\_ProjectileNotify.h

```
17 public:  
18     //Constructor  
19     ASuperSideScroller_Player();  
20  
21     void SpawnProjectile();  
22  
23 protected:  
24     //Override base character class function to setup our player input component  
25     virtual void SetupPlayerInputComponent(class UInputComponent* InputComponent) override;  
26  
27     //Sprinting  
28     void Sprint();  
29     //StopSprinting  
30     void StopSprinting();  
31  
32     //ThrowProjectile  
33     void ThrowProjectile();  
34  
35 private:  
36     //Bool to control if we are sprinting. Failsafe.  
37     bool bIsSprinting;  
38  
39     UPROPERTY(EditAnywhere)  
40     class UAnimMontage* ThrowMontage;  
41  
42     UPROPERTY(EditAnywhere)  
43     TSubclassOf<class APlayerProjectile> PlayerProjectile;
```

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+.)

솔루션 'SuperSideScroller' (2/2개 프로젝트)

- Engine
- UE4
- Games
- SuperSideScroller
  - 참조
  - 외부 종속성
  - Config
  - Source
    - SuperSideScroller
      - Anim\_ProjectileNotify.cpp
      - Anim\_ProjectileNotify.h
      - EnemyBase.cpp
      - EnemyBase.h
      - PlayerProjectile.cpp
      - PlayerProjectile.h
      - SuperSideScroller.Build.cs
      - SuperSideScroller.cpp
      - SuperSideScroller.h
      - SuperSideScroller\_Player.cpp
      - SuperSideScroller\_Player.h
      - SuperSideScrollerCharacter.cpp
      - SuperSideScrollerCharacter.h

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Super...ller — □ ×

Live Share

SuperSideScroller\_Player.cpp\* SuperSideScroller\_Player.h Anim\_ProjectileNotify.cpp Anim\_ProjectileNotify.h

```
SuperSideScroller
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "SuperSideScroller_Player.h"
5 #include "Components/InputComponent.h"
6 #include "GameFramework/CharacterMovementComponent.h"
7 #include "Animation/AnimInstance.h"
8 #include "PlayerProjectile.h"
9 #include "Engine/World.h"
10 #include "Components/SphereComponent.h"
11
12 ASuperSideScroller_Player::ASuperSideScroller_Player()
13 {
14     //Set sprinting to false by default.
15     bIsSprinting = false;
16
17     //Set our Max Walk Speed to 300.0f
18     GetCharacterMovement()->MaxWalkSpeed = 300.0f;
19 }
20
21 void ASuperSideScroller_Player::SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent)
22 {
23     //Not always necessary, but good practice to call the function in the base class with Super.
24     Super::SetupPlayerInputComponent(PlayerInputComponent);
25
26     //Bind pressed action Sprint to your Sprint function
27     PlayerInputComponent->BindAction("Sprint", IE_Pressed, this, &ASuperSideScroller_Player::Sprint);
28     //Bind released action Sprint to your StopSprinting function
29     PlayerInputComponent->BindAction("Sprint", IE_Released, this, &ASuperSideScroller_Player::StopSprinting);
30
31     //Bind pressed action ThrowProjectile to your ThrowProjectile function
32     PlayerInputComponent->BindAction("ThrowProjectile", IE_Pressed, this, &ASuperSideScroller_Player::ThrowProjectile);
33
34 }
```

100 % 문제가 검색되지 않음 출: 79 문자: 72 열: 84 탭 CRLF

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+Shift+F)

슬루션 탐색기 'SuperSideScroller' (2/2개 프로젝트)

- Engine
- UE4
- Games
- SuperSideScroller
  - 참조
  - 외부 종속성
  - Config
  - Source
    - SuperSideScroller
      - Anim\_ProjectileNotify.cpp
      - Anim\_ProjectileNotify.h
      - EnemyBase.cpp
      - EnemyBase.h
      - PlayerProjectile.cpp
      - PlayerProjectile.h
    - SuperSideScroller.Build.cs
    - SuperSideScroller.cpp
    - SuperSideScroller.h
    - SuperSideScroller\_Player.cpp
    - SuperSideScroller\_Player.h
    - SuperSideScrollerCharacte...
    - SuperSideScrollerCharacte...
    - SuperSideScrollerGameMc...
    - SuperSideScrollerGameMc...
    - SuperSideScroller.Target.cs
    - SuperSideScrollerEditor.Targe...
  - SuperSideScroller.uproject

슬루션 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(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Super...ller — □ ×

Live Share

SuperSideScroller\_Player.cpp\* SuperSideScroller\_Player.h Anim\_ProjectileNotify.cpp Anim\_ProjectileNotify.h

SuperSideScroller

```
54 void ASuperSideScroller_Player::ThrowProjectile()
55 {
56     //UE_LOG(LogTemp, Warning, TEXT("THROW PROJECTILE!"));
57     if (ThrowMontage) {
58         bool bIsMontagePlaying = GetMesh() ->GetAnimInstance() ->Montage_IsPlaying(ThrowMontage);
59         if (!bIsMontagePlaying) {
60             GetMesh() ->GetAnimInstance() ->Montage_Play(ThrowMontage, 2.0f);
61         }
62     }
63 }
64
65 void ASuperSideScroller_Player::SpawnProjectile()
66 {
67     if (PlayerProjectile) {
68         UWorld* World = GetWorld();
69         if (World) {
70             FActorSpawnParameters SpawnParams;
71             SpawnParams.Owner = this;
72             FVector SpawnLocation = this ->GetMesh() ->GetSocketLocation(FName("ProjectileSocket"));
73             FRotator Rotation = GetActorForwardVector().Rotation();
74
75             APlayerProjectile* Projectile = World ->SpawnActor<APlayerProjectile>(PlayerProjectile,
76                 SpawnLocation, Rotation, SpawnParams);
77             if (Projectile) {
78                 Projectile->CollisionComp->MoveIgnoreActors.Add(SpawnParams.Owner);
79             }
80         }
81     }
82 }
83
84 }
```

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+Shift+F)

- 슬루션 'SuperSideScroller' (2/2개 프로젝트)
  - Engine
    - UE4
  - Games
    - SuperSideScroller
      - 참조
      - 외부 종속성
      - Config
      - Source
        - SuperSideScroller
          - Anim\_ProjectileNotify.cpp
          - Anim\_ProjectileNotify.h
          - EnemyBase.cpp
          - EnemyBase.h
          - PlayerProjectile.cpp
          - PlayerProjectile.h
        - SuperSideScroller.Build.cs
        - SuperSideScroller.cpp
        - SuperSideScroller.h
        - SuperSideScroller\_Player.cpp
        - SuperSideScroller\_Player.h
        - SuperSideScrollerCharacter.cpp
        - SuperSideScrollerCharacter.h
        - SuperSideScrollerGameMode.cpp
        - SuperSideScrollerGameMode.h
        - SuperSideScroller.Target.cs
        - SuperSideScrollerEditor.Target.cs

SuperSideScroller.uproject

준비

문제가 검색되지 않음

줄: 79 문자: 72 열: 84 템 CRLF

↑ 소스 제어에 추가 ↑



# Exercise 14.06: Updating the Anim\_ProjectileNotify Class

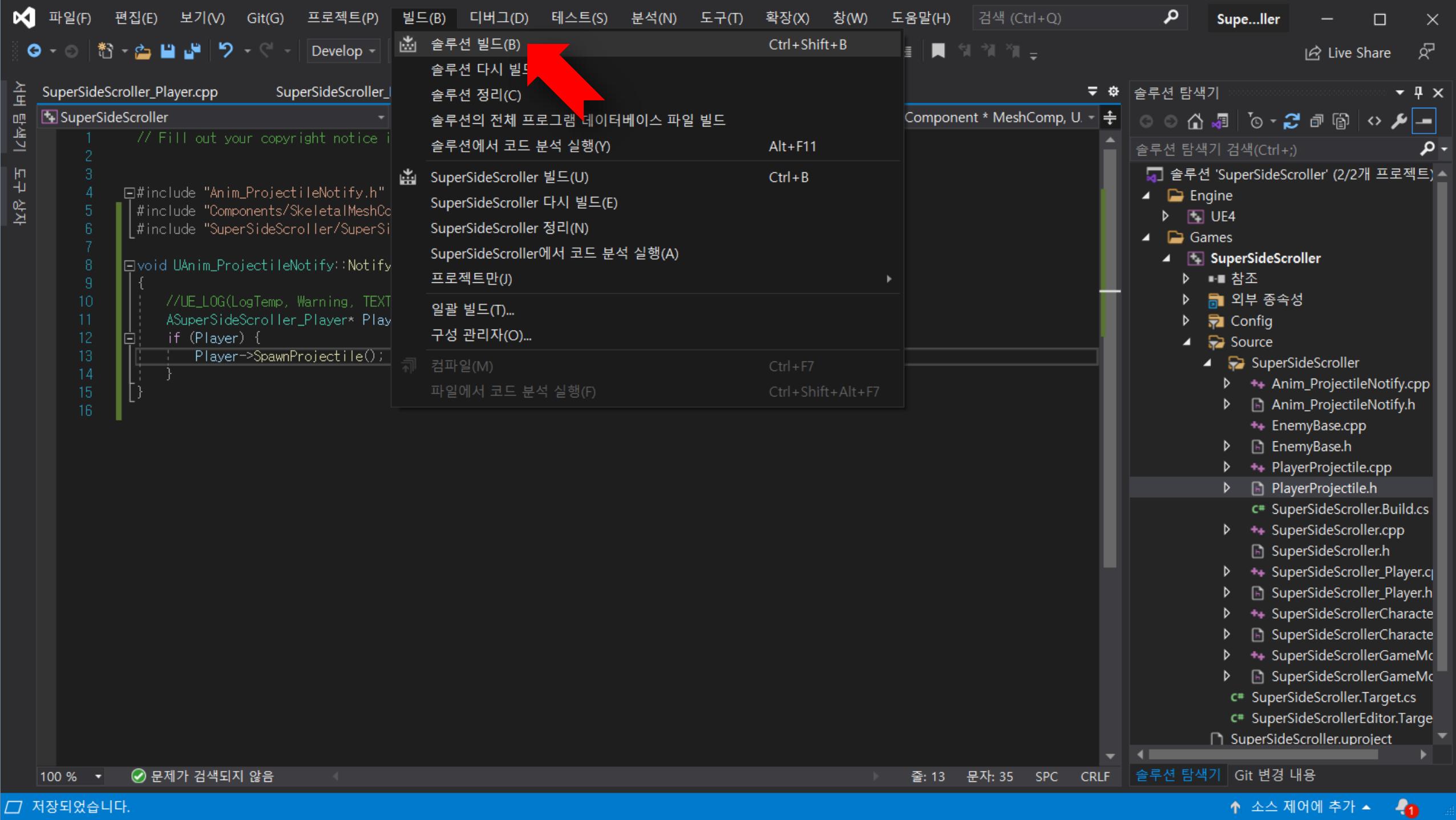
The screenshot shows the Unreal Engine Editor interface. The title bar includes the Unreal Engine logo, file menu, and various development tools. The code editor window displays `Anim_ProjectileNotify.cpp`, which contains the following code:

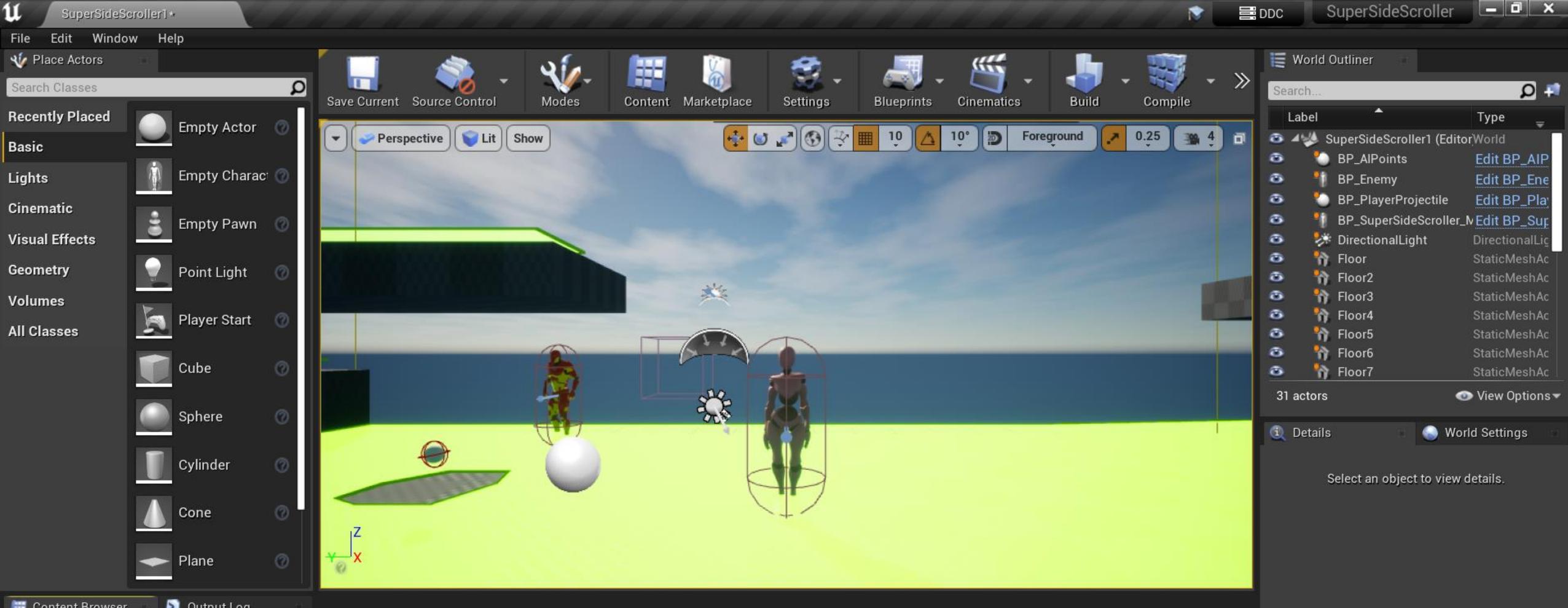
```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "Anim_ProjectileNotify.h"
5 #include "Components/SkeletalMeshComponent.h"
6 #include "SuperSideScroller/SuperSideScroller_Player.h"
7
8 void UAnim_ProjectileNotify::Notify(USkeletalMeshComponent* MeshComp, UAnimSequenceBase* Animation)
9 {
10     //UE_LOG(LogTemp, Warning, TEXT("Throw Notify"));
11     ASuperSideScroller_Player* Player = Cast<ASuperSideScroller_Player>(MeshComp->GetOwner());
12     if (Player) {
13         Player->SpawnProjectile();
14     }
15 }
```

The code editor has several UI elements like tabs, status bar, and toolbars. The right side features the Solution Explorer showing the project structure:

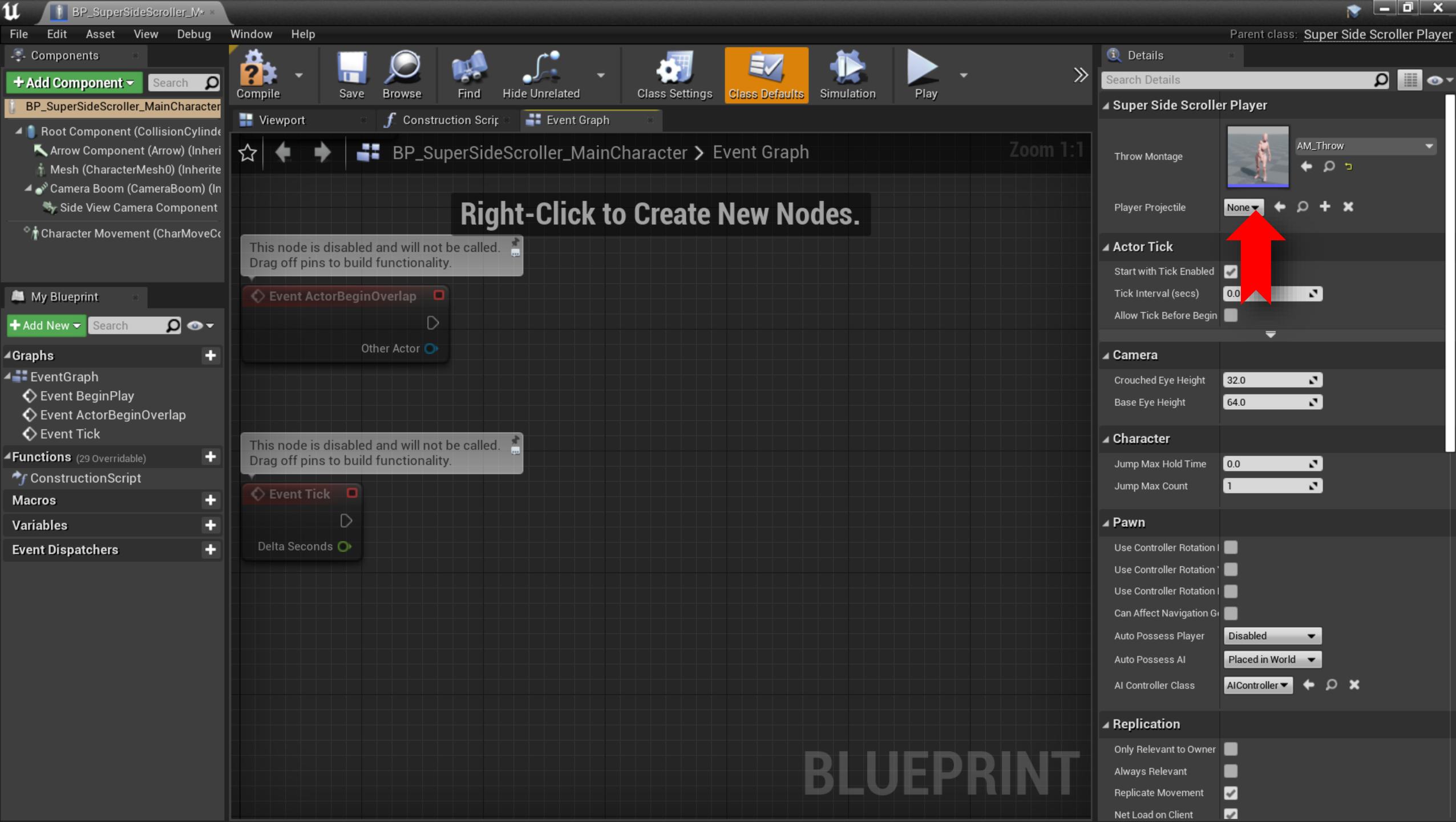
- SuperSideScroller
- Engine
- UE4
- Games
- SuperSideScroller
- Source
- SuperSideScroller
- Anim\_ProjectileNotify.cpp
- Anim\_ProjectileNotify.h
- EnemyBase.cpp
- EnemyBase.h
- PlayerProjectile.cpp
- PlayerProjectile.h
- SuperSideScroller.Build.cs
- SuperSideScroller.cpp
- SuperSideScroller.h
- SuperSideScroller\_Player.cpp
- SuperSideScroller\_Player.h
- SuperSideScrollerCharacter.cpp
- SuperSideScrollerCharacter.h

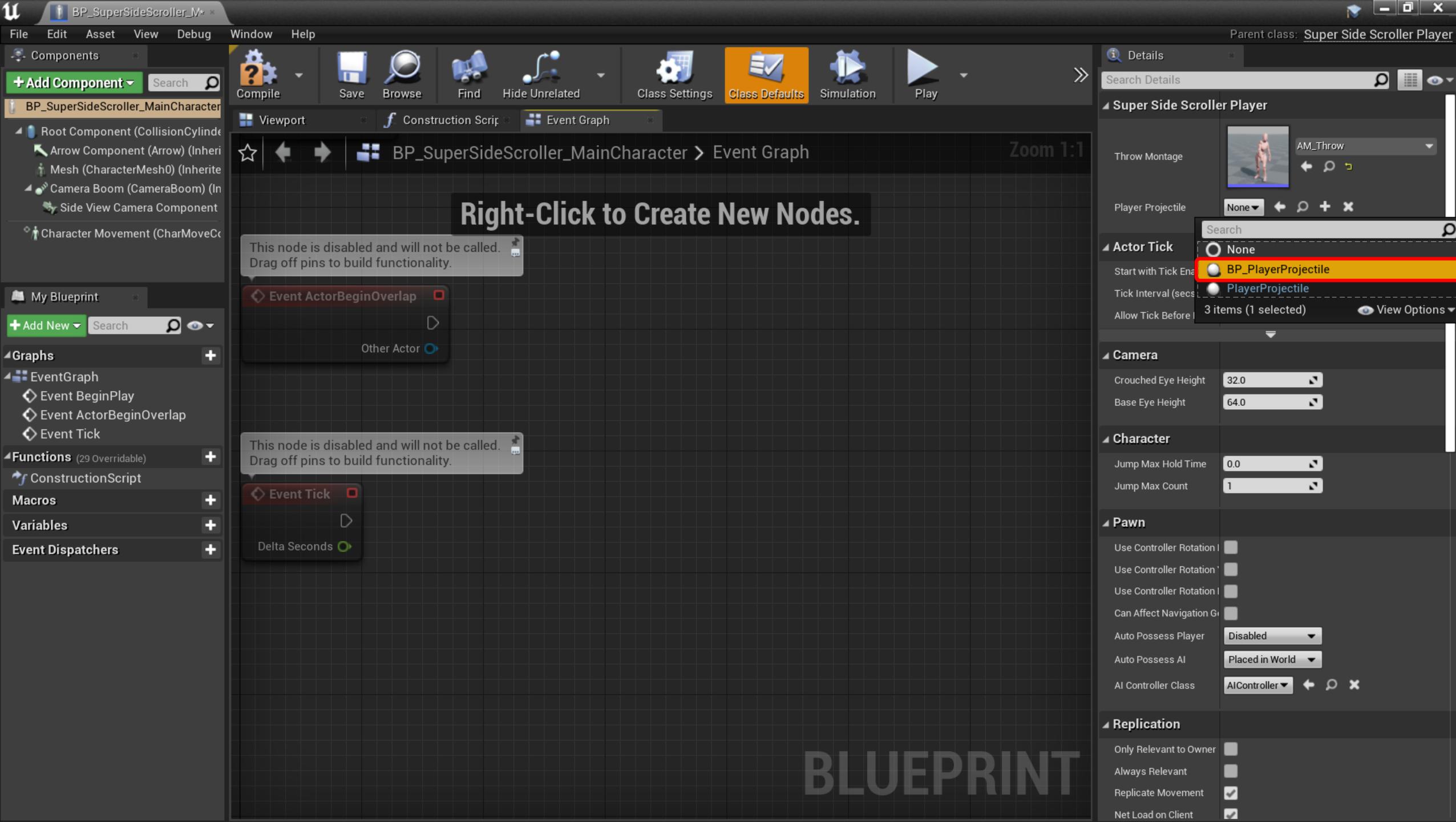
At the bottom right of the code editor, the text **Ctrl+S** is displayed in red.

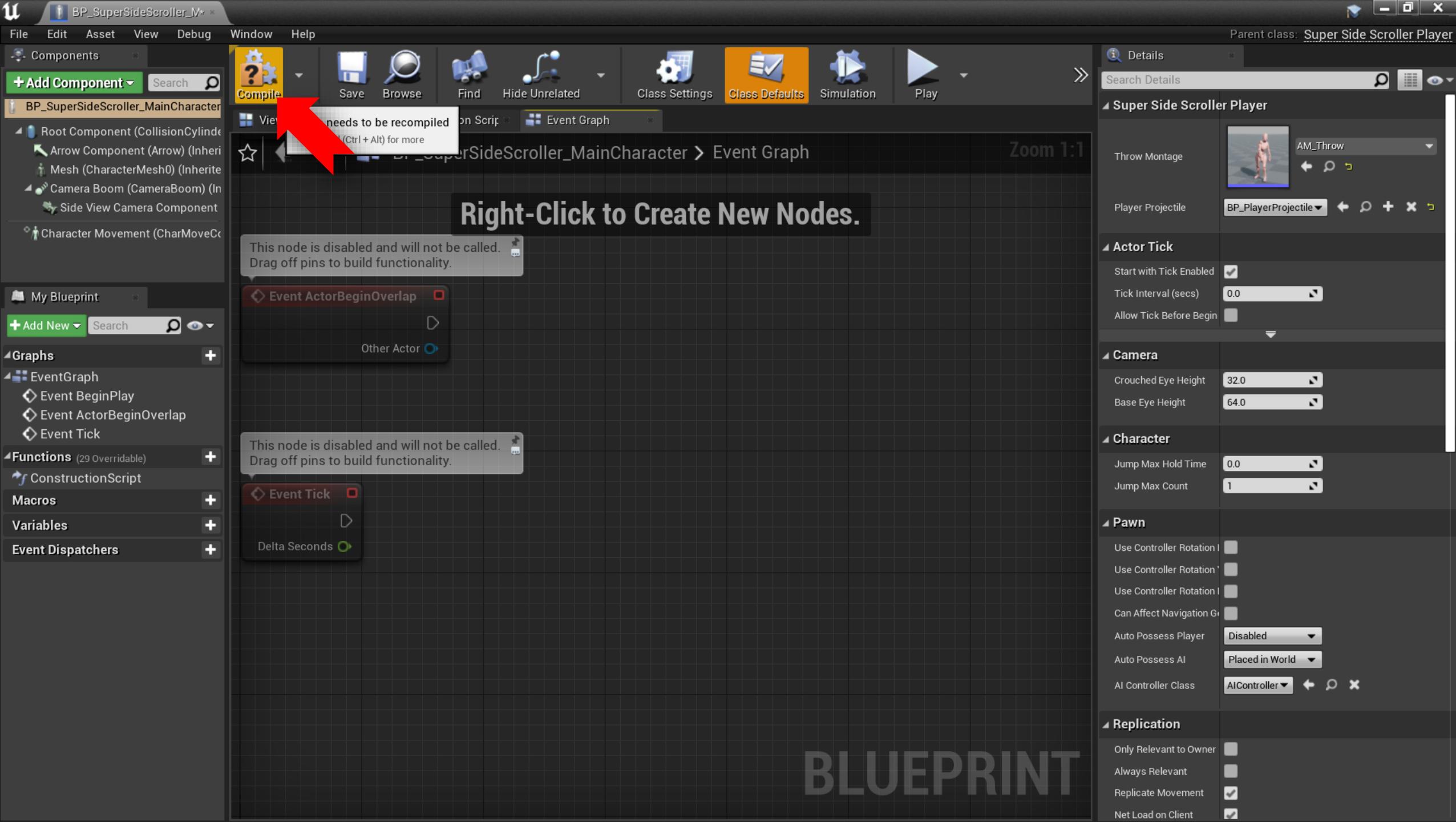


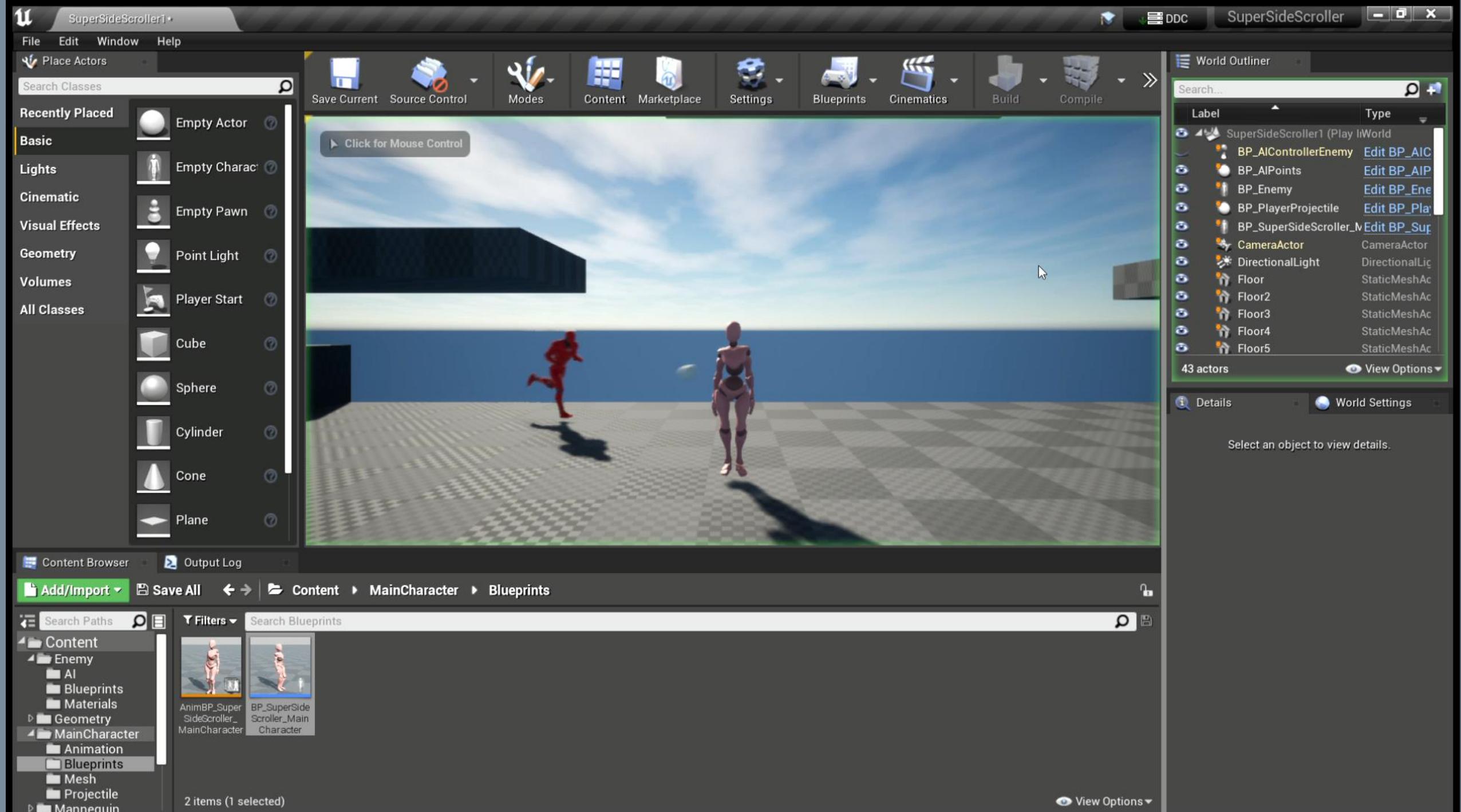


The screenshot shows the Content Browser interface. The top navigation bar includes Add/Import, Save All, Content, MainCharacter, and Blueprints. The left sidebar shows the Content tree with nodes for Content, Enemy, AI, Blueprints, Materials, Geometry, MainCharacter (selected), Animation, Blueprints, Mesh, Projectile, and Mannequin. The main area displays two Blueprint assets: AnimBP\_SuperSideScroller\_MainCharacter and BP\_SuperSideScroller\_MainCharacter. A red arrow points to the BP\_SuperSideScroller\_MainCharacter asset, which is highlighted with a yellow border. The status bar at the bottom indicates "2 items (1 selected)".











# Destroy Actors

- › The player character uses the **UWorld** class in order to spawn the projectile.
- › Unreal Engine 4 and its base **Actor** class come with a default function that you can use to destroy, or remove, an actor from the game world:

```
bool AActor::Destroy(bool bNetForce, bool bShouldModifyLevel);
```



# Exercise 14.07: Creating the DestroyEnemy() Function

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Super...ller - □ ×

Develop Win64 로컬 Windows 디버거 Live Share

EnemyBase.cpp EnemyBase.h\* SuperSideScroller\_Player.cpp SuperSideScroller\_Player.h Anim\_ProjectileNotify.cpp

SuperSideScroller AEnemyBase

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "SuperSideScrollerCharacter.h"
7 #include "EnemyBase.generated.h"
8
9 /**
10 * 
11 */
12 UCLASS()
13 class SUPERSIDESCRROLLER_API AEnemyBase : public ASuperSideScrollerCharacter
14 {
15     GENERATED_BODY()
16
17     public:
18         void DestroyEnemy();
19     };
20 }
21
22 }
```

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+I)

SuperSideScroller (2/2개 프로젝트)

- Engine
- UE4
- Games
- SuperSideScroller
  - 참조
  - 외부 종속성
  - Config
  - Source
    - SuperSideScroller
      - Anim\_ProjectileNotify.cpp
      - Anim\_ProjectileNotify.h
      - EnemyBase.cpp
      - EnemyBase.h
      - PlayerProjectile.cpp
      - PlayerProjectile.h
      - SuperSideScroller.Build.cs
      - SuperSideScroller.cpp
      - SuperSideScroller.h
      - SuperSideScroller\_Player.cpp
      - SuperSideScroller\_Player.h
      - SuperSideScrollerCharacter.cpp
      - SuperSideScrollerCharacter.h

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Super...ller

Develop Win64 로컬 Windows 디버거 Live Share

EnemyBase.cpp\* EnemyBase.h SuperSideScroller\_Player.cpp SuperSideScroller\_Player.h Anim\_ProjectileNotify.cpp

SuperSideScroller AEnemyBase DestroyEnemy()

```
1 // Fill out your copyright notice in the Description page of Project Settings.  
2  
3 #include "EnemyBase.h"  
4  
5 void AEnemyBase::DestroyEnemy()  
6 {  
7     Destroy();  
8 }  
9  
10
```

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+.)

솔루션 탐색기 'SuperSideScroller' (2/2개 프로젝트)

- Engine
- UE4
- Games
- SuperSideScroller
- 참조
- 외부 종속성
- Config
- Source
  - SuperSideScroller
    - Anim\_ProjectileNotify.cpp
    - Anim\_ProjectileNotify.h
    - EnemyBase.cpp
    - EnemyBase.h
    - PlayerProjectile.cpp
    - PlayerProjectile.h
    - SuperSideScroller.Build.cs
    - SuperSideScroller.cpp
    - SuperSideScroller.h
    - SuperSideScroller\_Player.cpp
    - SuperSideScroller\_Player.h
    - SuperSideScrollerCharacter
    - SuperSideScrollerCharacter
    - SuperSideScrollerGameMode
    - SuperSideScrollerGameMode
    - SuperSideScroller.Target.cs
    - SuperSideScrollerEditor.Target.cs
- SuperSideScroller.uproject

Ctrl+S

100 % 문제가 검색되지 않음 줄: 8 문자: 12 열: 15 템 CRLF

저장되었습니다. ↑ 소스 제어에 추가 ↗



# Exercise 14.08: Destroying Projectiles

The screenshot shows the Unreal Engine Editor interface with the following details:

- Toolbar:** 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).
- Search Bar:** Super...ller.
- Sidebar:** 허브 탐색기 (한국어) and 도구 상자.
- Project Explorer:** 솔루션 탐색기 (SuperSideScroller) and 솔루션 탐색기 검색(Ctrl+I). It lists the project structure: Engine, UE4, Games, SuperSideScroller, Source, and various source files like Anim\_ProjectileNotify.cpp, EnemyBase.cpp, PlayerProjectile.cpp, etc.
- Code Editor:** The main window displays the `PlayerProjectile.h` file. The code defines the `APlayerProjectile` class, which inherits from `AActor`. It includes properties for a collision component and a function `void ExplodeProjectile()`.
- Code Area:** A red box highlights the `ExplodeProjectile()` function definition.
- Status Bar:** Ctrl+S is displayed in the bottom right corner.

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) 🔍 Supe...ller

Develop Win64 로컬 Windows 디버거

PlayerProjectile.cpp\* ✘ X PlayerProjectile.h EnemyBase.cpp EnemyBase.h SuperSideScroller\_Player.cpp

SuperSideScroller → APlayerProjectile ExplodeProjectile()

```
16 // Set as root component
17 RootComponent = CollisionComp;
18
19 // Use a ProjectileMovementComponent to govern this projectile's movement
20 ProjectileMovement = CreateDefaultSubobject<UProjectileMovementComponent>(TEXT("ProjectileComp"));
21 ProjectileMovement->UpdatedComponent = CollisionComp;
22 ProjectileMovement->ProjectileGravityScale = 0.0f;
23 ProjectileMovement->InitialSpeed = 800.0f;
24 ProjectileMovement->MaxSpeed = 800.0f;
25
26 MeshComp = CreateDefaultSubobject<UStaticMeshComponent>(TEXT("MeshComp"));
27 MeshComp->AttachToComponent(RootComponent, FAttachmentTransformRules::KeepWorldTransform);
28
29 InitialLifeSpan = 3.0f;
30
31 }
32
33
34 void APlayerProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp,
35 FVector NormalImpulse, const FHitResult& Hit)
36 {
37 UE_LOG(LogTemp, Warning, TEXT("HIT"));
38 }
39
40 void APlayerProjectile::ExplodeProjectile()
41 {
42 Destroy();
43 }
```

Ctrl+S

100 % ☑ 문제가 검색되지 않음 줄: 42 문자: 12 열: 15 템 CRLF

설루션 탐색기 Live Share 🔍

설루션 탐색기 검색(Ctrl+;) 🔍

설루션 탐색기 'SuperSideScroller' (2/2개 프로젝트)

- Engine
- UE4
- Games
- SuperSideScroller
  - 참조
  - 외부 종속성
  - Config
  - Source
    - SuperSideScroller
      - Anim\_ProjectileNotify.cpp
      - Anim\_ProjectileNotify.h
      - EnemyBase.cpp
      - EnemyBase.h
      - PlayerProjectile.cpp
      - PlayerProjectile.h
    - SuperSideScroller.Build.cs
    - SuperSideScroller.cpp
    - SuperSideScroller.h
    - SuperSideScroller\_Player.cpp
    - SuperSideScroller\_Player.h
    - SuperSideScrollerCharacter.cpp
    - SuperSideScrollerCharacter.h
    - SuperSideScrollerGameMode.cpp
    - SuperSideScrollerGameMode.h
    - SuperSideScroller.Target.cs
    - SuperSideScrollerEditor.Target.cs
  - SuperSideScroller.uproject

설루션 탐색기 Git 변경 내용 ↑ 소스 제어에 추가 ↻ 저장되었습니다.



# Activity 14.01: Projectile Destroying Enemies

The screenshot shows the Unreal Engine Editor interface with the following details:

- Toolbar:** 파일(F), 편집(E), 보기(V), Git(G), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(I), 확장(X), 창(W), 도움말(H), 검색 (Ctrl+Q), Super...ller.
- File Explorer:** Live Share icon.
- Solution Explorer:** Shows the project structure:
  - SuperSideScroller
  - Engine
  - UE4
  - Games
  - SuperSideScroller
    - 참조
    - 외부 종속성
    - Config
    - Source
      - SuperSideScroller
        - Anim\_ProjectileNotify.cpp
        - Anim\_ProjectileNotify.h
        - EnemyBase.cpp
        - EnemyBase.h
        - PlayerProjectile.cpp
        - PlayerProjectile.h
        - SuperSideScroller.Build.cs
        - SuperSideScroller.cpp
        - SuperSideScroller.h
        - SuperSideScroller\_Player.cpp
        - SuperSideScroller\_Player.h
        - SuperSideScrollerCharacter.cpp
        - SuperSideScrollerCharacter.h
- Code Editor:** The file `PlayerProjectile.cpp` is open. A red box highlights the line `#include "EnemyBase.h"`. The code defines the `APlayerProjectile` class, which initializes collision and movement components and sets its root component to the collision component.

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Super...ller — □ ×

Live Share

PlayerProjectile.cpp\* × PlayerProjectile.h EnemyBase.cpp EnemyBase.h SuperSideScroller\_Player.cpp

SuperSideScroller → APlayerProjectile OnHit(UPrimitiveComponent \* HitComp, AActor \* OtherActor, UPrimitiveComponent \* OtherComp, FHitResult & Hit)

```
20 // Use a ProjectileMovementComponent to govern this projectile's movement
21 ProjectileMovement = CreateDefaultSubobject<UProjectileMovementComponent>(TEXT("ProjectileComp"));
22 ProjectileMovement->UpdatedComponent = CollisionComp;
23 ProjectileMovement->ProjectileGravityScale = 0.0f;
24 ProjectileMovement->InitialSpeed = 800.0f;
25 ProjectileMovement->MaxSpeed = 800.0f;
26
27 MeshComp = CreateDefaultSubobject<UStaticMeshComponent>(TEXT("MeshComp"));
28 MeshComp->AttachToComponent(RootComponent, FAttachmentTransformRules::KeepWorldTransform);
29
30 InitialLifeSpan = 3.0f;
31
32 }
33
34
35 void APlayerProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp,
36 FHitResult & Hit)
37 {
38 //UE_LOG(LogTemp, Warning, TEXT("HIT"));
39 AEnemyBase* Enemy = Cast<AEnemyBase>(OtherActor);
40 if (Enemy)
41 {
42 Enemy->DestroyEnemy();
43 }
44 ExplodeProjectile();
45 }
46
47 void APlayerProjectile::ExplodeProjectile()
48 {
49 Destroy();
50 }
```

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+I)

슬루션 'SuperSideScroller' (2/2개 프로젝트)

- Engine
- UE4
- Games
- SuperSideScroller
  - 참조
  - 외부 종속성
  - Config
  - Source
    - SuperSideScroller
      - Anim\_ProjectileNotify.cpp
      - Anim\_ProjectileNotify.h
      - EnemyBase.cpp
      - EnemyBase.h
      - PlayerProjectile.cpp
      - PlayerProjectile.h
    - SuperSideScroller.Build.cs
    - SuperSideScroller.cpp
    - SuperSideScroller.h
    - SuperSideScroller\_Player.cpp
    - SuperSideScroller\_Player.h
    - SuperSideScrollerCharacter.cpp
    - SuperSideScrollerCharacter.h
    - SuperSideScrollerGameMode.cpp
    - SuperSideScrollerGameMode.h
    - SuperSideScroller.Target.cs
    - SuperSideScrollerEditor.Target.cs
  - SuperSideScroller.uproject

준비

문제가 검색되지 않음

출: 41 문자: 25 열: 31 템 CRLF

↑ 소스 제어에 추가 ↑

The screenshot shows the Microsoft Visual Studio interface with the Korean localization. The main window displays the code for `PlayerProjectile.cpp`. The top navigation bar includes tabs for 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), and 검색 (Ctrl+Q). A search bar is located at the top right. The title bar shows the project name "SuperSideScroller". The left sidebar has sections for 서버, 멤버 보기, and 최근 상자. The main editor area shows the following code:

```
// Use a ProjectileMovementComp
ProjectileMovement = CreateDefaultSubobject<UPrimitiveComponent>(ProjectileMovement);
ProjectileMovement->UpdatedComponent();
ProjectileMovement->ProjectileType = EProjectileType::P_Enemy;
ProjectileMovement->InitialSpeed = 1000.0f;
ProjectileMovement->MaxSpeed = 1000.0f;
ProjectileMovement->MinSpeed = 0.0f;
MeshComp = CreateDefaultSubobject<UStaticMeshComponent>(MeshComp);
MeshComp->AttachToComponent(RocketBody, FAttachmentTransformRules::KeepRelativeTransform);
InitialLifeSpan = 3.0f;

void APlayerProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp, FVector NormalImpulse, const FHitResult& Hit)
{
    //UE_LOG(LogTemp, Warning, TEXT("HIT"));
    AEnemyBase* Enemy = Cast<AEnemyBase>(OtherActor);
    if (Enemy)
        Enemy->DestroyEnemy();
}

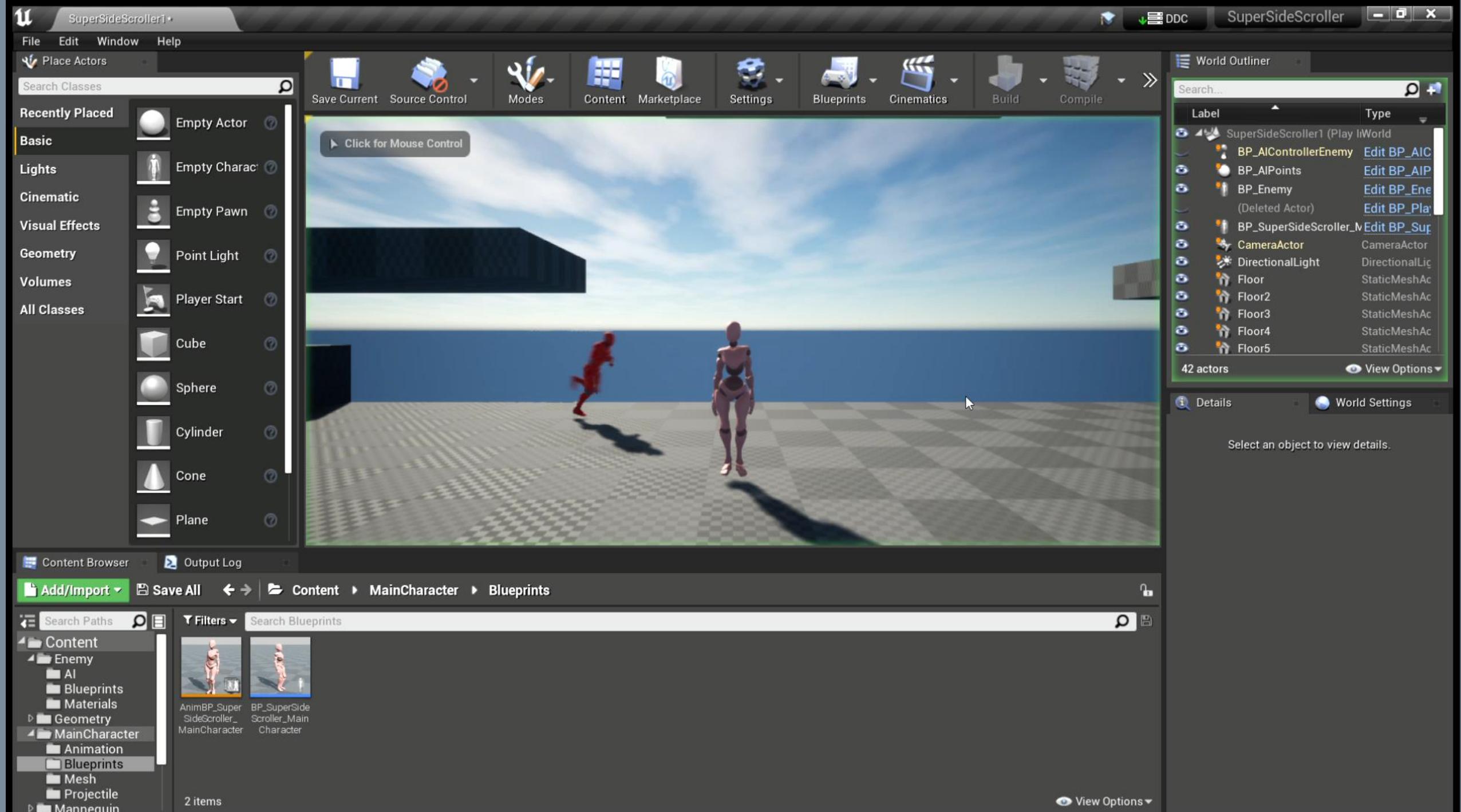
void APlayerProjectile::ExplodeProjectile()
{
    Destroy();
}
```

The Solution Explorer on the right lists the project structure:

- SuperSideScroller (Selected)
- Engine
- UE4
- Games
- SuperSideScroller
  - 참조
  - 외부 종속성
  - Config
  - Source
    - SuperSideScroller
      - Anim\_ProjectileNotify.cpp
      - Anim\_ProjectileNotify.h
      - EnemyBase.cpp
      - EnemyBase.h
      - PlayerProjectile.cpp
      - PlayerProjectile.h
    - SuperSideScroller.Build.cs
    - SuperSideScroller.cpp
    - SuperSideScroller.h
    - SuperSideScroller\_Player.cpp
    - SuperSideScroller\_Player.h
    - SuperSideScrollerCharacter.cpp
    - SuperSideScrollerCharacter.h
    - SuperSideScrollerGameMode.cpp
    - SuperSideScrollerGameMode.h
    - SuperSideScroller.Target.cs
    - SuperSideScrollerEditor.Target.cs
  - SuperSideScroller.uproject

A red arrow points to the "솔루션 빌드(B)" (Solution Build) option in the Build menu.

Bottom status bar: 100%, 문제가 검색되지 않음 (No problems found), 줄: 41, 문자: 25, 열: 31, 탭, CRLF, 솔루션 탐색기, Git 변경 내용.





# 연습 과제

- › Activity 14.01까지 모두 완성한 **SuperSideScroller** 프로젝트를 제출하시오.
- › 제출 방법: “프로젝트 폴더” 전체 압축  
→ 압축 파일 내에서 다음 “5개 폴더” 삭제
  - 1) Content 폴더 안 StarterContent 폴더
  - 2) Intermediate 폴더
  - 3) Saved 폴더
  - 4) Binary 폴더
  - 5) .vs 폴더  
→ 압축파일 업로드