

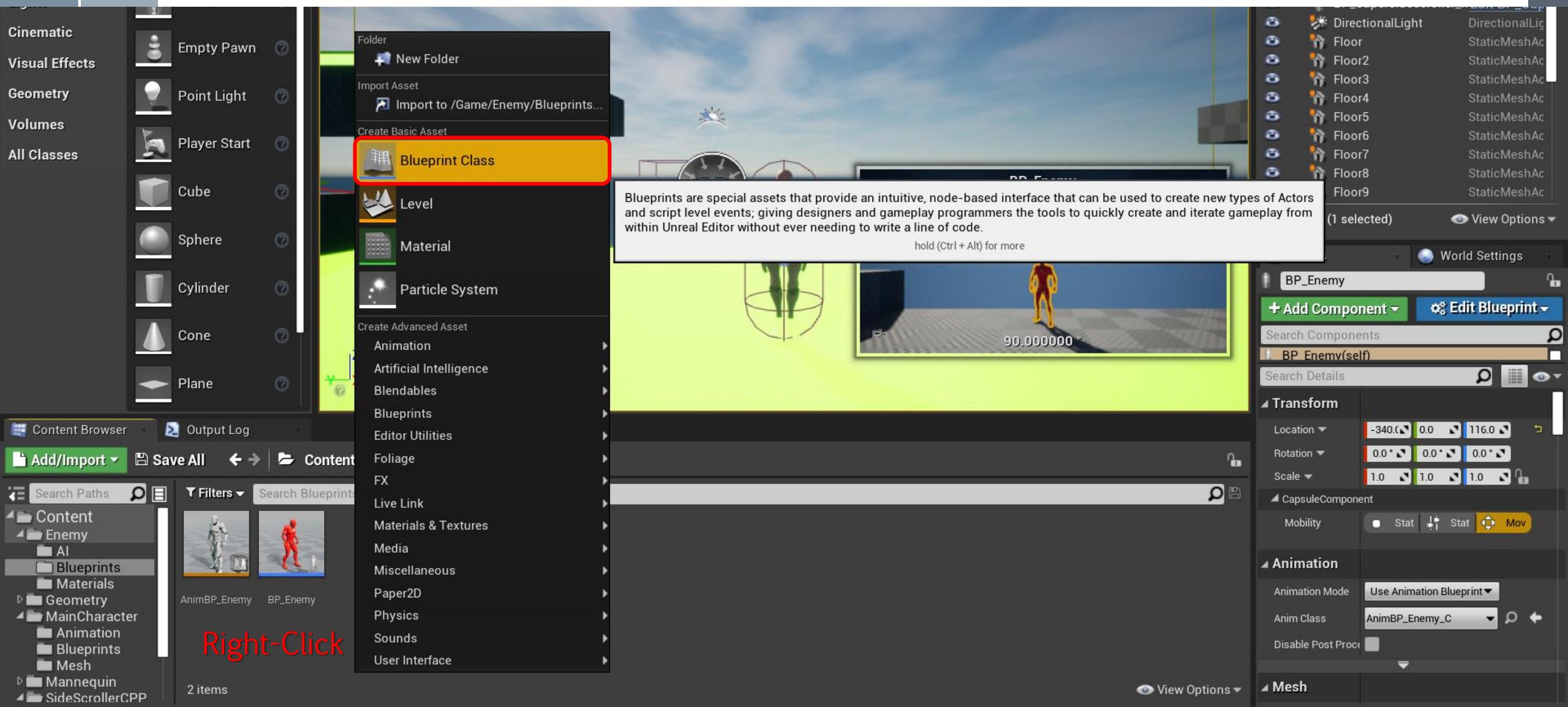
# Enemy Artificial Intelligence (2)

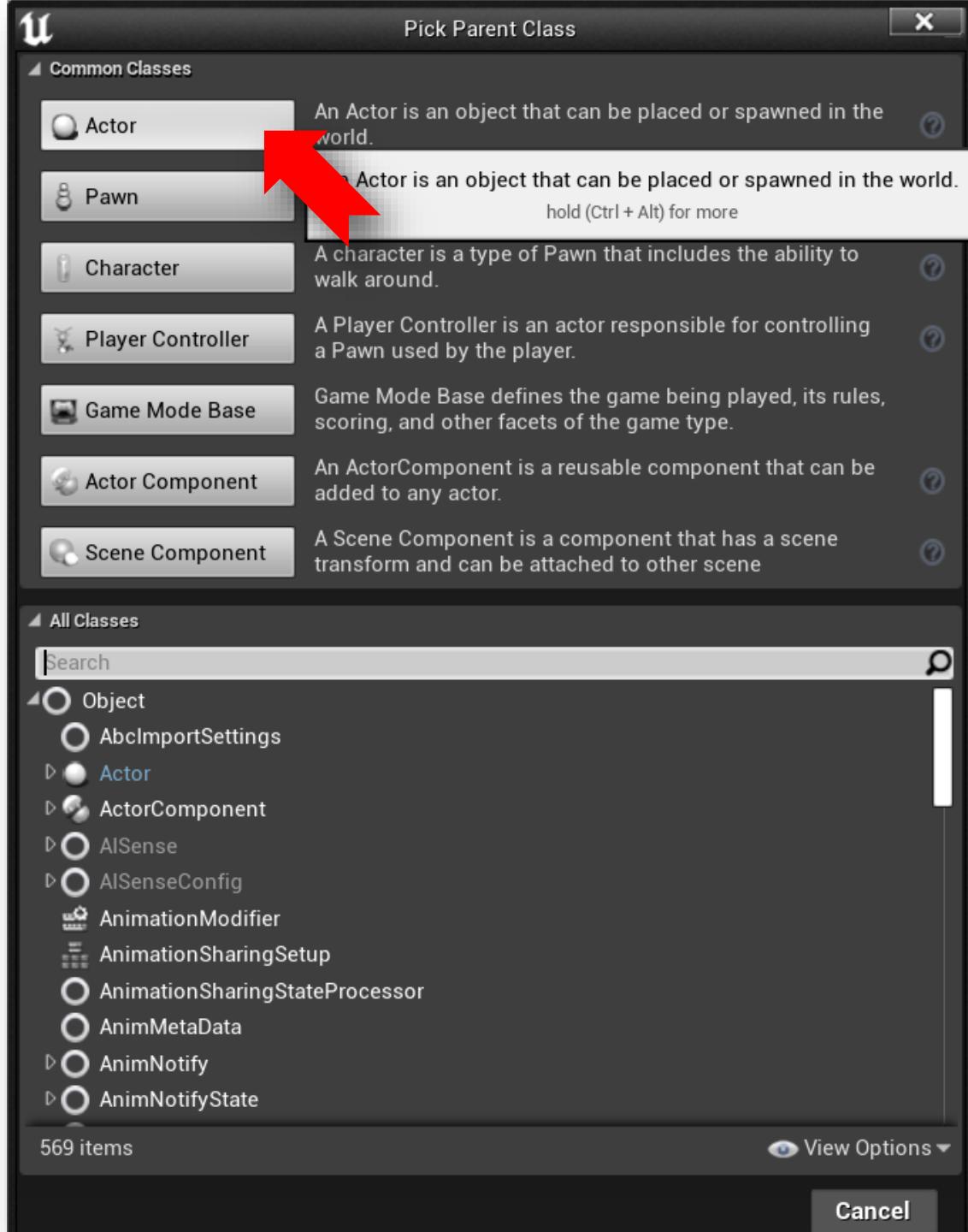
13<sup>th</sup> Week, 2021

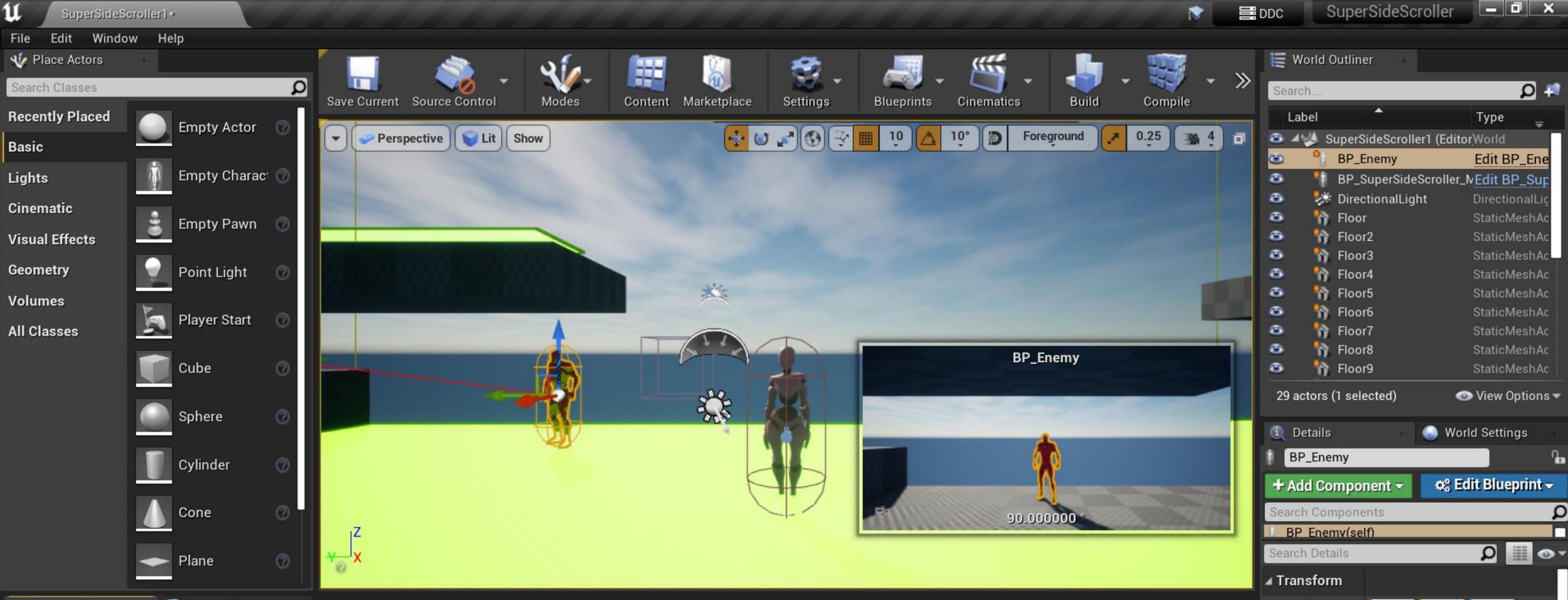


UNREAL  
ENGINE

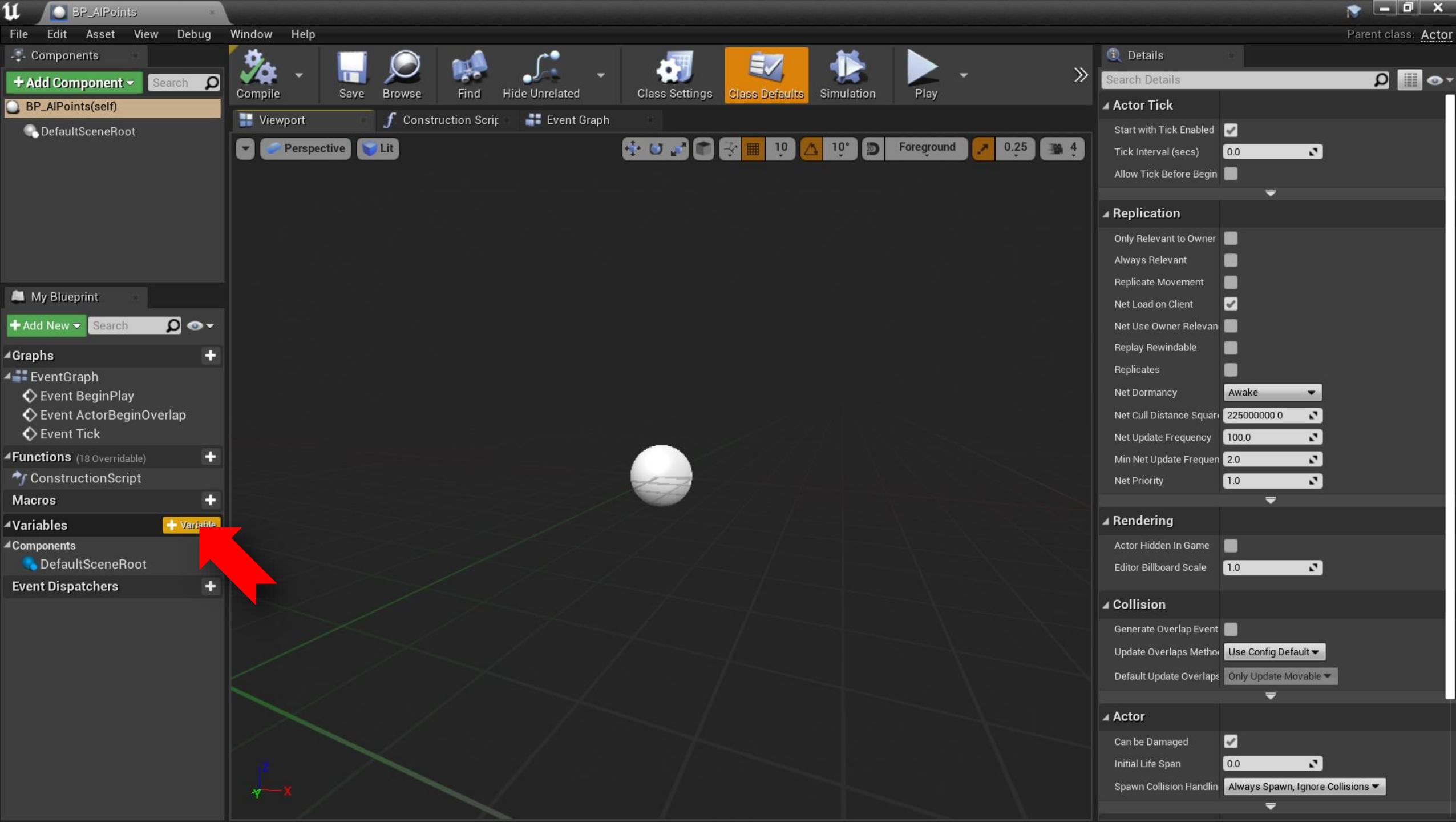
# Exercise 13.07: Creating the Enemy Patrol Location

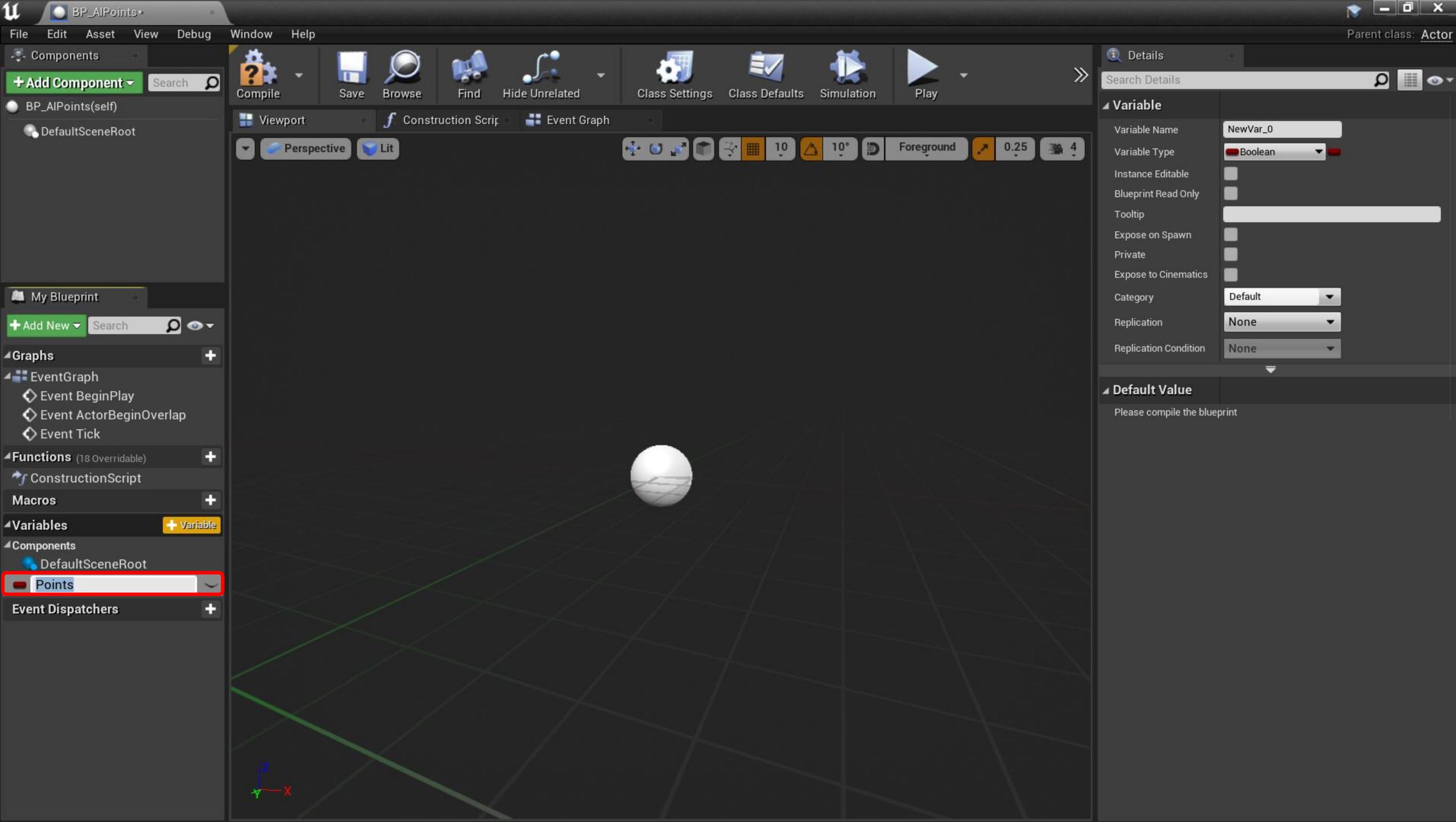


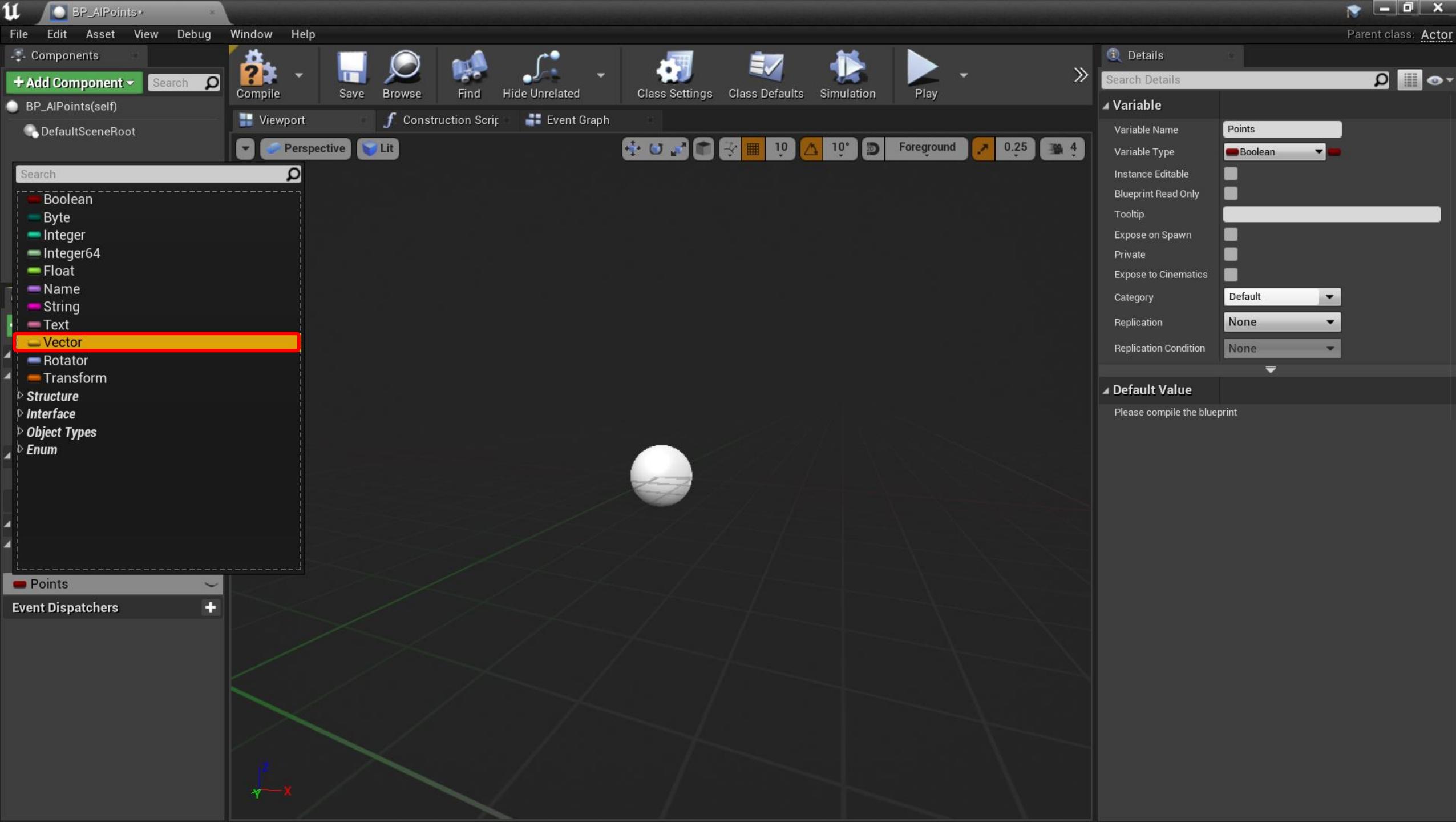


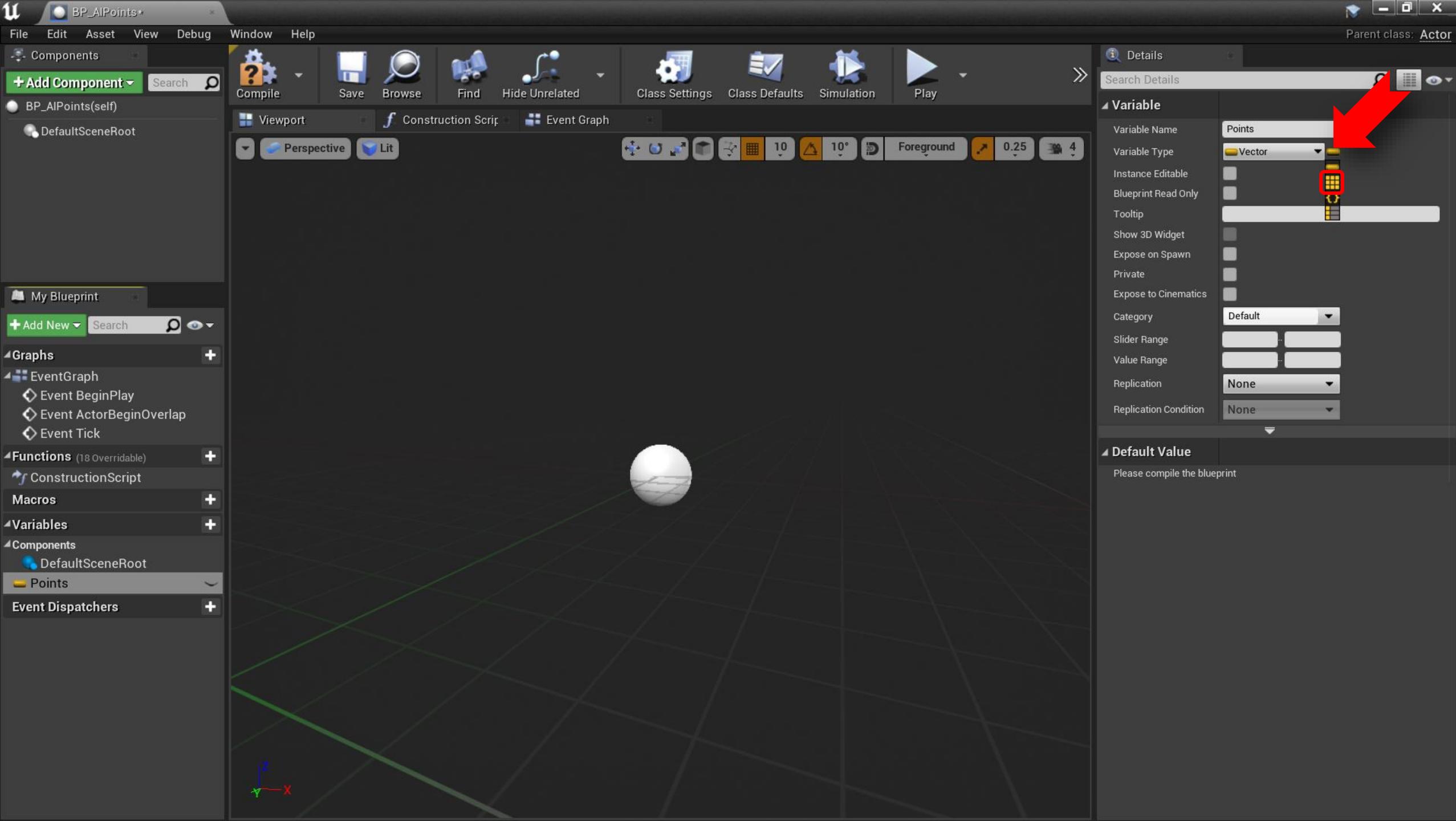


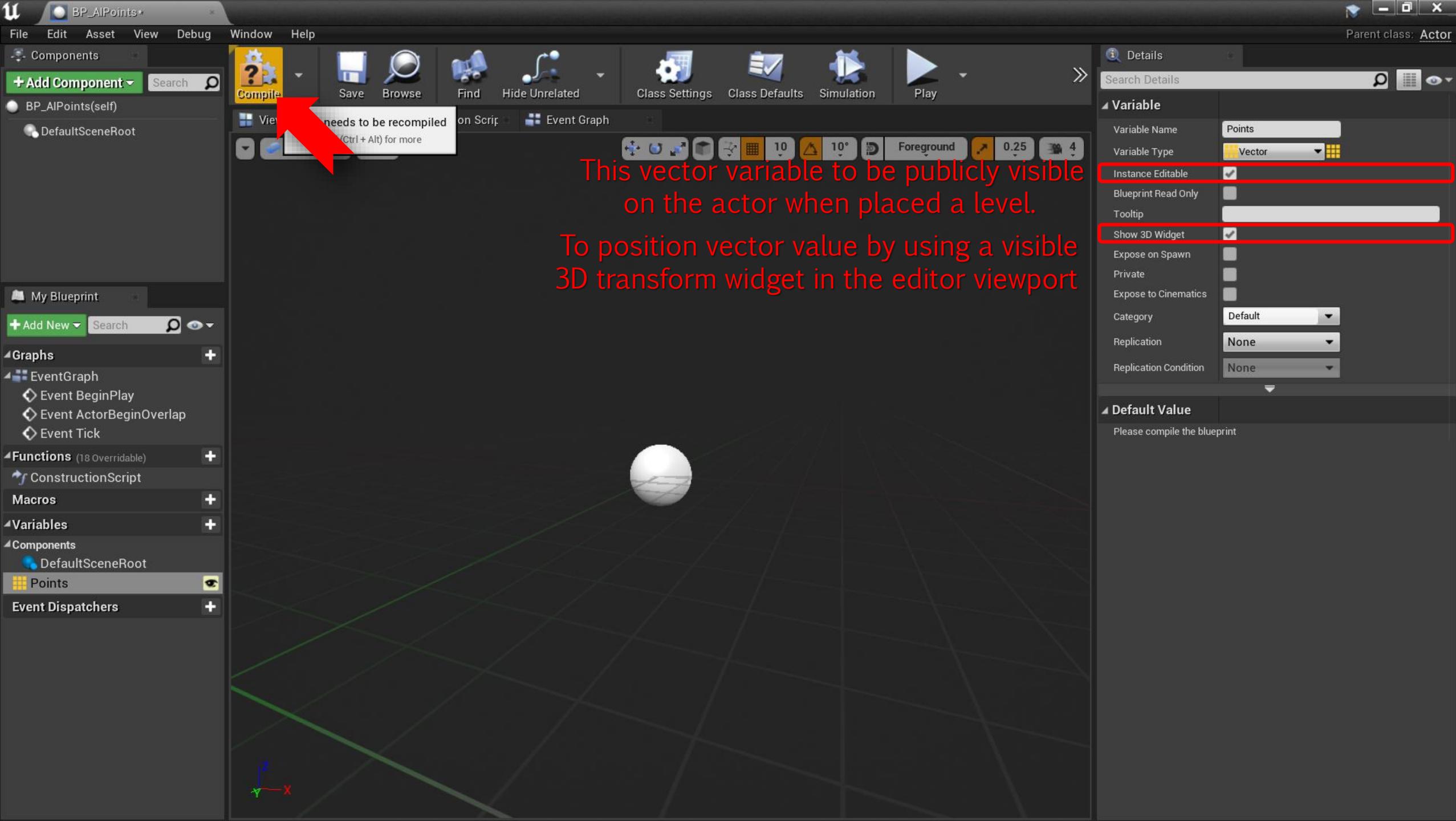
The Content Browser at the bottom left shows the file structure: Content > Enemy > AI > Blueprints. It displays two blueprints: "AnimBP\_Enemy" and "BP\_AIPoints". The "BP\_AIPoints" blueprint is highlighted with a red box. The bottom status bar indicates "3 items (1 selected)".

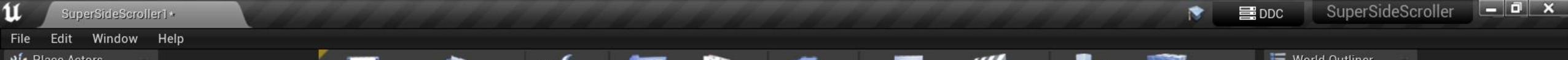












File Edit Window Help

Place Actors

Search Classes

Recently Placed

Basic

Lights

Cinematic

Visual Effects

Geometry

Volumes

All Classes



Empty Actor

?



Empty Charac:

?



Empty Pawn

?



Point Light

?



Player Start

?



Cube

?



Sphere

?



Cylinder

?



Cone

?



Plane

?



Save Current



Source Control



Modes



Content



Marketplace



Settings



Blueprints



Cinematics



Build



Compile



World Outliner



Search...



Content Browser   Output Log

Add/Import Save All Content > Enemy > Blueprints

Search Paths Filters Search Blueprints

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

AnimBP\_Enemy BP\_AIPoints BP\_Enemy

3 items (1 selected) View Options

World Outliner

Label Type

BP\_Enemy Edit BP\_Ene

BP\_SuperSideScroller\_N Edit BP\_Sup

DirectionalLight

Floor

Floor2

Floor3

Floor4

Floor5

Floor6

Floor7

Floor8

Floor9

29 actors (1 selected) View Options

Details World Settings

BP\_Enemy

+ Add Component Edit Blueprint

Search Components

BP\_Enemy(self)

Search Details

Transform

Location: -340.0 0.0 116.0

Rotation: 0.0° 0.0° 0.0°

Scale: 1.0 1.0 1.0

CapsuleComponent

Mobility: Stat Stat Mov

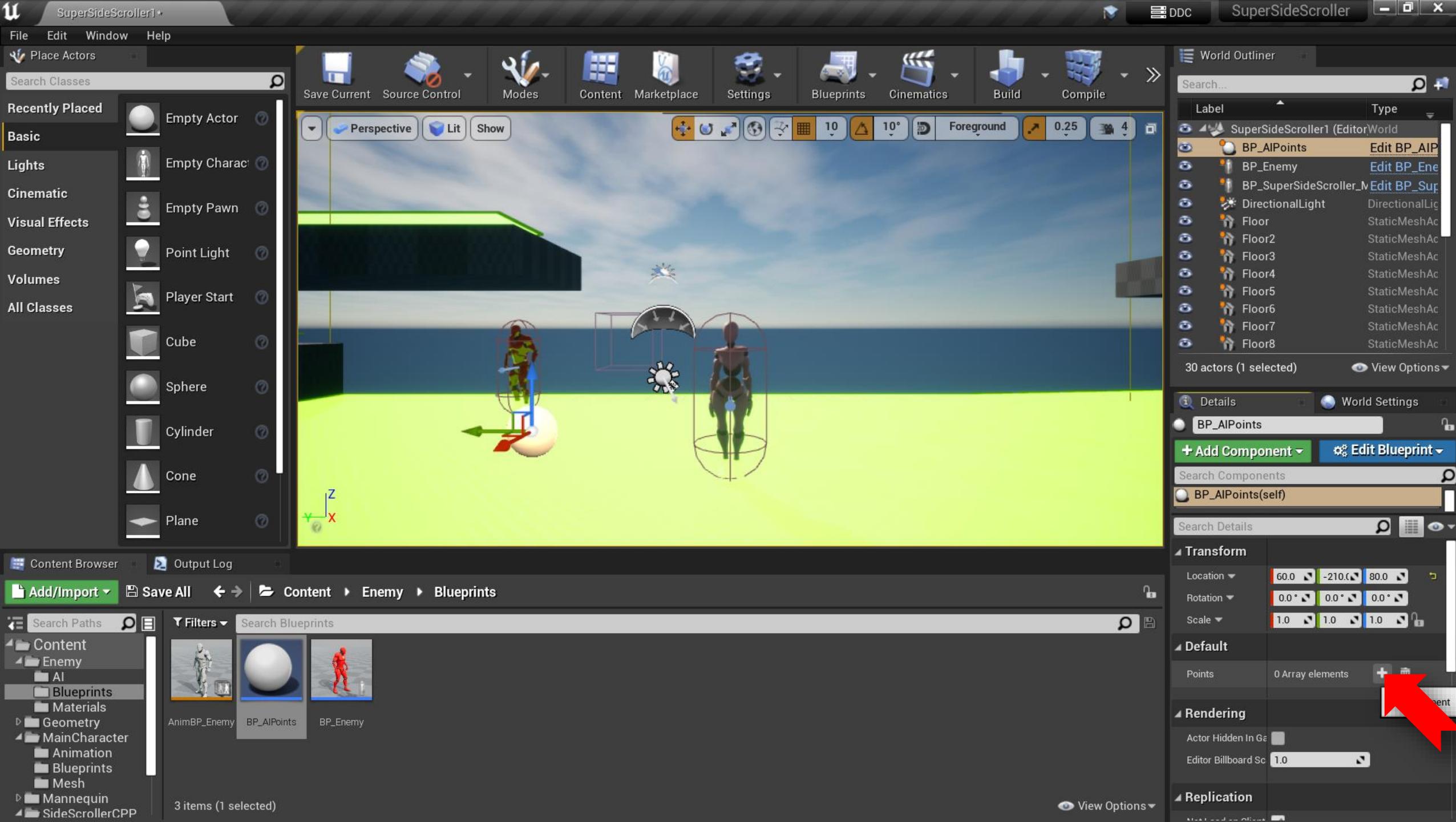
Animation

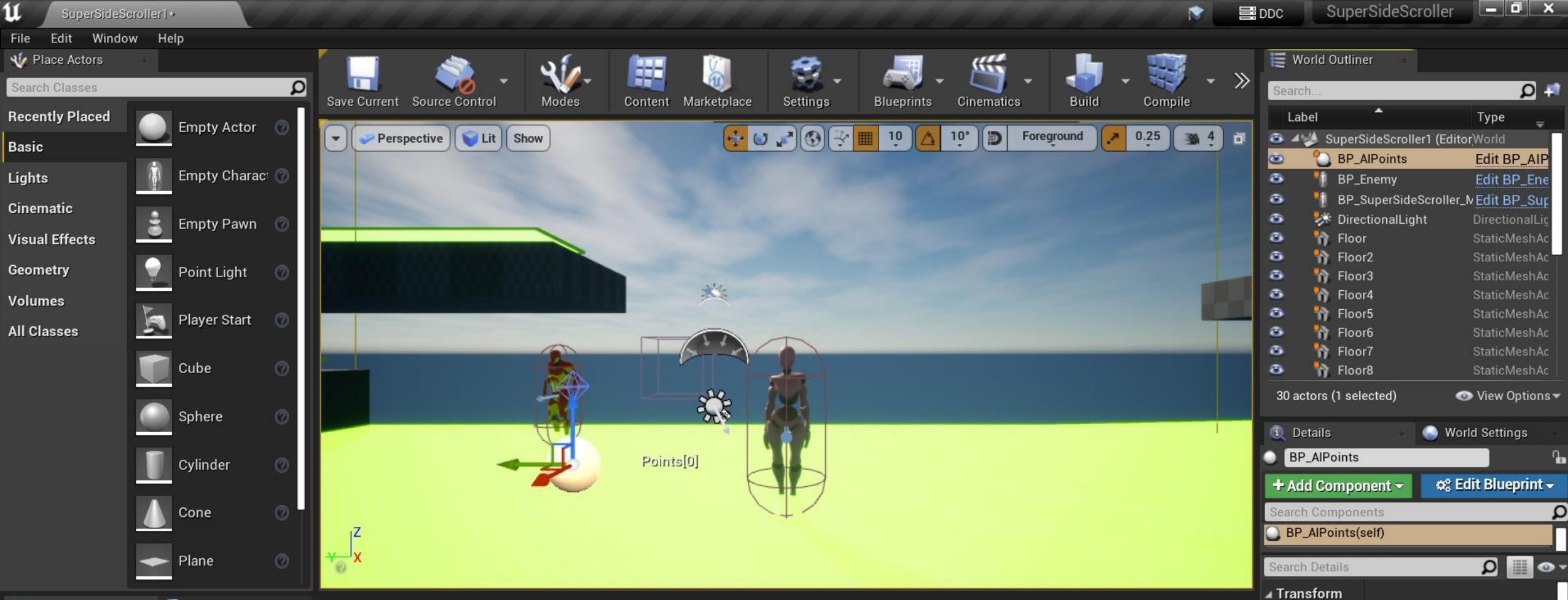
Animation Mode: Use Animation Blueprint

Anim Class: AnimBP\_Enemy\_C

Disable Post Process

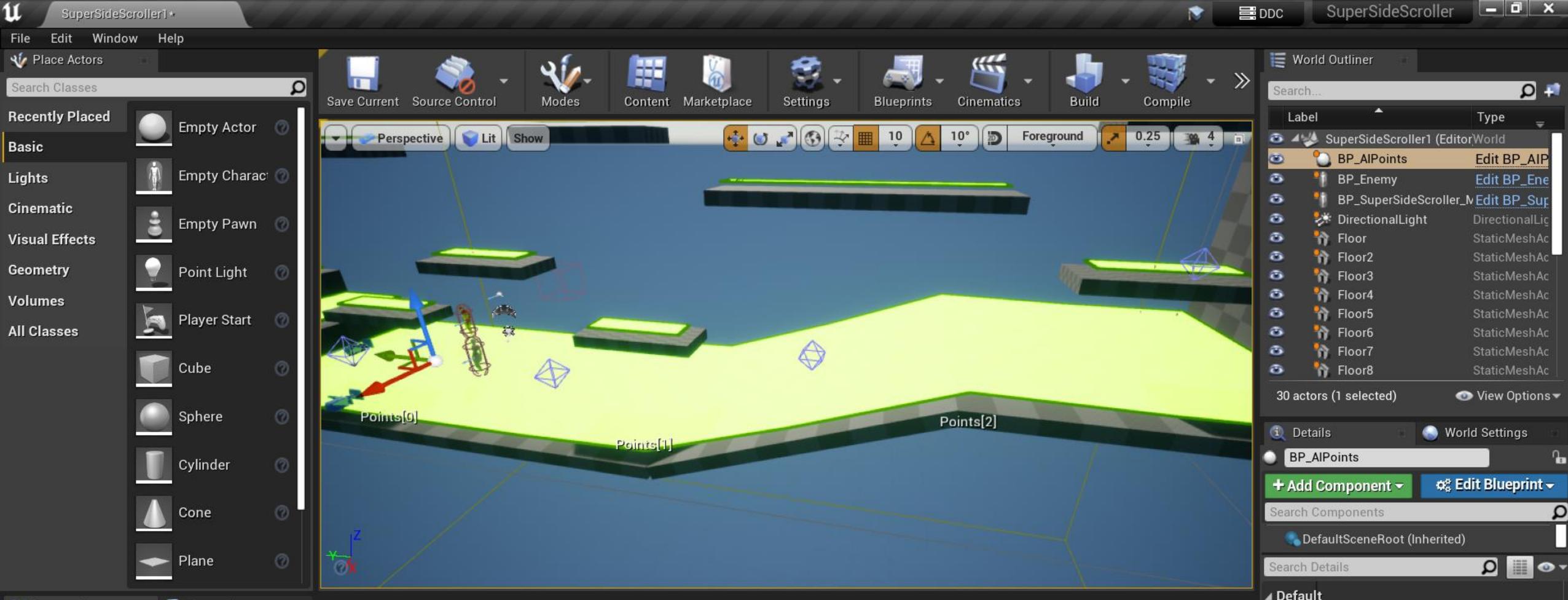
Mesh





You can have many elements inside of an array, but the larger the array, the more memory is allocated





Content Browser Output Log

Add/Import Save All Content > Enemy > Blueprints

Search Paths Filters Search Blueprints

Content Enemy AI Blueprints Materials

Geometry MainCharacter Animation Blueprints Mesh

Mannequin SideScrollerCPP

3 items (1 selected)

View Options

AnimBP\_Enemy BP\_AIPoints BP\_Enemy

The bottom-left corner of the interface shows the Content Browser with a tree view of project assets. The "Content" folder contains "Enemy" and "MainCharacter" subfolders. "Enemy" contains "AI", "Blueprints", and "Materials". "MainCharacter" contains "Animation", "Blueprints", "Mesh", "Mannequin", and "SideScrollerCPP". The "Blueprints" folder under "Enemy" contains three items: "AnimBP\_Enemy", "BP\_AIPoints", and "BP\_Enemy". The status bar at the bottom indicates "3 items (1 selected)".



# Vector Transformation

- › What are world space and local space?

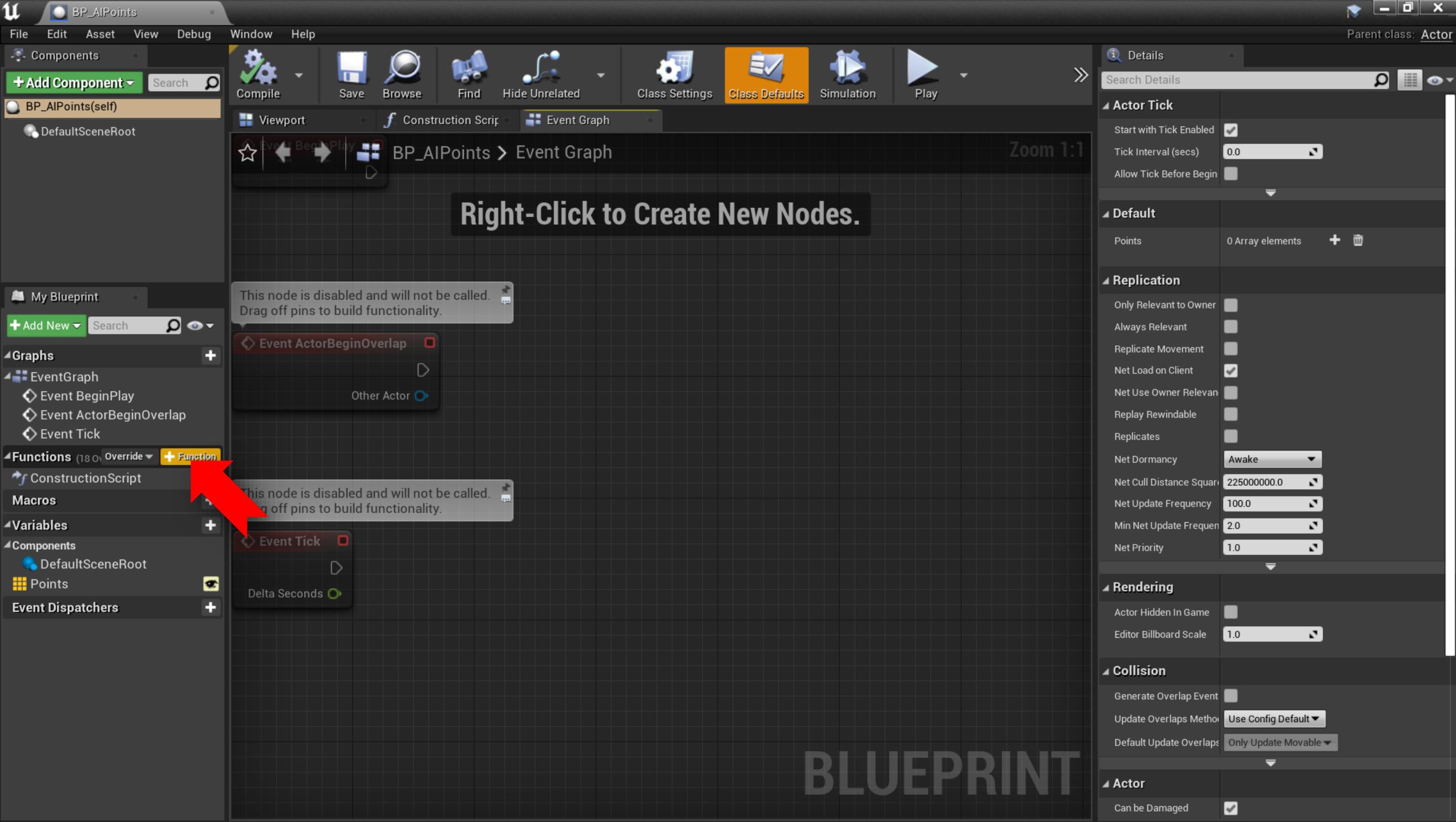
Default		4 Array elements					
Points		X	Y	Z			
	D 0	X 0.0	Y 510.0	Z 0.0			
	D 1	X 0.0	Y -2900.0	Z 670.0			
	D 2	X 0.0	Y -1770.0	Z 0.0			
	D 3	X 0.0	Y -620.0	Z 0.0			

The local-space position Vectors of the Points array, relative to the world-position of the BP\_AIPoints actor

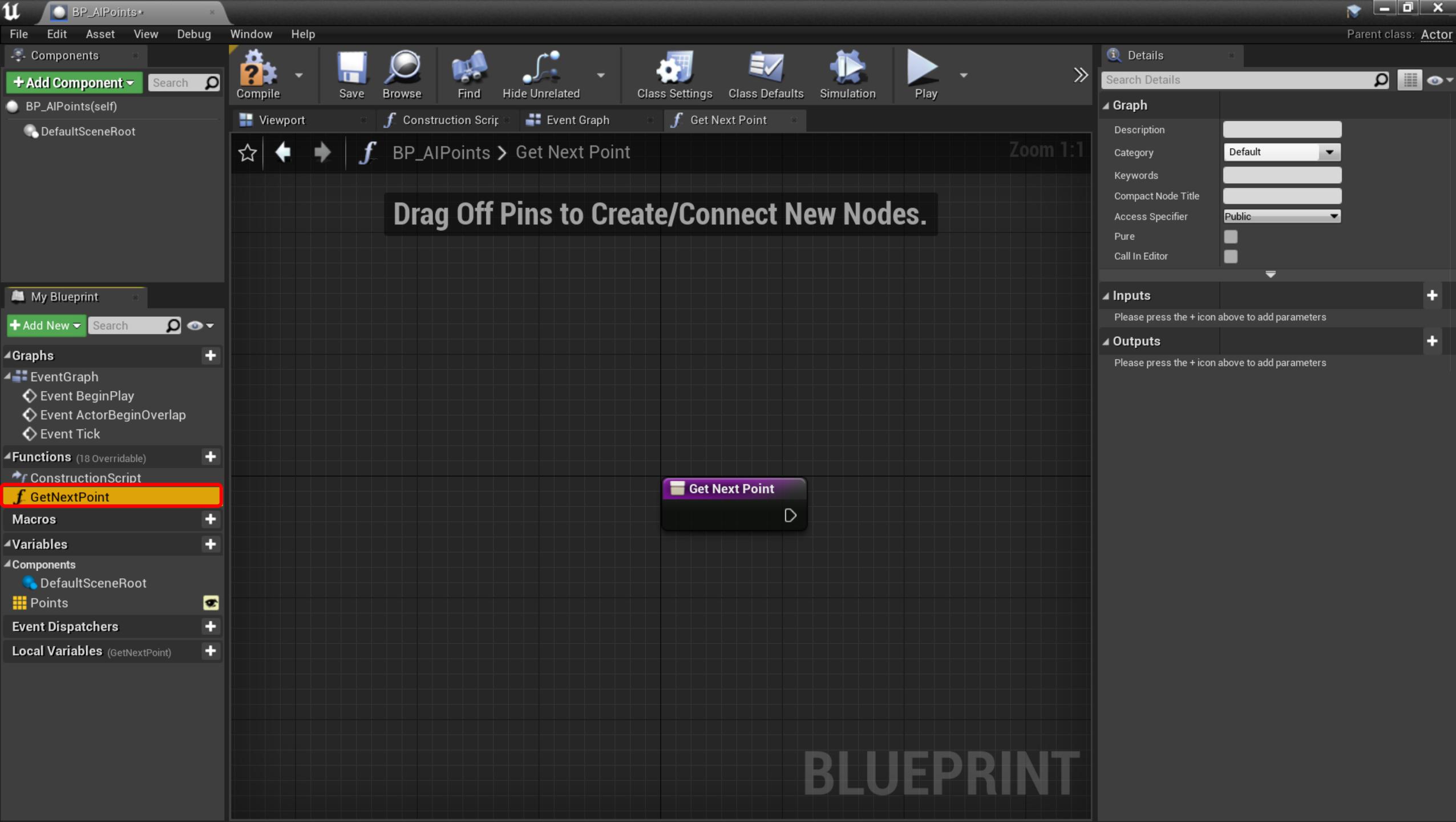
- › **Transform Location** function takes in two parameters
  - **T**: to convert the vector location parameter from a local-space into a world-space value
  - **Location**: to be converted from local space to world space

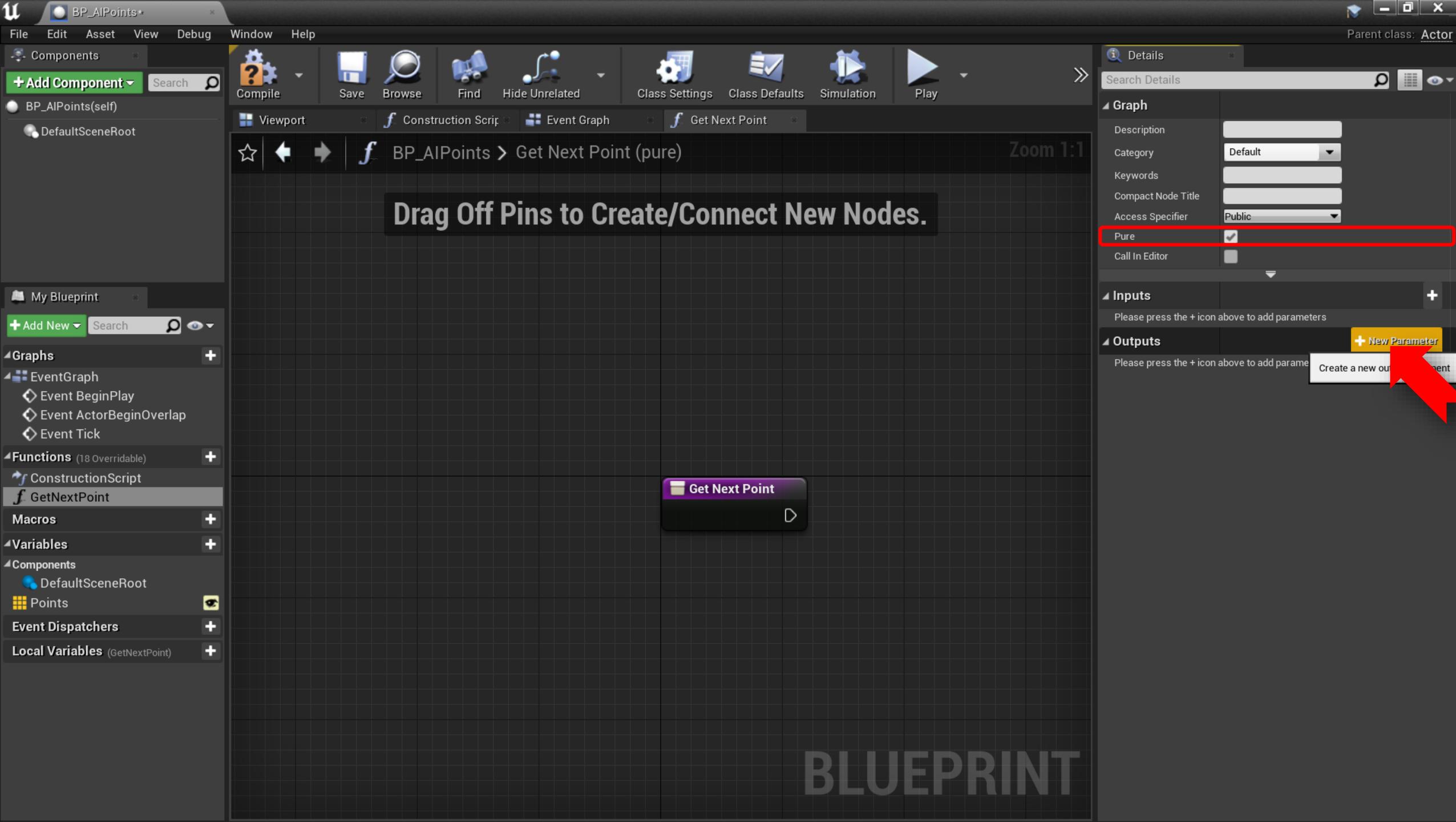
# Exercise 13.08: Selecting a Random Point in an Array

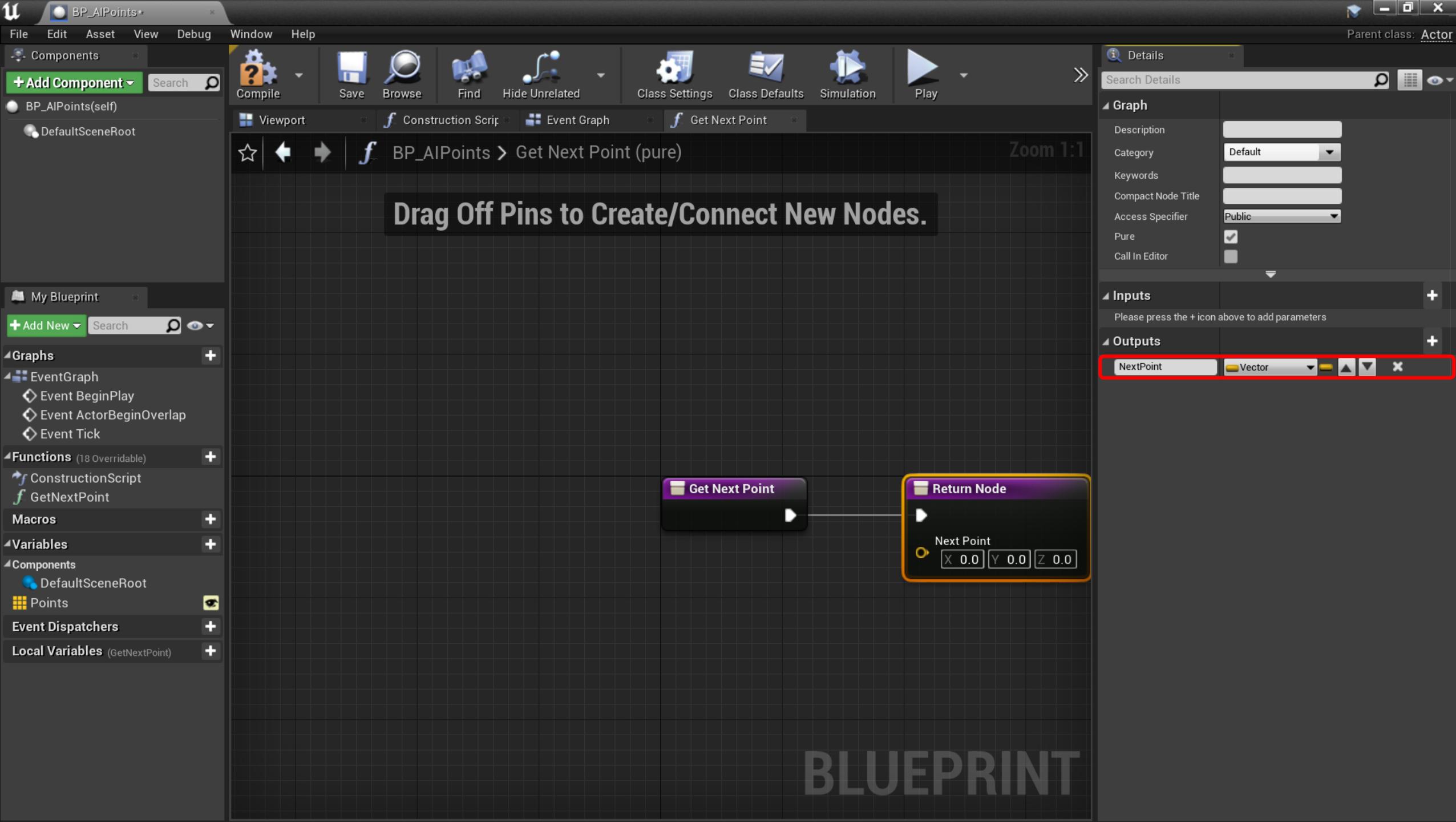


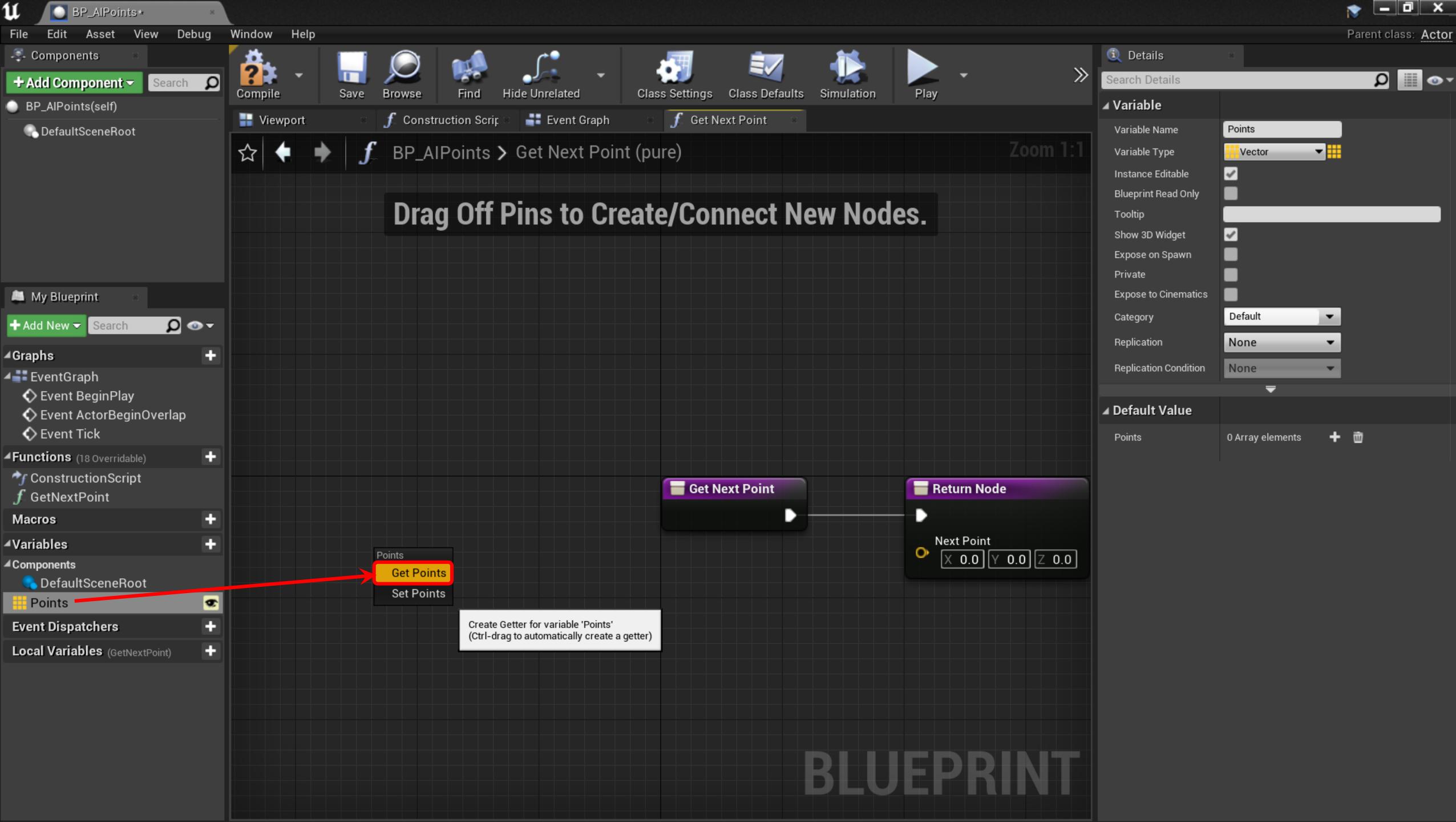


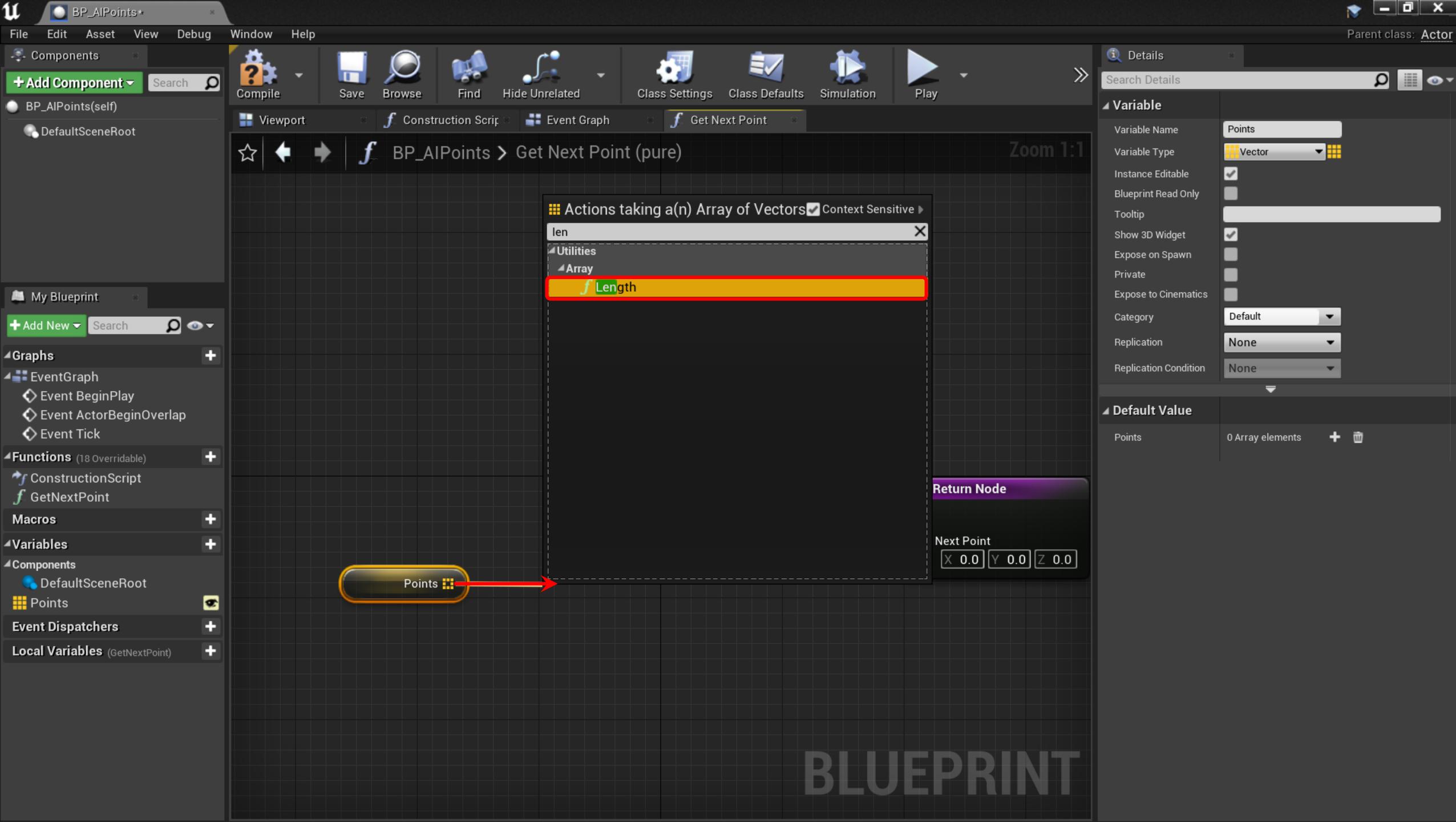
# BLUEPRINT

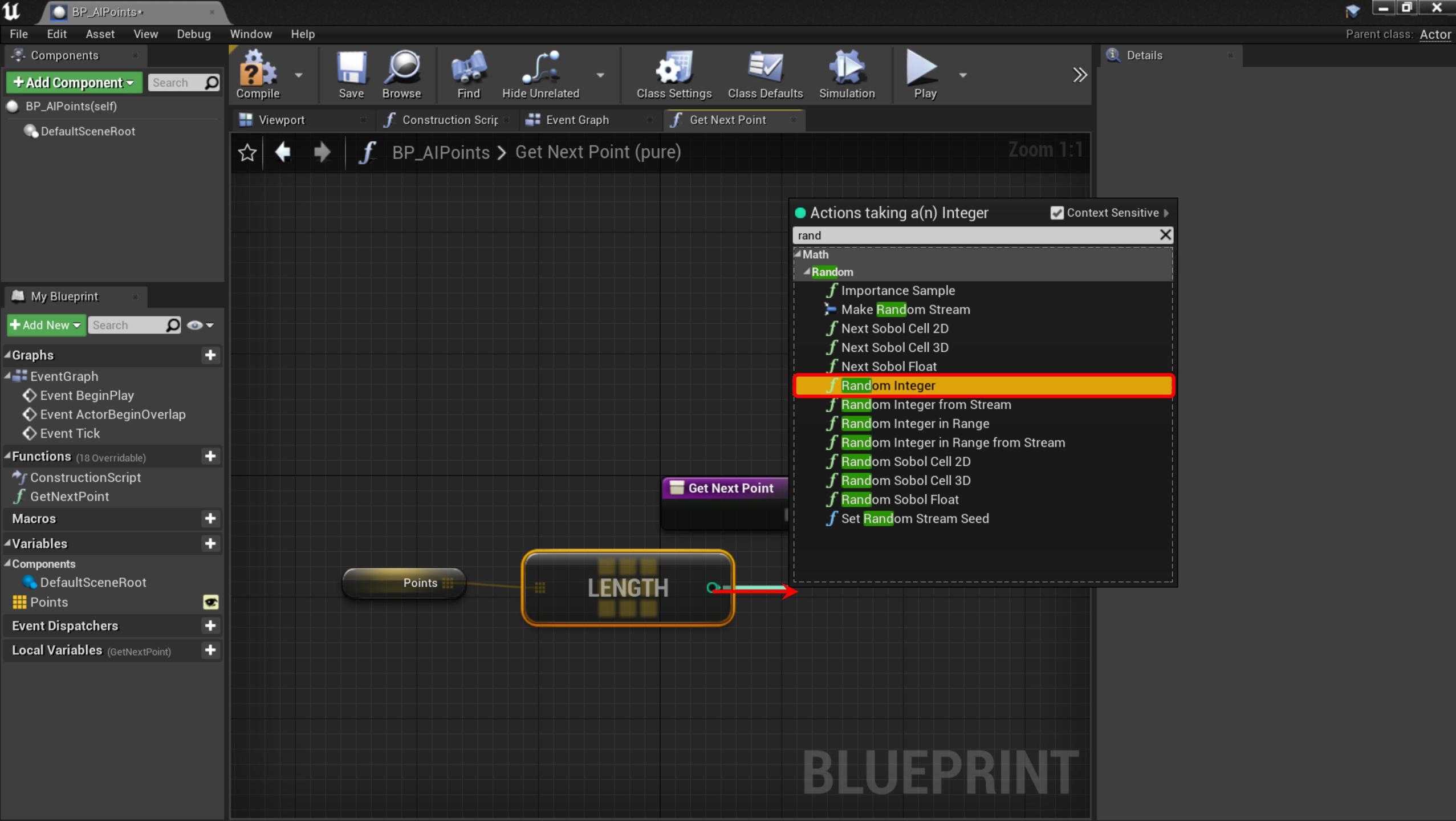


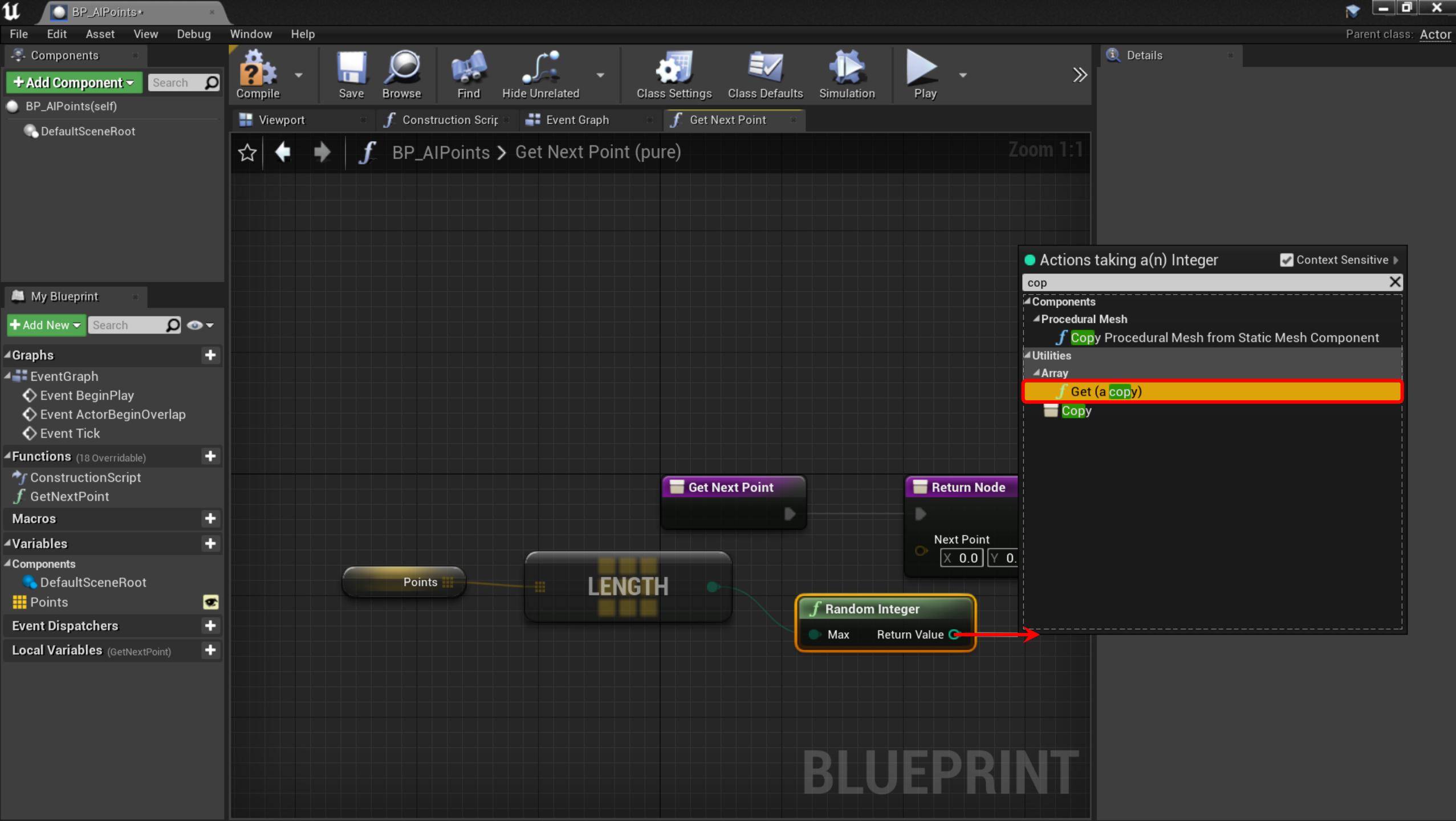


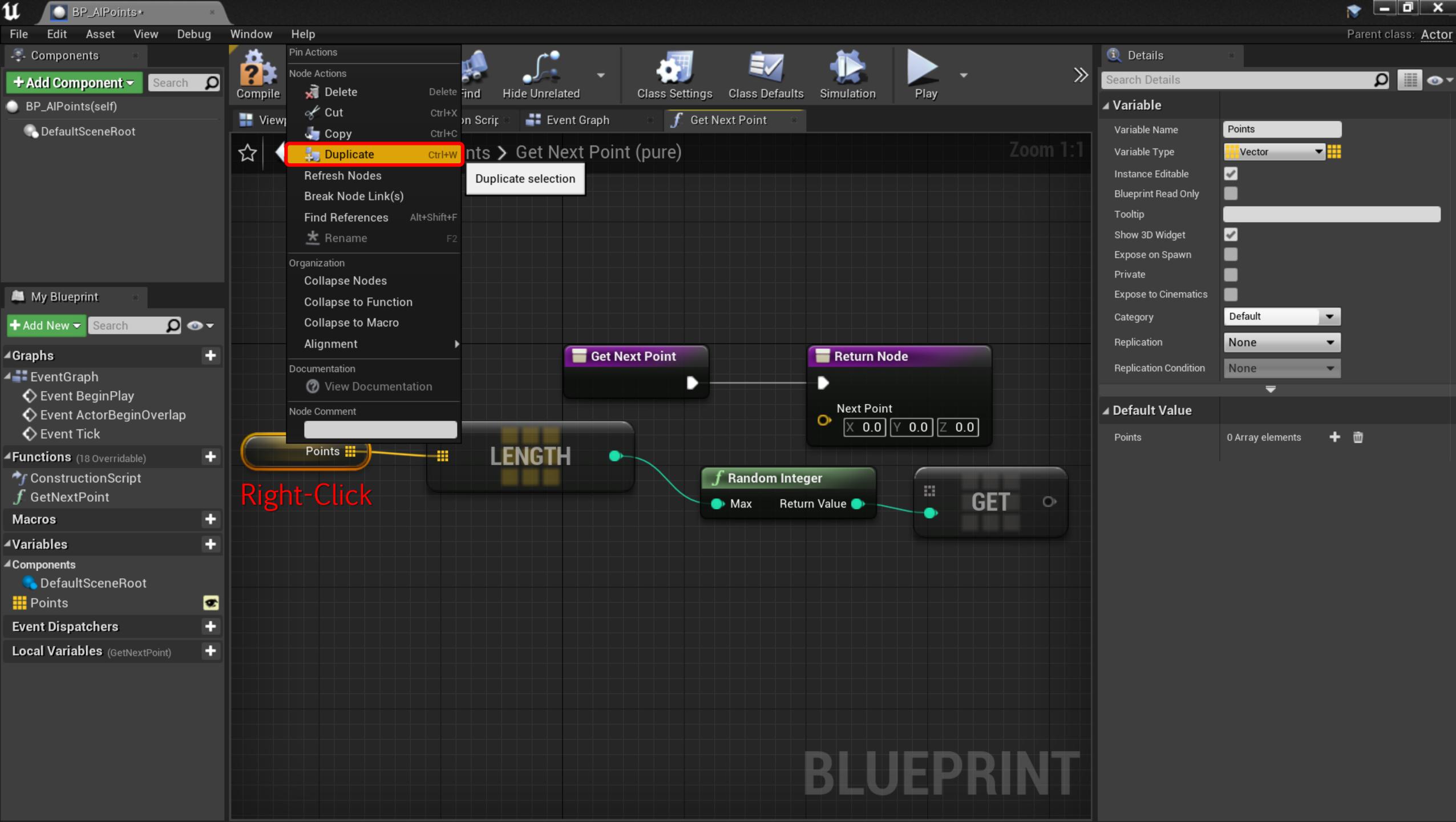


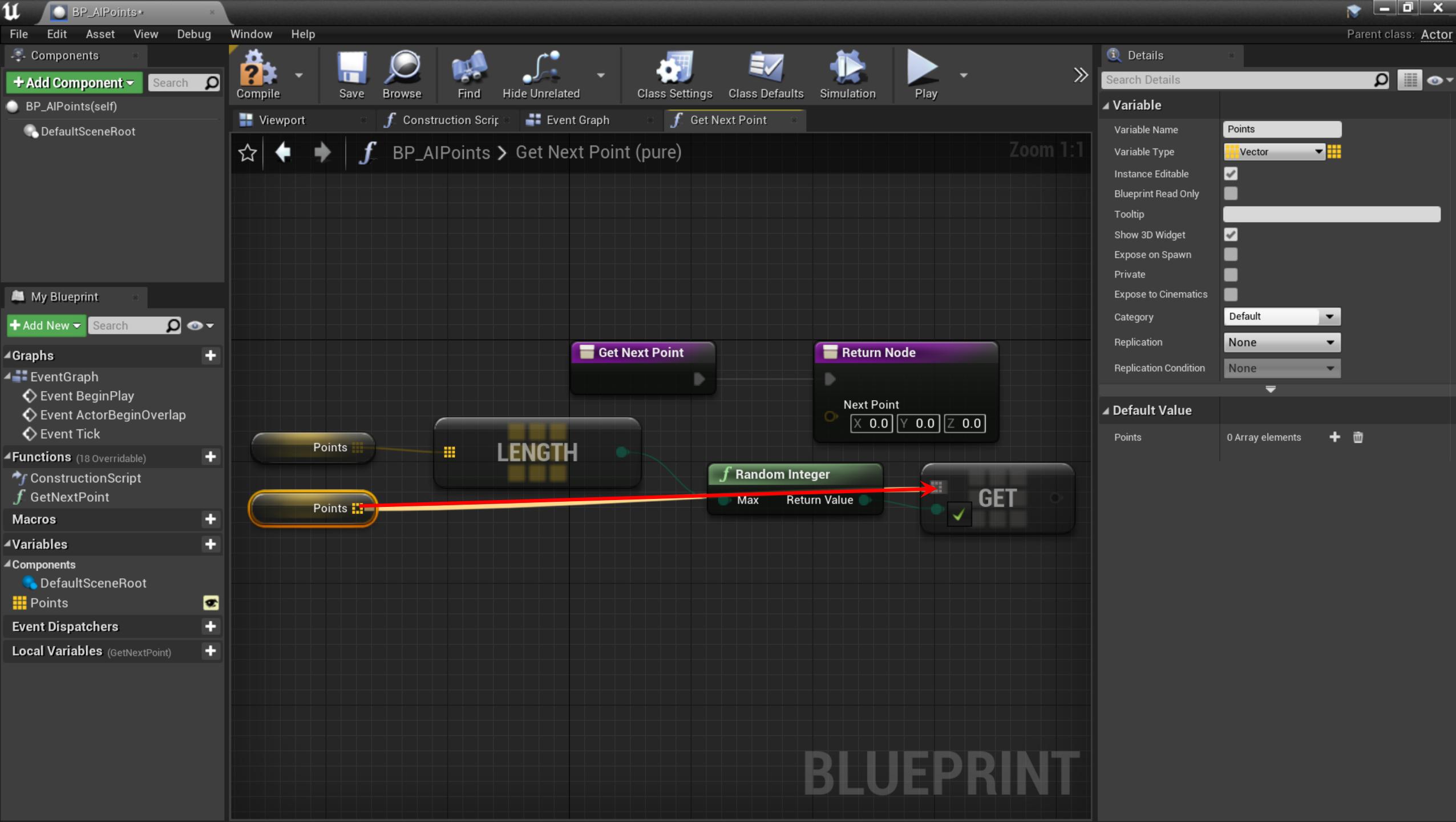


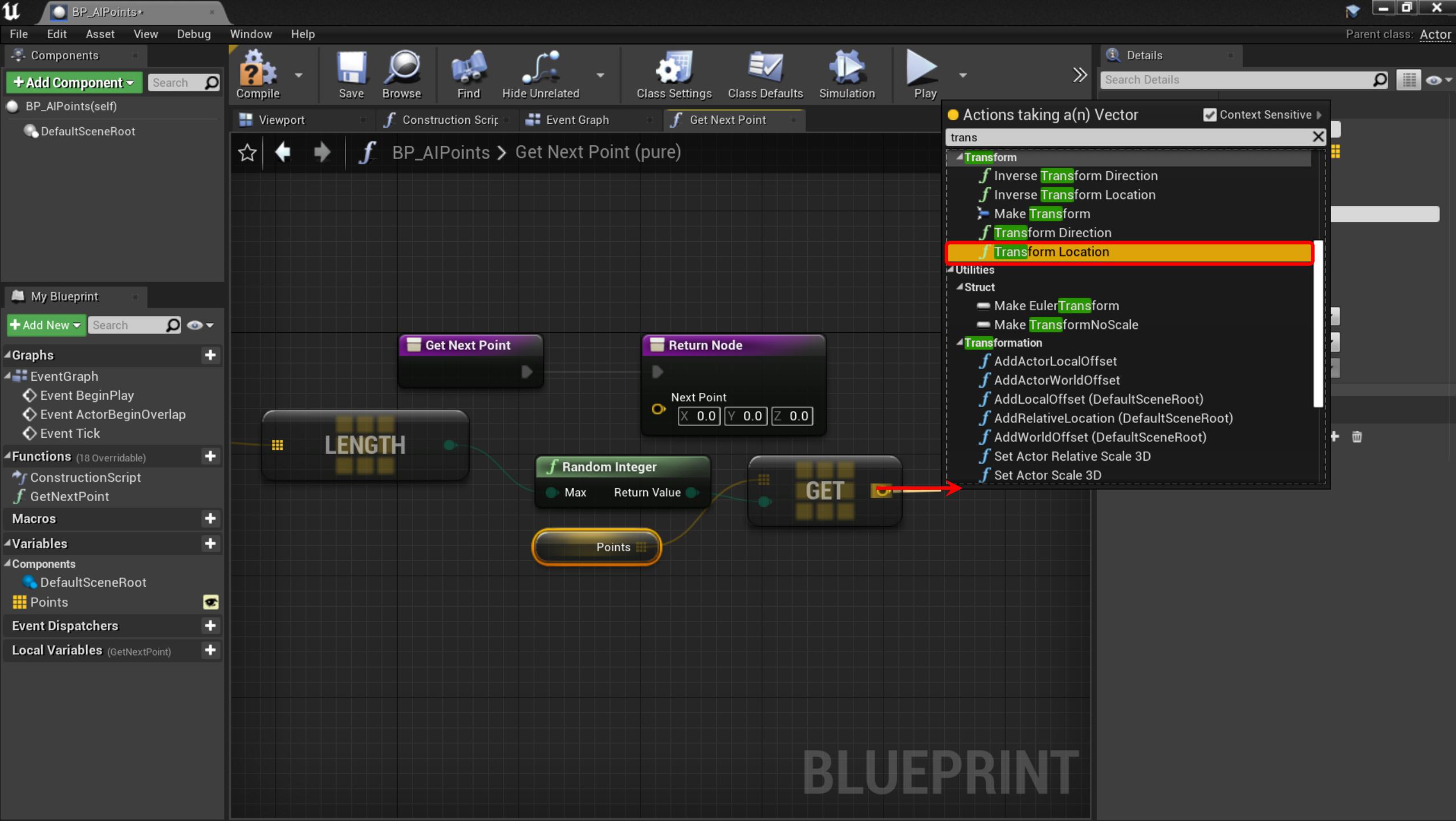


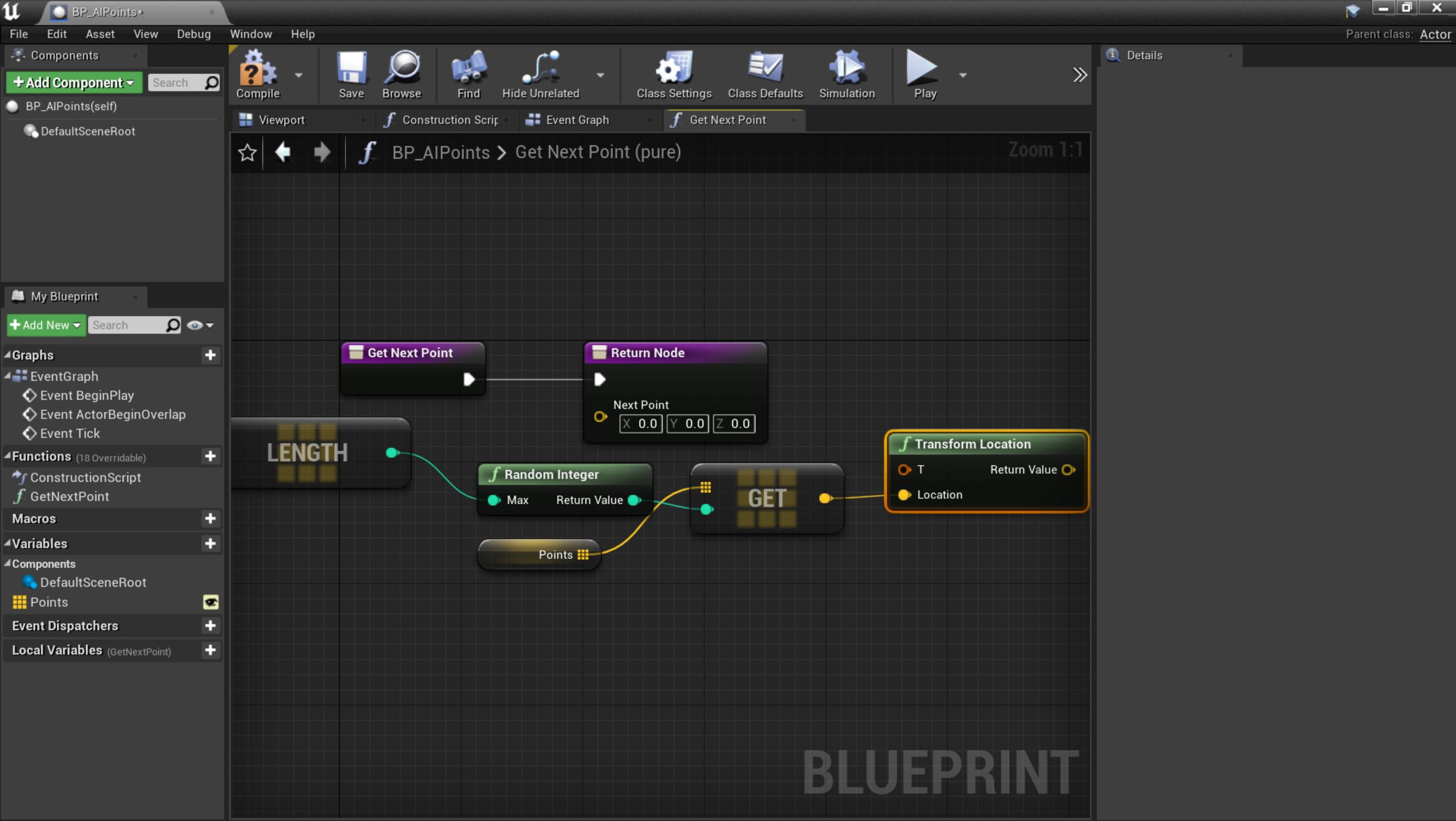


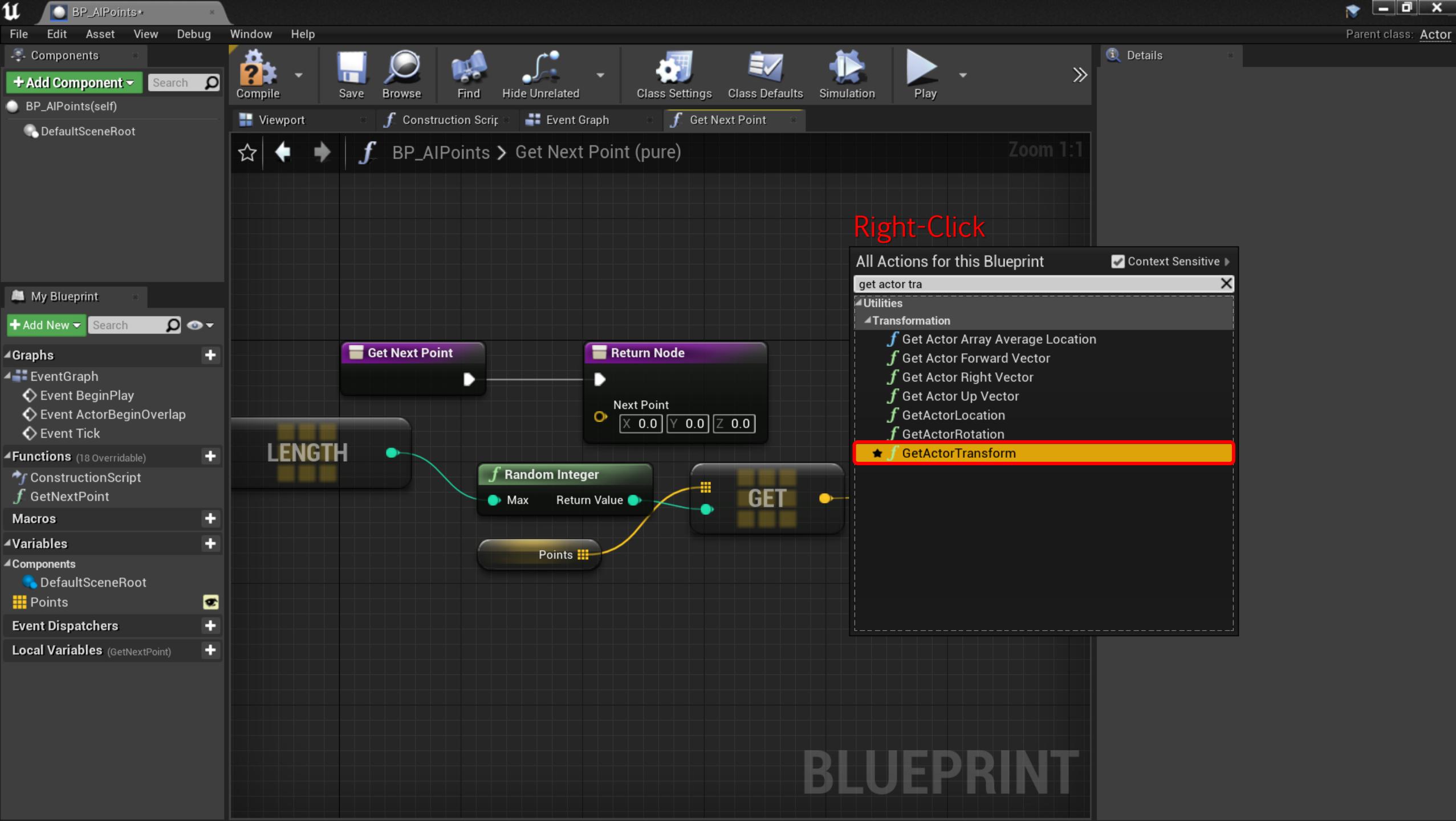


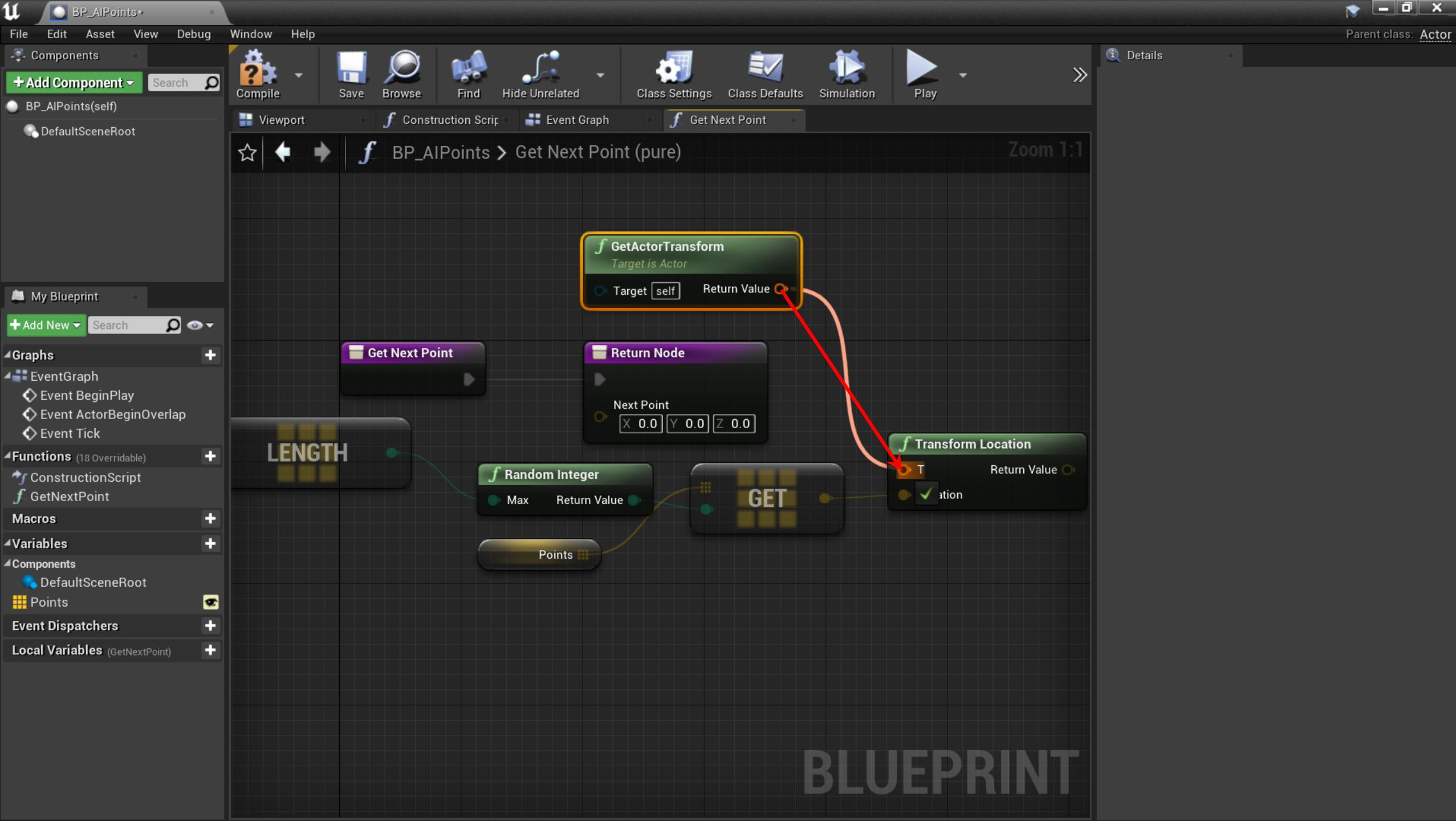


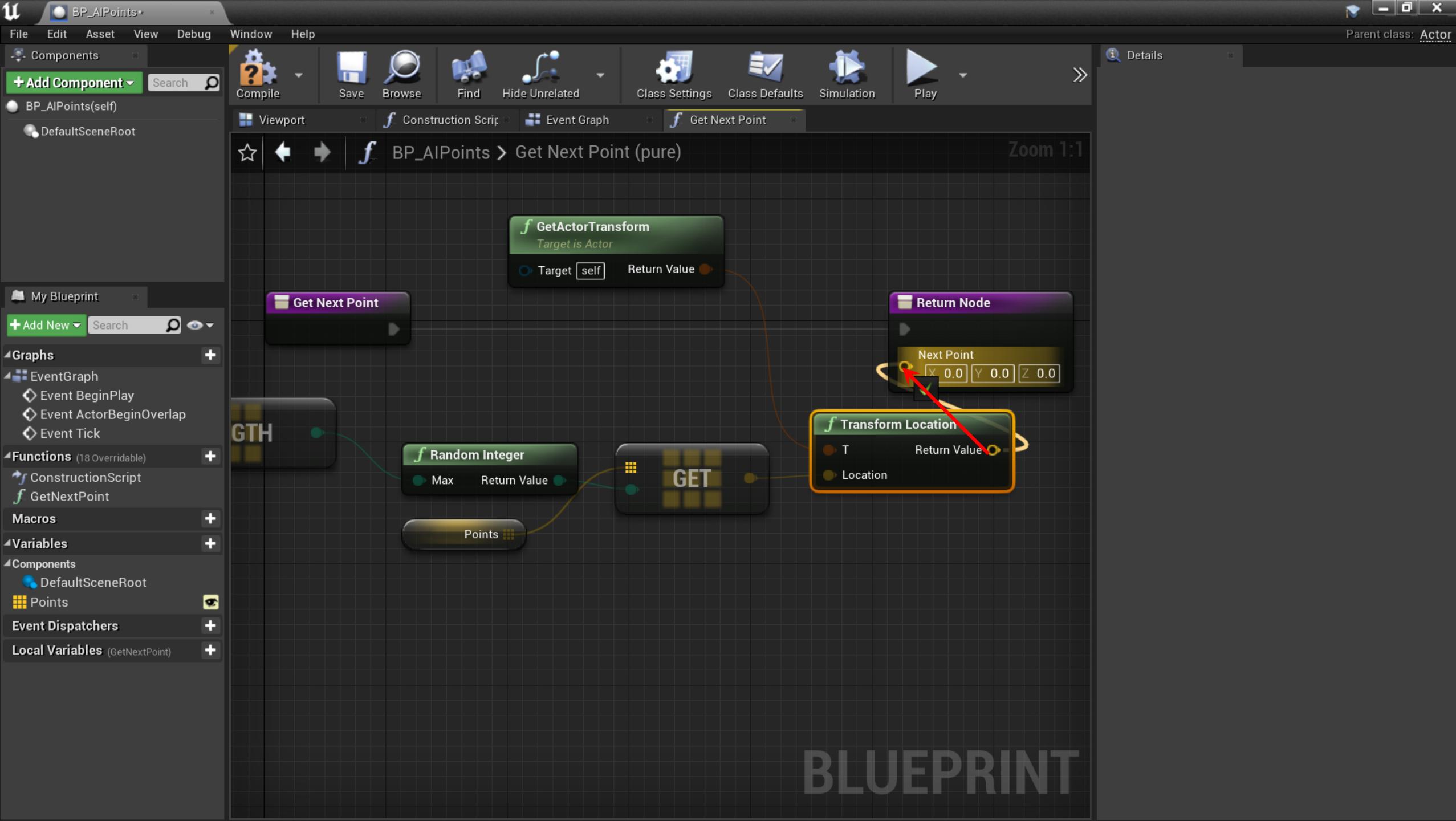


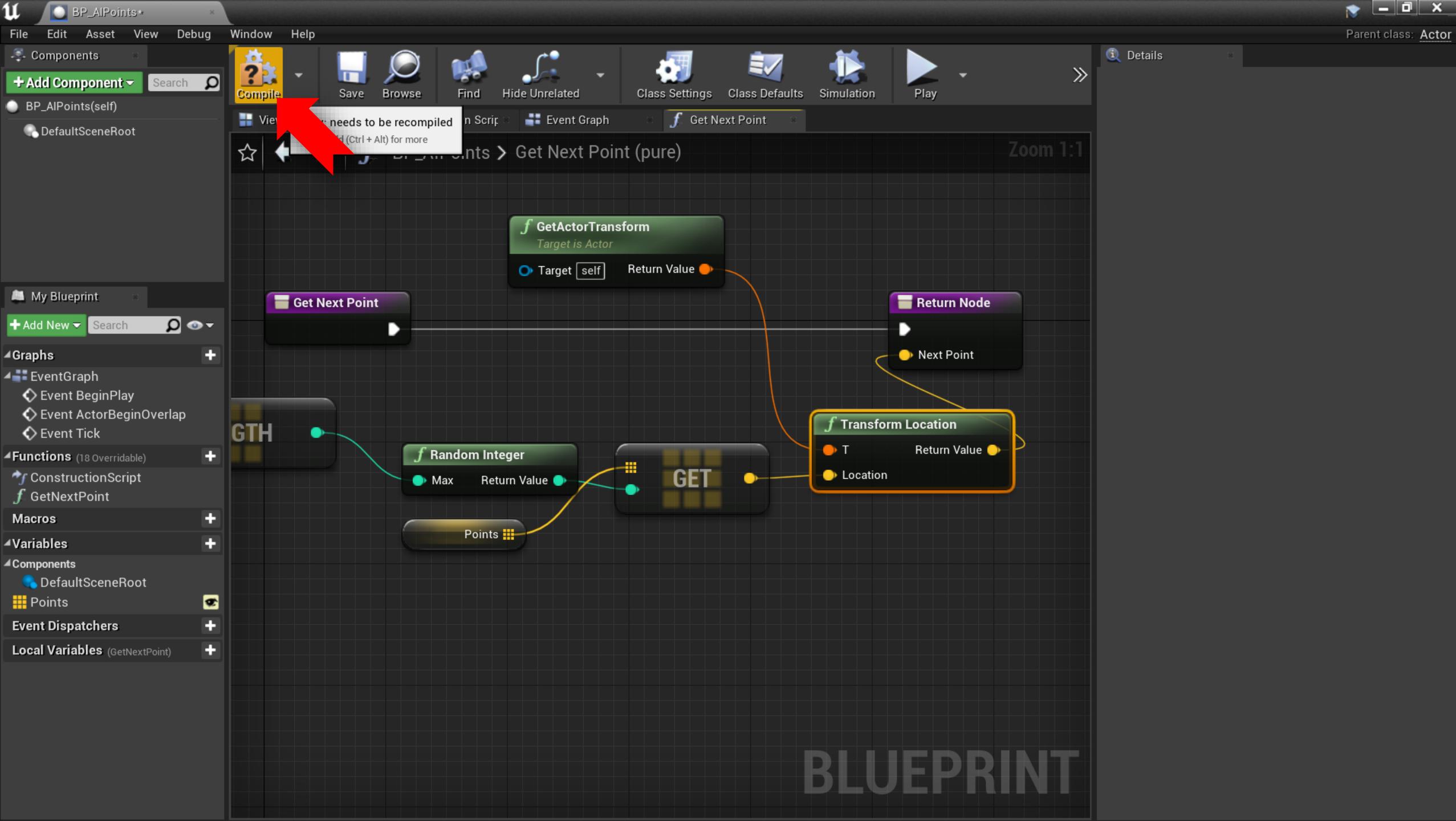




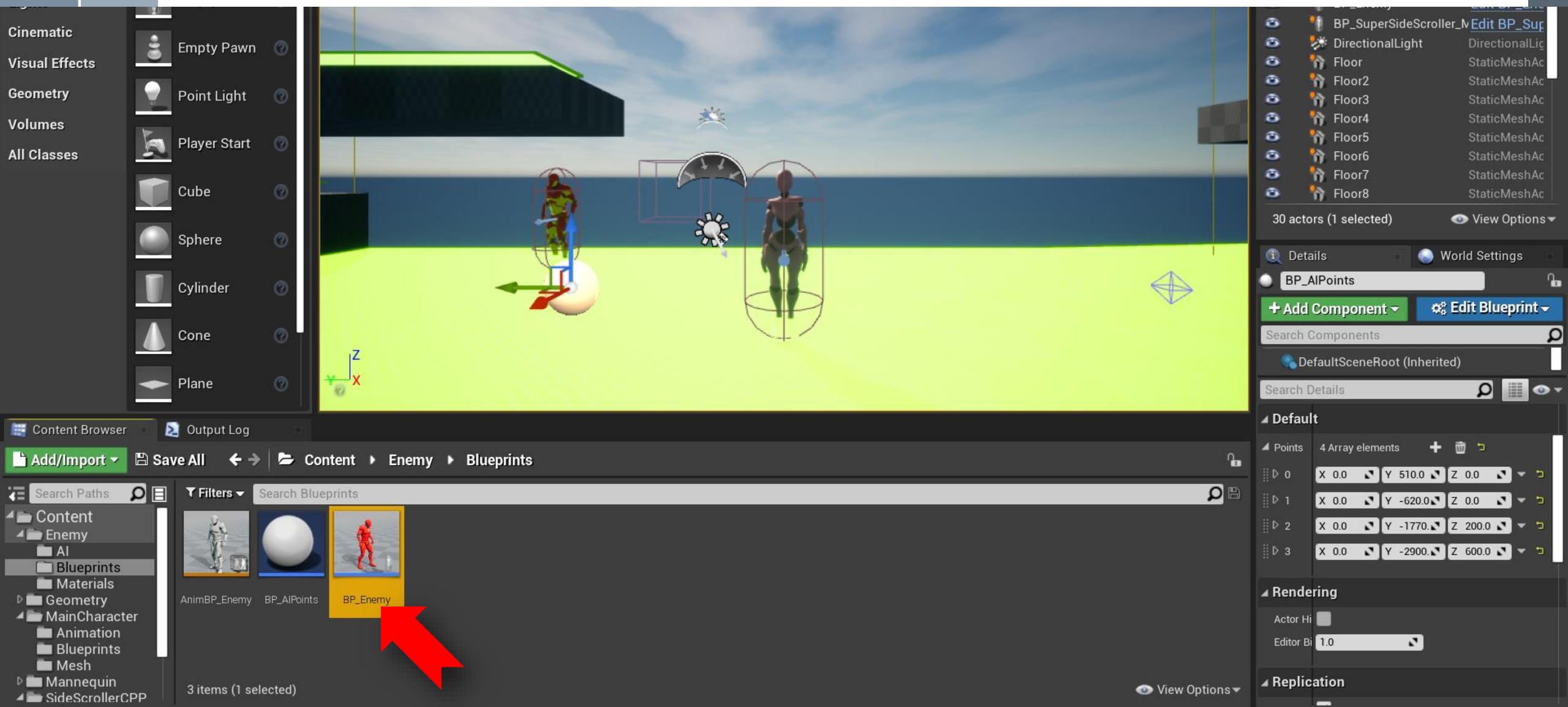


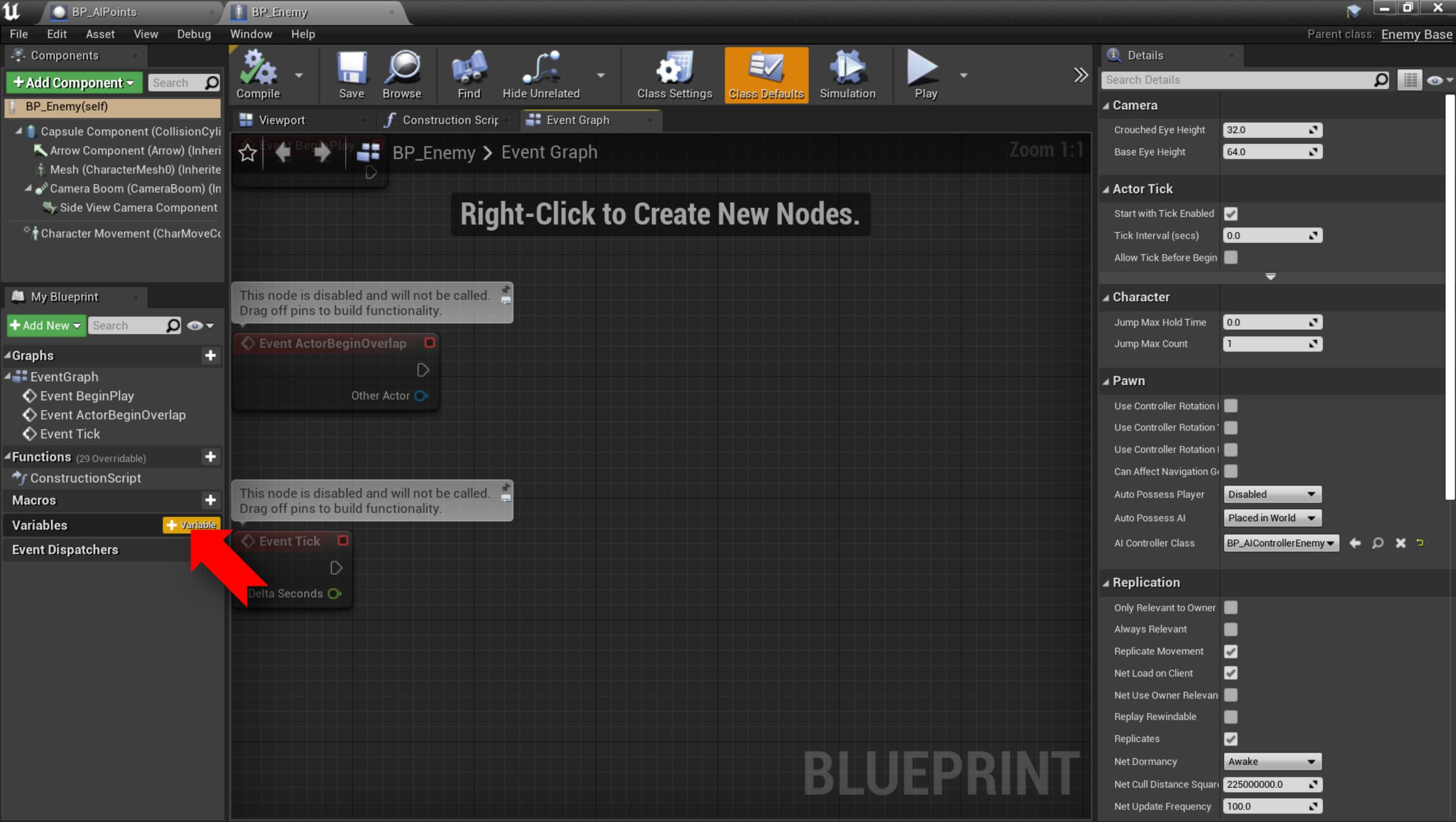


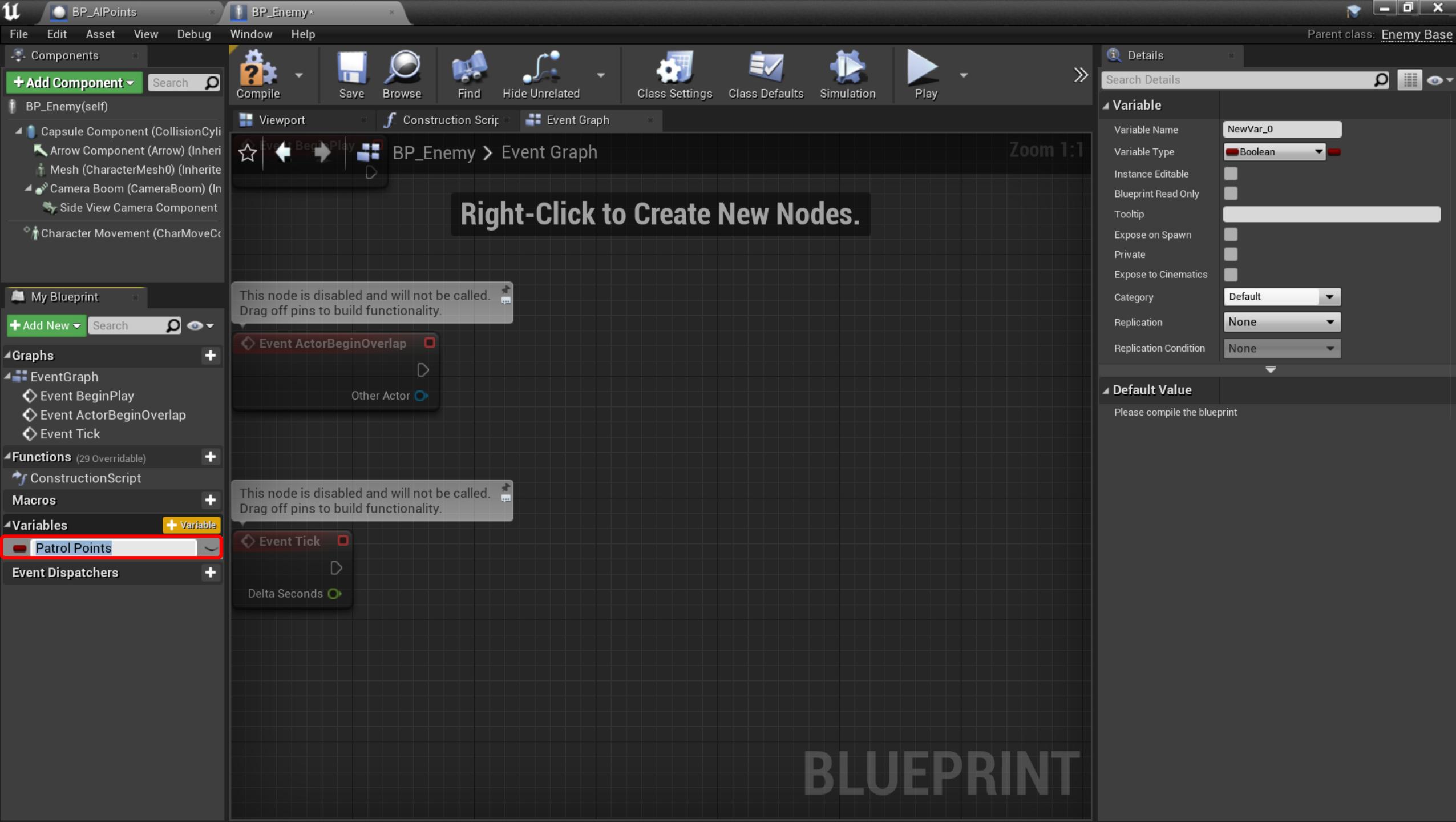


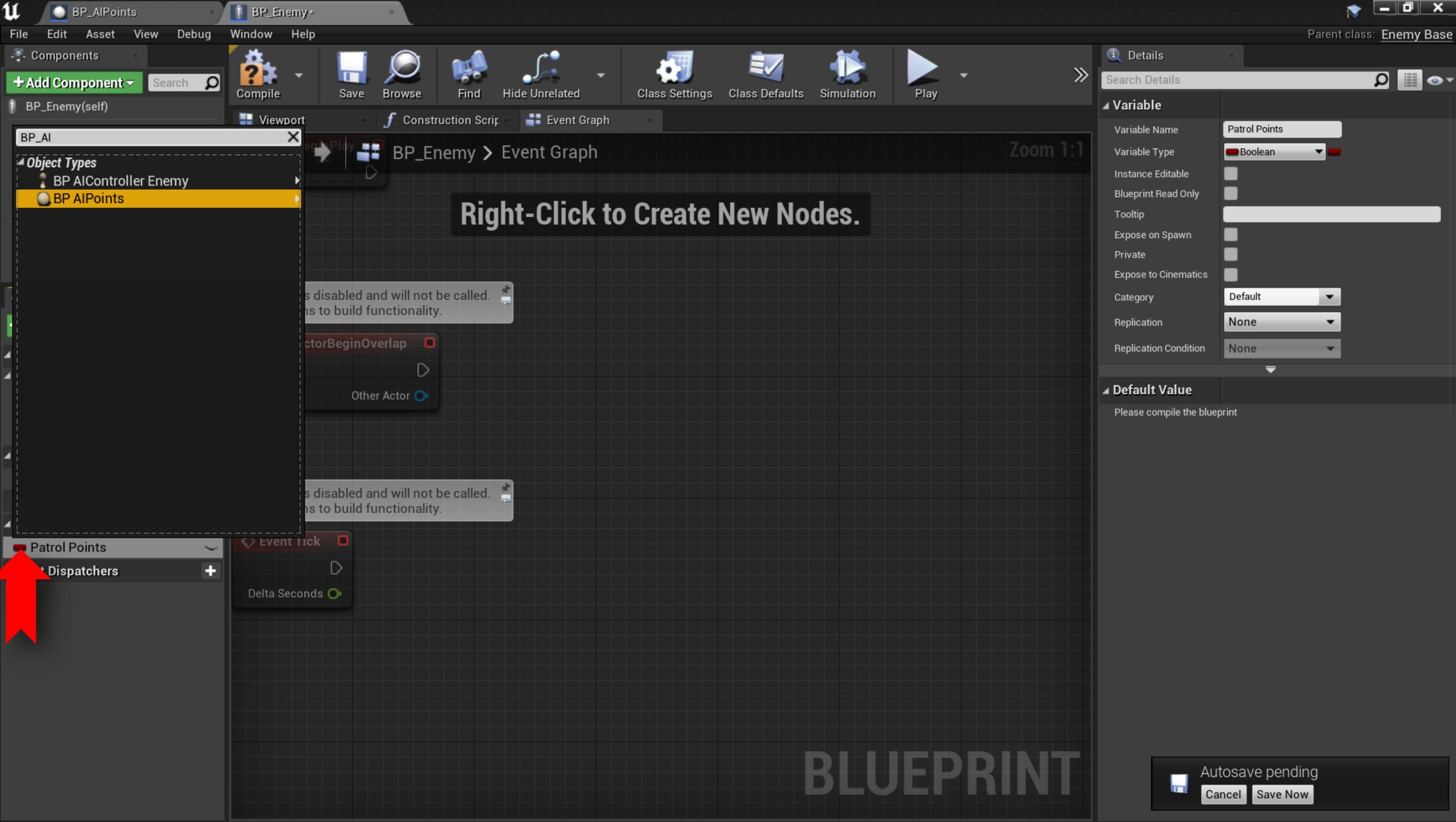


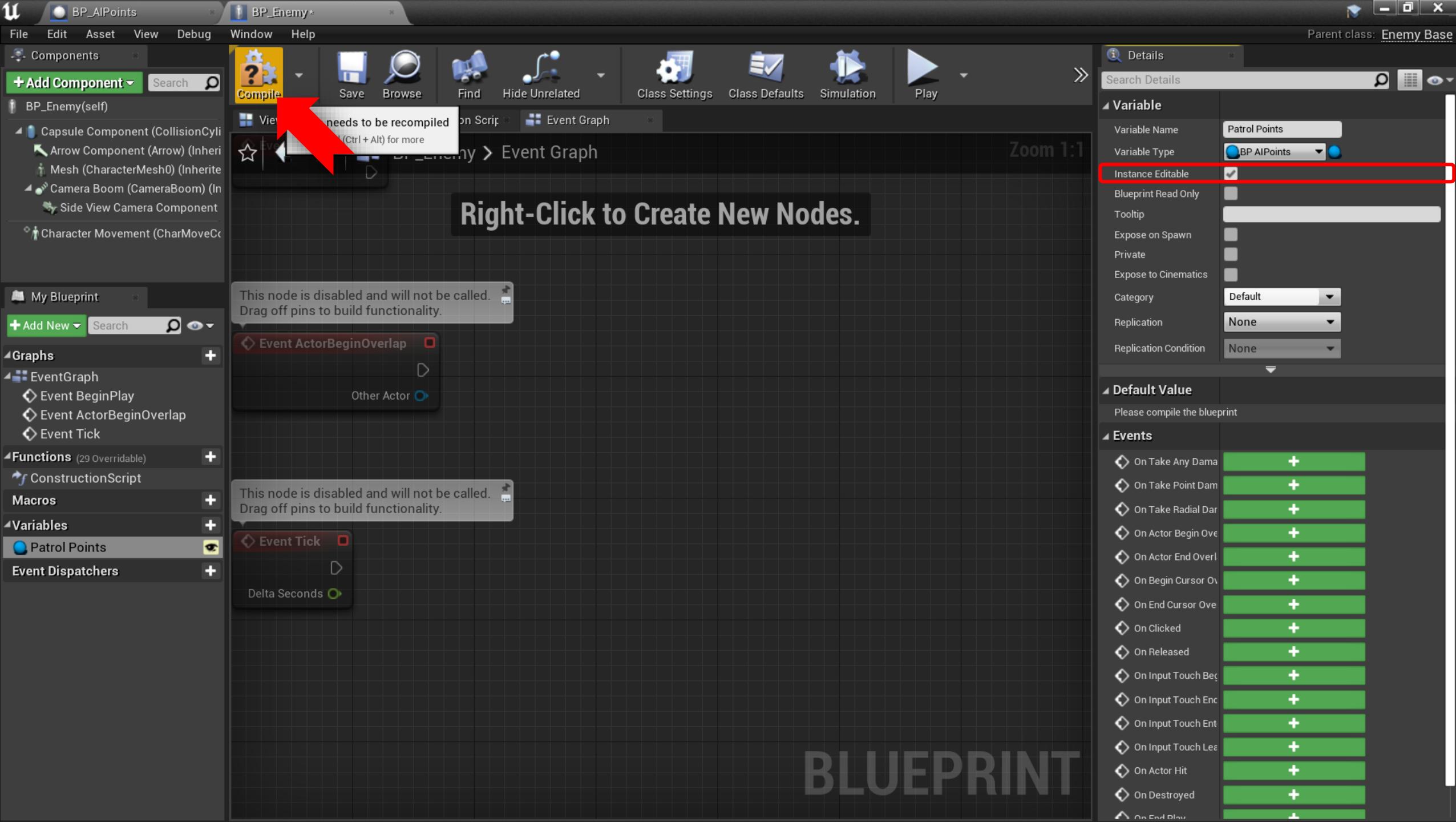
# Exercise 13.09: Referencing the Patrol Point Actor

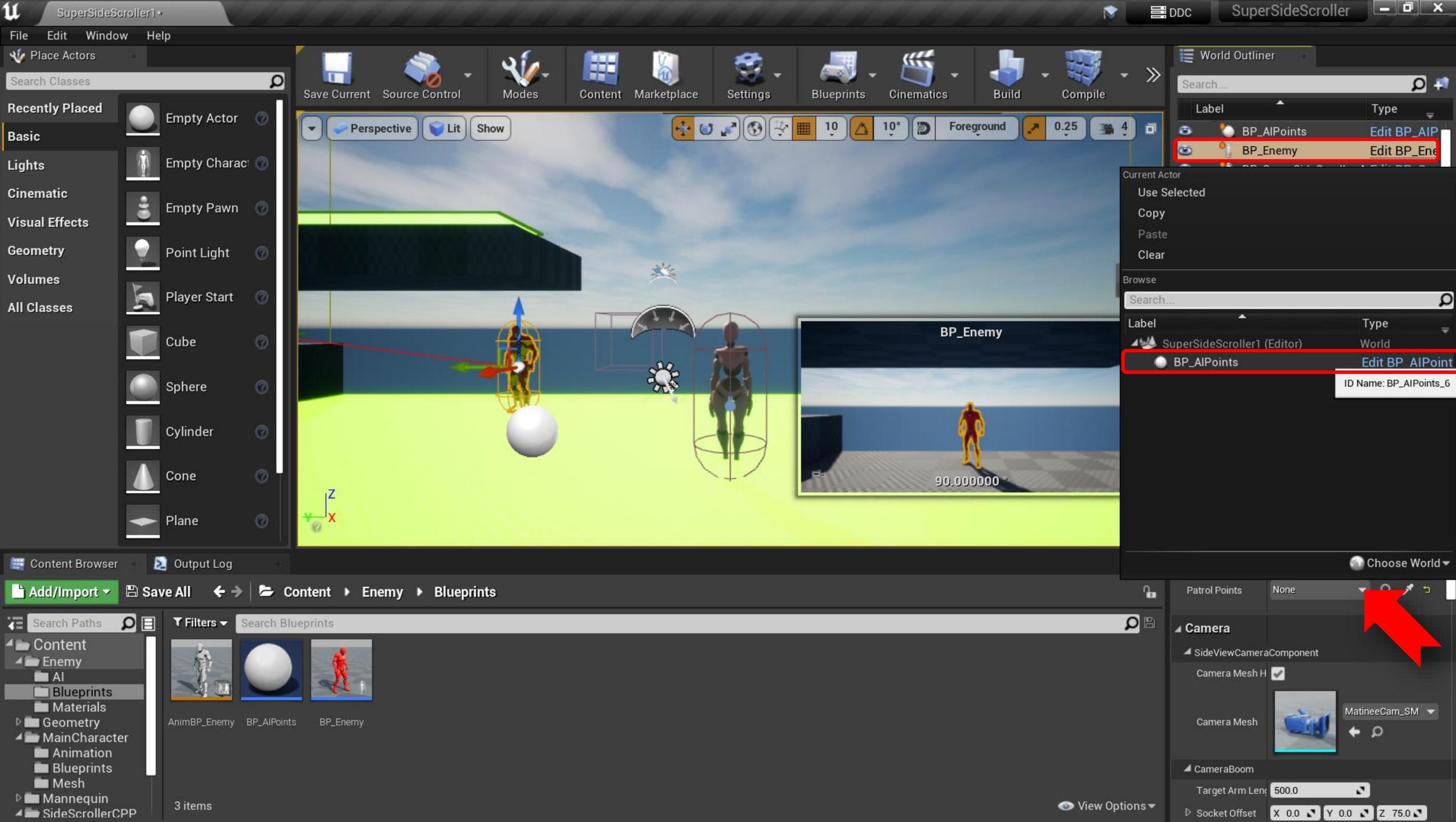










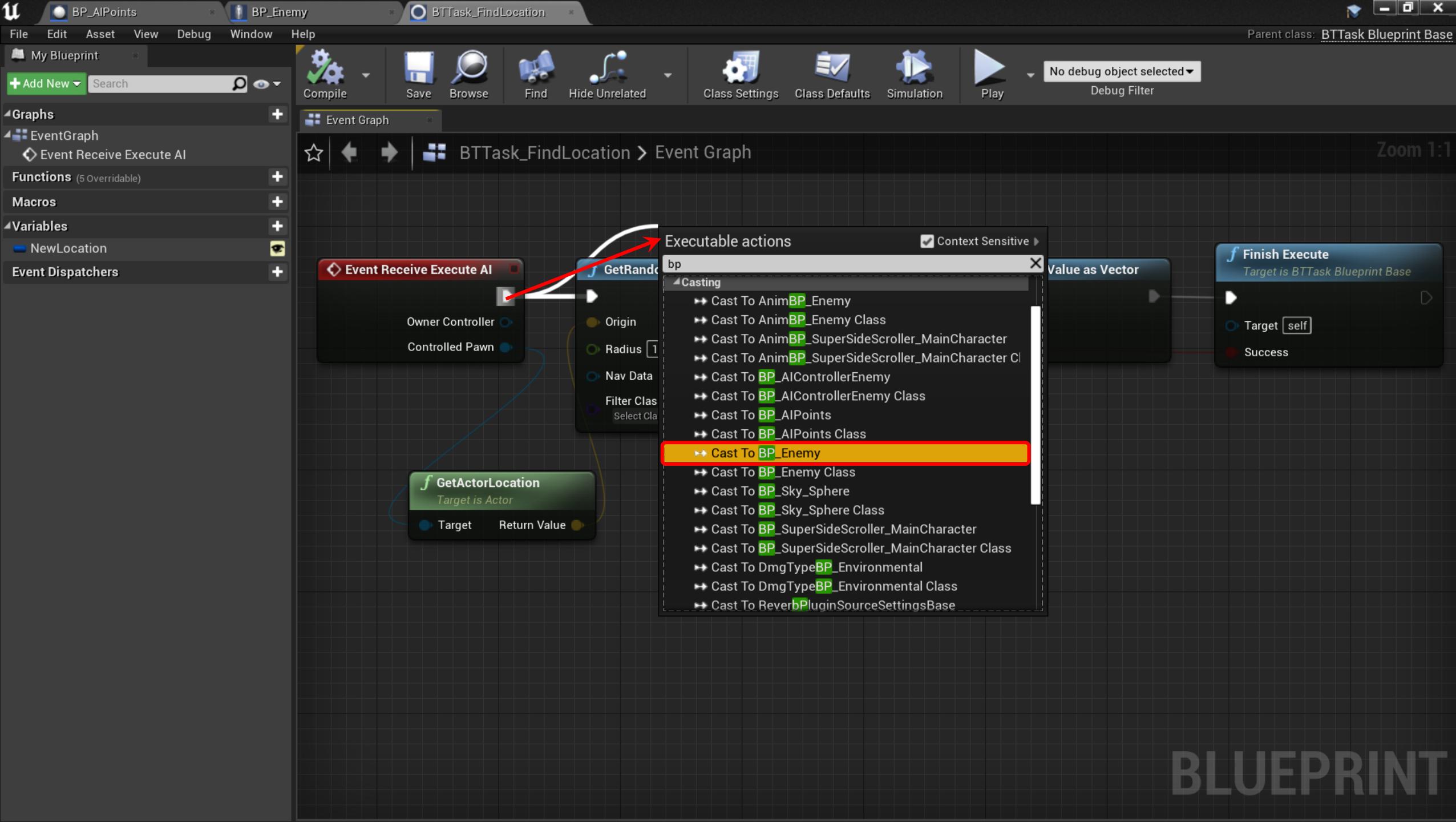


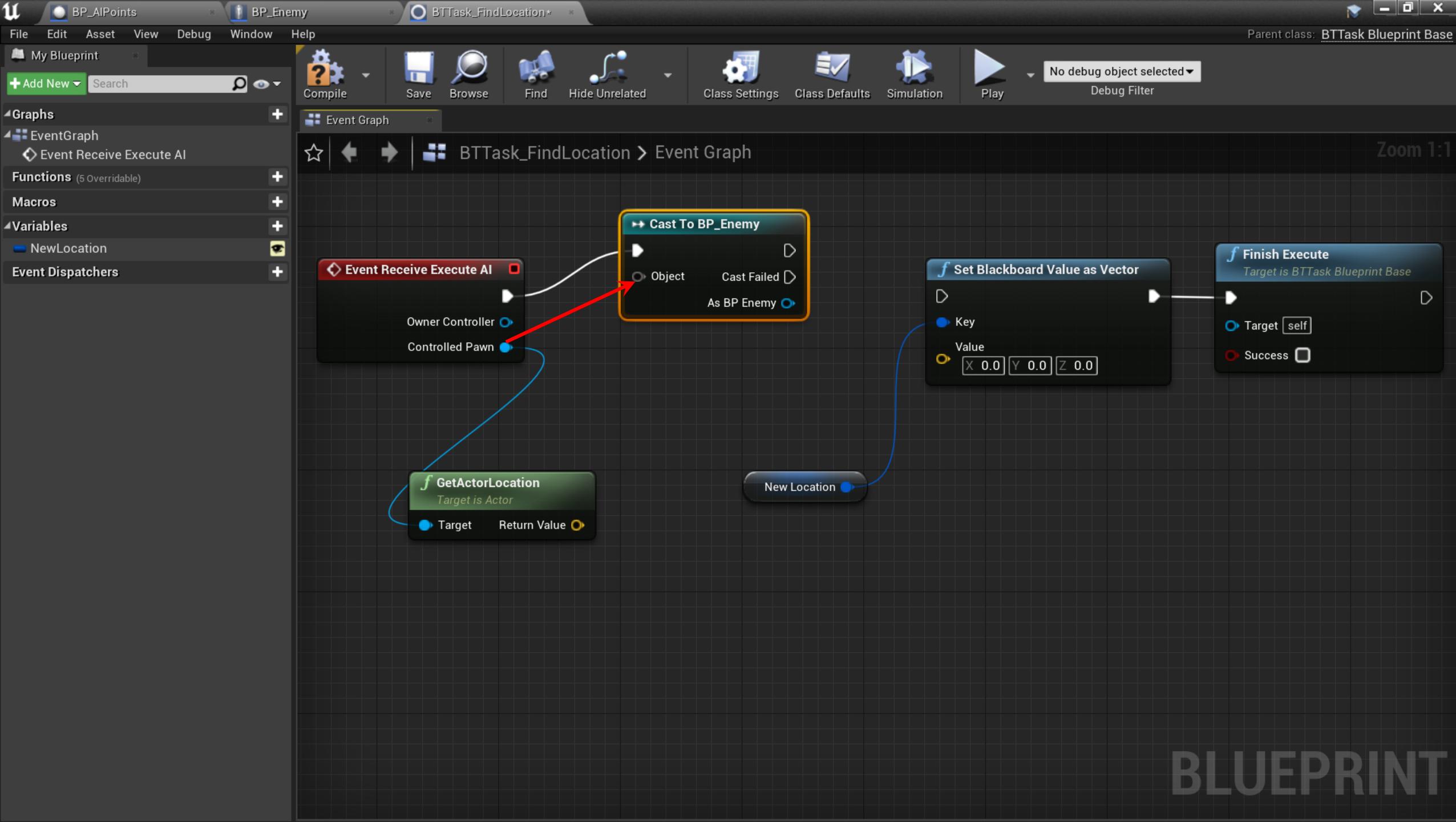


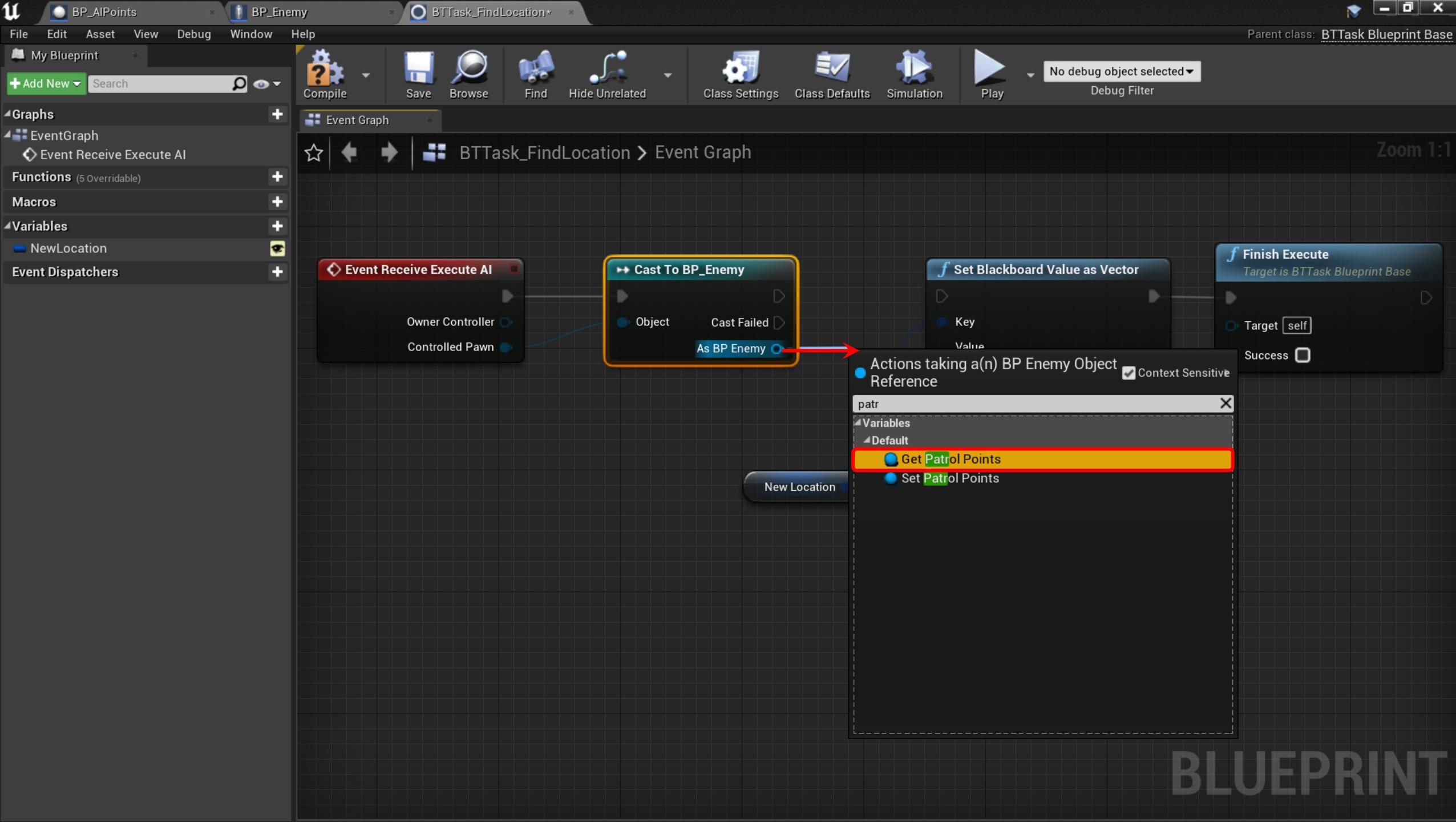
# Exercise 13.10: Updating BTTask\_FindLocation

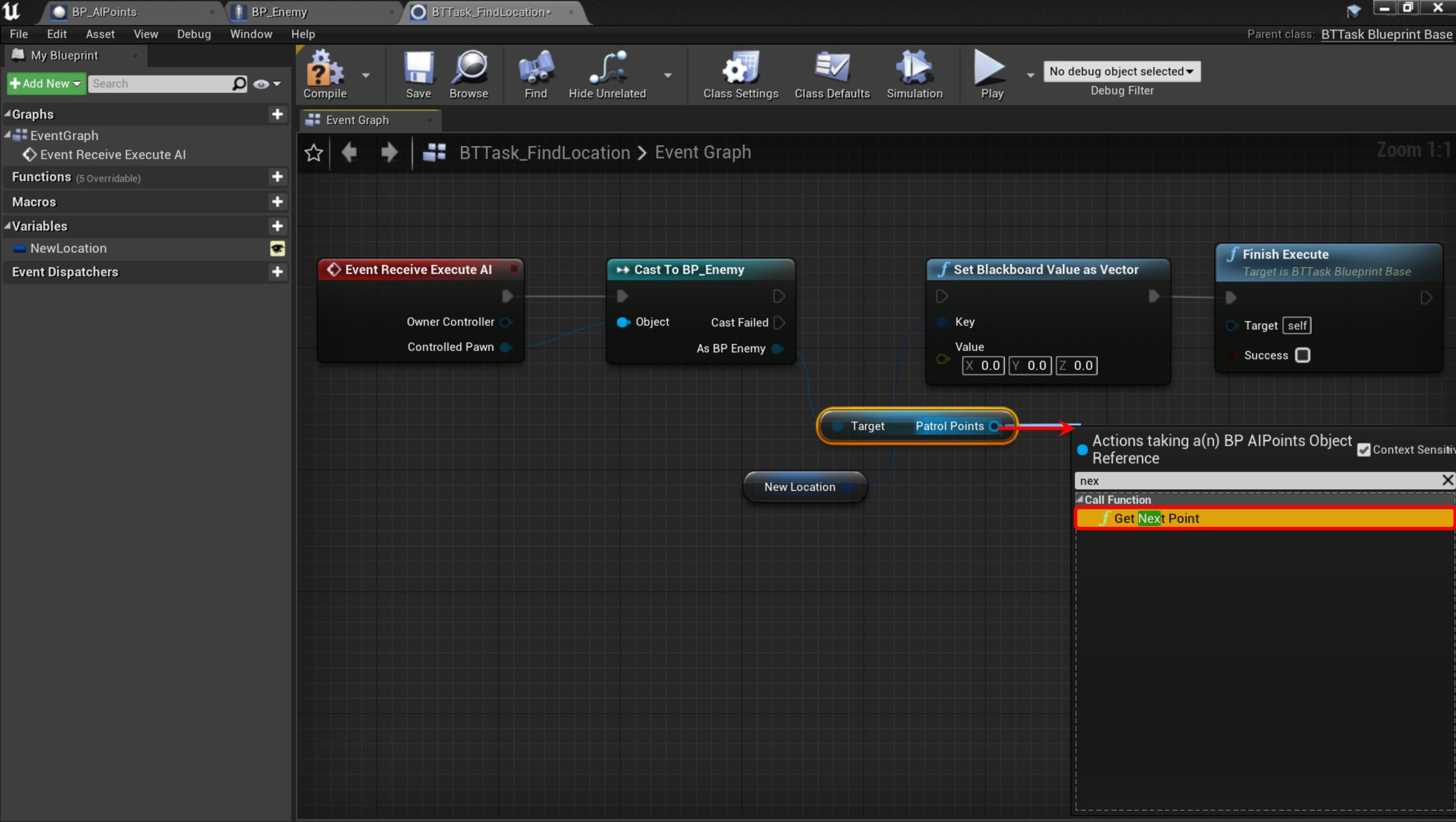
The screenshot shows the Unreal Engine Editor interface with the following components:

- Content Browser:** On the left, it displays a tree view of assets under "Content/Enemy/AI". The "BTTask\_Find Location" asset is selected and highlighted with a yellow border. A red arrow points to this asset.
- Editor Viewport:** The central area shows a 3D scene with a green floor and blue sky. It features a green cube, a white sphere, and two humanoid characters. One character is highlighted with a green bounding box and a camera icon, while the other is highlighted with a red bounding box and a gear icon. A small inset window titled "BP\_Enemy" shows a third-person view of the character with a rotation indicator of "90.000000 °".
- Details Panel:** On the right, it shows the "BP\_Enemy" component details. Under the "Default" tab, "Patrol Points" are set to "BP\_AIPoints". Under the "Camera" tab, "SideViewCameraComponent" is selected, with "Camera Mesh H" checked and "MatineeCam\_SM" assigned. Under the "CameraBoom" tab, "Target Arm Length" is set to "500.0".









My Blueprint

+ Add New Search

Graphs

EventGraph

Event Receive Execute AI

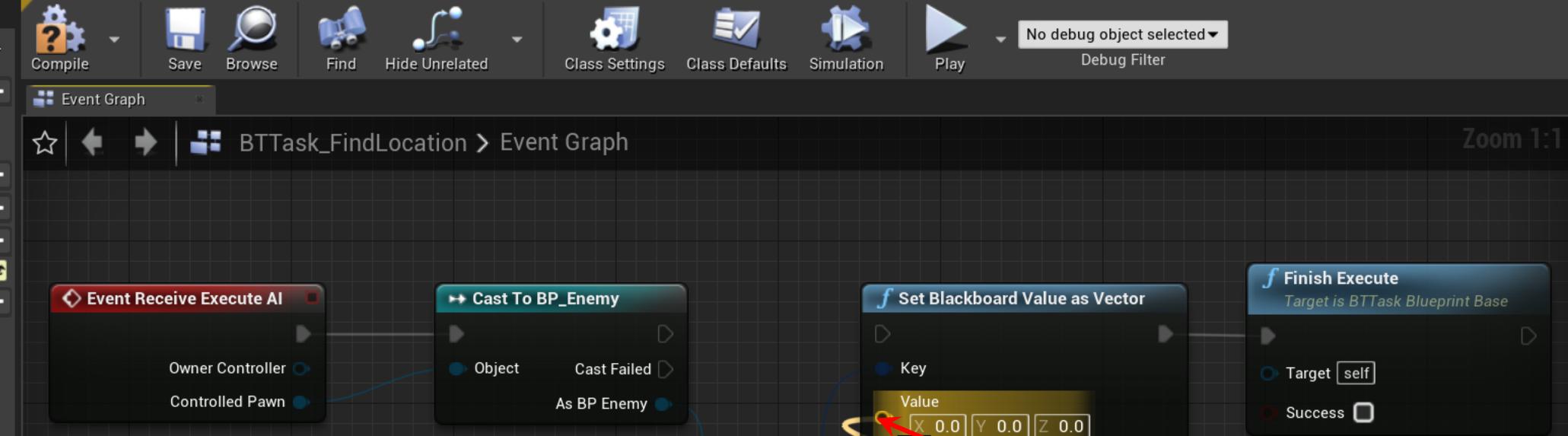
Functions (5 Overridable)

Macros

Variables

NewLocation

Event Dispatchers



BLUEPRINT

My Blueprint

+ Add New Search

Graphs

EventGraph

Event Receive Execute AI

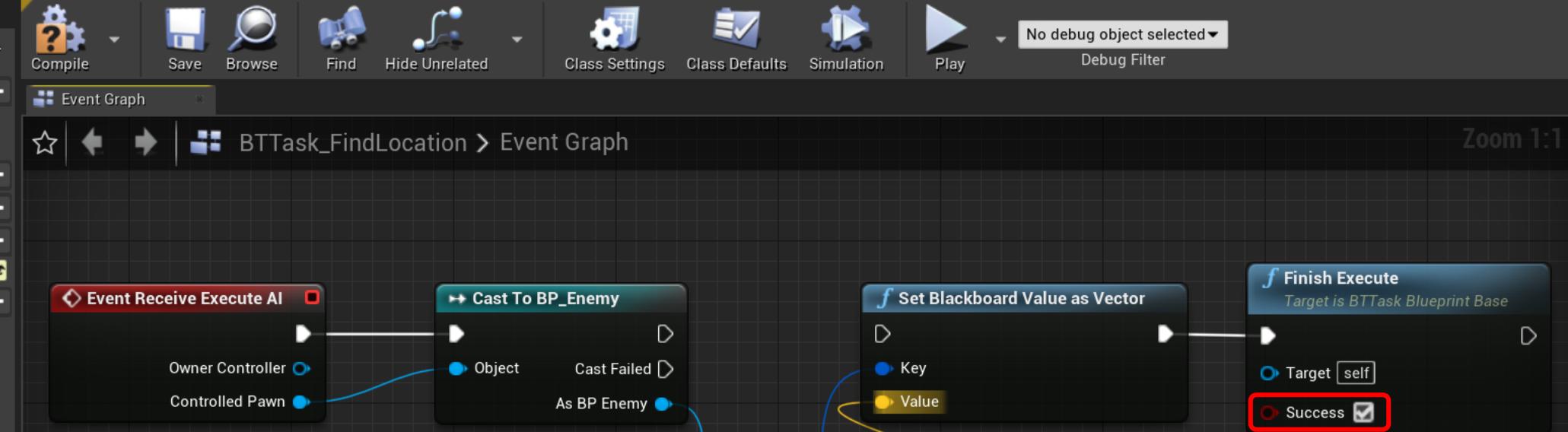
Functions (5 Overridable)

Macros

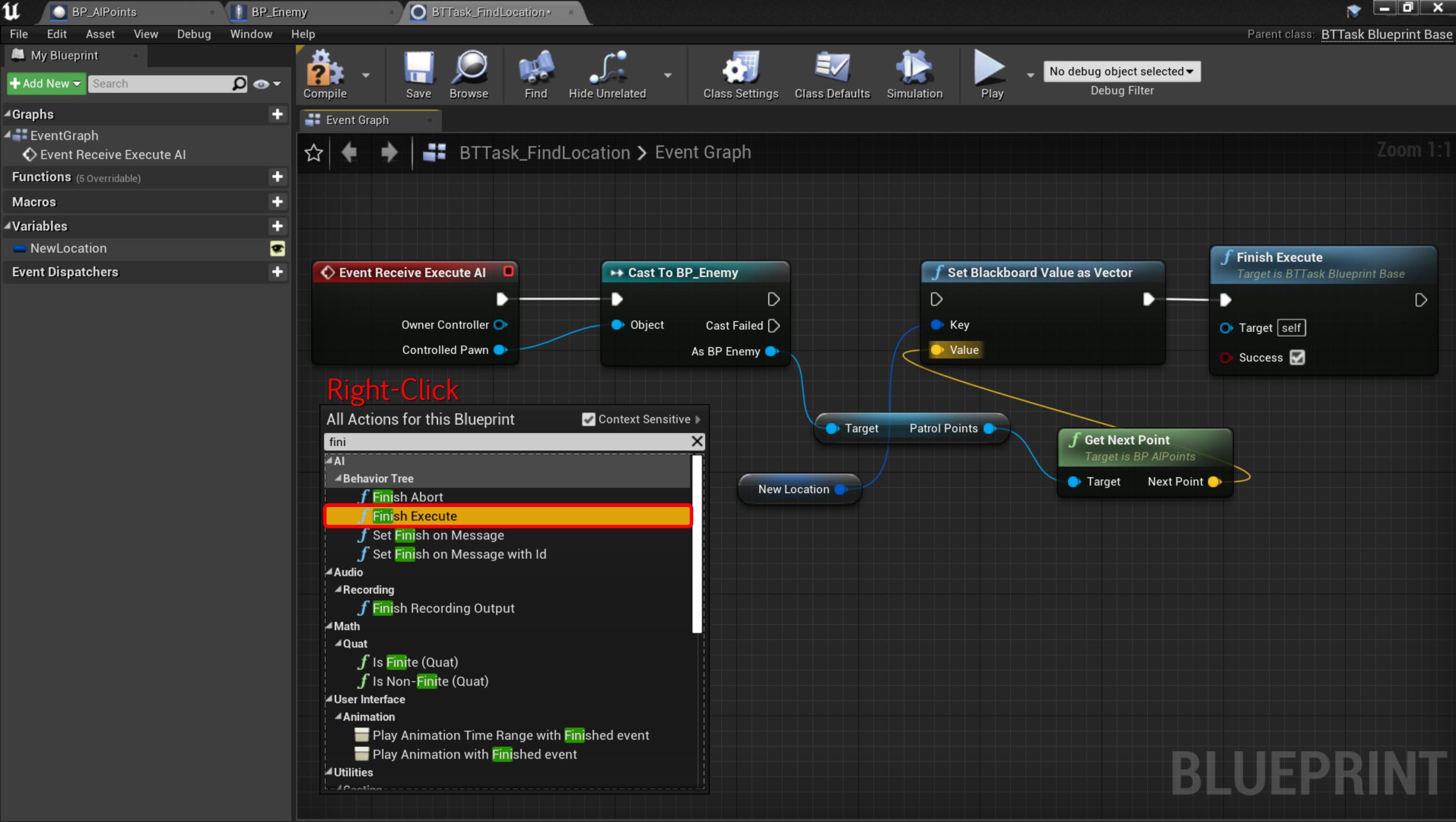
Variables

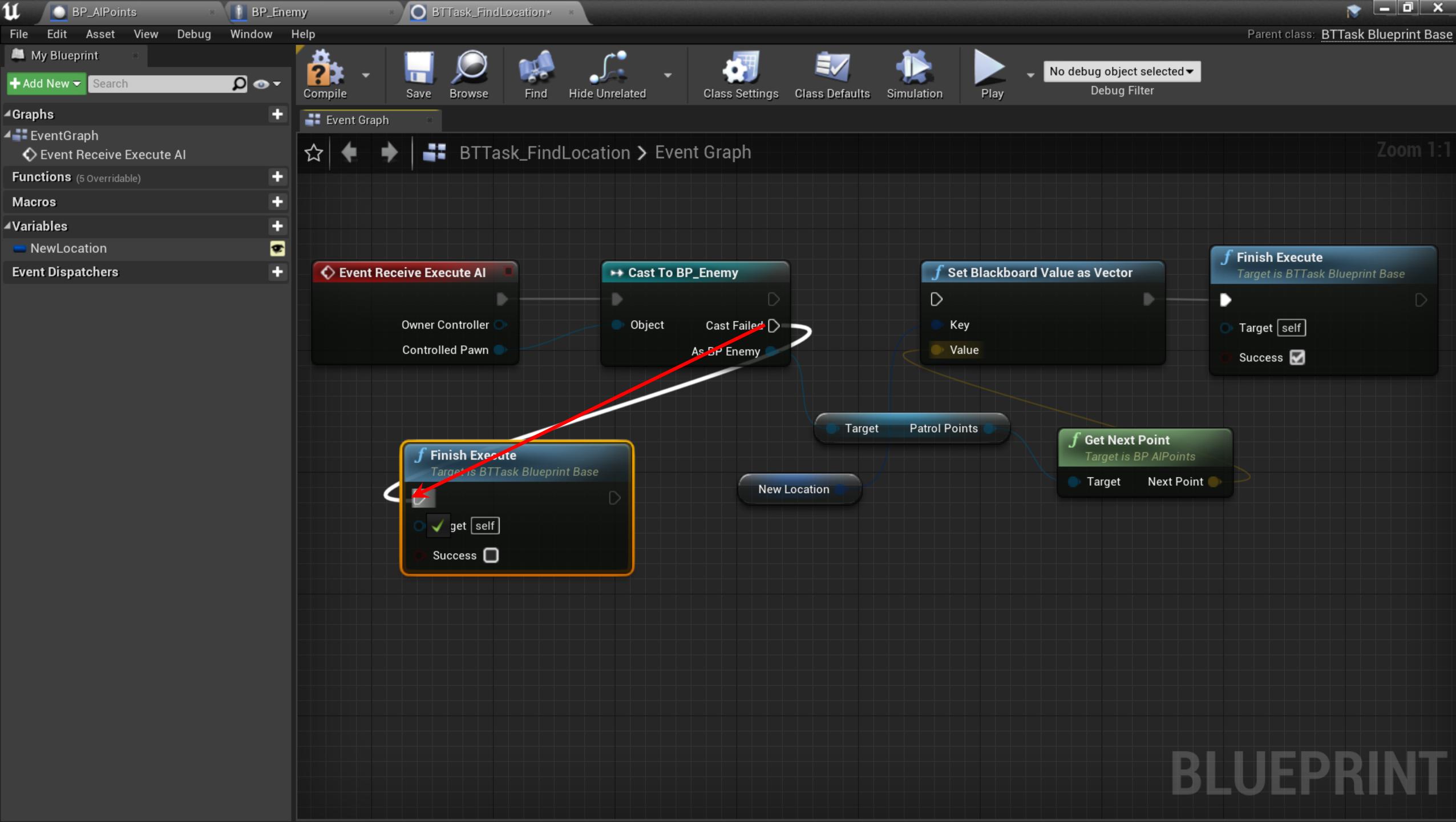
NewLocation

Event Dispatchers



BLUEPRINT





My Blueprint

+ Add New Search

Graphs

EventGraph

Event Receive Execute AI

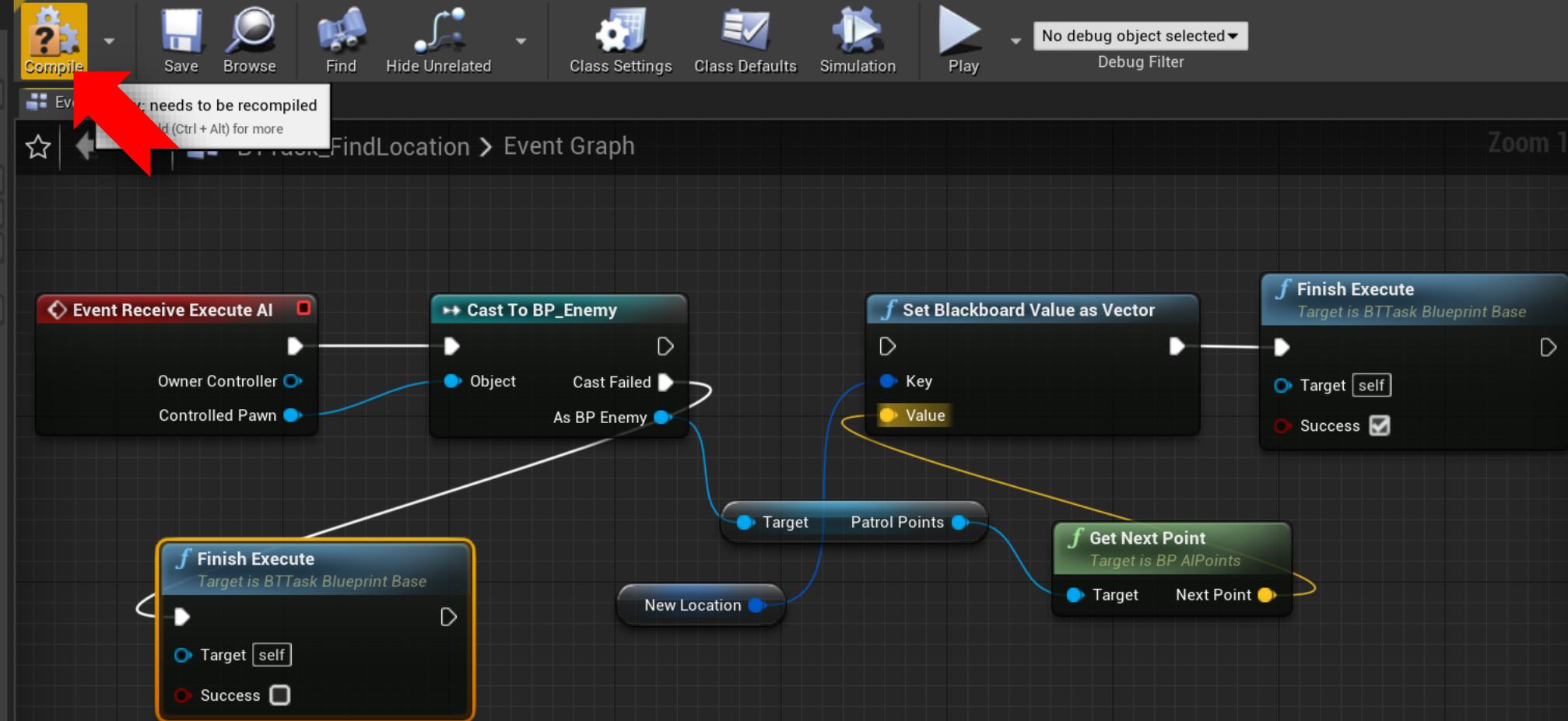
Functions (5 Overridable)

Macros

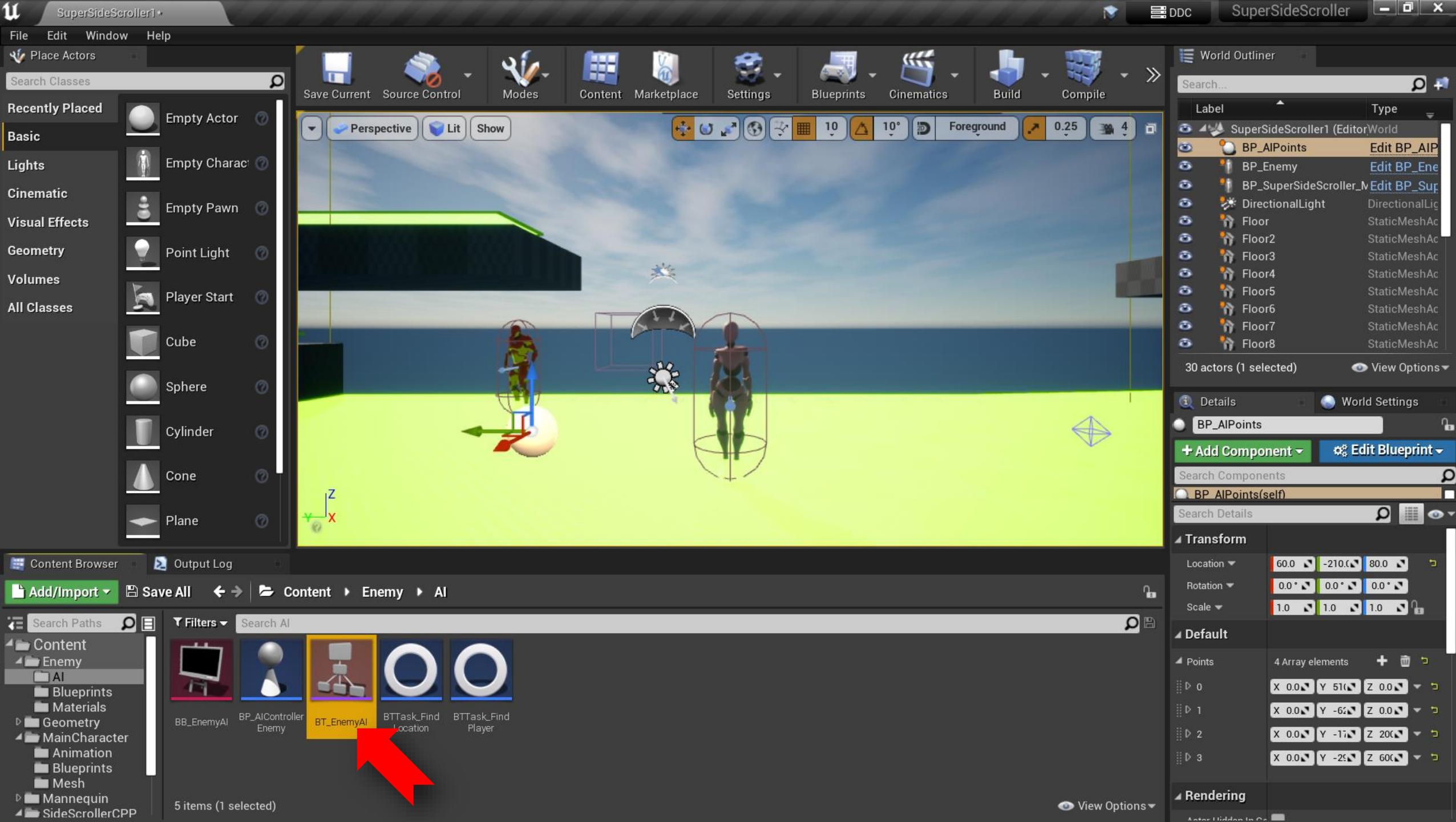
Variables

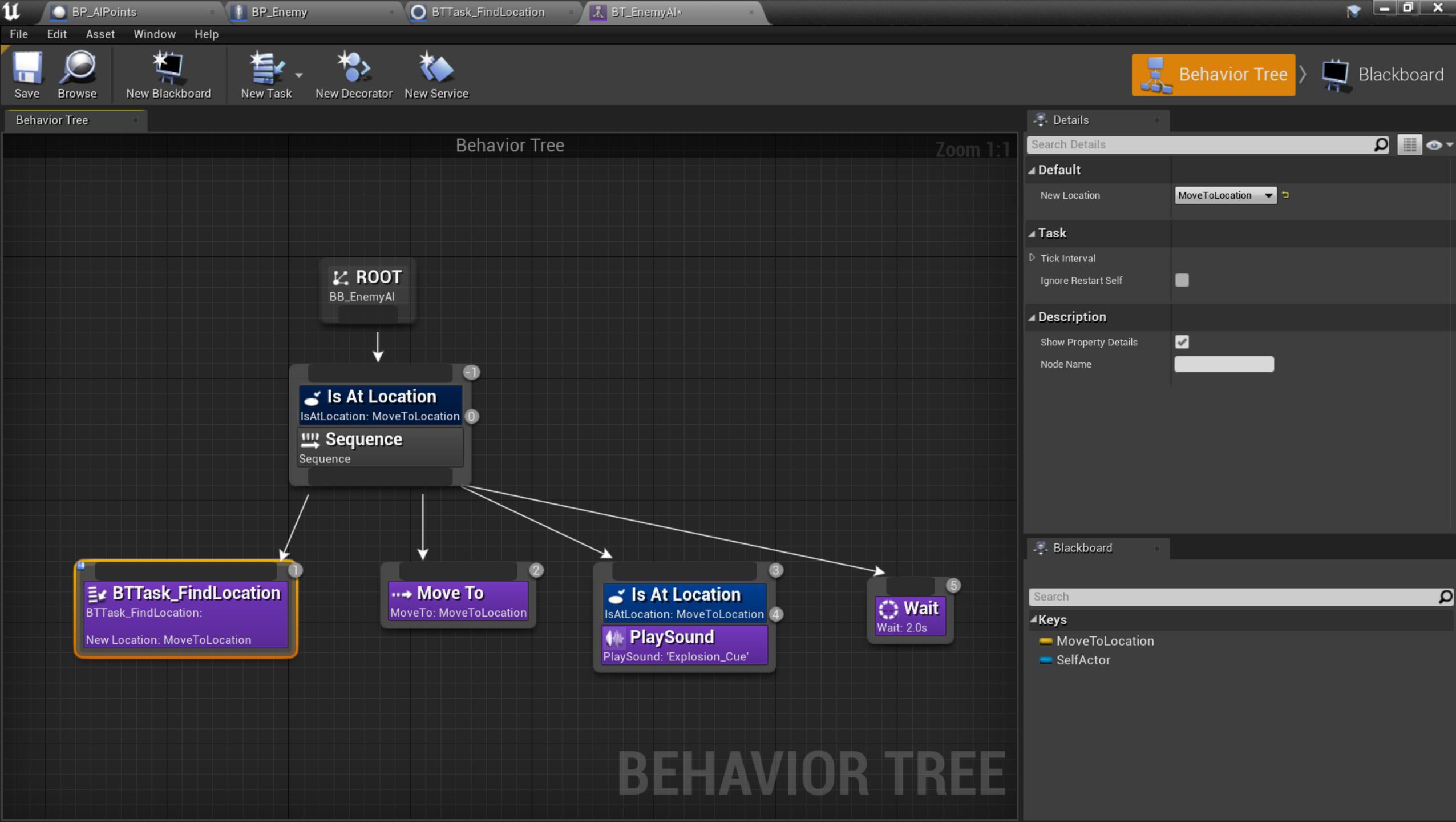
NewLocation

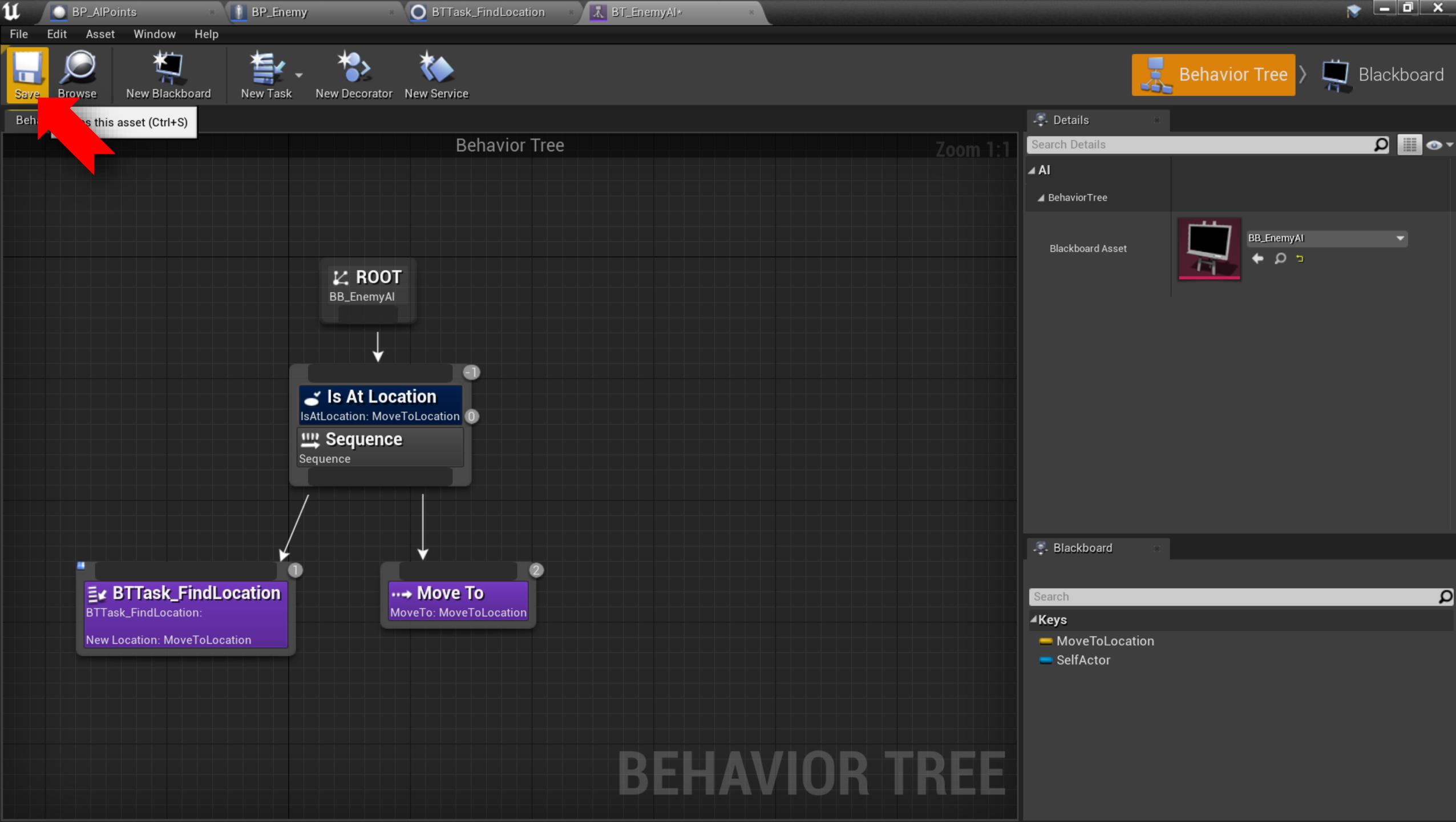
Event Dispatchers

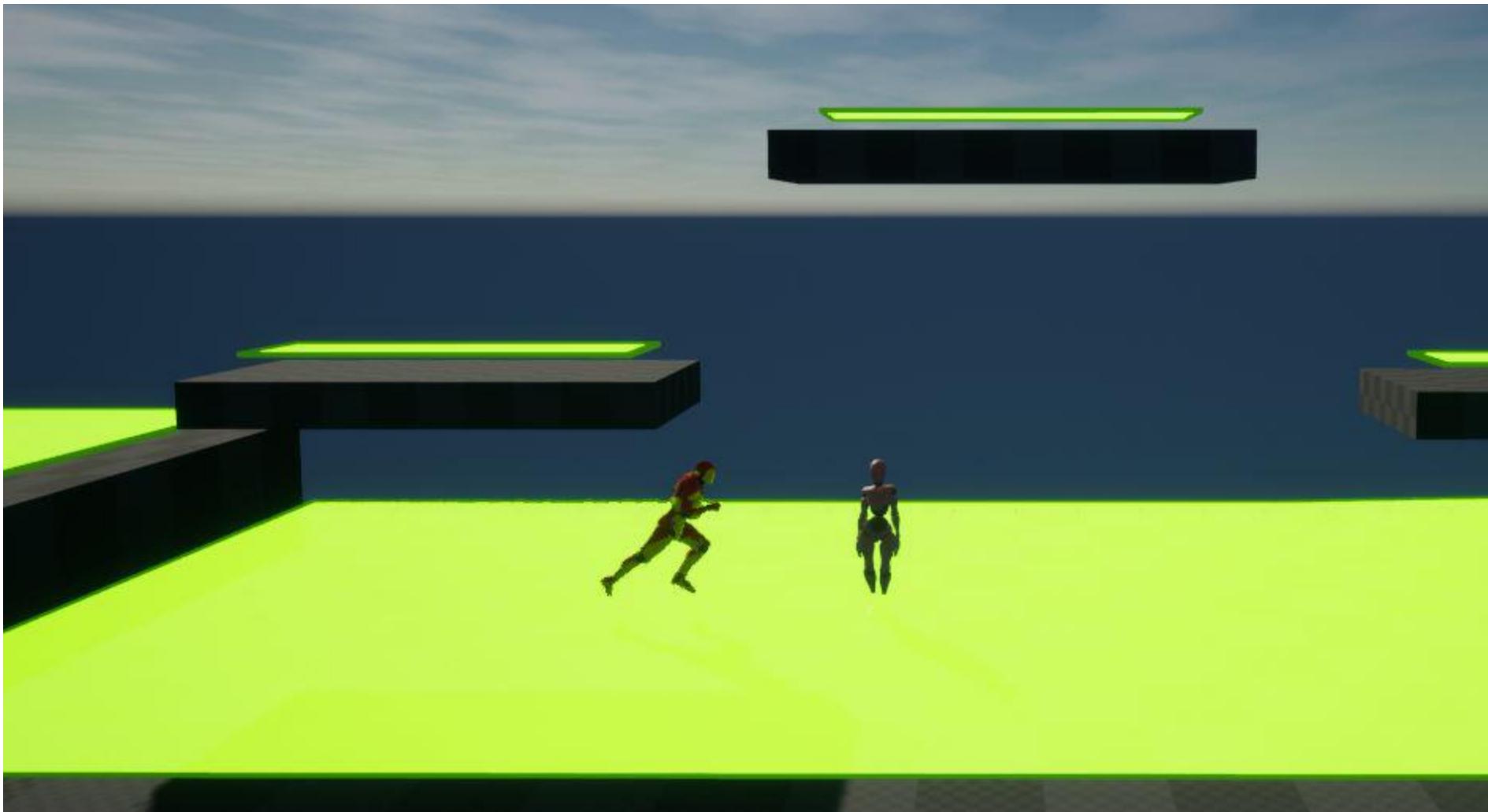


BLUEPRINT









The enemy AI now moving between the patrol point locations in the level.



# Player Projectile

- › Creating the base of the player projectile, which can be used to destroy enemies
  - To create the appropriate actor class
  - To introduce the required collision and projectile movement components to the class
  - To set up the necessary parameters for the projectile's motion behavior
- › The projectile will not use gravity, will destroy enemies with one hit, and the projectile itself will be destroyed on hitting any surface; it will not bounce off walls, for example.



# Exercise 13.11: Creating the Player

The screenshot shows the Unreal Engine 4 Editor interface with the project "SuperSideScroller" open. The left sidebar contains the "File" menu, which has "New C++ Class..." highlighted with a red box. A tooltip for this option states: "Adds C++ code to the project. The code can only be compiled if you have Visual Studio installed." The main workspace displays a 3D scene with a character pawn and a camera. To the right, the "World Outliner" panel lists various game objects, including "BP\_AIPoints", "BP\_Enemy", and several "Floor" objects. The "Content" browser at the bottom shows the path "Content > Enemy > AI".



# Choose Parent Class

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

Show All Classes

## None

An empty C++ class with a default constructor and destructor.

## 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.

Selected Class

Actor

Selected Class Source

Actor.h



Next >

Create Class

Cancel



# Name Your New Actor

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="PlayerProjectile"/> <span style="border: 2px solid red; padding: 2px;">PlayerProjectile</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/PlayerProjectile.h			
Source File	C:/Users/sunje/Desktop/Unreal/SuperSideScroller/Source/SuperSideScroller/PlayerProjectile.cpp			

< Back

Create Class 

Cancel

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\* ✎ X

SuperSideScroller APlayerProjectile

```
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/Actor.h"
7 #include "PlayerProjectile.generated.h"
8
9 UCLASS()
10 class SUPERSIDESCROLLER_API APlayerProjectile : public AActor
11 {
12     GENERATED_BODY()
13
14 public:
15     // Sets default values for this actor's properties
16     APlayerProjectile();
17
18 protected:
19     // Called when the game starts or when spawned
20     //virtual void BeginPlay() override;
21
22 public:
23     // Called every frame
24     //virtual void Tick(float DeltaTime) override;
25
26 };
27
```

Delete

Ctrl+S

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+Shift+F)

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
    - SuperSideScroller.uproject
  - Visualizers
  - UE4.natvis

준비

문제가 검색되지 않음

줄: 24 문자: 4 열: 7 템 CRLF

소스 제어에 추가

The screenshot shows the Unreal Engine 4 Editor interface. The main window displays the code for `PlayerProjectile.cpp`. A red box highlights the tab bar, and a red circle highlights the file name `PlayerProjectile.cpp*`. A red bracket on the right side of the code area encloses several lines of code, with the word "Delete" written in red at the bottom right of the bracket. The code includes comments about copyright notices and default values for the `APlayerProjectile` class.

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #include "PlayerProjectile.h"
4
5 // Sets default values
6 APlayerProjectile::APlayerProjectile()
7 {
8     // Set this actor to call Tick() every frame. You can turn this off to improve performance if you don't need it.
9     PrimaryActorTick.bCanEverTick = true;
10 }
11
12
13 // Called when the game starts or when spawned
14 //void APlayerProjectile::BeginPlay()
15 //{
16 //    Super::BeginPlay();
17 //
18 //}
19
20 // Called every frame
21 //void APlayerProjectile::Tick(float DeltaTime)
22 //{
23 //    Super::Tick(DeltaTime);
24 //
25 //}
26
27
28
```

The Solution Explorer on the right lists the project structure for `'SuperSideScroller' (2/2 개 프로젝트)`, including the `Engine` and `Games` folders, and the `SuperSideScroller` module with its source files like `EnemyBase.cpp` and `PlayerProjectile.cpp`.

Bottom status bar: 100% (문제가 검색되지 않음), 줄: 26, 문자: 3, 혼합, CRLF, 슬루션 탐색기, Git 변경 내용.

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

PlayerProjectile.cpp PlayerProjectile.h\* X

SuperSideScroller APlayerProjectile::UStaticMeshComponent

```
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/Actor.h"
7 #include "PlayerProjectile.generated.h"
8
9 UCLASS()
10 class SUPERSIDESCRROLLER_API APlayerProjectile : public AActor
11 {
12     GENERATED_BODY()
13
14 public:
15     // Sets default values for this actor's properties
16     APlayerProjectile();
17
18     //Sphere collision component
19     UPROPERTY(VisibleDefaultsOnly, Category = Projectile)
20     class USphereComponent* CollisionComp;
21
22     //called when projectile hits something
23     UFUNCTION()
24     void OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp, FVector NormalImpulse,
25                 const FHitResult& Hit);
26
27 private:
28
29     //Projectile movement component
30     UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Movement, meta = (AllowPrivateAccess = "true"))
31     class UProjectileMovementComponent* ProjectileMovement;
32
33     //Static mesh component
34     UPROPERTY(VisibleDefaultsOnly, Category = Projectile)
35     class UStaticMeshComponent* MeshComp;
36
37 };
```

Ctrl+S

100 % 문제가 검색되지 않음 줄: 35 문자: 39 열: 42 탭 CRLF

슬루션 탐색기

슬루션 탐색기 검색(Ctrl+Shift+F)

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
    - SuperSideScroller.uproject
  - Visualizers
  - UE4.natvis

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

Develop Win64 로컬 Windows 디버거 Live Share

서버  
템플릿  
도구상자

PlayerProjectile.cpp\* PlayerProjectile.h

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

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #include "PlayerProjectile.h"
4
5 // Sets default values
6 APlayerProjectile::APlayerProjectile()
7 {
8     // Set this actor to call Tick() every frame. You can turn this off to improve performance if you don't need it.
9     PrimaryActorTick.bCanEverTick = true;
10 }
11
12
13 void APlayerProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp,
14     FVector NormalImpulse, const FHitResult& Hit)
15 {
16 }
17
18 }
```

Ctrl+S

100 % 문제가 검색되지 않음 줄: 17 문자: 5 혼합 CRLF

서버 템플릿 도구상자

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 탐색기 '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
- SuperSideScroller.uproject

Visualizers

UE4.natvis

솔루션 탐색기 Git 변경 내용

저장되었습니다. ↑ 소스 제어에 추가 ↗

The screenshot shows the Unreal Engine 4 Editor interface. The top navigation bar includes tabs like 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 and a user icon are also present. The main workspace displays code for PlayerProjectile.cpp and PlayerProjectile.h. On the right, the Solution Explorer shows the project structure for SuperSideScroller, including Engine, Games, and Source folders with various files like EnemyBase.cpp and SuperSideScroller.h. A large red arrow points to the 'Build Solution' option in the Build menu.

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 Live Share

PlayerProjectile.cpp PlayerProjectile.h

SuperSideScroller

```
1 // Fill out your copyright notice i
2
3 #include "PlayerProjectile.h"
4
5 // Sets default values
6 APlayerProjectile::APlayerProjectile()
7 {
8     // Set this actor to call Tick() on every frame. You can turn this off in the Inspector if you don't need it.
9     PrimaryActorTick.bCanEverTick = true;
10 }
11
12 void APlayerProjectile::OnHit(UPrimitiveComponent * HitComp, AActor * Actor, FVector NormalImpulse, const FHitResult & Hit)
13 {
14 }
```

솔루션 빌드(B) 솔루션 다시 빌드 Ctrl+Shift+B

솔루션 정리(C)

솔루션의 전체 프로그램 데이터베이스 파일 빌드

솔루션에서 코드 분석 실행(Y) Alt+F11

SuperSideScroller 빌드(U) Ctrl+B

SuperSideScroller 다시 빌드(E)

SuperSideScroller 정리(N)

SuperSideScroller에서 코드 분석 실행(A)

프로젝트만(I)

일괄 빌드(T)...

구성 관리자(O)...

컴파일(M) Ctrl+F7

파일에서 코드 분석 실행(F) Ctrl+Shift+Alt+F7

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 'SuperSideScroller' (2/2개 프로젝트)

- Engine
- 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
  - SuperSideScroller.uproject
- Visualizers
- UE4.natvis

100 % 문제가 검색되지 않음 줄: 17 문자: 5 혼합 CRLF

저장되었습니다. ↑ 소스 제어에 추가 ↗

# Exercise 13.12: Initializing Player Projectile Settings

The screenshot shows the Unreal Engine Editor interface. The top bar includes the Unreal Engine logo, menu items (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)), a search bar (검색 (Ctrl+Q)), and a Super...ller button. Below the menu is a toolbar with various icons. The main area shows two files open: PlayerProjectile.cpp\* and PlayerProjectile.h. The code editor has a red box highlighting the include statements in PlayerProjectile.cpp. To the right is the Project Browser, which displays the project structure for 'SuperSideScroller'.

```
PlayerProjectile.cpp* + X PlayerProjectile.h
SuperSideScroller (전역 범위)
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "PlayerProjectile.h"
5 #include "GameFramework/ProjectileMovementComponent.h"
6 #include "Components/SphereComponent.h"
7 #include "Components/StaticMeshComponent.h"
8
9 // Sets default values
10 APlayerProjectile::APlayerProjectile()
11 {
12     // Set this actor to call Tick() every frame. You can turn this off to improve performance if you don't need it.
13     PrimaryActorTick.bCanEverTick = true;
14 }
15
16 void APlayerProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp,
17 FVector NormalImpulse, const FHitResult& Hit)
18 {
19 }
20
21
22
23
```

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+I)

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.cpp
    - SuperSideScrollerCharacter.h
    - SuperSideScrollerGameMode.cpp
    - SuperSideScrollerGameMode.h

Live Share

PlayerProjectile.cpp\* PlayerProjectile.h

SuperSideScroller → APlayerProjectile → APlayerProjectile()

```
1 // Fill out your copyright notice in the Description page of Project Settings.  
2  
3  
4 #include "PlayerProjectile.h"  
5 #include "GameFramework/ProjectileMovementComponent.h"  
6 #include "Components/SphereComponent.h"  
7 #include "Components/StaticMeshComponent.h"  
8  
9 // Sets default values  
10 APlayerProjectile::APlayerProjectile()  
11 {  
12     // Set this actor to call Tick() every frame. You can turn this off to improve performance if you don't need it.  
13     //PrimaryActorTick.bCanEverTick = true;  
14 }  
15  
16  
17 void APlayerProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp,  
18 FVector NormalImpulse, const FHitResult& Hit)  
19 {  
20 }  
21  
22  
23
```

Delete

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 탐색기 '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
    - SuperSideScroller.uproject
  - Visualizers
  - UE4.natvis

100 % 문제가 검색되지 않음 줄: 13 문자: 4 열: 7 혼합 CRLF

빌드에 성공했습니다. 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

PlayerProjectile.cpp\* PlayerProjectile.h

SuperSideScroller → APlayerProjectile → APlayerProjectile()

```
1 // Fill out your copyright notice in the Description page of Project Settings.  
2  
3  
4 #include "PlayerProjectile.h"  
5 #include "GameFramework/ProjectileMovementComponent.h"  
6 #include "Components/SphereComponent.h"  
7 #include "Components/StaticMeshComponent.h"  
8  
9 // Sets default values  
10 APlayerProjectile::APlayerProjectile()  
11 {  
12     CollisionComp = CreateDefaultSubobject<USphereComponent>(TEXT("SphereComp"));  
13     CollisionComp->InitSphereRadius(15.0f);  
14     CollisionComp->BodyInstance.SetCollisionProfileName("BlockAll");  
15     CollisionComp->OnComponentHit.AddDynamic(this, &APlayerProjectile::OnHit);  
16  
17     // Set as root component  
18     RootComponent = CollisionComp;  
19  
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 void APlayerProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp,  
34 FVector NormalImpulse, const FHitResult& Hit)
```

Ctrl+S

100 % 문제가 검색되지 않음 줄: 30 문자: 25 열: 28 템 CRLF

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

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

- Engine
- 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
    - SuperSideScroller.uproject
  - Visualizers
  - UE4.natvis

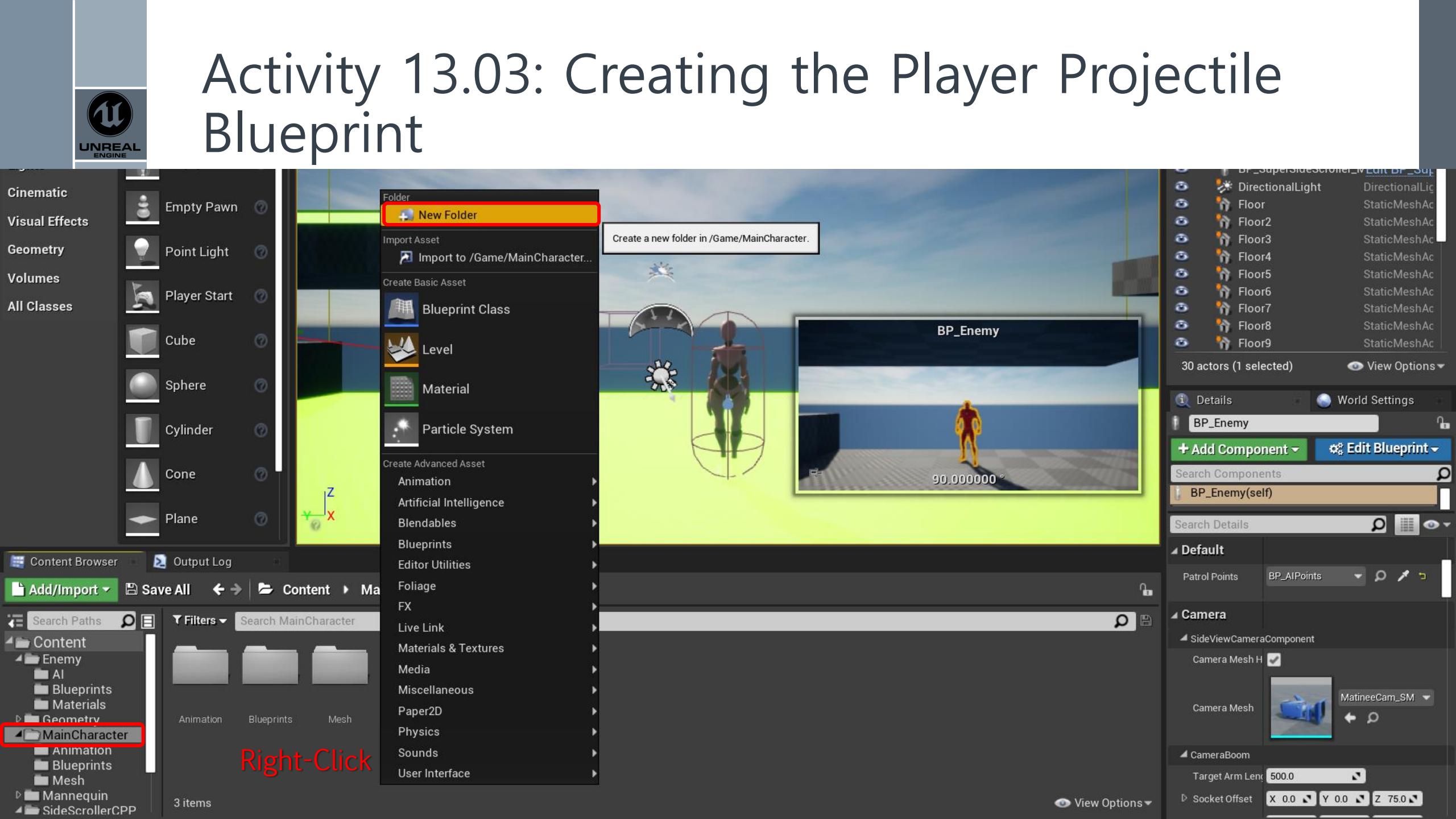
솔루션 탐색기 Git 변경 내용

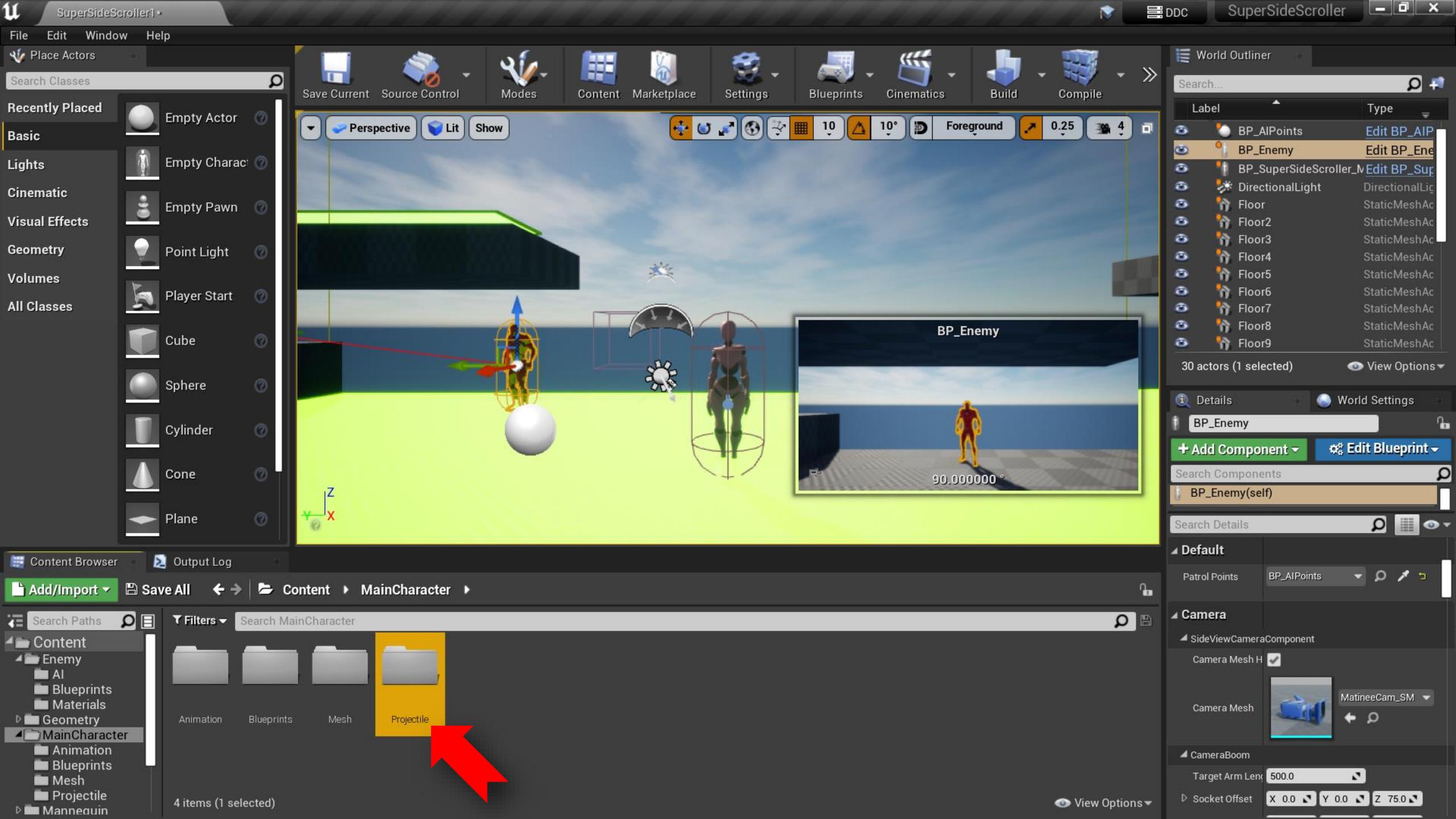
빌드에 성공했습니다. ↑ 소스 제어에 추가 ↗

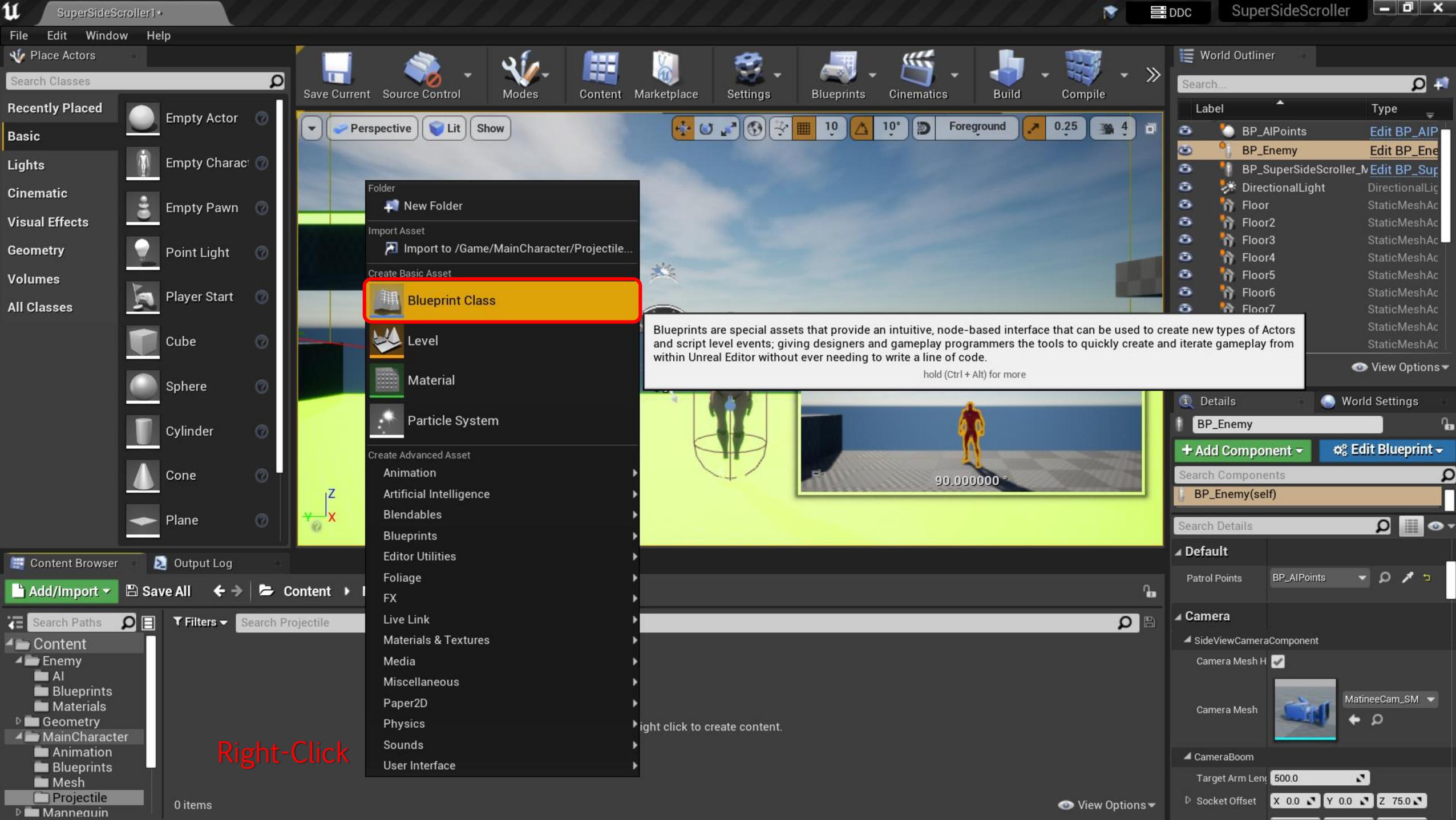
The screenshot shows the Microsoft Visual Studio interface with the following details:

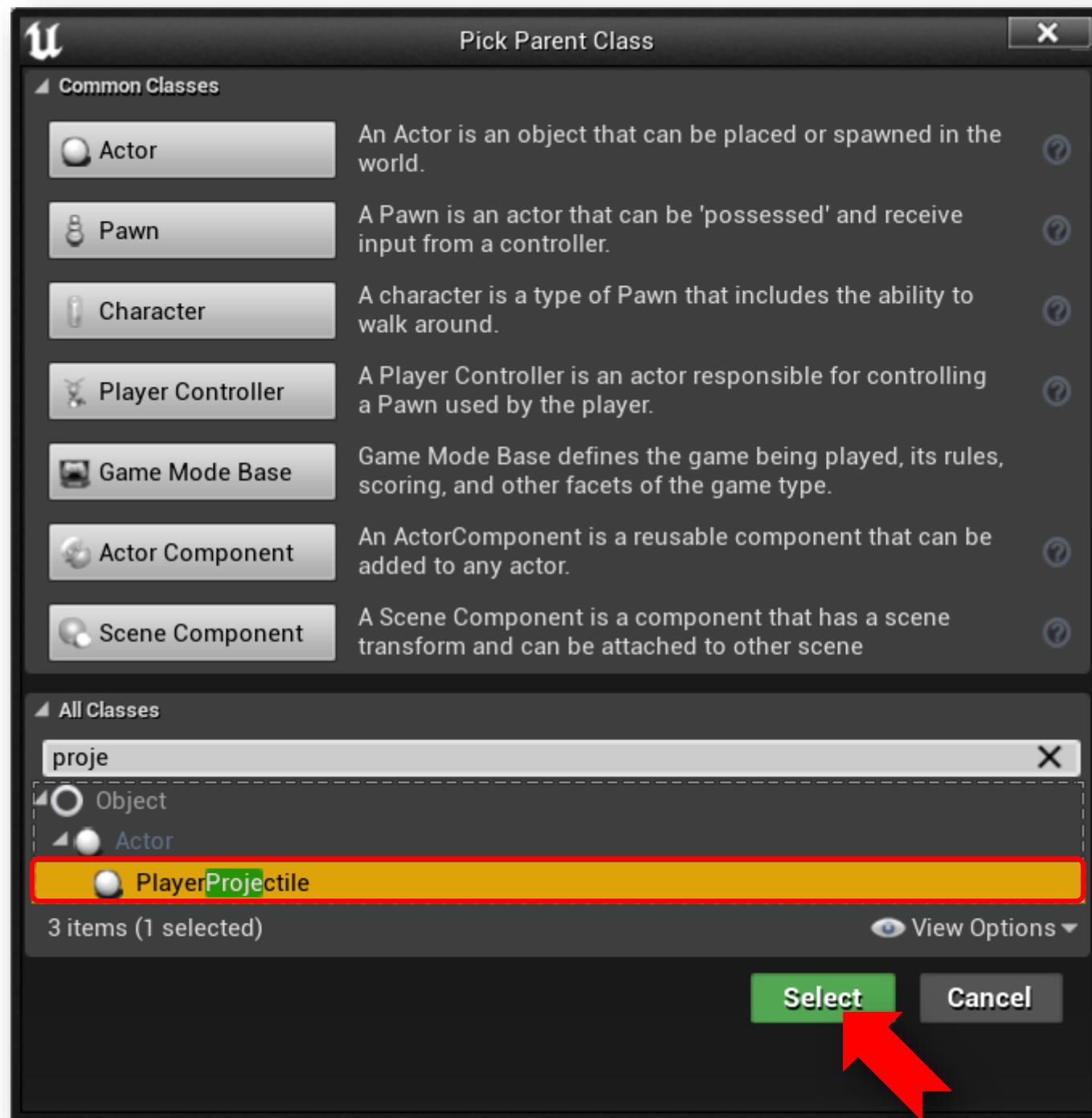
- Top Bar:** 파일(F), 편집(E), 보기(V), Git(G), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(T), 확장(X), 창(W), 도움말(H), 검색 (Ctrl+Q), Live Share.
- Left Sidebar:** 최근 파일 목록 (최근 5개).
- Code Editor:** 파일명: PlayerProjectile.cpp, 내용: C++ 코드 (PlayerProjectile.h 포함). 텍스트 표시: 100%, 문제 검색: 문제 없음.
- Build Menu (Open):** 솔루션 빌드(B) (선택됨), 솔루션 다시 빌드, 솔루션 정리(C), 솔루션의 전체 프로그램 데이터베이스 파일 빌드, 솔루션에서 코드 분석 실행(Y), SuperSideScroller 빌드(U), SuperSideScroller 다시 빌드(E), SuperSideScroller 정리(N), SuperSideScroller에서 코드 분석 실행(A), 프로젝트만(), 일괄 빌드(T)..., 구성 관리자(O)..., 컴파일(M), 파일에서 코드 분석 실행(F).
- Right Sidebar:** 솔루션 탐색기 (SuperSideScroller 프로젝트), 솔루션 탐색기 검색(Ctrl+;), 파일 목록: Engine, Games, SuperSideScroller (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), Visualizers, UE4.natvis.
- Bottom Status Bar:** 줄: 30, 문자: 25, 열: 28, 탭, CRLF, 슬루션 탐색기 Git 변경 내용.

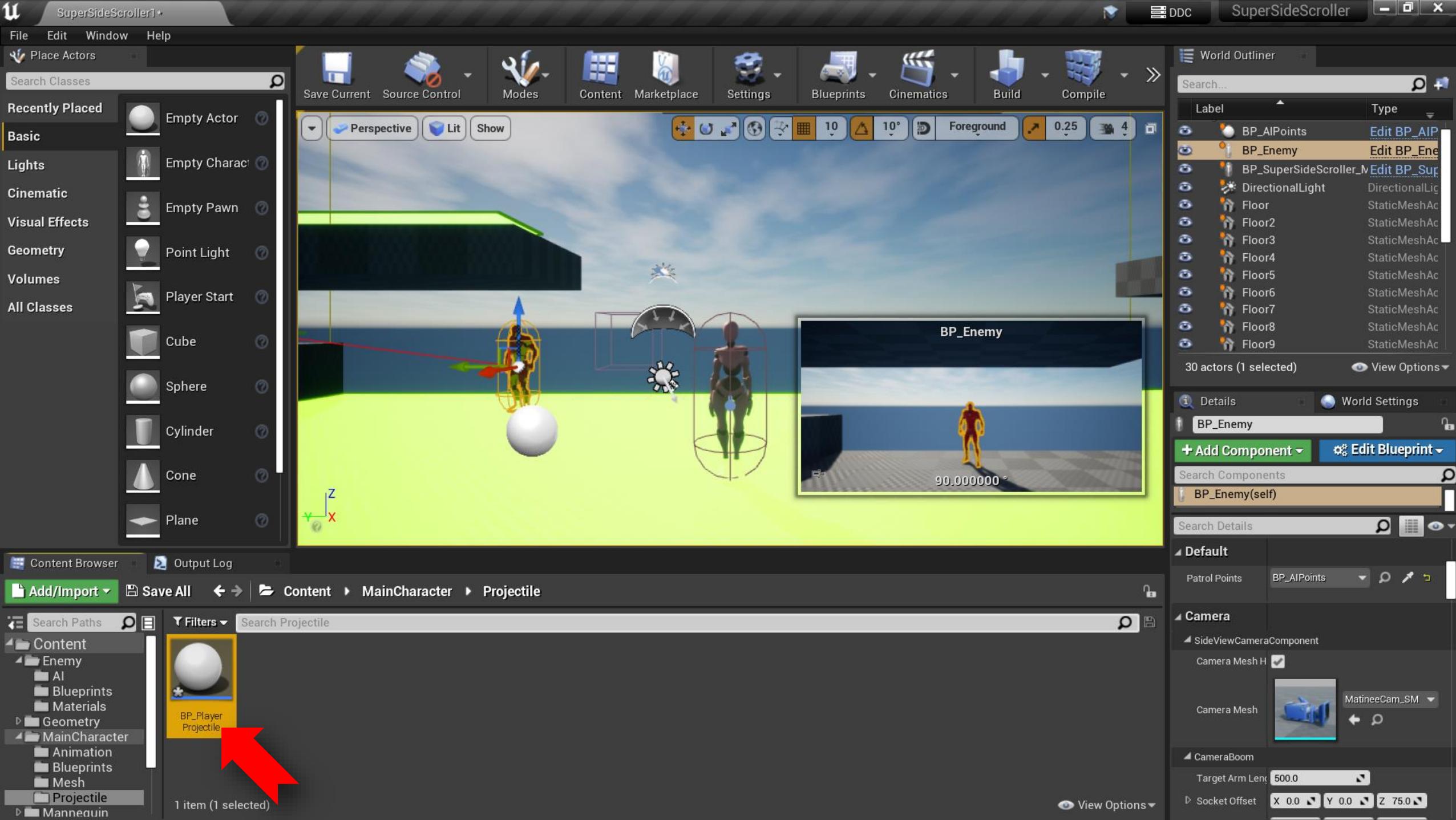
# Activity 13.03: Creating the Player Projectile Blueprint

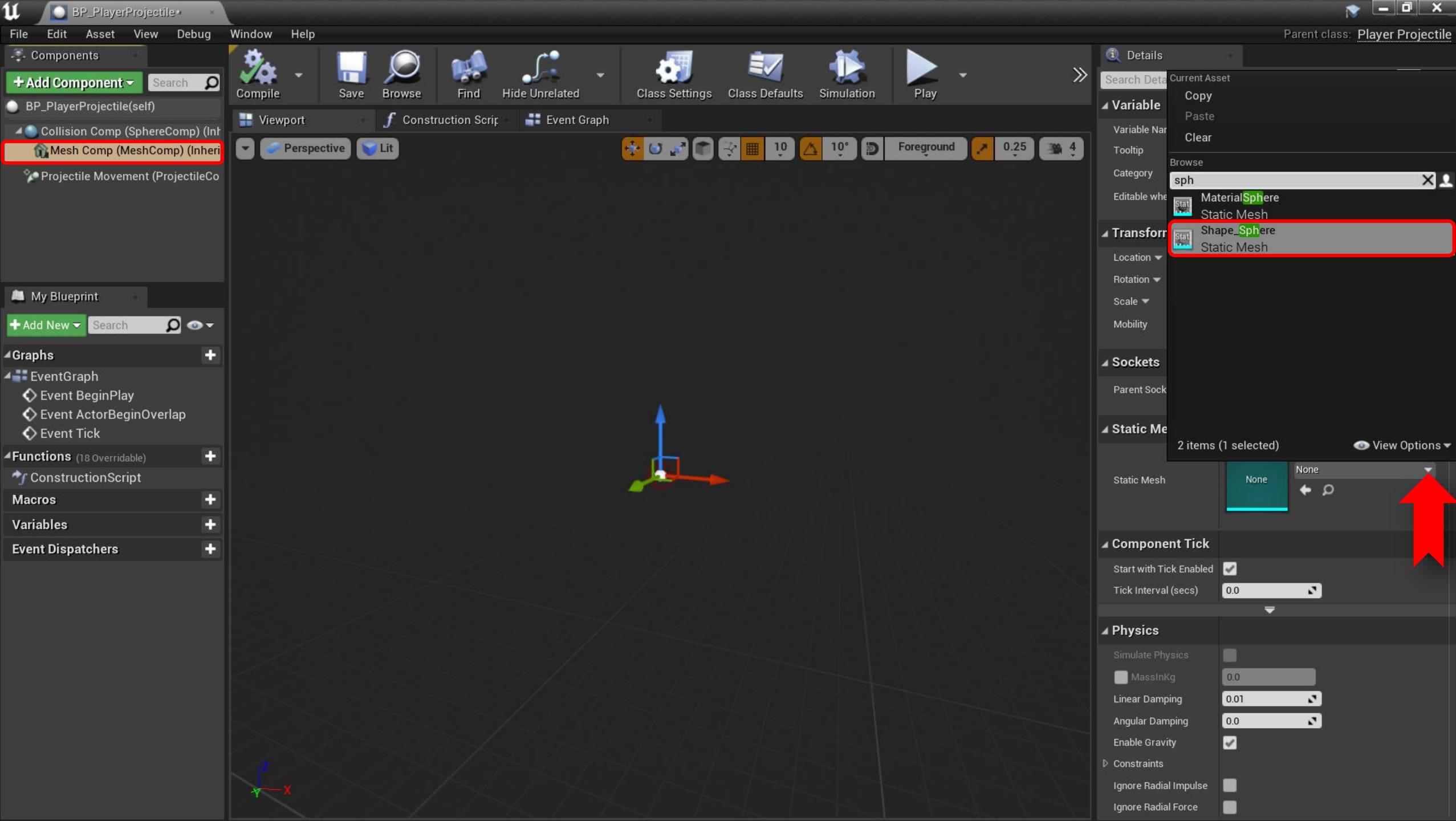


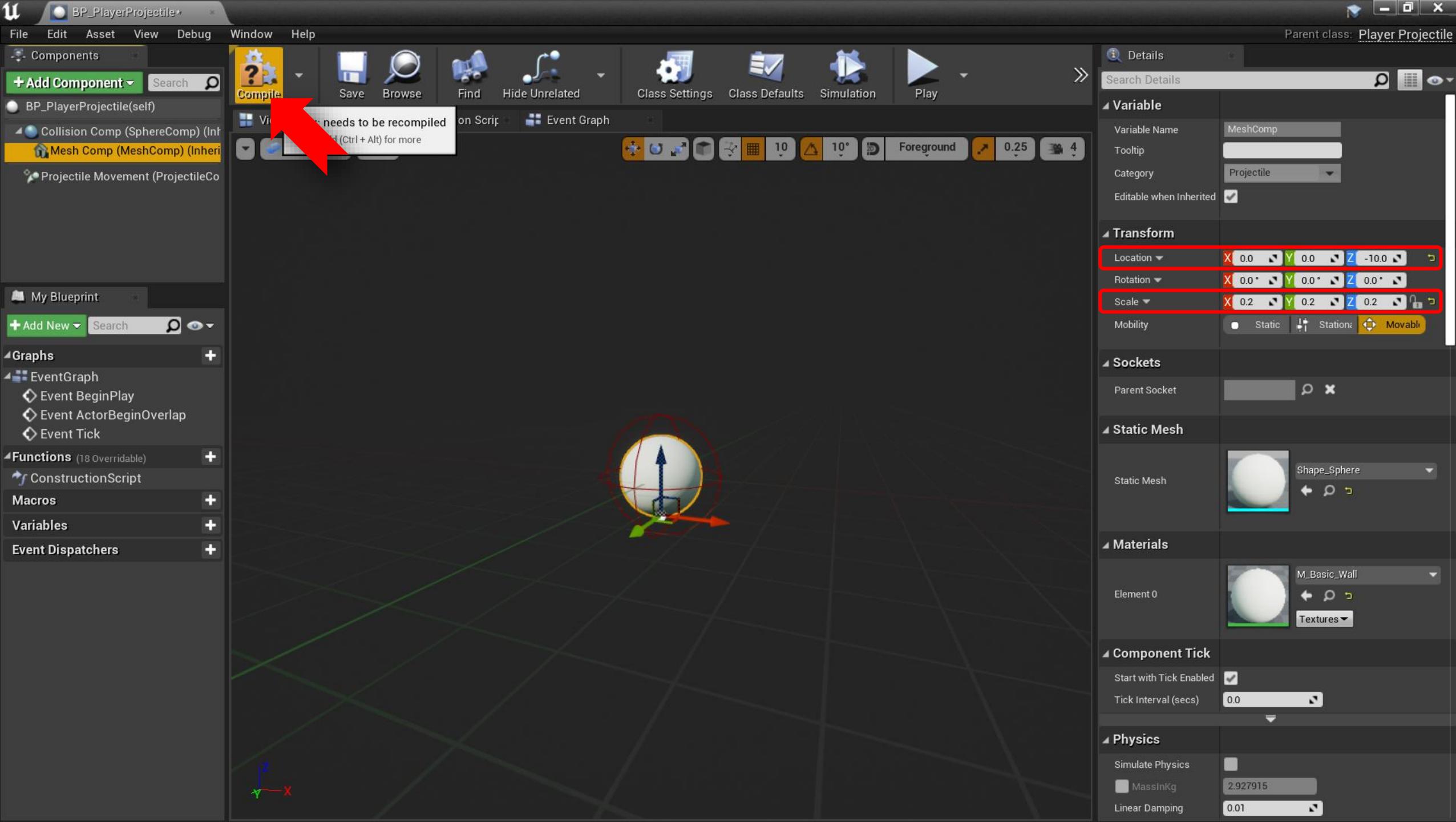












파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(I) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+Q) Supe...ller

Develop Win64 로컬 Windows 디버거 Live Share

PlayerProjectile.cpp\* PlayerProjectile.h

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

```
9 // Sets default values
10 APlayerProjectile::APlayerProjectile()
11 {
12     CollisionComp = CreateDefaultSubobject<USphereComponent>(TEXT("SphereComp"));
13     CollisionComp->InitSphereRadius(15.0f);
14     CollisionComp->BodyInstance.SetCollisionProfileName("BlockAll");
15     CollisionComp->OnComponentHit.AddDynamic(this, &APlayerProjectile::OnHit);
16
17     // Set as root component
18     RootComponent = CollisionComp;
19
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 void APlayerProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp,
34                             FVector NormalImpulse, const FHitResult& Hit)
35 {
36     UE_LOG(LogTemp, Warning, TEXT("HIT"));
37 }
```

100 % 문제가 검색되지 않음 출: 37 문자: 40 열: 43 템 CRLF

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 탐색기 '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
    - SuperSideScroller.uproject
  - Visualizers
  - UE4.natvis

소스 제어에 추가

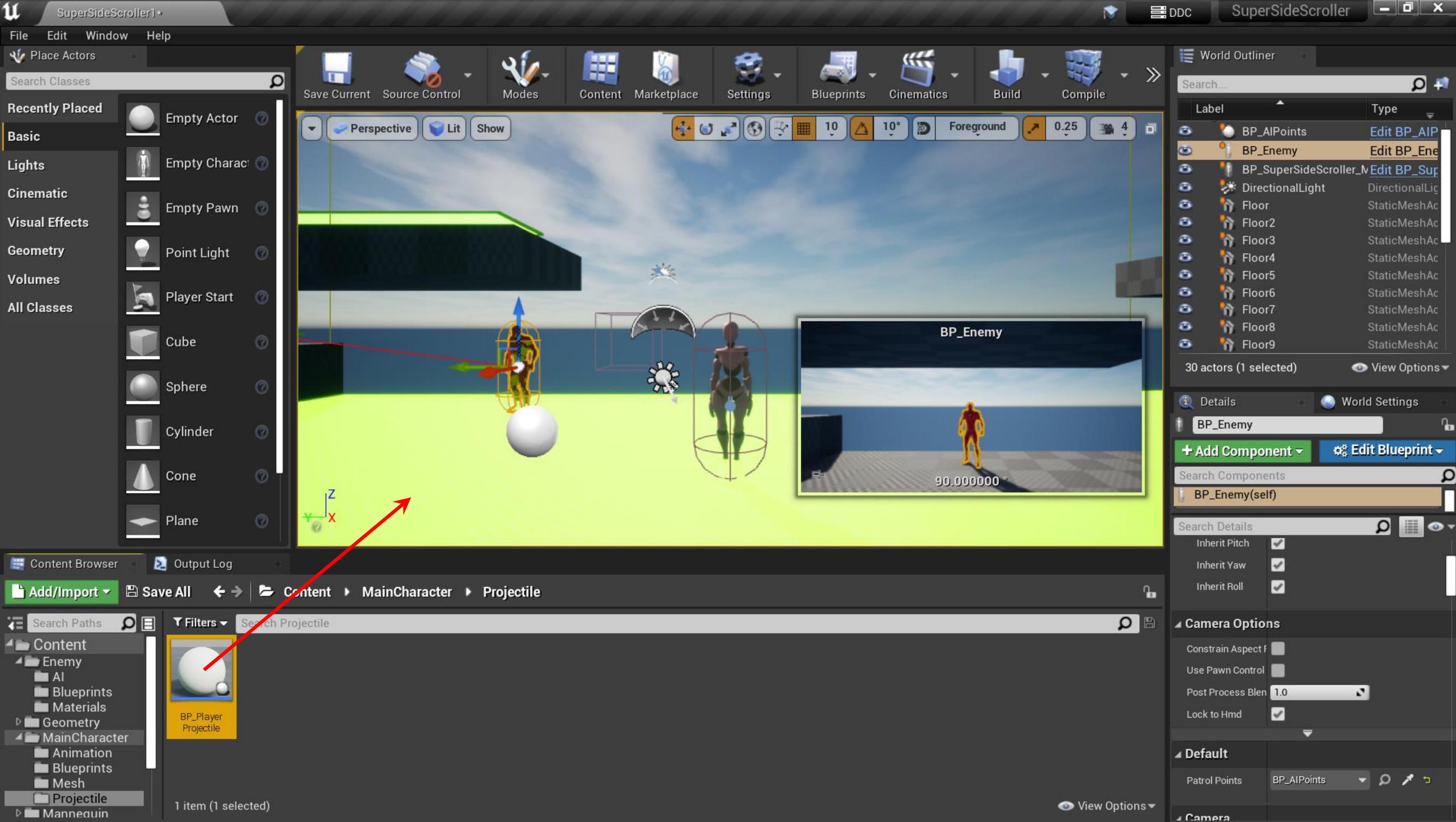
The screenshot shows the Microsoft Visual Studio interface with the following details:

- Top Bar:** 파일(F), 편집(E), 보기(V), Git(G), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(T), 확장(X), 창(W), 도움말(H), 검색 (Ctrl+Q), Super...ller.
- Left Sidebar:** 최근 파일 목록 (파일 목록, 최근 파일, 최근 작업).
- Central Area:** 코드 편집기 창 (PlayerProjectile.cpp)과 편집기 상태 표시줄 (100%, 문제가 검색되지 않음).
- Build Menu (Visible):** 솔루션 빌드(B) (선택됨), 솔루션 다시 빌드, 솔루션 정리(C), 솔루션의 전체 프로그램 데이터베이스 파일 빌드, 솔루션에서 코드 분석 실행(Y), SuperSideScroller 빌드(U), SuperSideScroller 다시 빌드(E), SuperSideScroller 정리(N), SuperSideScroller에서 코드 분석 실행(A), 프로젝트만(J), 일괄 빌드(T)..., 구성 관리자(O)..., 컴파일(M), 파일에서 코드 분석 실행(F).
- Right Sidebar:** 솔루션 탐색기 (SuperSideScroller 프로젝트 내 파일 목록), Live Share.
- Bottom Status Bar:** 줄: 37, 문자: 40, 열: 43, 템, CRLF, 솔루션 탐색기, Git 변경 내용.

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

```
9     // Sets default values
10    APlayerProjectile::APlayerProjectile()
11    {
12        CollisionComp = CreateDefaultSubobject(TEXT("CollisionComp"));
13        CollisionComp->InitSphereRadius(150.0f);
14        CollisionComp->BodyInstance.SetCollisionProfileName("OverlapAll");
15        CollisionComp->OnComponentHit.AddDynamic(this, &APlayerProjectile::OnHit);
16
17        // Set as root component
18        RootComponent = CollisionComp;
19
20        // Use a ProjectileMovementComponent
21        ProjectileMovement = CreateDefaultSubobject(TEXT("ProjectileMovement"));
22        ProjectileMovement->UpdatedComponent = CollisionComp;
23        ProjectileMovement->ProjectileEmissions = 100.0f;
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    void APlayerProjectile::OnHit(UPrimitiveComponent* HitComp, AActor* OtherActor, UPrimitiveComponent* OtherComp,
35    FVector NormalImpulse, const FHitResult& Hit)
36    {
37        UE_LOG(LogTemp, Warning, TEXT("HIT"));
38    }

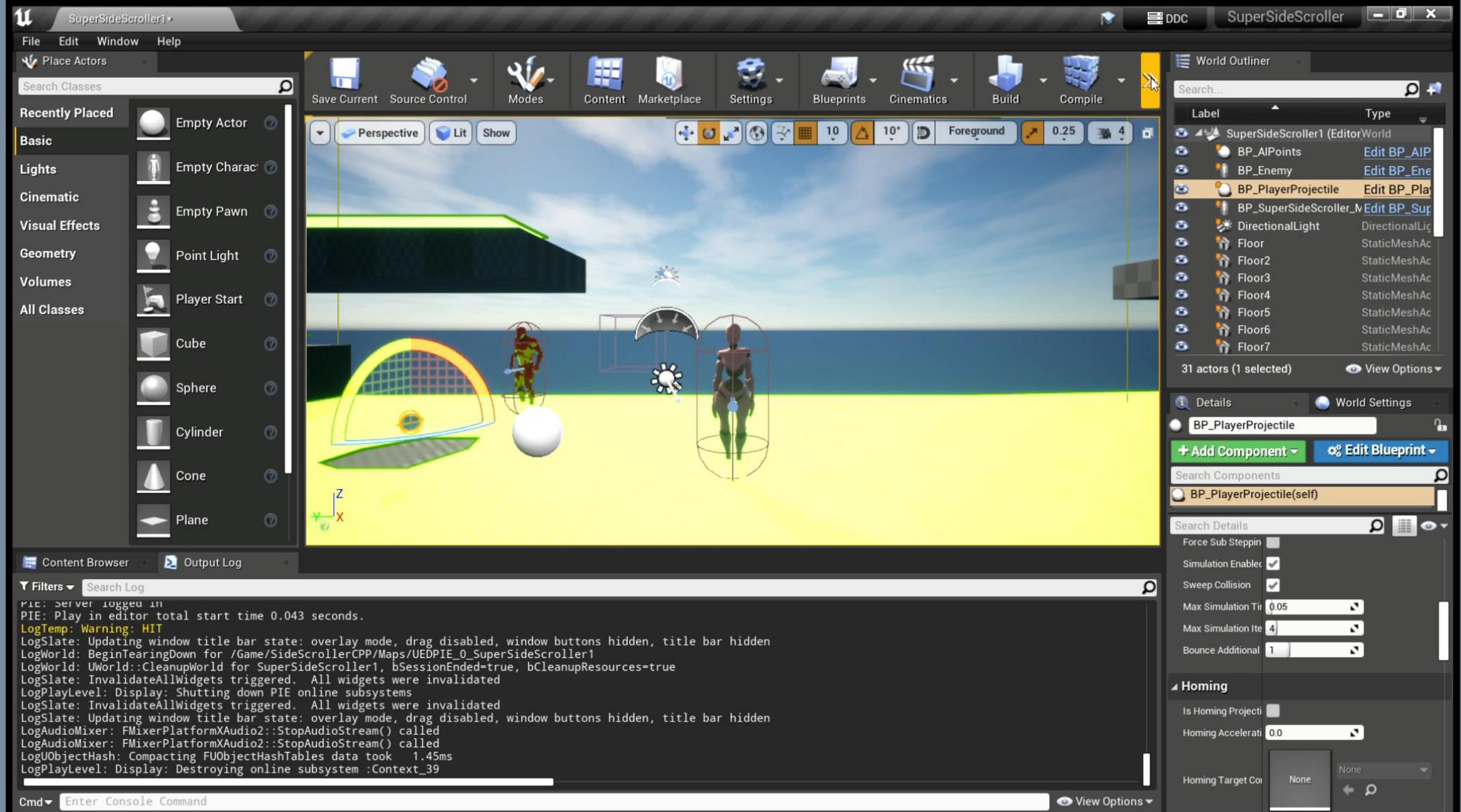
```







```
PIE: Server logged in
PIE: Play in editor total start time 0.043 seconds.
LogTemp: Warning: HIT
LogSlate: Updating window title bar state: overlay mode, drag disabled, window buttons hidden, title bar hidden
LogWorld: BeginTearingDown for /Game/SideScrollerCPP/Maps/UEDPIE_0_SuperSideScroller1
LogWorld: UWorld::CleanupWorld for SuperSideScroller1, bSessionEnded=true, bCleanupResources=true
LogSlate: InvalidateAllWidgets triggered. All widgets were invalidated
LogPlayLevel: Display: Shutting down PIE online subsystems
LogSlate: InvalidateAllWidgets triggered. All widgets were invalidated
LogSlate: Updating window title bar state: overlay mode, drag disabled, window buttons hidden, title bar hidden
LogAudioMixer: FMixerPlatformXAudio2::StopAudioStream() called
LogAudioMixer: FMixerPlatformXAudio2::StopAudioStream() called
LogUObjectHash: Compacting FUObjectHashTables data took 1.45ms
LogPlayLevel: Display: Destroying online subsystem :Context_39
```





## 연습 과제

- › Activity 13.03까지 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