

Activity 3.01

Extending the C++ Character Class with Blueprint in the Animation Project

4th Week, 2021



UNREAL
ENGINE



Activity 3.01: Extending the C++ Character Class with Blueprint in the Animation Project

- › 1. Open the project from *Activity 2.01, Mixamo Character Animation*.
- › 2. Create a **Character** class in C++ that will initialize the character variables, including the camera associated with the player.
- › 3. Map the Jump input to the spacebar key in the project settings.
- › 4. Extend the created C++ class with a blueprint to add the associated assets and jump functionality.



Select or Create New Project

Recent Projects



Exercise3_01



Activity_Anim



CharAnim



Exercise2_01



Exercise1_01

[More](#)

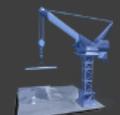
New Project Categories

**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.

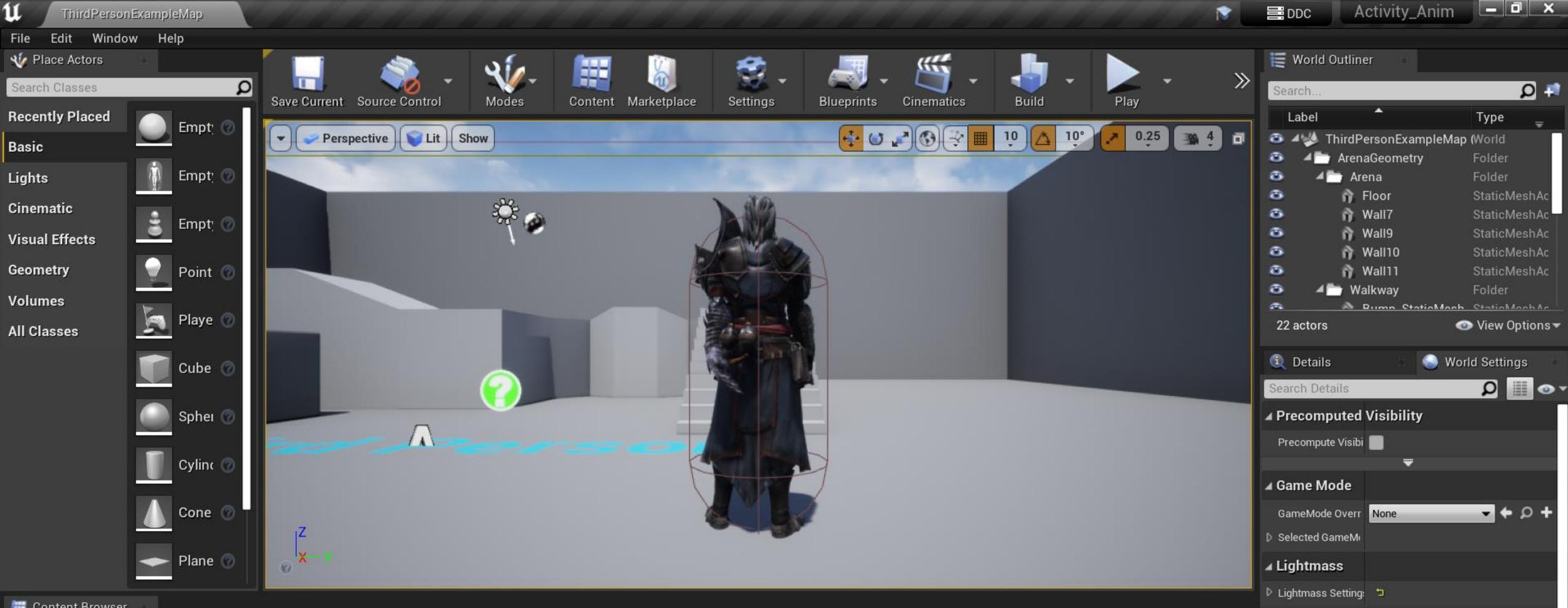
**Architecture, Engineering, and Construction**

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

**Automotive, Product Design, and Manufacturing**

Find templates for multi-user design reviews, photobooth studio environments, and product configurators.

[Open Project](#)[Cancel](#)



Add/Import Save All Content > ThirdPersonBP > Blueprints

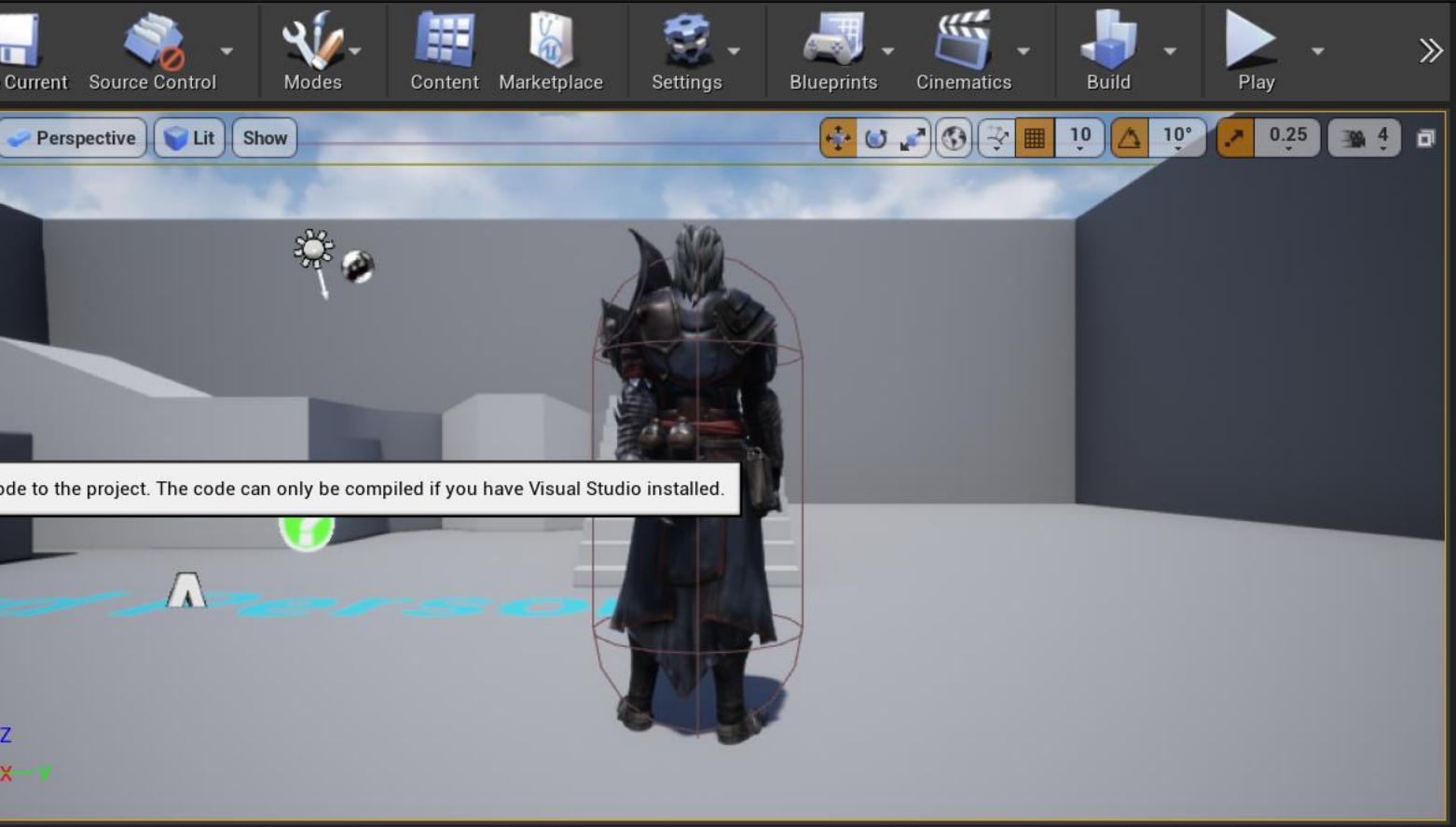
Filters Search Blueprints

Content

- Ganfault
- Geometry
- Mannequin
- ThirdPerson
- ThirdPersonBP
- Blueprints
- Maps

4 items

View Options

File Edit Window Help**Load and Save**
New Level... Ctrl+N
Open Level... Ctrl+O
Save Current Ctrl+S
Save Current As... Ctrl+Alt+S
Save All Levels
Open Asset... Ctrl+P
Save All Ctrl+Shift+S
Choose Files to Save...
Connect To Source Control...**Project**
New Project...
Open Project...
New C++ Class...
Package Project
Generate Visual Studio Project
Cook Content for Windows**Python**
Execute Python Script...
Recent Python Scripts**DataValidation**
Validate Data...**Actors**
Import Into Level...
Export All...
Export Selected...
Favorite Levels
Recent Levels
Recent Projects**Exit**
ThirdPerson
ThirdPersonBP
Blueprints
Maps

World Outliner

Label	Type
ThirdPersonExampleMap (World)	Folder
ArenaGeometry	Folder
Arena	Folder
Floor	StaticMeshActor
Wall7	StaticMeshActor
Wall9	StaticMeshActor
Wall10	StaticMeshActor
Wall11	StaticMeshActor
Walkway	Folder
Runn StaticMesh	StaticMeshActor

22 actors View Options

Details World Settings

Precomputed Visibility
Precompute Visibi

Game Mode
GameMode Overr None

Lightmass
Lightmass Setting:

World
Enable World Comp
Use Client Side Le
Kill Z

Physics
Override World Gr
Global Gravity Z

Broadphase
Override Default B



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

Character

Selected Class Source

Character.h

A large red arrow points to the "Next >" button at the bottom left of the dialog.

Next >

Create Class

Cancel



Name Your New Character

Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.

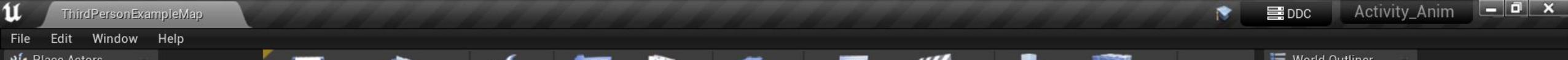
When you click the "Create" button below, a header (.h) file and a source (.cpp) file will be made using this name.

Name	<input type="text" value="MyThirdPersonChar"/>	Activity_Anim (Runtime) ▾	Public	Private
Path	<input type="text" value="C:/Users/sunje/Desktop/Unreal/Activity2_01/Source/Activity_Anim/"/> Choose Folder			
Header File	C:/Users/sunje/Desktop/Unreal/Activity2_01/Source/Activity_Anim/MyThirdPersonChar.h			
Source File	C:/Users/sunje/Desktop/Unreal/Activity2_01/Source/Activity_Anim/MyThirdPersonChar.cpp			

< Back

Create Class

Cancel



File Edit Window Help

Place Actors

Search Classes



Save Current Source Control

Modes



Content Marketplace



Settings



Blueprints Cinematics



Build Compile



World Outliner

Search...



Recently Placed



Empty



Basic



Empty



Lights



Empty



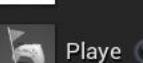
Cinematic



Empty



Visual Effects



Point



Geometry



Play



Volumes



Cube



All Classes



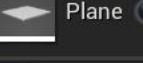
Sphere



Cylinder



Cone



Plane



Content Browser



Add/Import

Save All



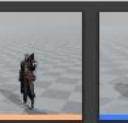
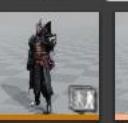
Content



Filters Search Blueprints



Search Blueprints



Anim_Ganfault BP_WalkRun ThirdPersonCharacter ThirdPersonGameMode

4 items

View Options

World Outliner

Search...



Label

Type

ThirdPersonExampleMap	(World)
ArenaGeometry	Folder
Arena	Folder
Floor	StaticMeshActor
Wall7	StaticMeshActor
Wall9	StaticMeshActor
Wall10	StaticMeshActor
Wall11	StaticMeshActor
Walkway	Folder
Runn	StaticMeshActor

22 actors

View Options

Details

World Settings

Search Details

View Options

Precomputed Visibility

Precompute Visibi

Game Mode

GameMode Overr

View Options

Selected GameM

Lightmass

Lightmass Setting

World

Enable World Comp

View Options

Use Client Side Le

View Options

Kill Z

View Options

Physics

Override World Gr

View Options

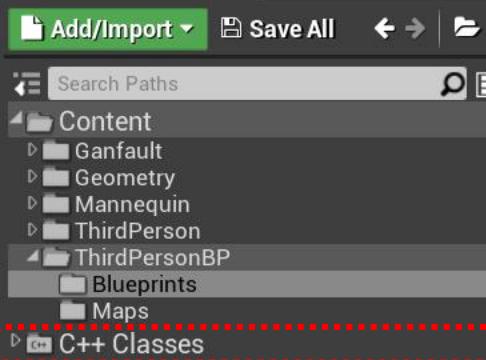
Global Gravity Z

View Options

Broadphase

Override Default B

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) Act...nim

Live Share

MyThirdPersonChar.cpp MyThirdPersonChar.h

Activity_Anim (전역 범위)

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #include "MyThirdPersonChar.h"
4
5 // Sets default values
6 AMyThirdPersonChar::AMyThirdPersonChar()
7 {
8     // Set this character 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 AMyThirdPersonChar::BeginPlay()
15 {
16     Super::BeginPlay();
17 }
18
19
20 // Called every frame
21 void AMyThirdPersonChar::Tick(float DeltaTime)
22 {
23     Super::Tick(DeltaTime);
24 }
25
26
27
28 // Called to bind functionality to input
29 void AMyThirdPersonChar::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
30 {
31     Super::SetupPlayerInputComponent(PlayerInputComponent);
32 }
33
34
35
```

100 % 문제가 검색되지 않음 출: 1 문자: 1 혼합 CRLF

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

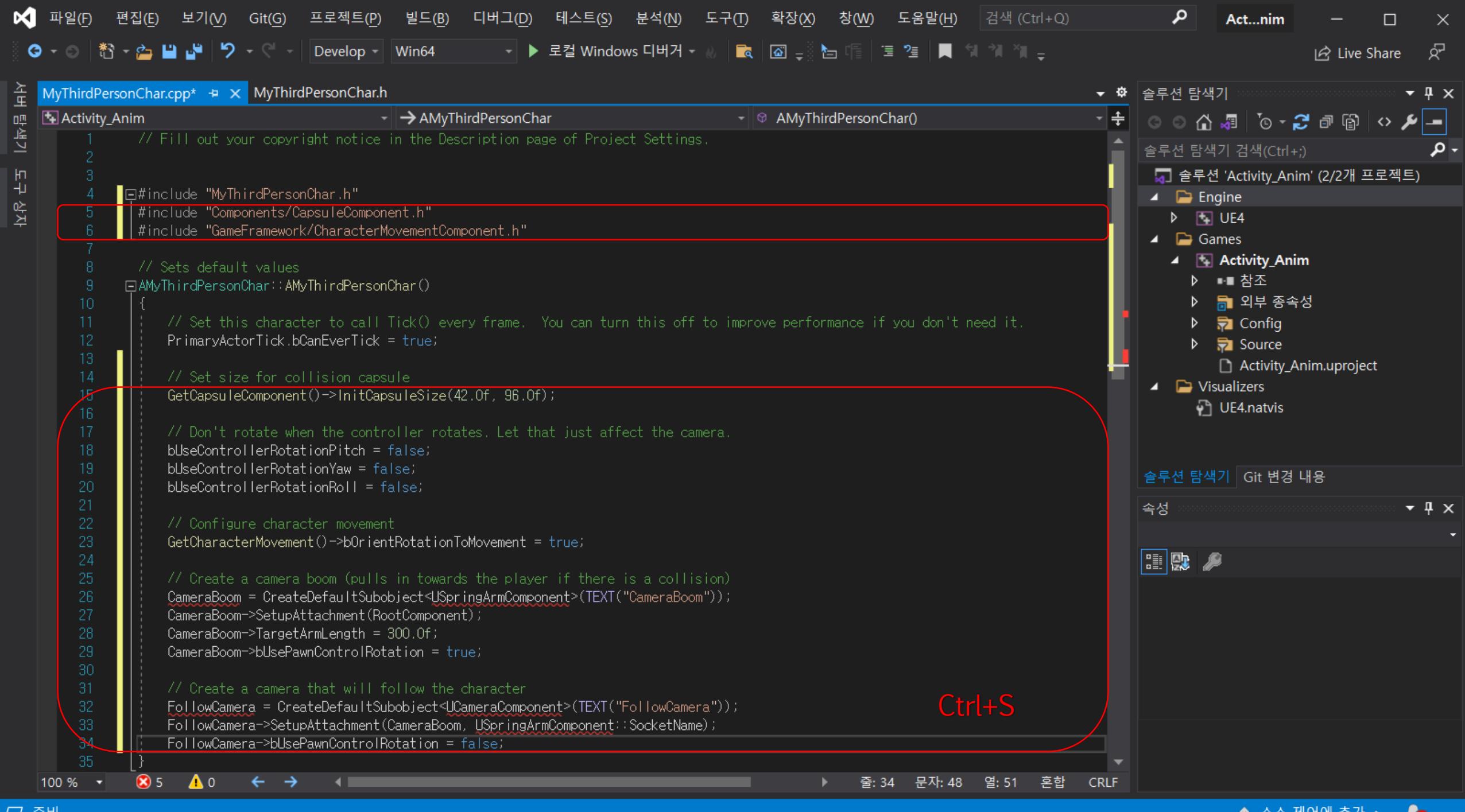
솔루션 탐색기 'Activity_Anim' (2/2개 프로젝트)

- Engine
 - UE4
- Games
 - Activity_Anim
 - 참조
 - 외부 종속성
 - Config
 - Source
- Visualizers
 - UE4.natvis

솔루션 탐색기 Git 변경 내용

속성

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) Act...nim Live Share

MyThirdPersonChar.cpp MyThirdPersonChar.h*

Activity_Anim AMyThirdPersonChar::UCameraComponent

```
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/SpringArmComponent.h"
7 #include "Camera/CameraComponent.h"
8 #include "GameFramework/Character.h"
9 #include "MyThirdPersonChar.generated.h"
10
11 UCLASS()
12 class ACTIVITY_ANIM_API AMyThirdPersonChar : public ACharacter
13 {
14     GENERATED_BODY()
15
16     // Spring arm component which will act as a placeholder for the player camera
17     UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = MyTPS_Cam, meta = (AllowPrivateAccess = "true"))
18     class USpringArmComponent* CameraBoom;
19
20     // Follow camera
21     UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = MyTPS_Cam, meta = (AllowPrivateAccess = "true"))
22     class UCameraComponent* FollowCamera;
23
24     public:
25         // Sets default values for this character's properties
26         AMyThirdPersonChar();
27
28     protected:
29         // Called when the game starts or when spawned
30         virtual void BeginPlay() override;
31
32     public:
33         // Called every frame
34         virtual void Tick(float DeltaTime) override;
35 }
```

Ctrl+S

100% 1 0 22 문자: 39 42 템 CRLF

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+Shift+F)

솔루션 탐색기 'Activity_Anim' (2/2개 프로젝트)

- Engine
- UE4
- Games
- Activity_Anim
- 참조
- 외부 종속성
- Config
- Source
- Activity_Anim.uproject

Visualizers

UE4.natvis

솔루션 탐색기 Git 변경 내용

속성

저장되었습니다.

↑ 소스 제어에 추가 ↗



ThirdPersonExampleMap DDC Activity_Anim

File Edit Window Help

Place Actors Search Classes

Recently Placed

- Basic
- Lights
- Cinematic
- Visual Effects
- Geometry
- Volumes
- All Classes

Content Browser Add/Import Save All Content > ThirdPersonBP > Blueprints

Save Current Source Control Modes Content Marketplace Settings Blueprints Cinematics Build Compile

Perspective Lit Show

Folder New Folder Import Asset Import to /Game/ThirdPersonBP/Blueprints... Create Basic Asset Blueprint Class Level Material Particle System

Blueprints are special assets that provide an intuitive, node-based interface that can be used to create new types of Actors and script level events; giving designers and gameplay programmers the tools to quickly create and iterate gameplay from within Unreal Editor without ever needing to write a line of code.

hold (Ctrl + Alt) for more

Details World Settings

Search Details

Precomputed Visibility Precompute Visibi

Game Mode GameMode Overr None Selected GameMode

Lightmass Lightmass Setting

World Enable World Comp Use Client Side Le Kill Z -1000.0

Physics Override World Gr Global Gravity Z 0.0

Broadphase Override Default B

Content Browser

Search Paths

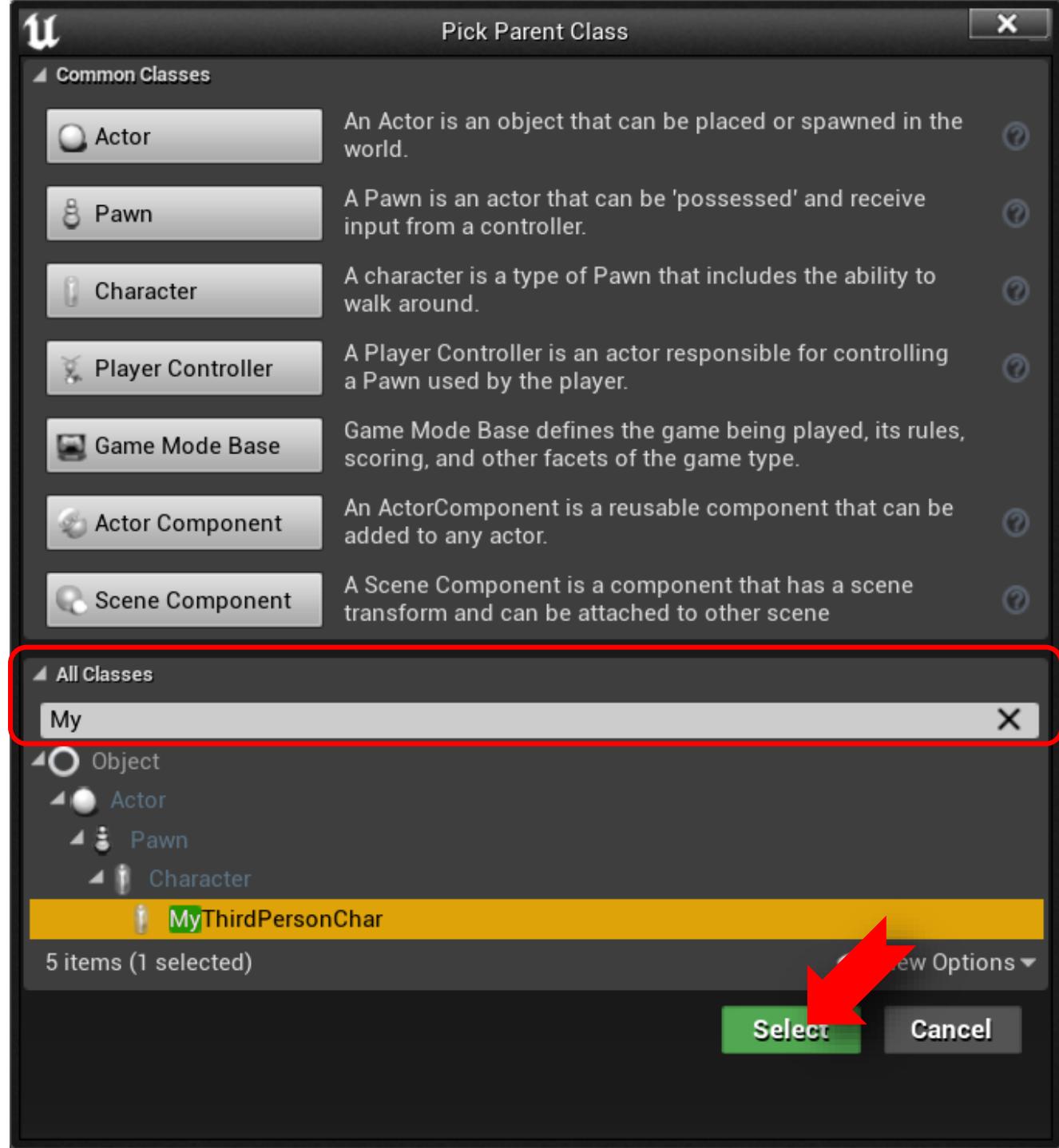
Filters Search Blueprints

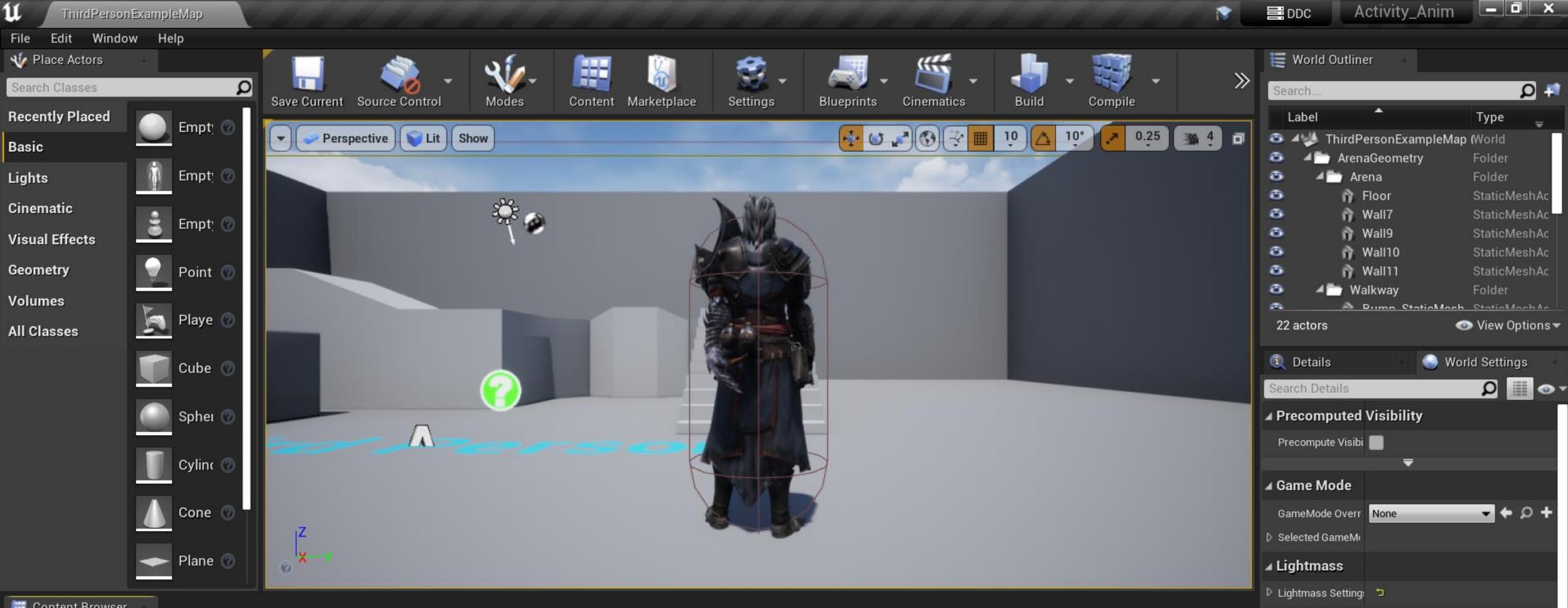
Content Ganfault Geometry Mannequin ThirdPerson ThirdPersonBP Blueprints Maps C++ Classes

Anim_Ganfault BP_WalkRun ThirdPerson Character ThirdPerson GameMode

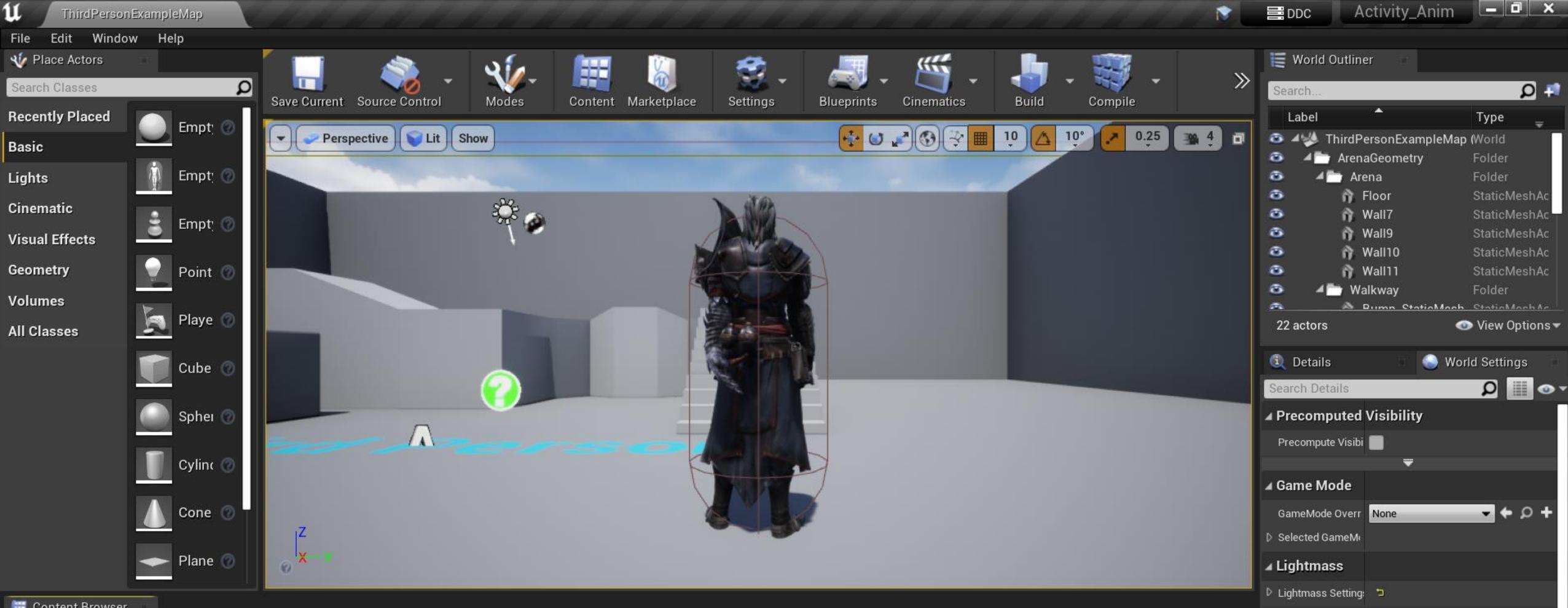
4 items View Options

The screenshot displays the Unreal Engine 4 Editor interface. The top bar includes the project name 'ThirdPersonExampleMap', session ID 'DDC', and tab 'Activity_Anim'. The toolbar contains icons for saving, source control, modes, content, marketplace, settings, blueprints, cinematics, build, and compile. The left sidebar lists recently placed assets like basic shapes and lights. The main workspace shows a 3D scene with a character model and a cube. A context menu is open over the scene, with 'Blueprint Class' selected. A tooltip provides information about Blueprints. The Content Browser shows the 'Blueprints' folder under 'ThirdPersonBP'. The bottom-left shows the file tree with 'Content' and 'C++ Classes' sections. The bottom-right is the Details panel with various world settings.

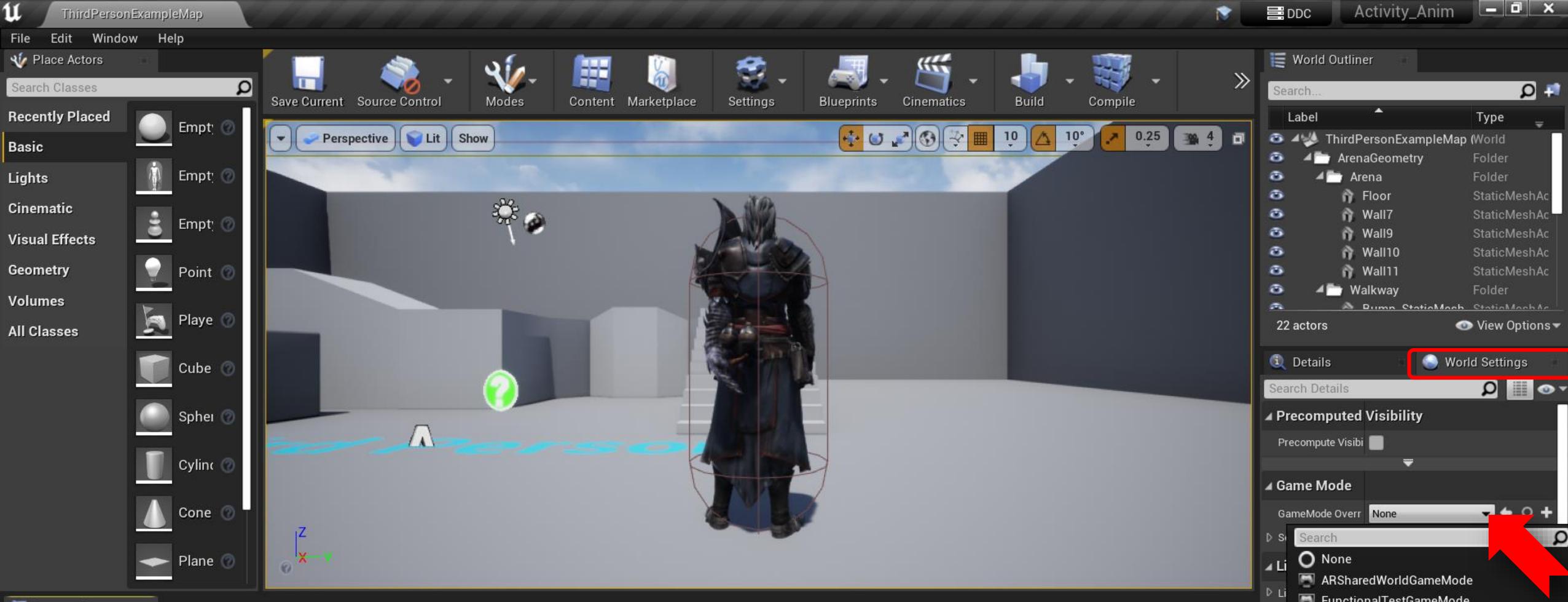




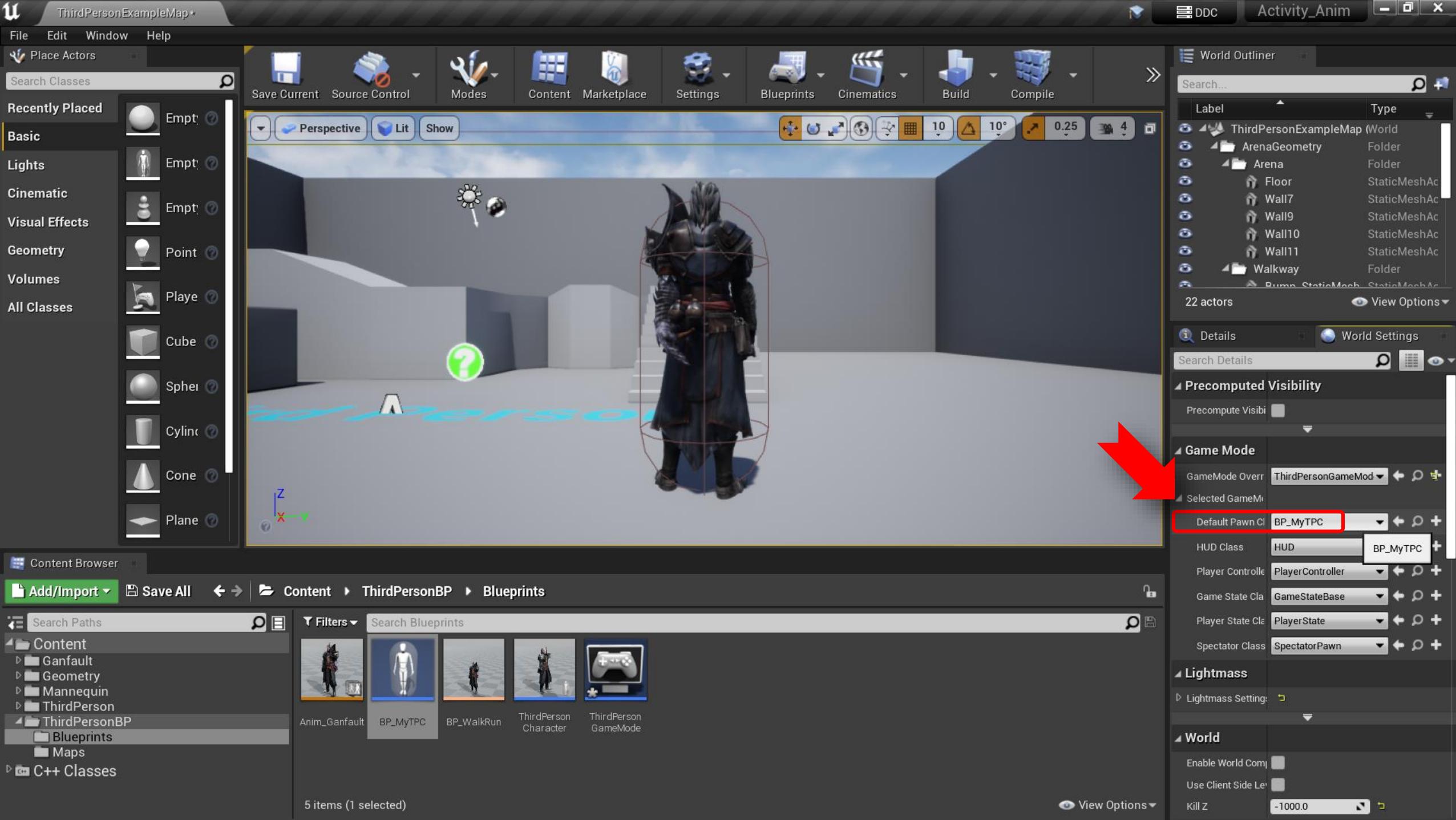
The screenshot shows the Content Browser panel of the Unreal Engine 4 Editor. The "Blueprints" section is currently selected. Inside, there are four entries: "Anim_Ganfault", "BP_WalkRun", "NewBlueprint" (which is highlighted with a red dashed box), and "ThirdPersonCharacter". The bottom status bar indicates "5 items (1 selected)". The left sidebar shows the project structure under "Content" and "ThirdPersonBP".

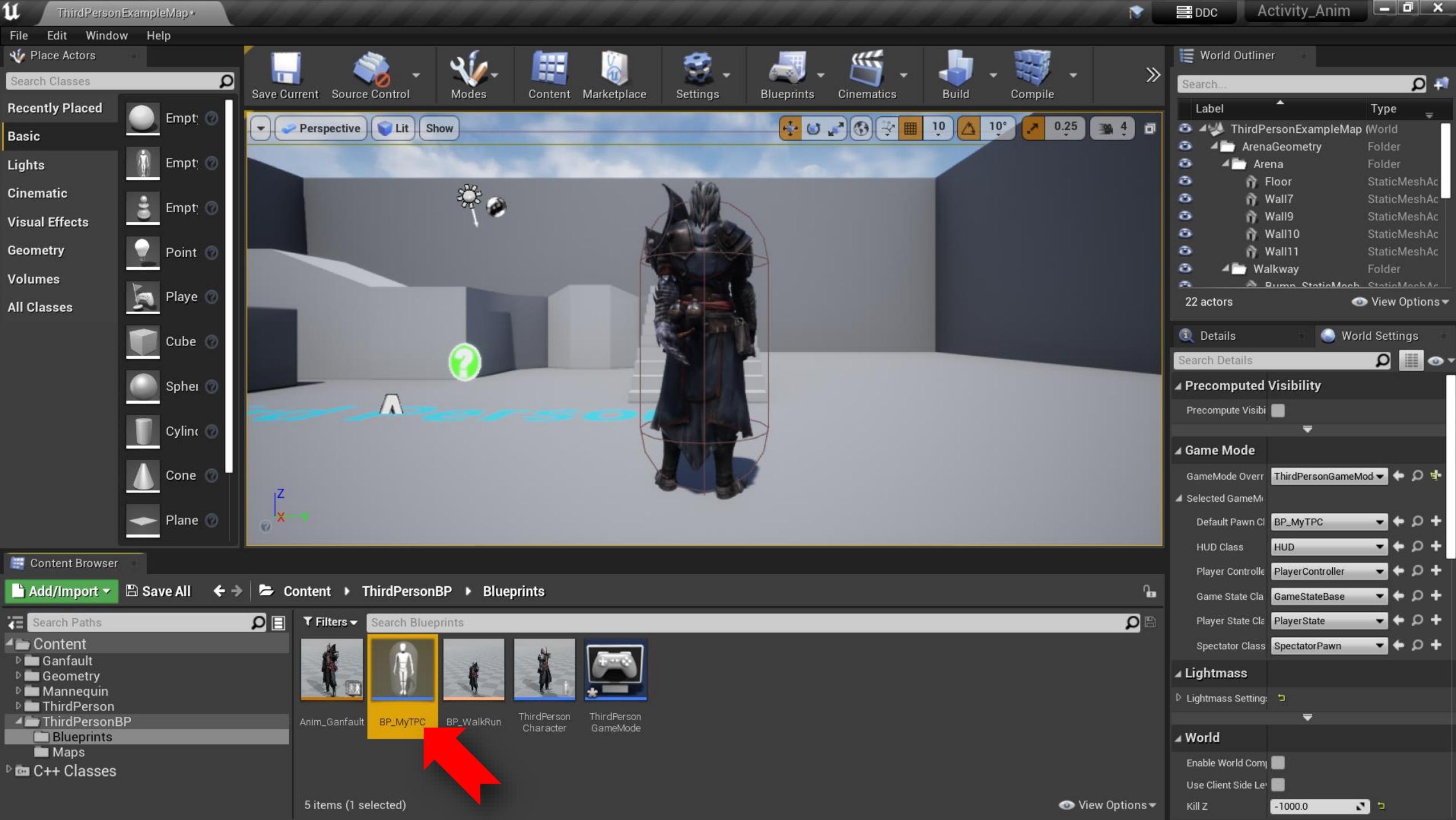


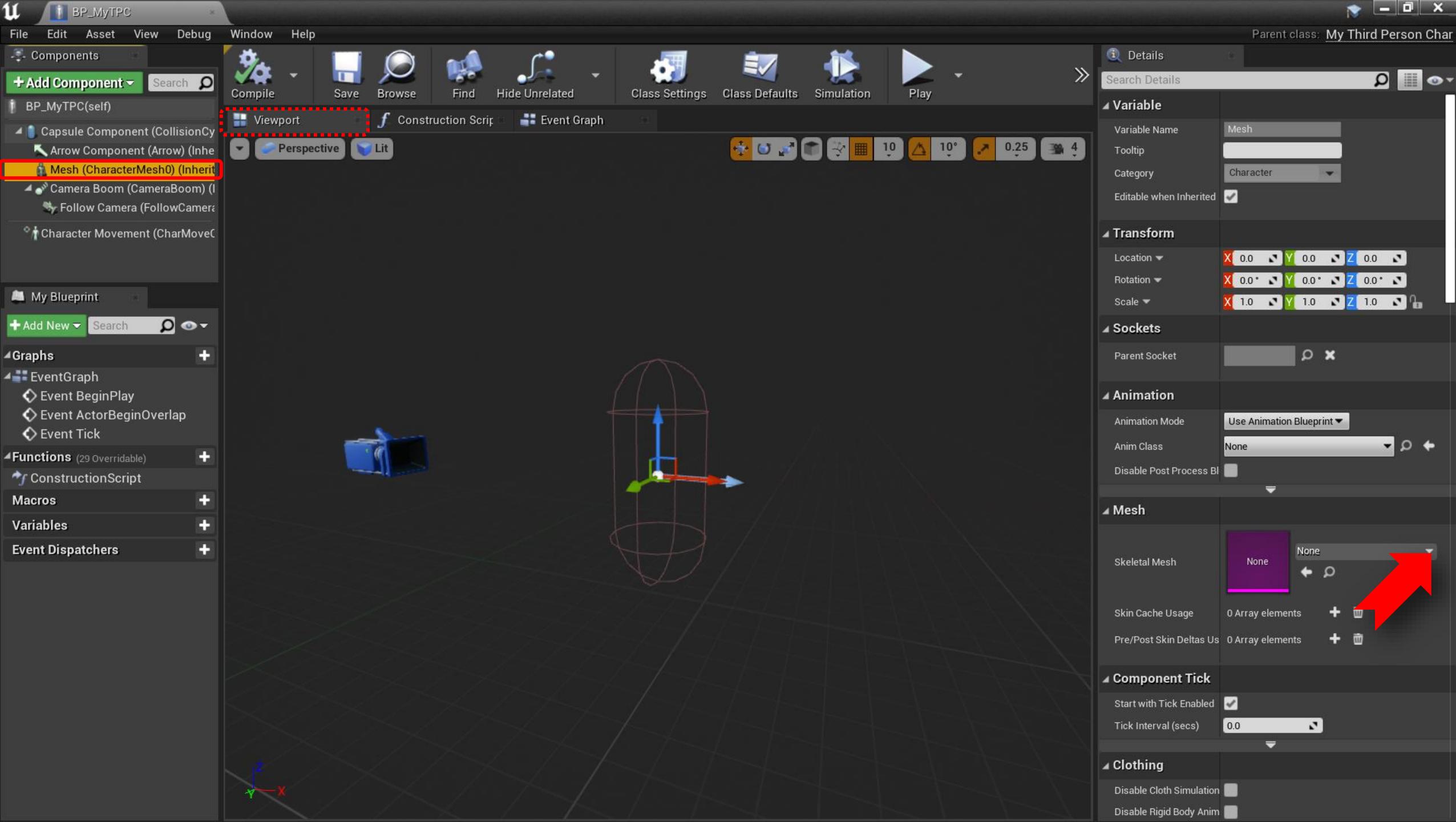
The Content Browser window is open, showing the "Blueprints" folder under "ThirdPersonBP". The list includes "Anim_Ganfault", "BP_MyTPC" (which is highlighted with a red border), "BP_WalkRun", "ThirdPersonCharacter", and "ThirdPersonGameMode". The status bar at the bottom indicates "5 items (1 selected)".

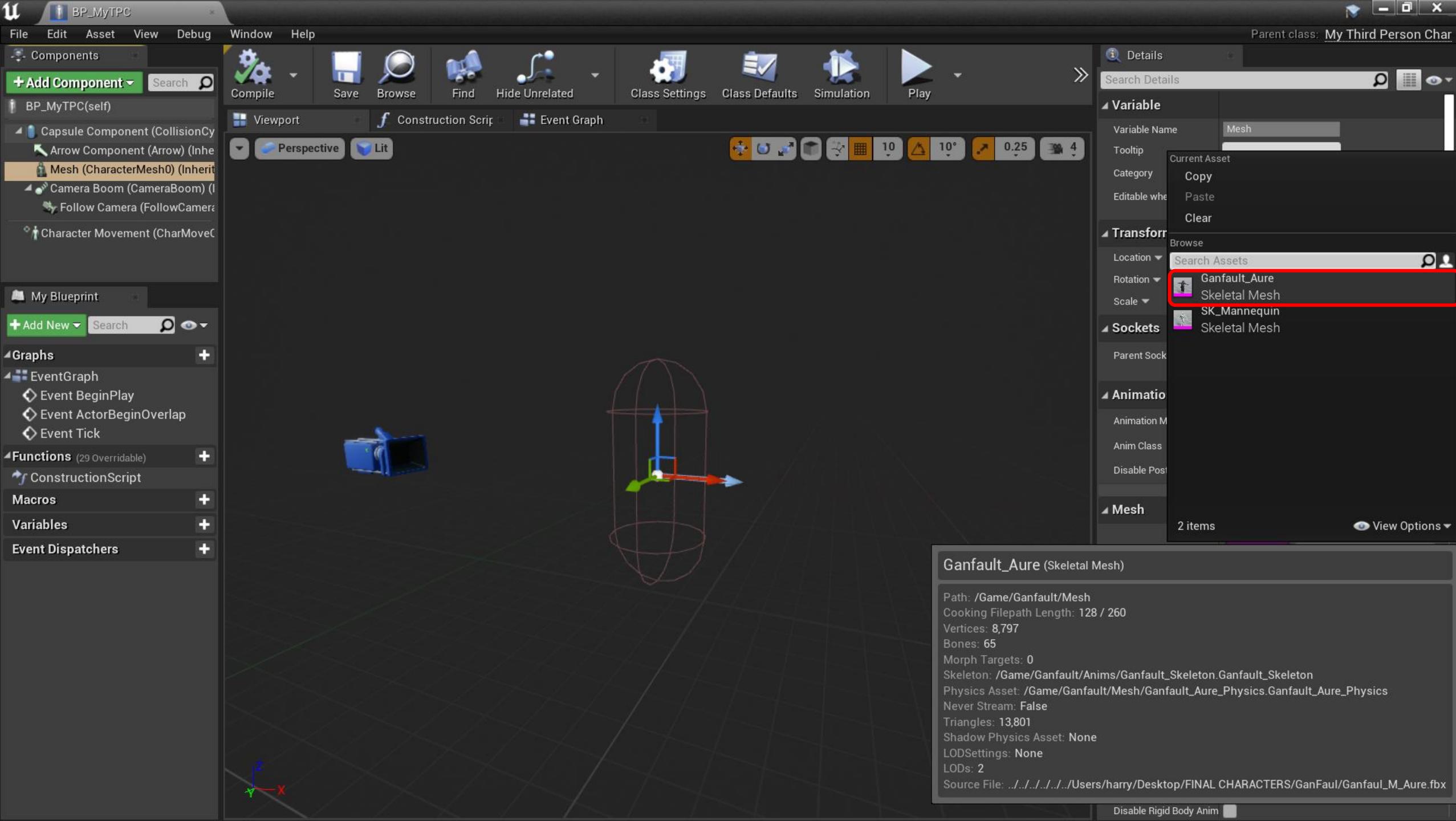


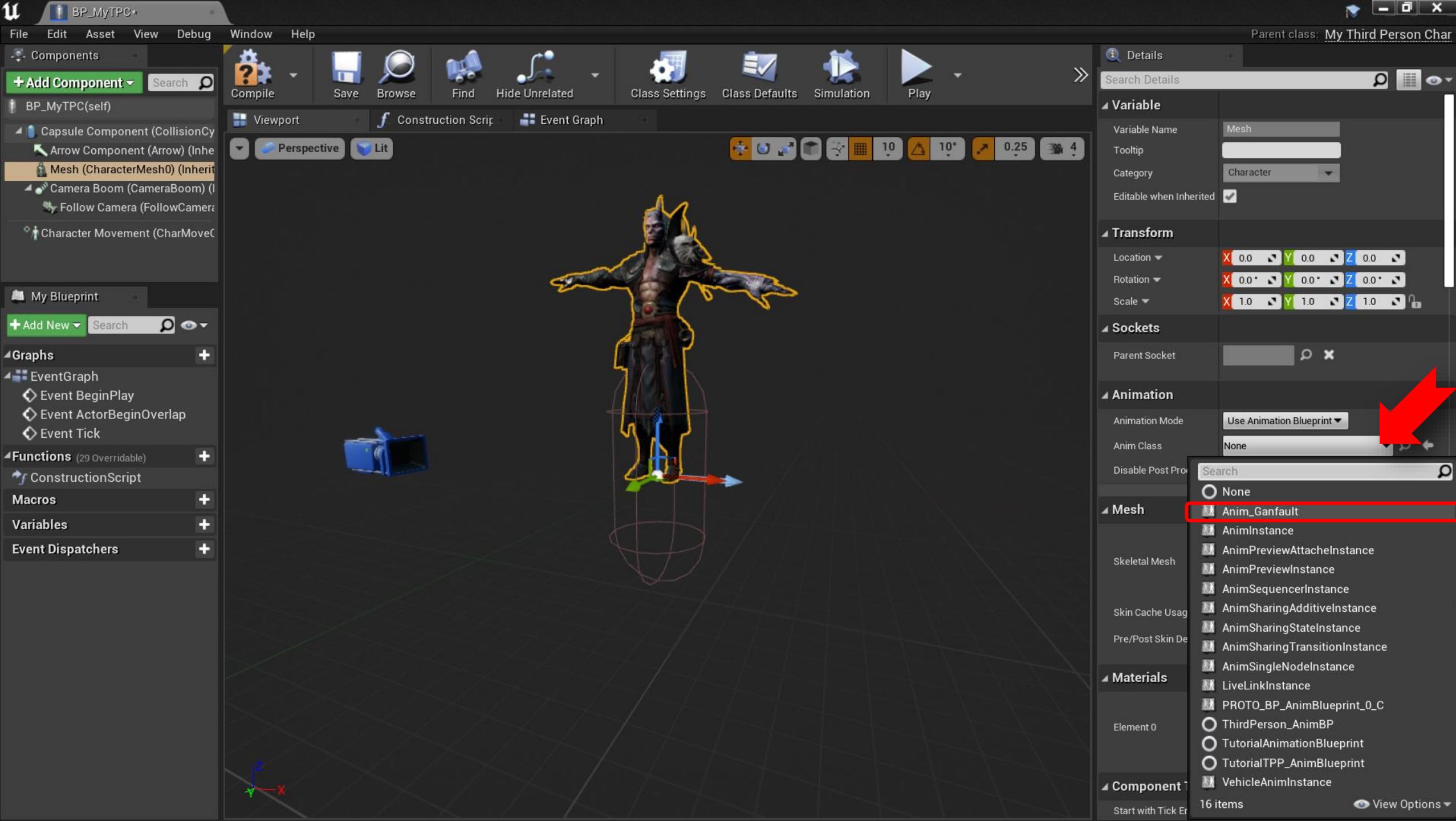
The screenshot shows the Content Browser interface. The top bar includes "Add/Import" and "Save All" buttons, and a navigation path: "Content > ThirdPersonBP > Blueprints". The left sidebar shows a tree view of "Content" with folders like "Ganfault", "Geometry", "Mannequin", "ThirdPerson", and "ThirdPersonBP" (which is selected). The main area displays five Blueprint assets: "Anim_Ganfault", "BP_MyTPC", "BP_WalkRun", "ThirdPerson Character", and "ThirdPerson GameMode". The "ThirdPerson GameMode" asset is currently selected. The bottom left shows a "Search Paths" field and a "Filters" dropdown. The bottom right contains "View Options" buttons. A red box highlights the "ThirdPerson GameMode" asset in the list.

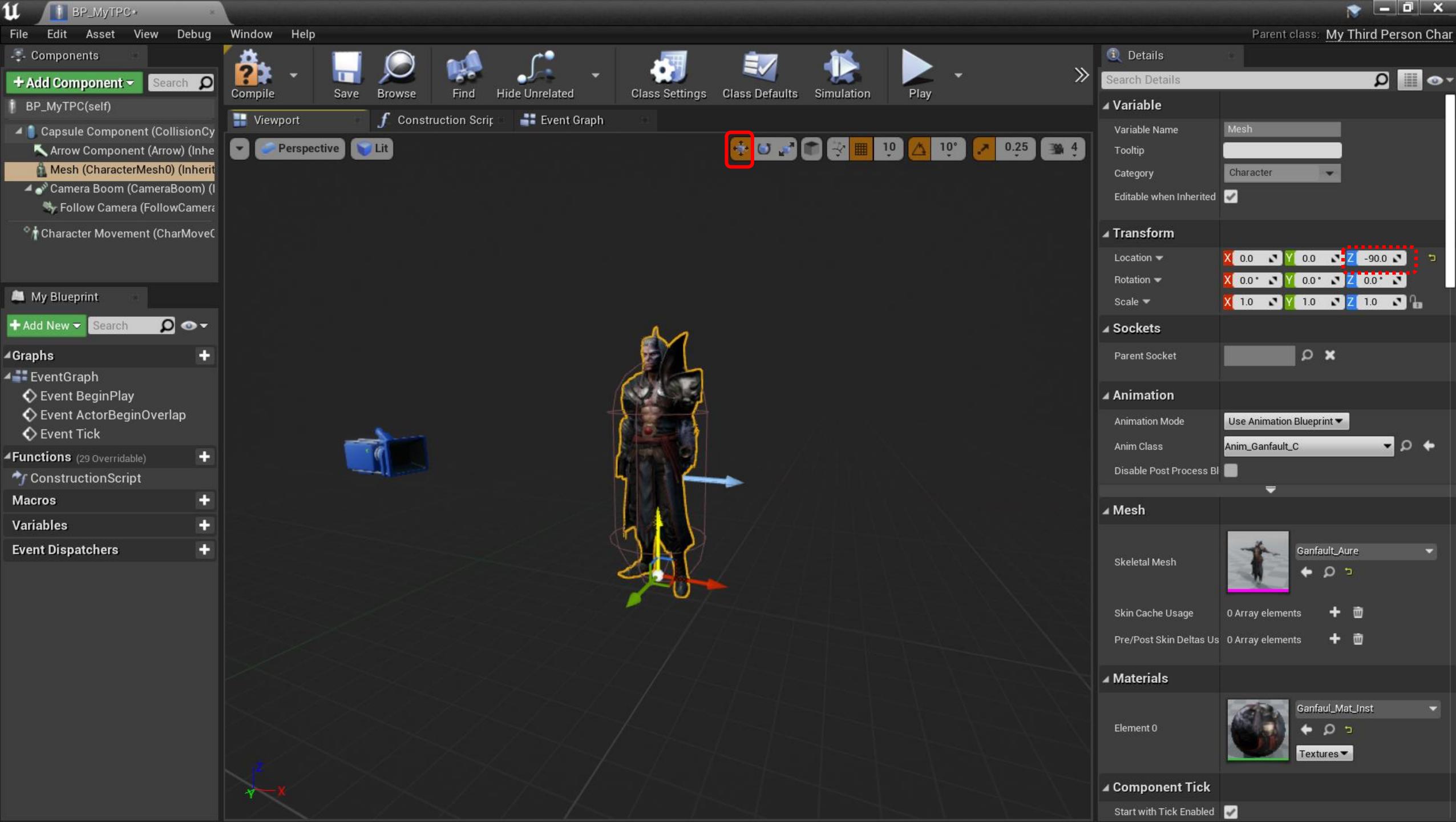


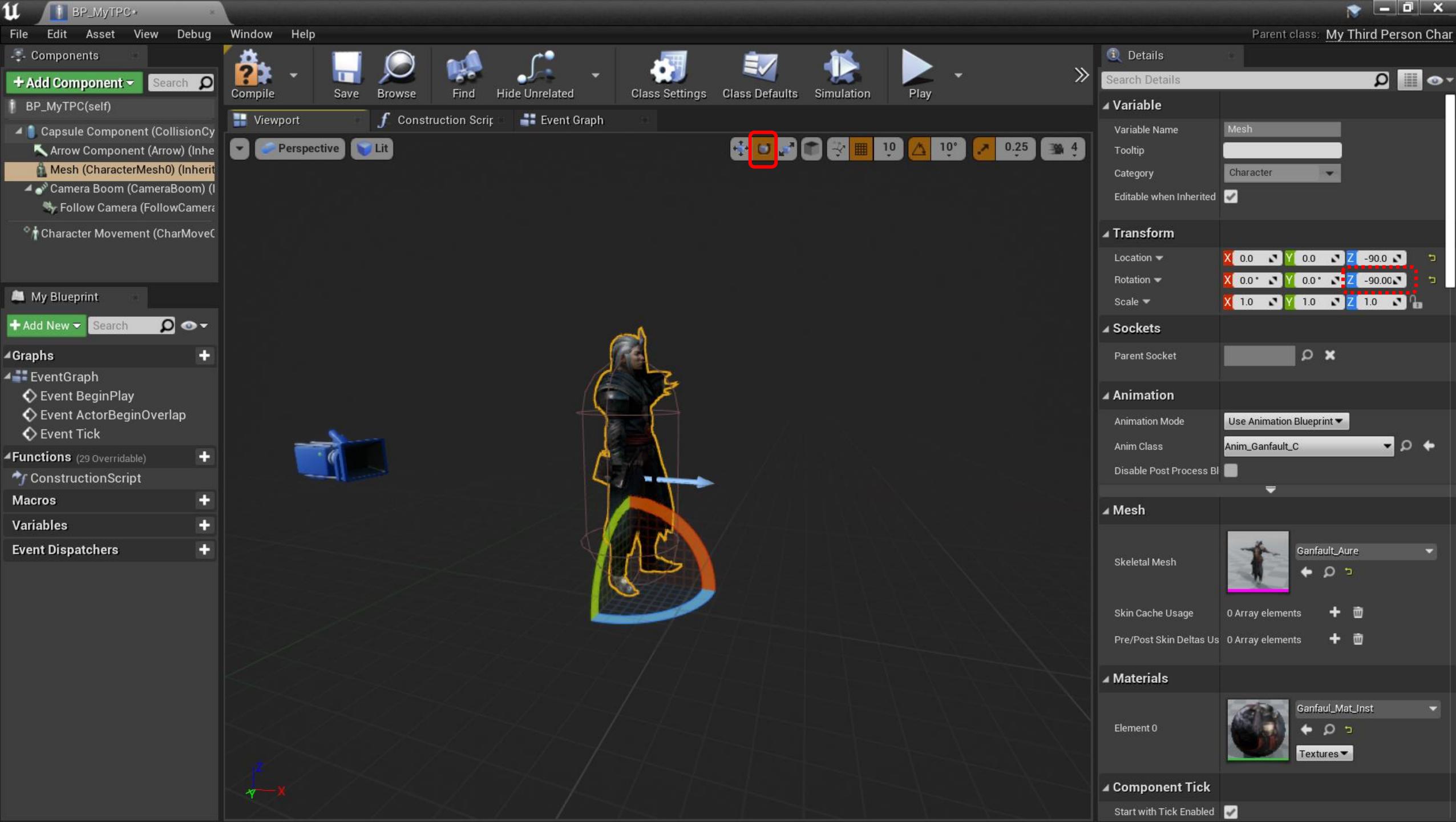


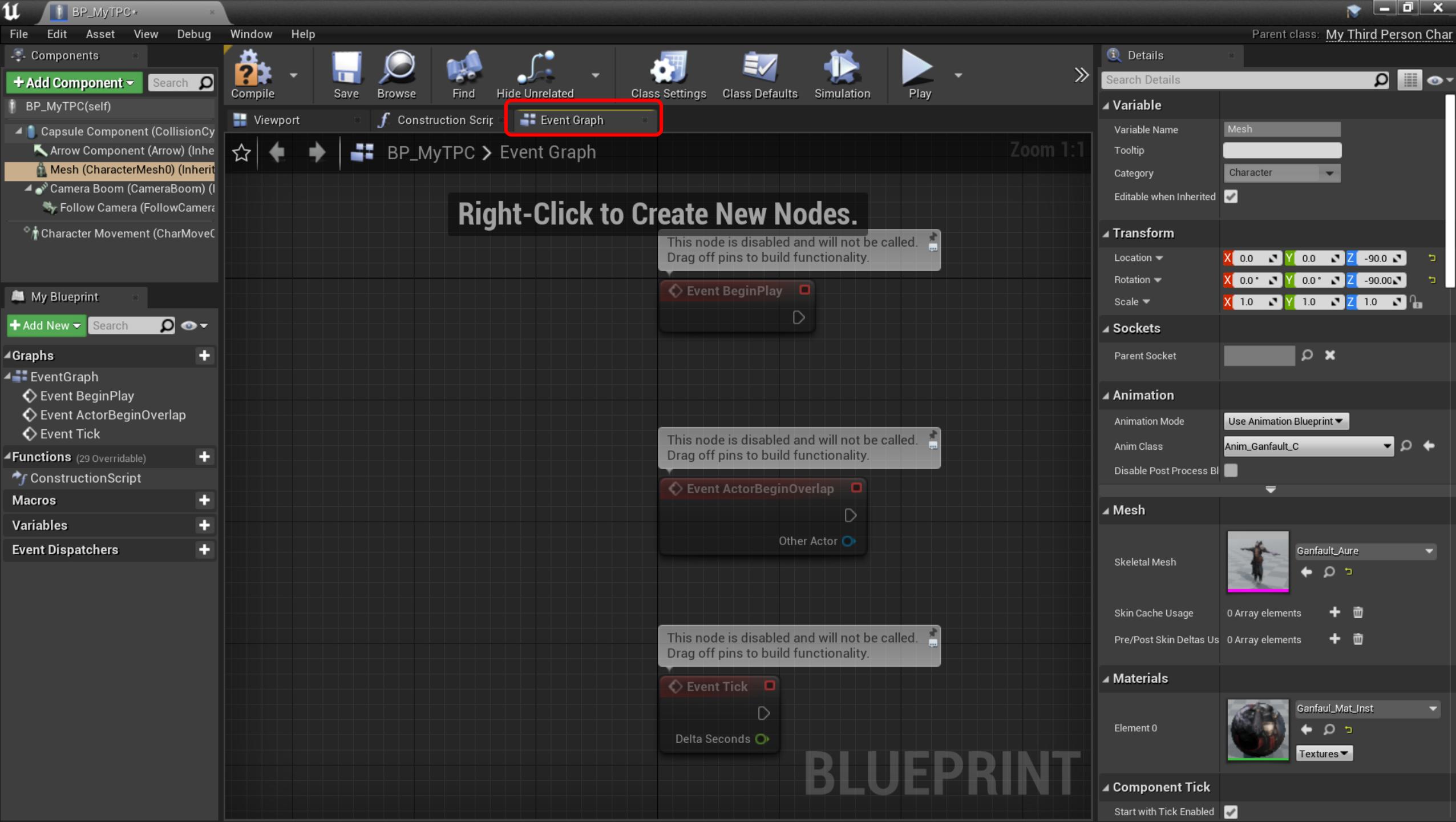


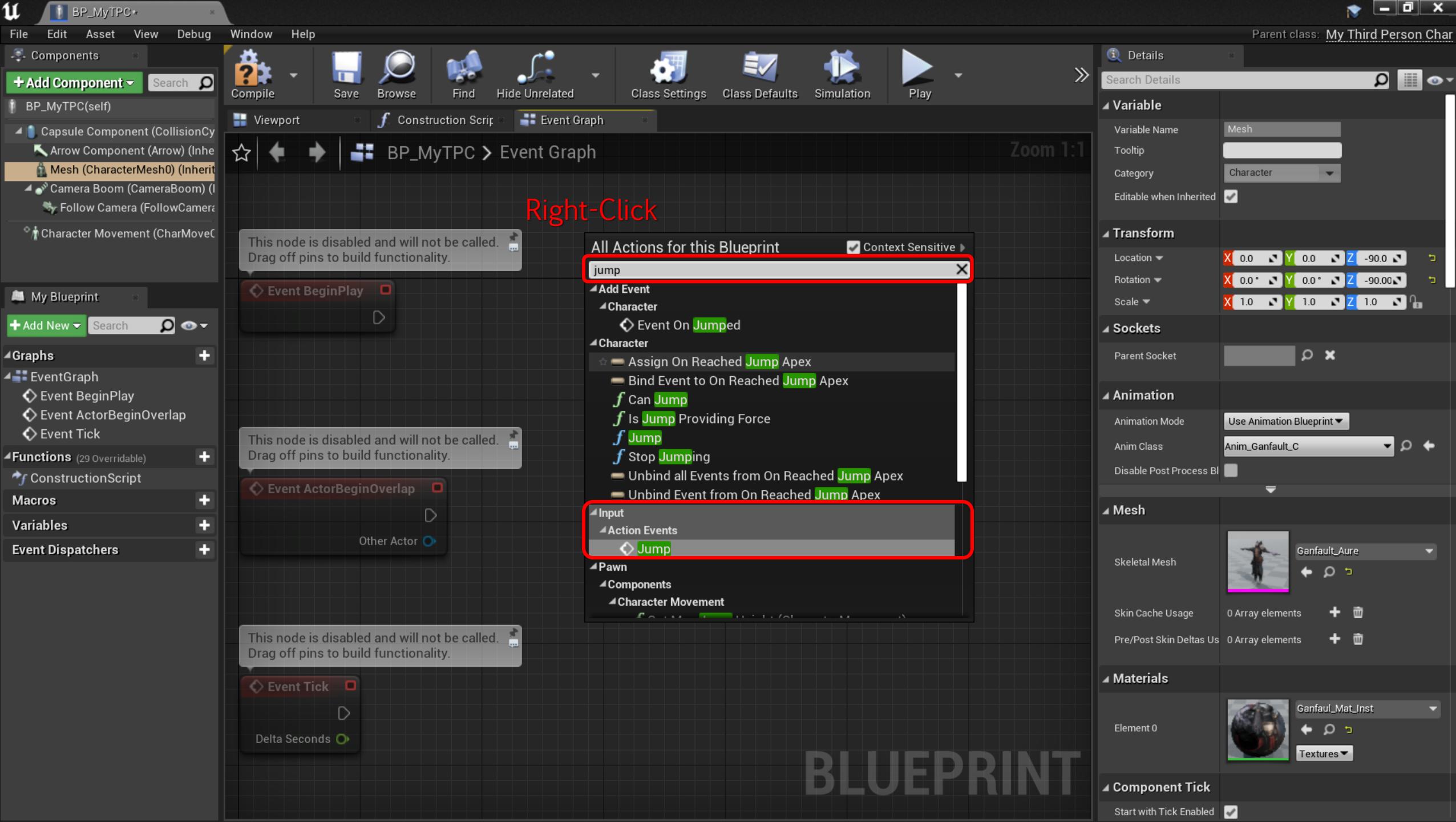


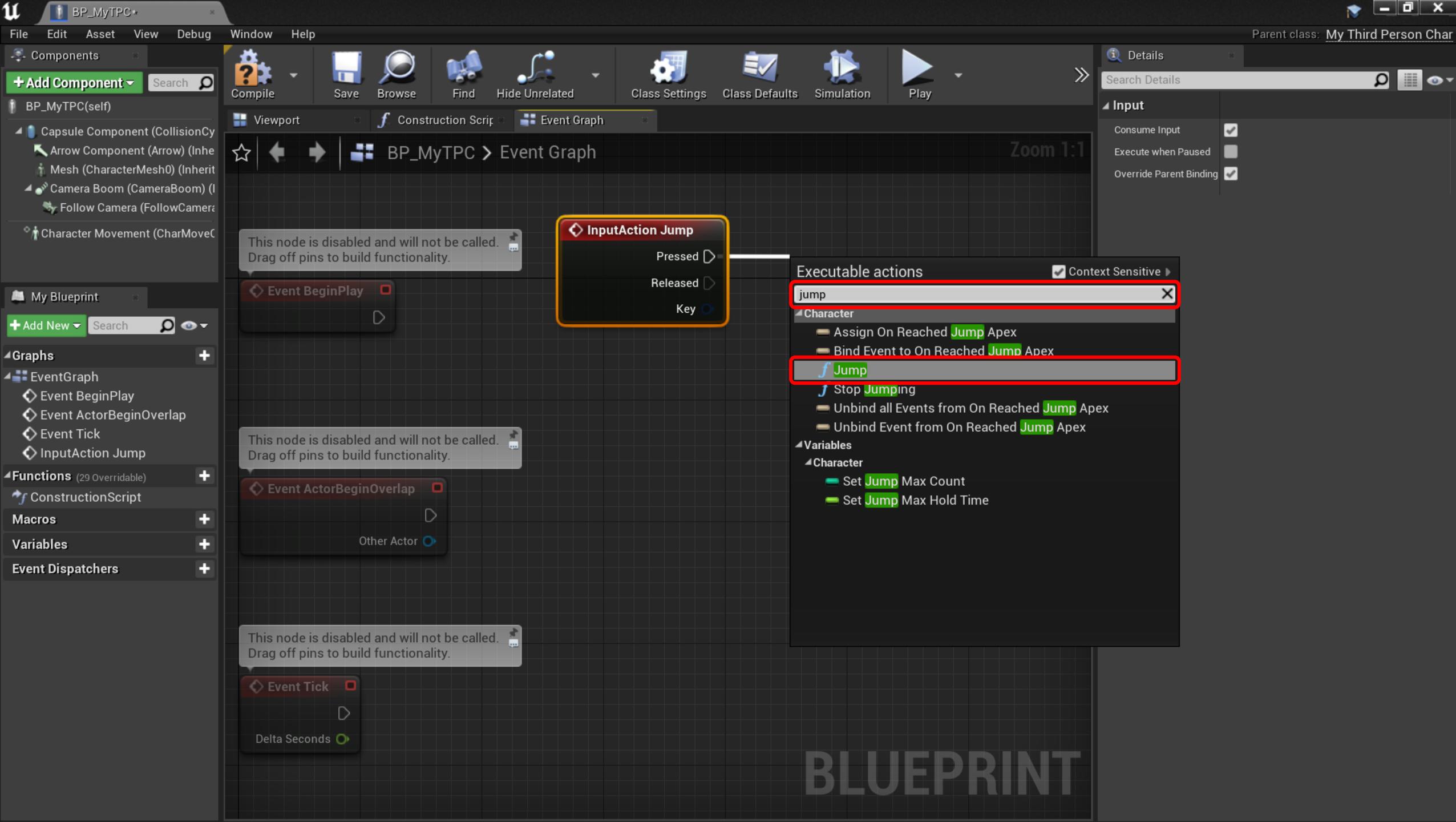


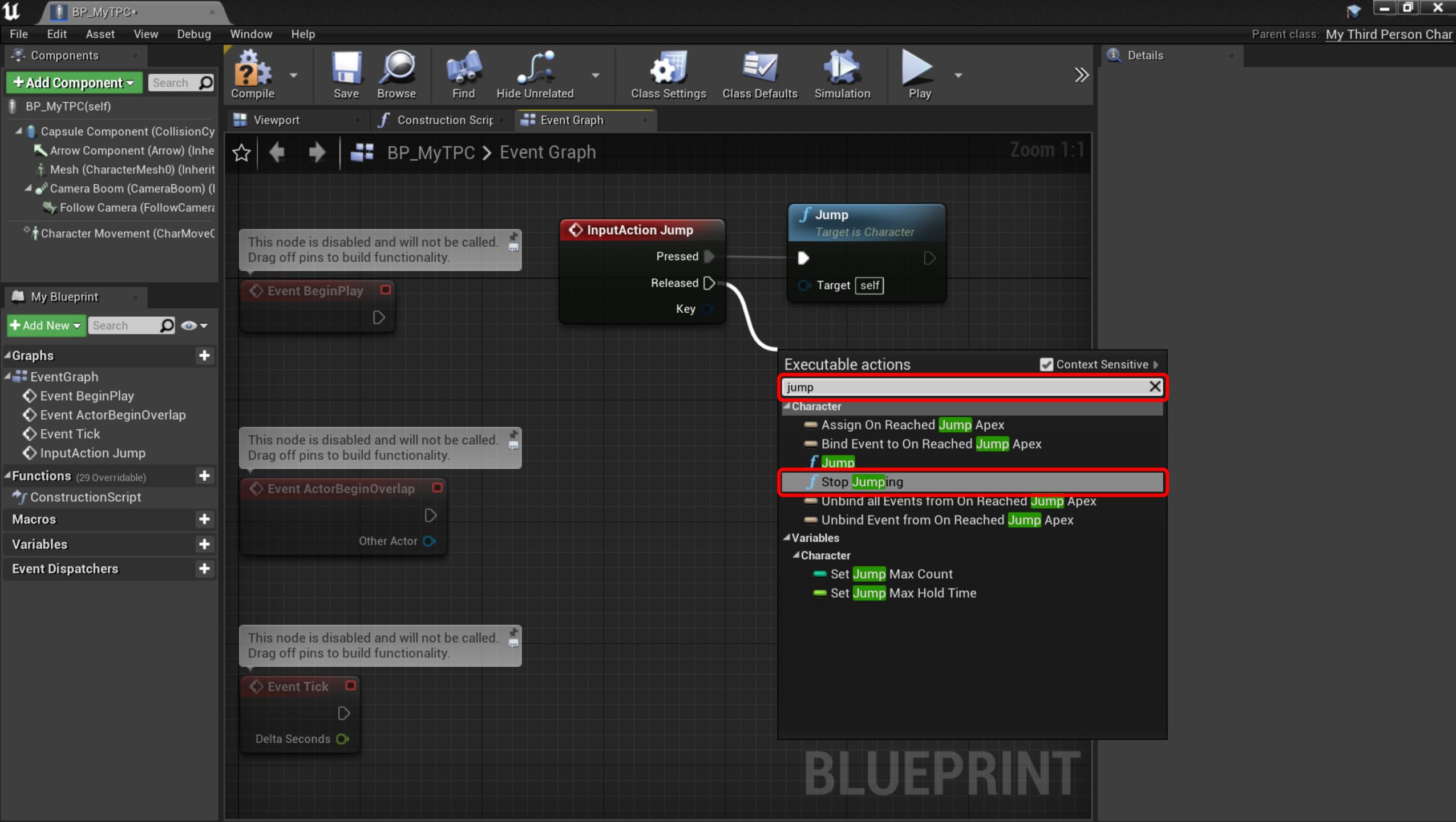


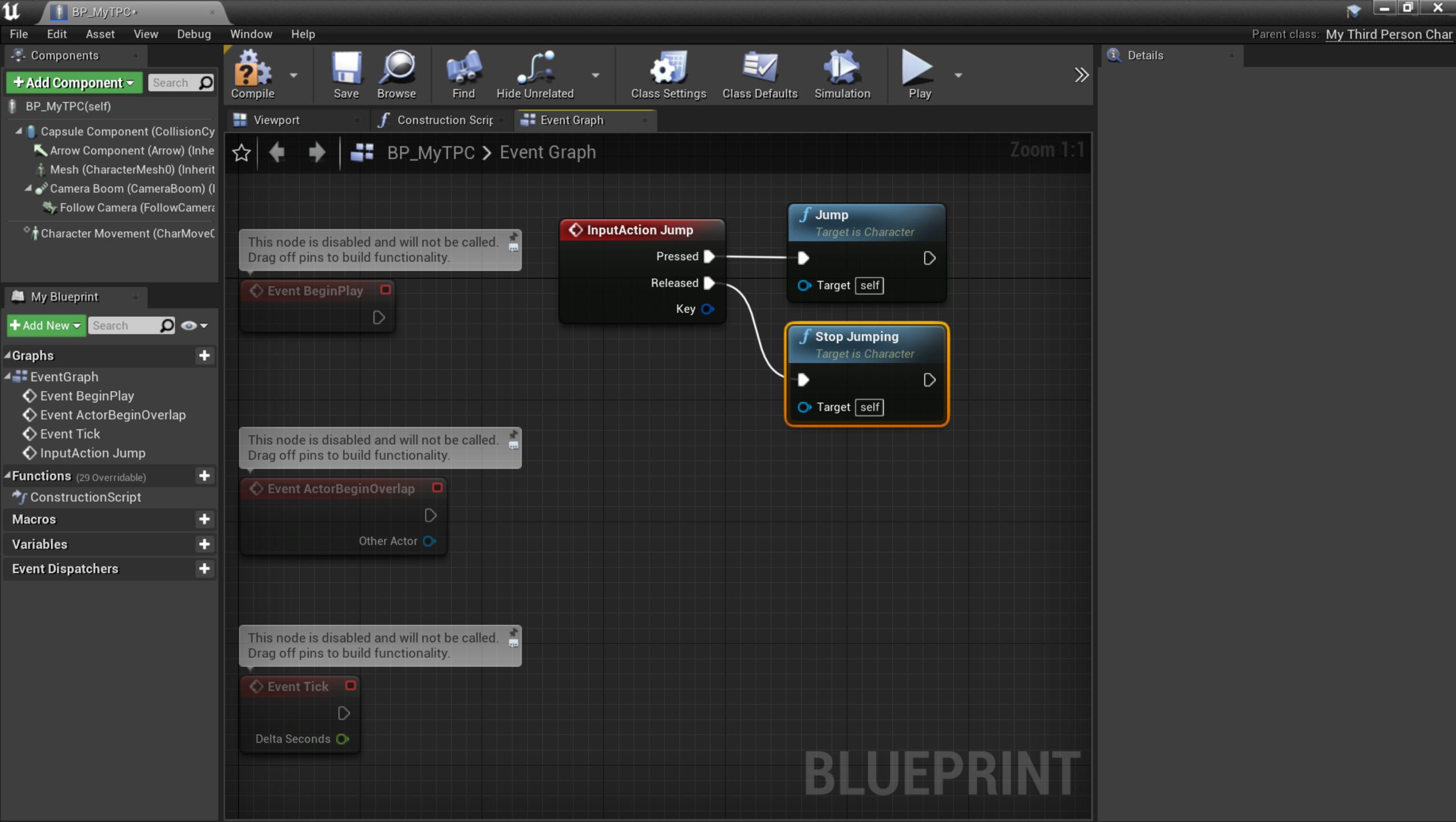


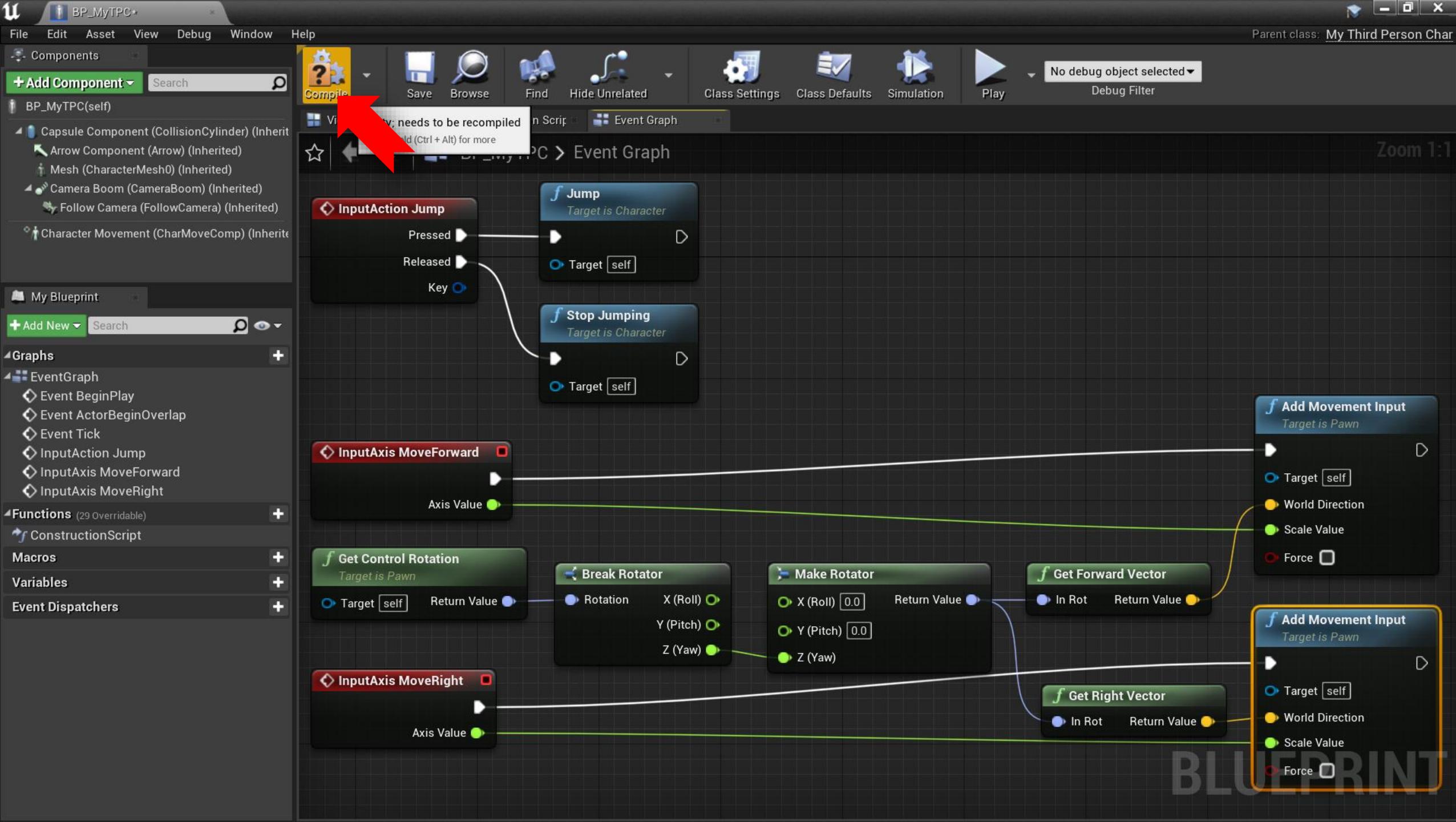


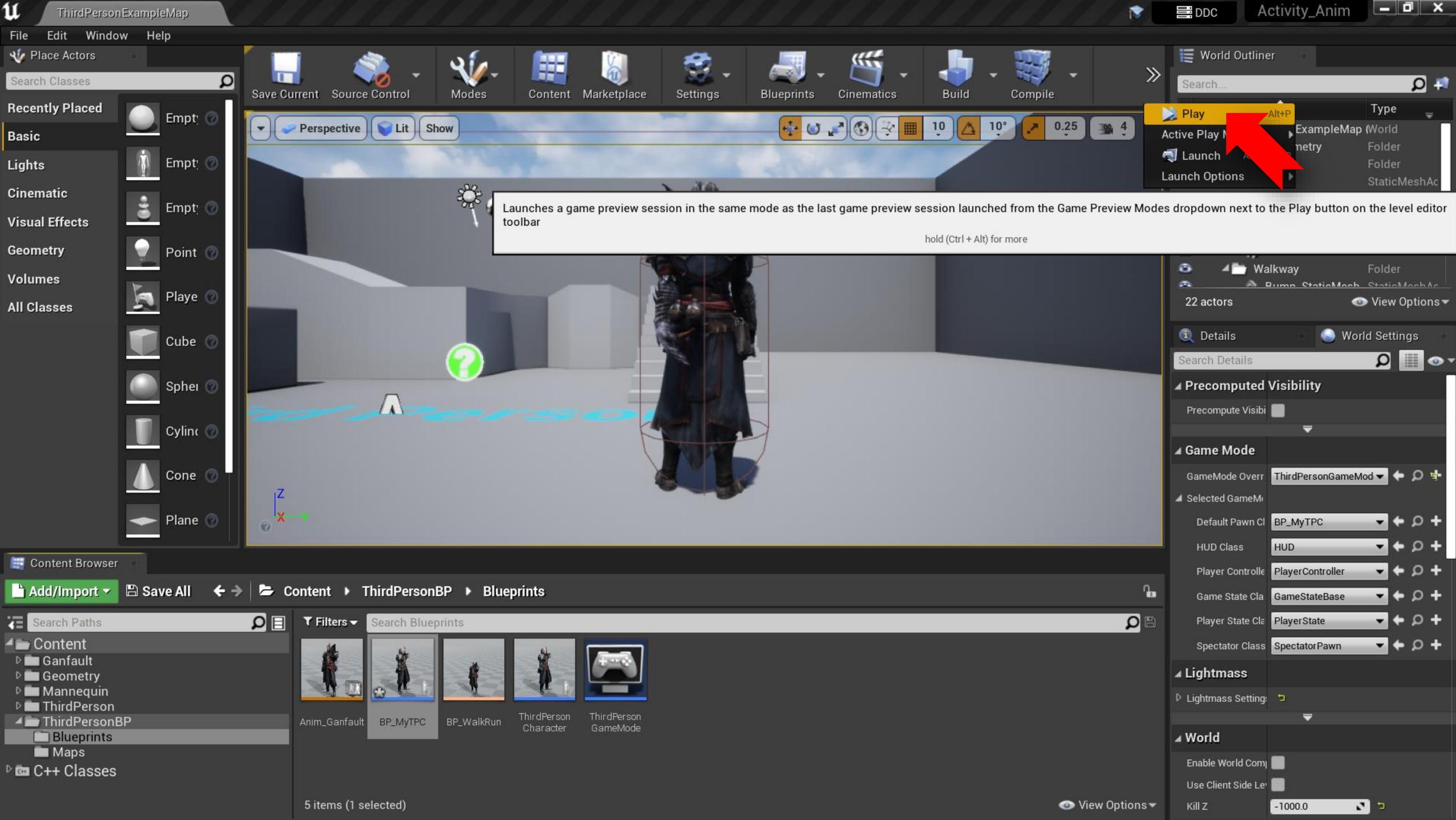












ThirdPersonExampleMap DDC Activity_Anim

File Edit Window Help

Place Actors Search Classes

Recently Placed

- Basic
- Lights
- Cinematic
- Visual Effects
- Geometry
- Volumes
- All Classes

Save Current Source Control Modes Content Marketplace Settings Blueprints Cinematics Build Compile

World Outliner

Search... 32 actors View Options

Details World Settings

Precomputed Visibility Precompute Visibi

Game Mode GameMode Overr ThirdPersonGameMod

Selected GameMode Default Pawn Cl BP_MyTPC

HUD Class HUD

Player Controller PlayerController

Game State Cl GameStateBase

Player State Cl PlayerState

Spectator Class SpectatorPawn

Lightmass Lightmass Setting

World Enable World Comp

Use Client Side Le

Kill Z -1000.0

Add/Import Save All Content > ThirdPersonBP > Blueprints

Content Browser

Content Paths

- Content
- Ganfault
- Geometry
- Mannequin
- ThirdPerson
- ThirdPersonBP
- Blueprints
- Maps

C++ Classes

Filters Search Blueprints

Anim_Ganfault BP_MyTPC BP_WalkRun ThirdPerson Character ThirdPerson GameMode

5 items (1 selected) View Options

A screenshot of the Unreal Engine 4 Editor interface. The central view shows a character in mid-air, wearing a dark, futuristic suit with glowing blue elements. The character is positioned above a white, geometrically shaped platform or floor. The background features a bright blue sky with scattered white clouds. On the left side, there's a toolbar with icons for saving, source control, modes, content, marketplace, settings, blueprints, cinematics, build, and compile. Below the toolbar is a 'Recently Placed' panel containing icons for various basic game objects like spheres, cubes, and planes. The bottom navigation bar shows the current project structure: Content > ThirdPersonBP > Blueprints. The right side of the screen contains several panels: 'World Outliner' listing 32 actors, 'Details' and 'World Settings' for game mode configuration, 'Lightmass' settings, and 'World' settings. The bottom left shows the file browser with a list of content assets like 'Anim_Ganfault', 'BP_MyTPC', etc.