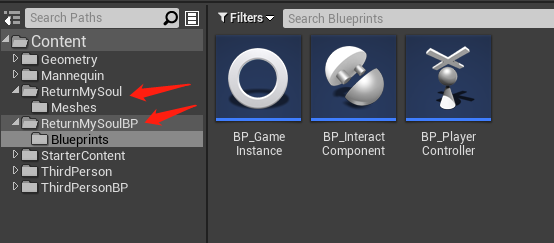
##### Overview

The project is made from the third person blueprint template. All the custom files we make specifically for Return My Soul project should be put into “Content/ReturnMySoul” and “Content/ReturnMySoulBP”.



##### Basic Concept

The whole idea of switching different objects is to use the same PlayerController to possess different Pawns. Different interact object should be derived from Pawn or its sub classes(for example, Character) and has its custom logic of movement, animation, etc.

Unreal official document about possessing:

<https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/HowTo/PossessPawns/>

##### Basic Blueprints

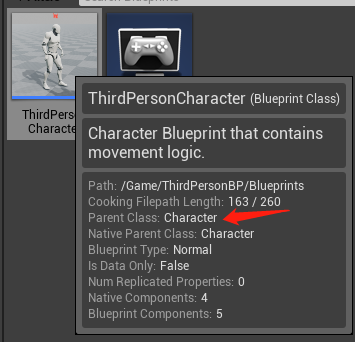
1. **BP\_PlayerController**: custom PlayerController for this project. It manages an array of pawns to keep track of the objects player can switch between and handles switching object input(Press keyboard F to switch).
2. **BP\_InteractComponent**: custom Actor Component to help BP\_PlayerController manage the pawn array by collision detection. When collision happens, it will add its owner(Pawn) to the pawn array. After collision, it will remove its owner from the array. It also automatically uses the default AIController to possess back to its owner when it is not possessed by PlayerController. **Every custom interact object we make should have this component!**
3. **BP\_GameInstance**: custom GameInstance. GameInstance is an instance that always exists as long as the game is running. **We should use this class to store global variables!**

##### Example: To set up an interact object step by step

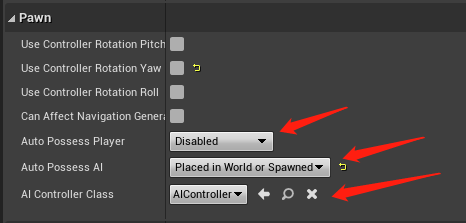
Please see the example level: Content/ThirdPersonBP/Maps/ThridPersonExampleMap

Note: When the game starts, you will find PlayerController does not possess any pawns in the scene. I set this on purpose because I have to wait for every pawn to spawn a default AI Controller. To start playing, we can hit “Tab” in the keyboard so that we can possess the first pawn we find in the scene. This is just an temporary solution.

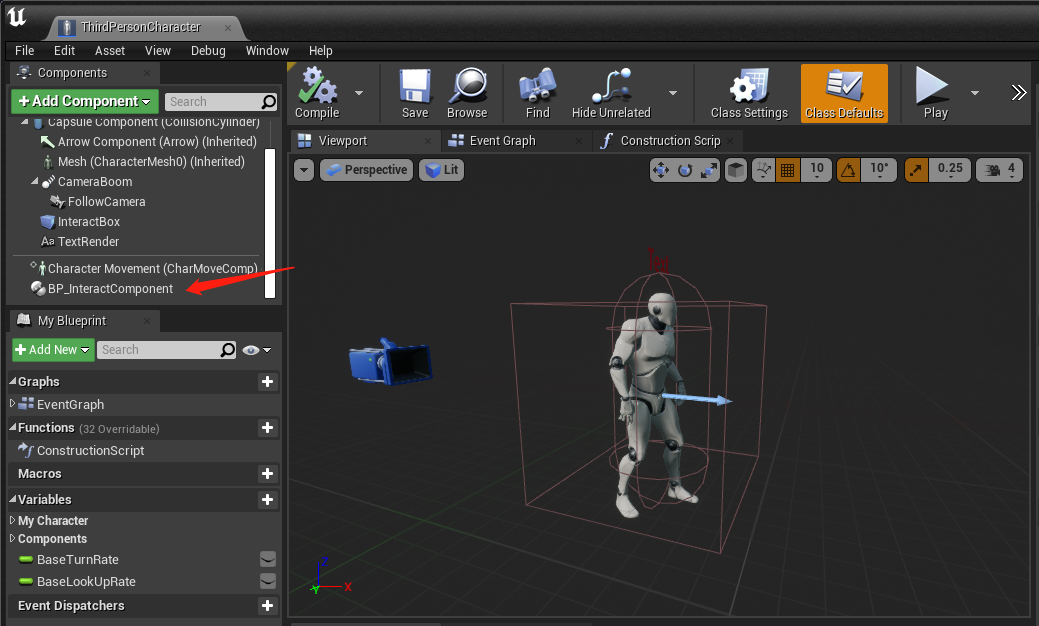
1. For every custom interact object, we make sure it is **derived from Pawn or its sub classes**. In this example, I use the original ThirdPersonCharacter from the template project.



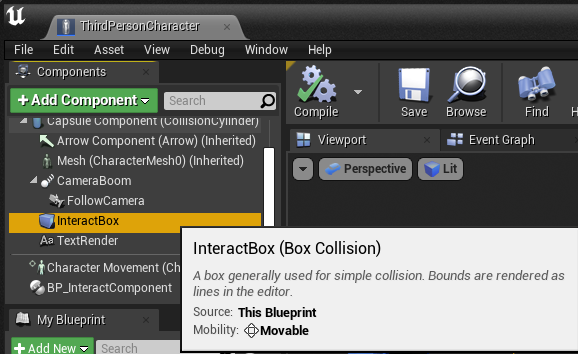
1. Open blueprint editor for the custom interact object, **set up the pawn section**:



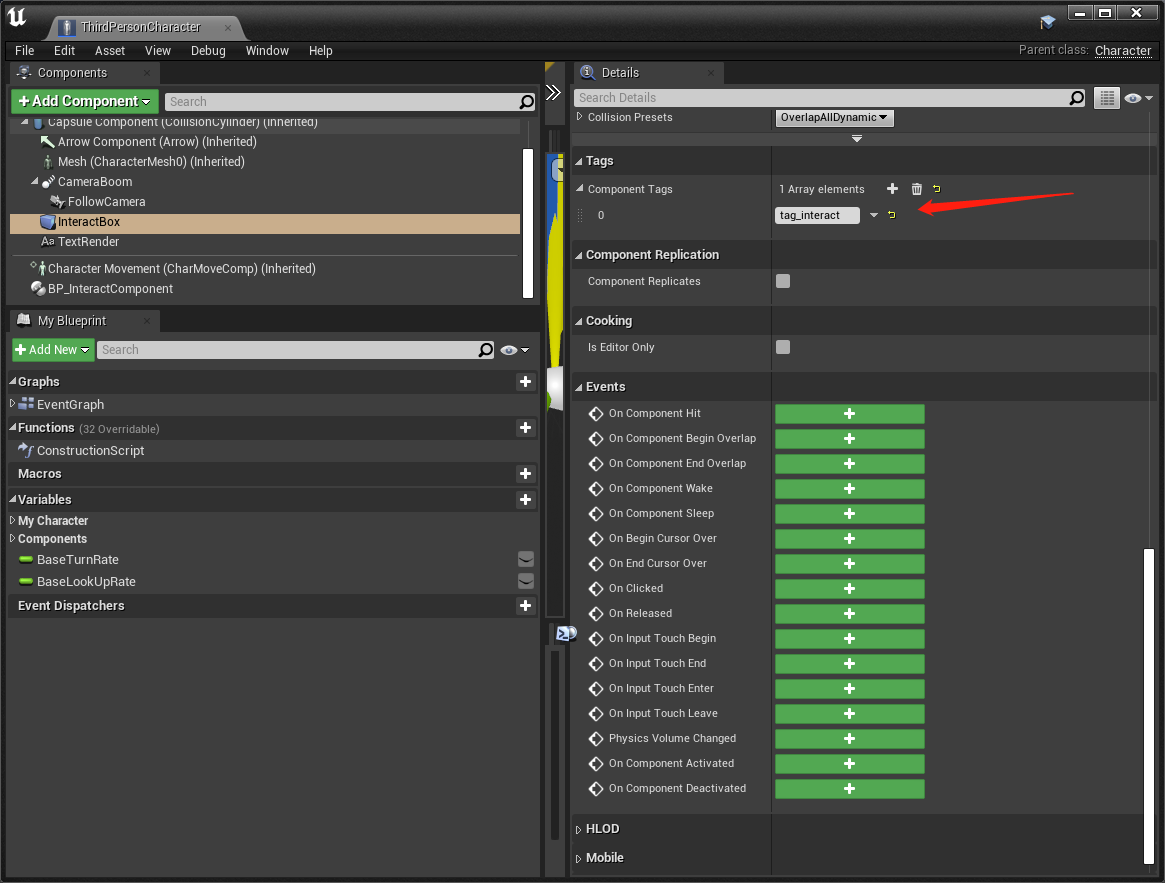
1. **Add BP\_InteractComponent**.

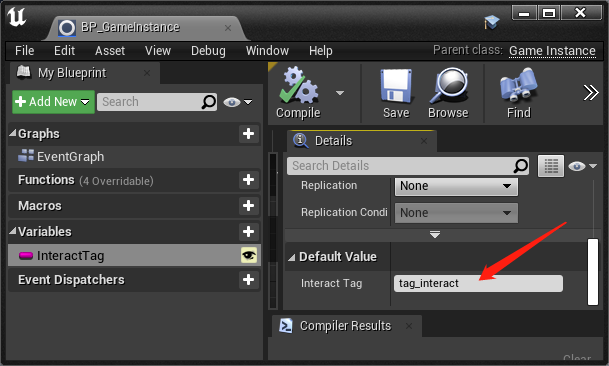


1. **Add collision component**. In this example, I add a Box Collision component and rename it to “InteractBox”. We can also use other collision shape such as Sphere Collision, Capsule Collision, or any other custom collision classes that derive from ShapeComponent (I use event OnComponentBeginOverlap() and OnComponentEndOverlap() from this component to detect collision)



1. **Add tag** to the collision component. The tag string is “tag\_interact”. Make sure this tag value also matches the variable value in BP\_GameInstance.





Also, when the game is running, you can see on the screen the debug information of how many interact objects are added to the pawn array.

