

Inventory System Notes, Part 1, Setting Up

First we need an ACTOR COMPONENT called **Inventory Component** Which will be the foundation for which the inventory will be a part of, all things we apply the inventory component too have it.

We also create a BLUEPRINT CLASS called **ITEM (CLASS)**

An **ItemStructure** is made, which holds all details our items will have.

	Type:	
Name	text	
IsStackable	boolean	these also have default values which we set as well.
Thumbnail	2D texture	
MaxStackSize	integer	
IsConsumable	boolean	
Durability	float	
Description	text	
Class	item class .	(this is a ref. to class we created)

Each Item, will have the Item Structure.

In Item class we set a Variable to ItemStructure. and its Type is ItemStructure we created

TheSlotStructure, will be what makes up the foundation of the inventory slot, it will contain details about each slot in the inventory

only 2 variables:

	Type:
ItemStructure	ref itemStructure (item that we store)
Quantity	integer

Now In **Inventory Component**

Variables:	Type
InventoryName	text
NumberOfSlots	integer
Inventory	Now inventory component will have SlotStructure Type (ARRAY)

AN Array because there will be multiple slot structures.

Now in **THIRDPERSONCHARACTER**, will have this **InventoryComponent** we have created.
details:

Its name will be **BACKPACK**

It will have 16 slots, use **INVENTORY** if you want player to start off with an item.

Now we will create an Interface, called the **Interact Interface**, (READ ONLY FUNCTIONS THAT ARE COMMON BETWEEN ALL CLASSES THAT USE THE INTERFACE. The only function in this interface will be INTERACT

we want ITEM CLASS to use the INTERACT Interface. we can make any class use it.

In ITEM class settings, Interfaces(implemented) and choose Interact Interface, now INTERACT will come as an event. And Our **THIRDPERSONCHARACTER** can use the interface.

ProjectSettings-->Input-->ActionMappings--->we will create an **Interact** one

ThirdPersonCharacter:

RickClick: ActionEvent: Interact (From Key Binding E we just made)

first we GetOverLappingActors(ClassFilter: Actor: Returns all actors we overlap)

→

→

ForEachLoopWithBreak

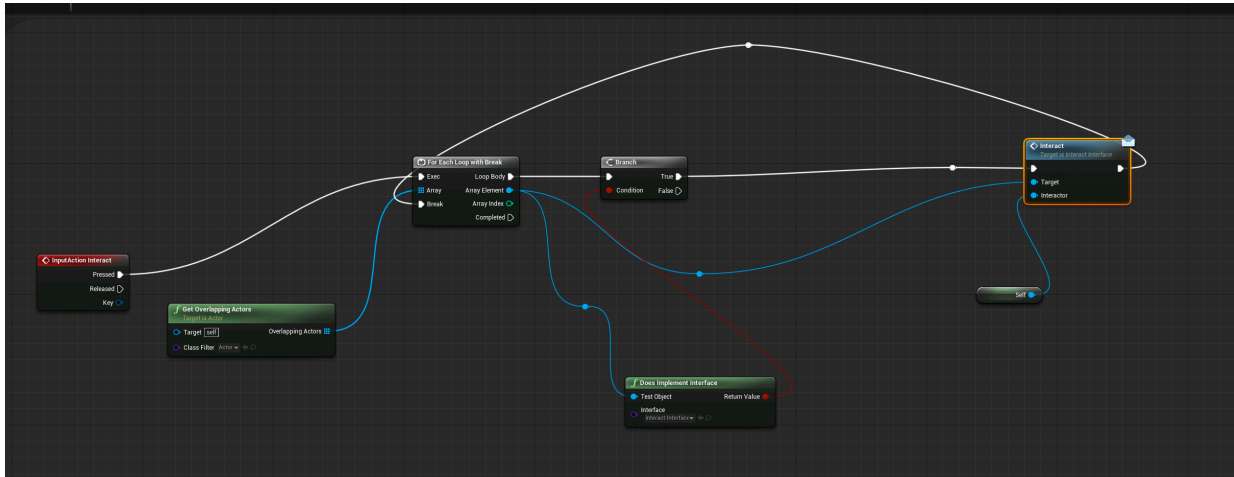
LoopBody: check if element has Interface

----> Interface(DoesImplement?) (Choose Interact Interface) (also goes to branch along with LoopBody)

----->Branch

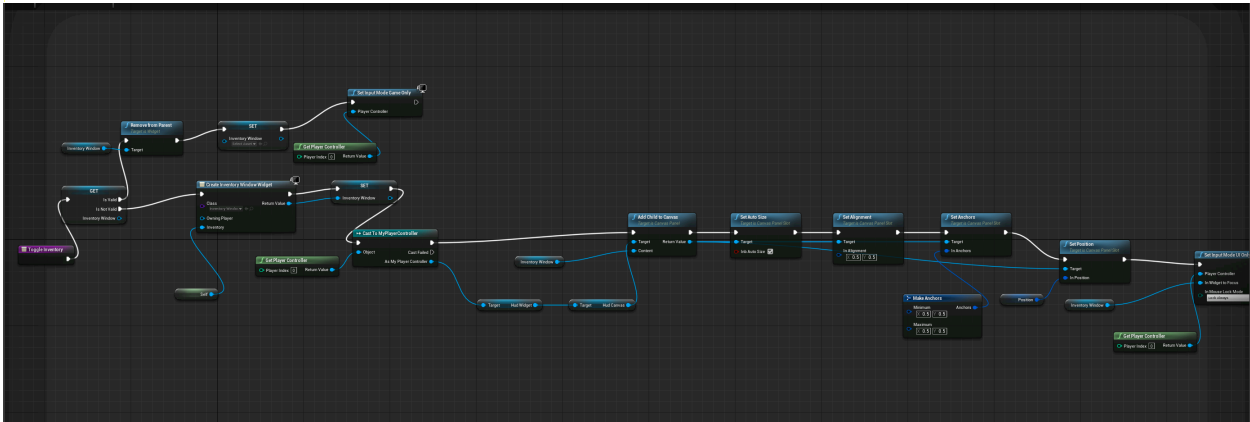
If True Interact, Target will be the element from the ForEachLoop (In Interact, it will now have an Interactor key, which is just itself, because it is the interactor)

Gets Looped Back To The Break

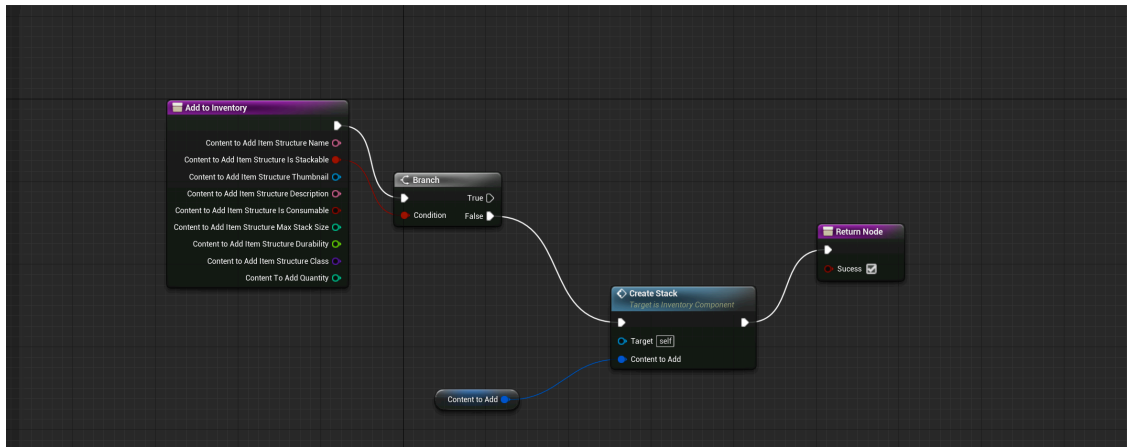


InventoryComponent Has 4 Functions

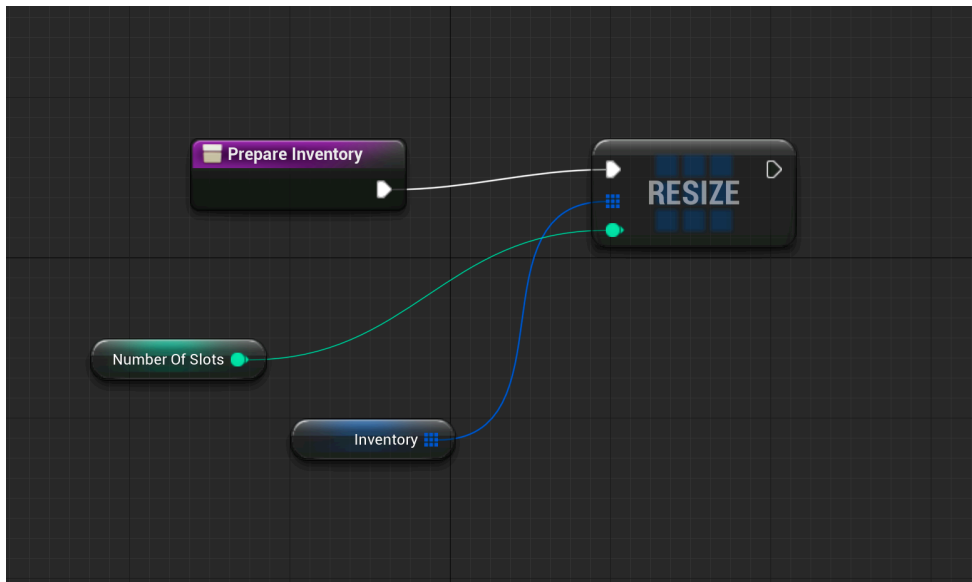
Toggle Inventory:



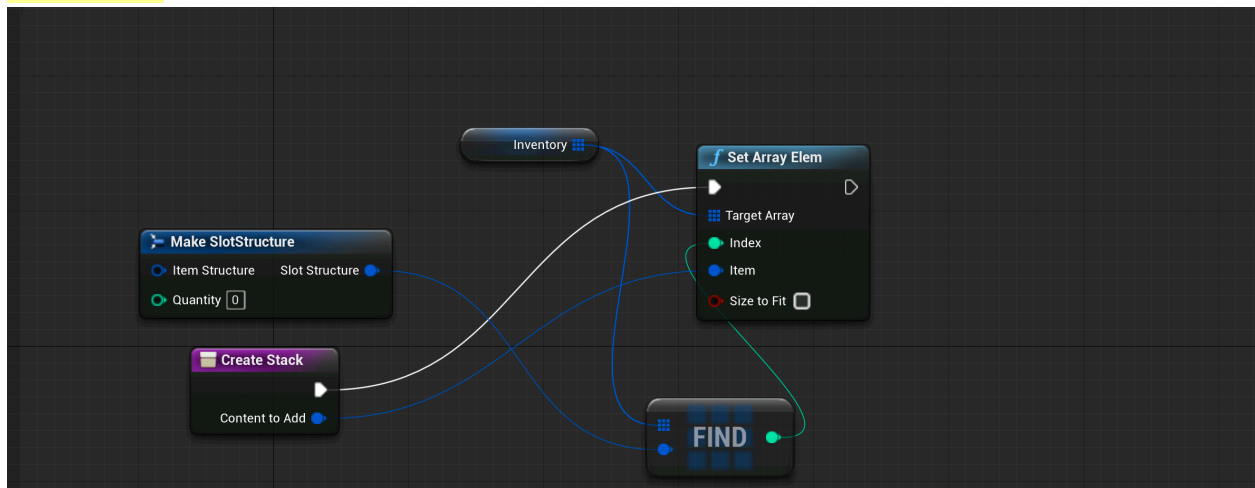
AddToInventory



PrepareInventory:



Create Stack:



Variables In Inventory:



InventoryUI