InventorySystem

An advanced Drag 'n Drop inventory system

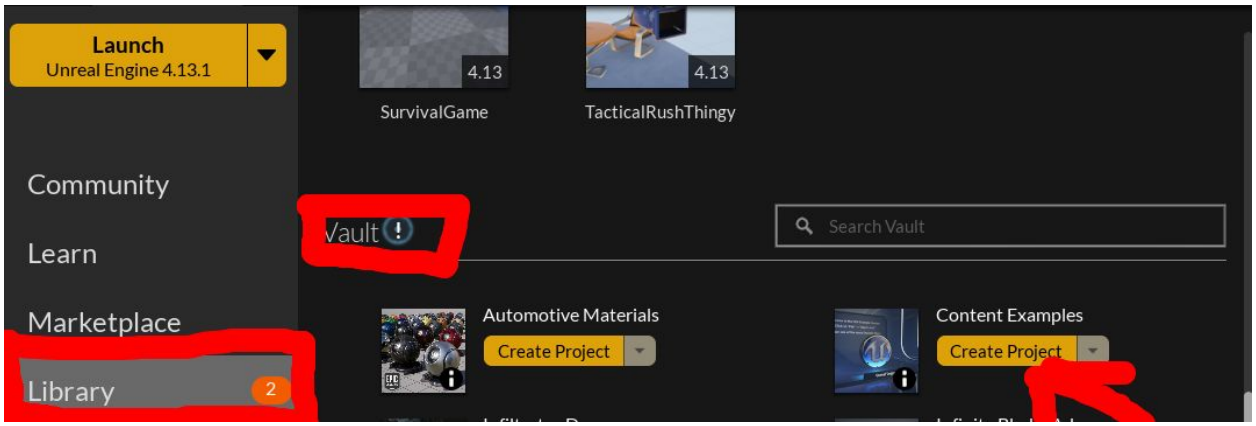
Documentation

11.06.2016

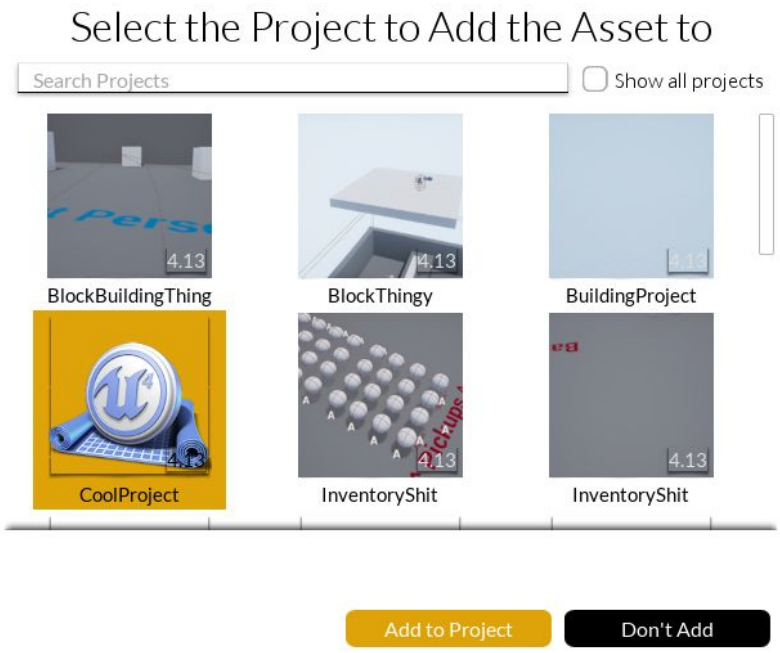
A HowToCompute Publication

Getting Up And Running

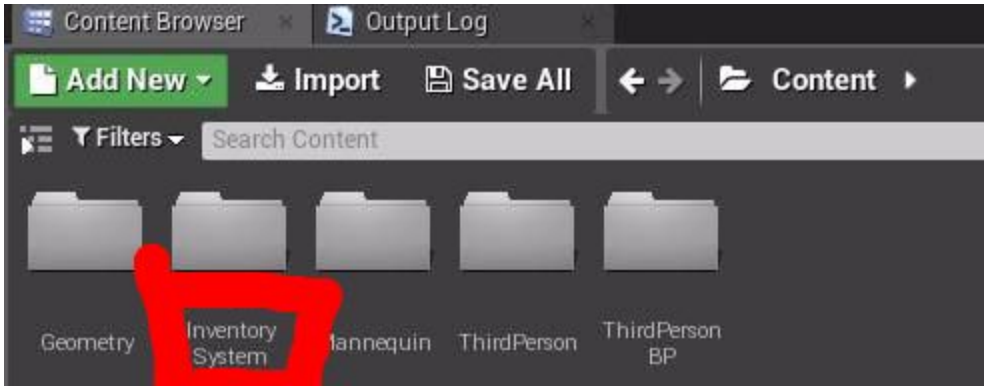
Start by adding InventorySystem to your project. After you have purchased InventorySystem from the unreal engine marketplace, it will end up in your vault. So open up the epic games launcher, go to the unreal engine tab, Library and if you scroll to the bottom, there is a tab named vault. Find InventorySystem and click on Add To Project (TODO update this immage once I have it on the marketplace).



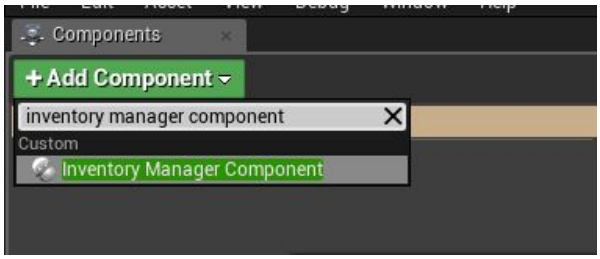
Now select the project you want to add it to, in my case this will be CoolProject. For reference, this is just the 3rd person template with a custom playerstate.



Now in your content browser, you should have the InventorySystem file, if not be sure you are not in a subfolder!

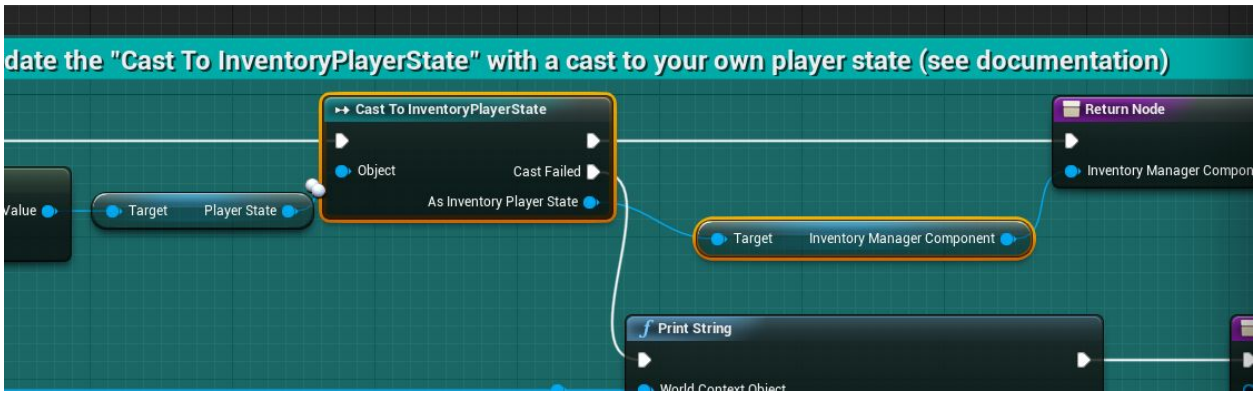


So the first thing that you will want to do is add an InventoryManager component to your PlayerState. For this open up your playerstate and add an “Inventory Manager Component”. (AddComponent, and go through the list)



Once you have added that hit Compile and Save.

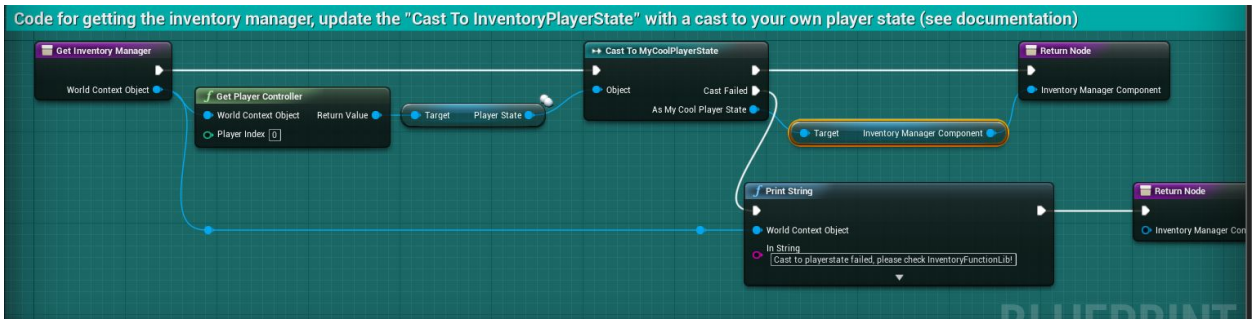
Now head over to InventorySystem > Blueprints > InventoryFunctionLib (blueprint file). And find the cast node.



Delete the highlighted, and drag out of the playerstate. Type in cast to <your playerstate>. Then drag out of that and type in “get Inventory Manager Component”

and press the one under variables > default. Now connect the output of that to the return node, and take the execution pin from “Get Inventory Manager” to the cast node. Now connect the execution pin without text to the return node, and the one with text to the Print String.

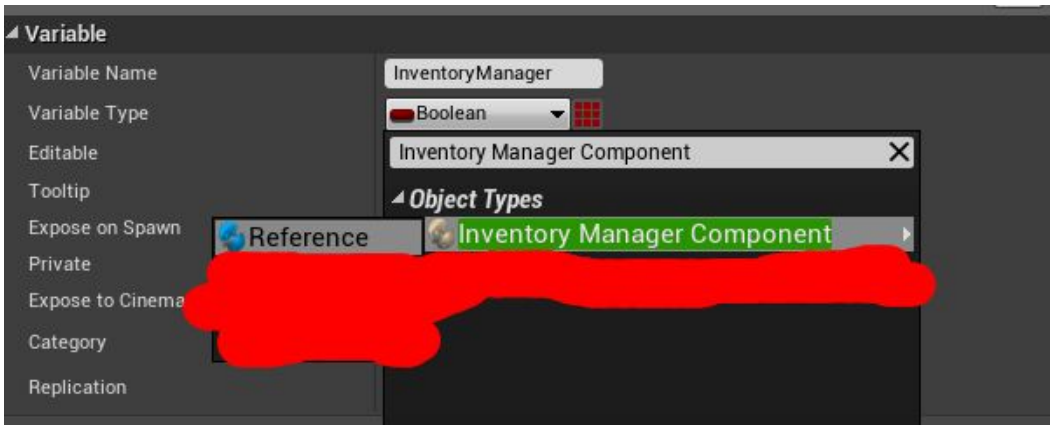
In the end you should end up with something like the following (MyCoolPlayerstate being the name of your playerstate):



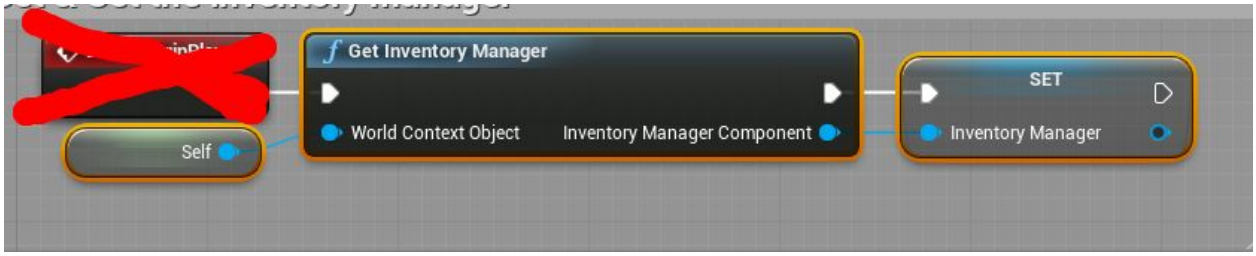
Now head over to InventorySystem > Example Project > InventorySystemCharacter. Head over to the “Inventory System code” part. Now head back to your content browser (don’t close this tab! Just minimize it) and open up your character (in my case the 3rd person character). If your character doesn’t yet have a begin play event, add an “Event BeginPlay” node. Now find the Variables tab on your left and press the +.

Name it “InventoryManager”. Be sure to spell this name correctly, or stuff will go wrong!

Give it the type “Inventory Manager Component” by Pressing the existing Variable Type (probably Boolean), typing in “Inventory Manager Component”, clicking it and pressing reference. Be sure to quickly compile and save.



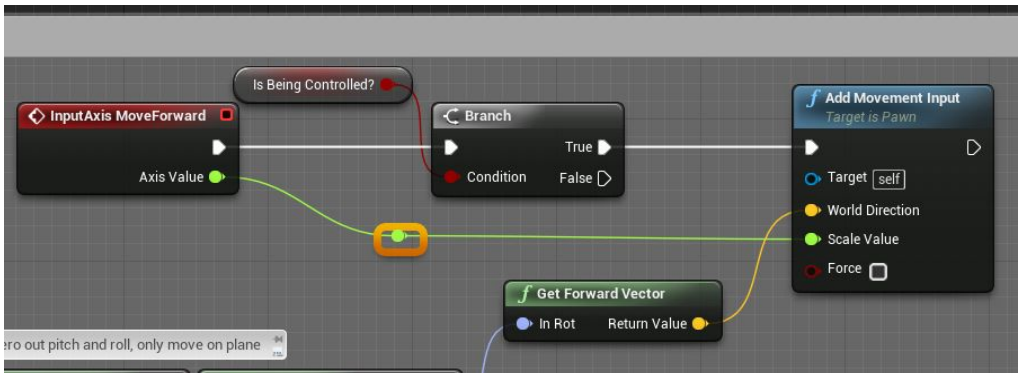
Head back to the InventorySystemCharacter and copy the below selected code.



Now go back into your character, and paste this right after begin play (to avoid the case in which it would not be called!). Be sure to connect the input from begin play to the Get Inventory Manager node. Don't forget to connect the output from the SET node to the rest of your own BeginPlay code.

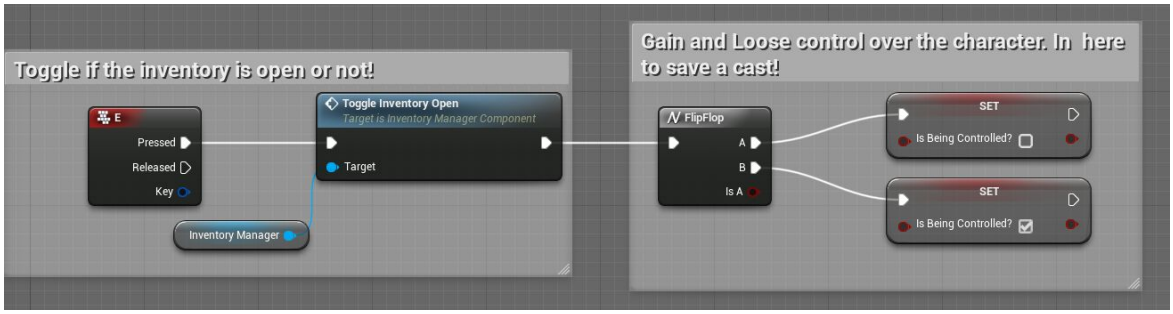
Now create a new variable of type Boolean and the name "IsBeingControlled?". Copy the following piece of code over to all of your input functions (unless you already have a system for loosing control in place, in which case you will need to modify later code). In the case that the checkbox is not check, please check it or you won't have any control over your character!

You should end up with code like this:



[explanation] Before your character can move or do anything it will check if it is being controlled, this allows the system to not let the user look around etc when they have their inventory open.

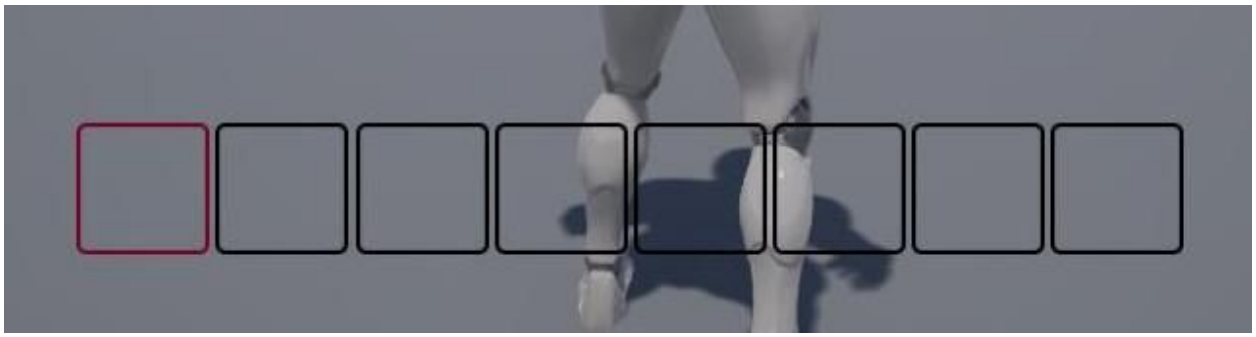
And now for the easiest part! Find the piece of code that toggles the inventory in the InventorySystemCharacter.



Copy that and paste it in a free spot of your own character. Compile and Save your character and we are ready to test!

Testing!

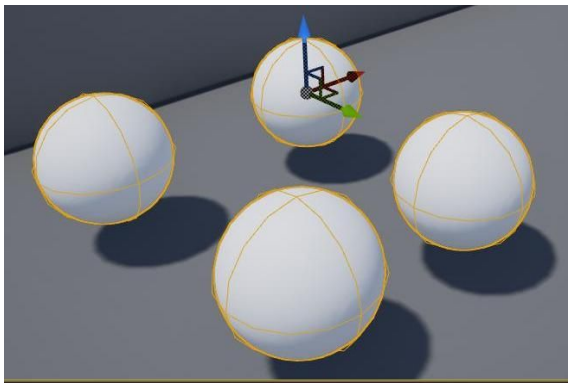
Now fire up one of your maps with your just edited character in it. When you press play you should see something like the following:



Now press E and you should get something like:

This means the basics work! Now let's try and add/remove some items!

Let's place a few pickups by going to InventorySystem > Mysc and dragging a few "Pickup"s into the scene.



Now you can adjust the pickup's settings, change the item and the amount. I'd personally recommend making 2 of them apples, and 2 of them burgers. I'd also make at least one less than 0 to test if removing items works. Now pick up items and drag them around in your inventory and move them to your hotbar. Try dropping a few and picking them up. (see the below article on controls).

If everything worked as expected, congratulations, you have successfully ported InventorySystem to your project!

Controls

Use E to open/close your inventory

Use keys 1-9 or your scrollwheel to select hotbar slots.

Use Q to drop an item, use CTRL+Q to drop all the items that are remaining in the selected slot.

Common Errors:

I get the error: "Cast to playerstate failed, please check InventoryFunctionLib"!

This means that it could not cast the current PlayerState to whichever cast node you specified. This is most commonly caused by setting the incorrect playerstate. Head over to your gamemode, and set the correct one. Did that not work? Check if you are overriding the playerstate in your world settings!

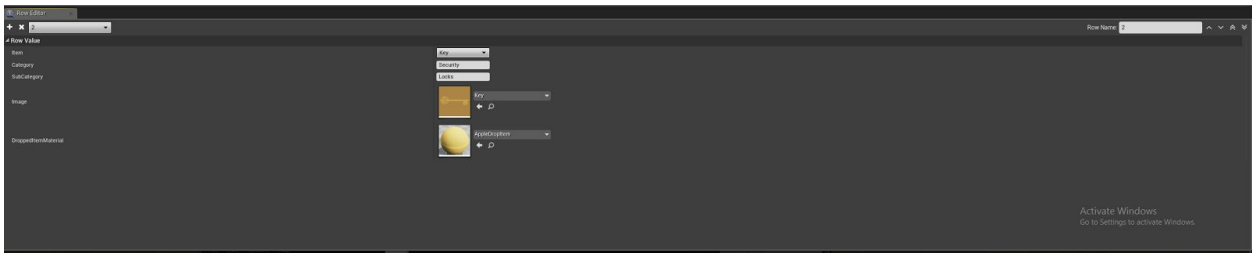
Adding your own items!

Adding items is easy!

First open up the EItems enum (in InventorySystem > Enums). Now Press the new button on the top right. Add as many as you want, giving them sensible names.

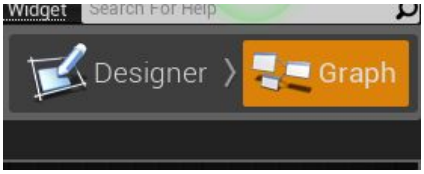
Now open up the Items datatable (in InventorySystem > Datatables). For each item press the + in the row editor. The current version of the system counts on the fact that you have ordered your enums and your datatable the same. So first of all give your row the name: previous row+1 (so if the last item had the rowname 2, this one will be

3). Now populate it with the wanted data. Category, SubCategory and DroppedItemMaterial are optional, tho setting a material is strongly recommended! Your row should look something like the following:



Now place a pickup in the your scene of choice, and set it's item to your newly created object. Now pick it up and see if it works as intended! Have you already downloaded the bonus assets? Click here.

Customizing the UI



If you head over to the UMG folder (Inventory System > UMG). For most of the widgets (atleast the ones that contain amounts) if you head to the graph, you can set the amount's color in the ItemAmmountText_Color variable!

More UI customization is possible, but not officially supported. Please stay tuned for future updates to feature more exposed variables to change the color of the slots etc!

Conclusion

This will conclude the documentation of the InventorySytem, if you still have any questions, don't be afraid to contact me at howtocompute123@gmail.com!