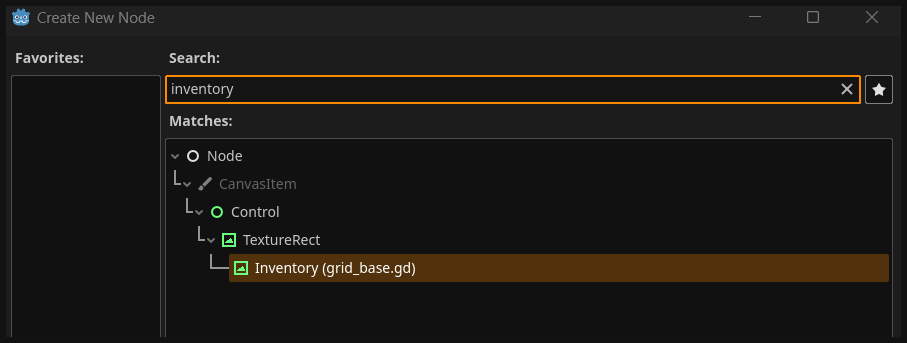
**Grid based inventory  
*Warning: this is the first time I make a documentation of something so excuse any retardation you may come across***

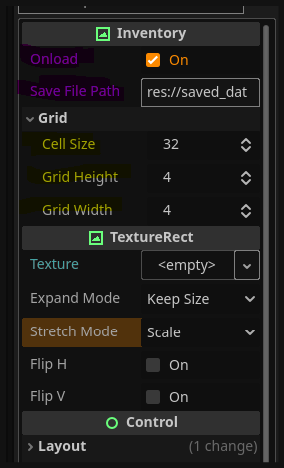
**Introduction:**

This is a simple guide on how to use the inventory for your game, if there is anything that I didn’t cover or you still have some questions you can send a comment on youtube or post an issue on the github repo.

**Setting up the Inventory Node:**

Press Ctrl + A to add a new node, type inventory and the node should be in your view





Once that’s done make sure to set it up correctly:

- Onload will load the items as soon as the scene loads.

- Save File path is where the path to the file you want to save the items.

* Cell size is the how big each grid will be.
* Grid height is how many grids are in the y axis.
* Grid width is how many grids are in the x axis.
* Assign whatever texture you want for the grids.

\*NOTE: I’ve added hover\_texture under grid so you can add a texture to the inventory cursor

**Setting Up ItemsDB.gd:**

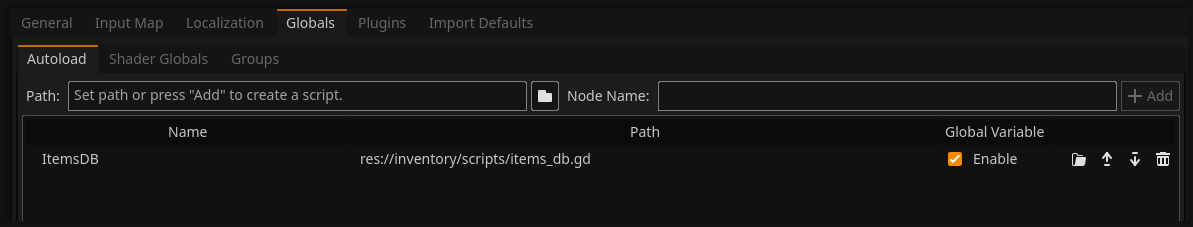
\*This script is a singleton responsible for managing the data of the items in here, you can add all the items you need but I little setup is needed first.

Go to Project > Project Settings > Globals and add the script.

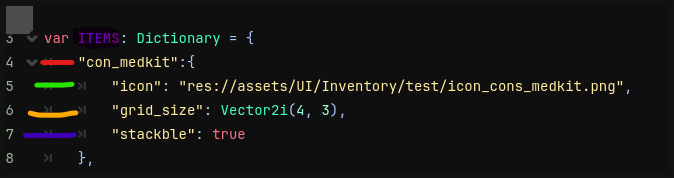
Add the items\_db.gd script by default it should be in inventory/scripts folder then give it ItemsDB as a name

**Note:** the name of the singleton must be ItemsDB because that’s how the inventory identify it in the code, you can change it if you really need to but you’ll have to change the code too, Also your path may vary from what I have so you don’t need to have the exact path

Once that’s done your Autoload should look like this (excluding any other autoload that you have in your project)



Now let’s go ahead and take a look at the items\_db.gd script itself, we will take the medkit as an example of how I added it to be used



ITEMS is the dictionary that will hold all the data of the items in your game in it there is also dictionaries each represent an item

For an example I gave the first item the ID of Con\_medkit, if later we want to add the item we would call *add\_item(“con\_medkit”)*

Icon is the path to the icon that will be displayed, just drag the image to the editor and it will do the rest

Grid\_size is well the size of the item, do keep in mind that despite the image above showing 4x3 that doesn’t mean it will be 4x3 pixels it will be multiplied by the Cell size of the inventory node

Stackable is a property for items that can be stacked like bullets or money…

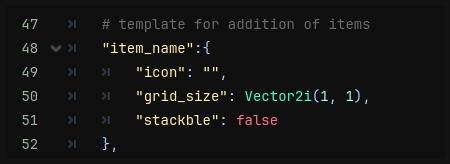
So let’s say you’d want to add a new item, first give it an ID then the Icon and then Grid\_size and so on…

**NOTE:** you are not bound to these properties you can add as many as you need, these are just to provide the basic functionalities.

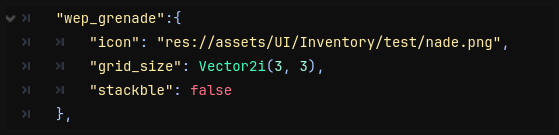
Now to add an item simply call the add\_item(id) method that exist in inventory node with the id being the one you defined in ItemsDB

Example: I’m going to add a hand grenade as an item to the inventory

1. Adding the grenade data inside ItemsDB ITEMS variable, first I opened the items\_db.gd script in there is a template that should be used for all the items



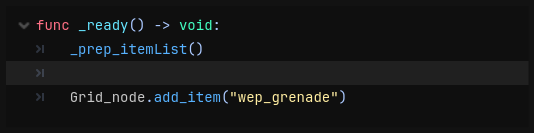
Here’s the template it should be in the bottom of the list



After filling it here’s how it will look like

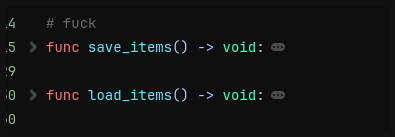
Again remember that the icon path must be correct or you will get an error

1. Now in the inventory script I will call the add\_item(id) function, MAKE SURE to call it after the inventory has been fully loaded



1. That’s it when the scene loads the grenade will pop there

**Saving / Loading:**

I’ve added the saving and loading functionality to this version and all you need to do is just call either save\_items or load\_items in the inventory node

My recommendation is to save it when your game save, it’ll be a waste of resource if you save every time you close the inventory.

**Few tips:**

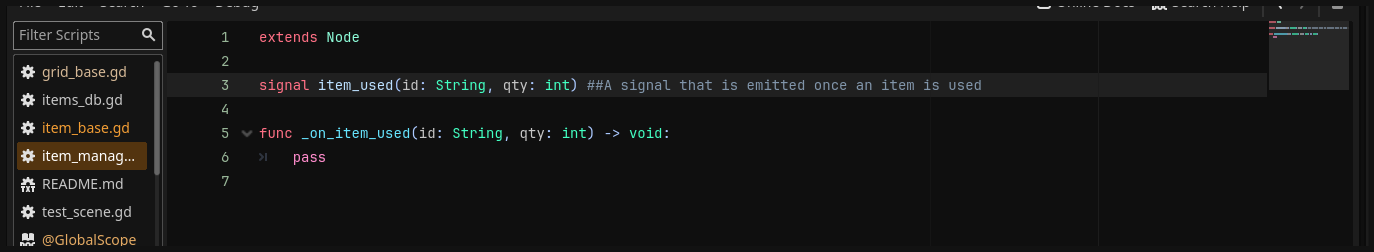
If you want to add functionality to those items in the inventory, you could use the item\_used signal but you also need to set up a script that will hold all the functionalities of said items

My idea is to have a singleton that acts as an overseer for all the items, it has a signal that gets emitted once the item is used

NOTE: My implementation is by no means the best but it is the one that is working you can change it if you have a better idea

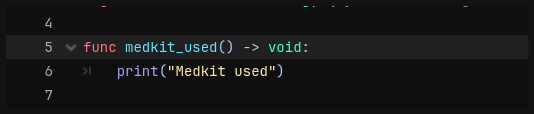
Example: making the medkit heal us

1)I will first open the ItemManager Script, it should look like this

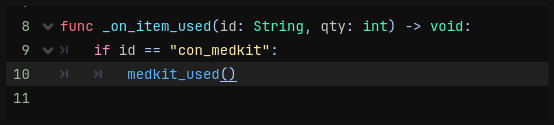


Notice the signal and the function? Those will be called every time you press the use button in the inventory you will also notice he parameters those can be used to identify what item is used and execute the corresponding function to it.

As an example I will add a function that represent a medkit being used, in this example it’s just going to print a line but for your case you’ll have to do more work like getting the reference to who will receive the medkit benefit and so on…



Now that’s Done we need to call this function, to do that we will use the item id that was passed through the signal to identify if a medkit is being used.



Now Let’s go thought what will happen when we press the use button on the medkit, once the button is pressed a signal will be emitted the ItemManager will pick up that signal and read the parameters that came with it then it will execute the \_on\_item\_used function passing the parameters it read from there you will need to give it the functionalities

While this method is fully functional and won’t cause problems it will become tedious if your game has hundreds of items as you will have to make an if statement for every individual item effectively turning you into a yandre dev, a way you could combat this is by globalizing all the functions into one, for an example you have a lot of items that alter the player or another entity, instead making a function for EVERY item you could make one that alter the attributes by passing new parameters like an array of dictionaries where each one will do something (add health/ depletes health/ add resistance….)

I hope I made some god damn sense.

**Conclusion:**

I hope this “Guide” helped you understand this system better if it didn’t well as I said before Im not good at explaining this, you can message me on discord (hexadotz) if you have questions.

Good luck on you game :)

