Performance Evaluation

# System Performance

I genuinely have no idea how to test the performance of my system, and I have very little idea about how performance can be adjusted. Every change that gets made to performance in a project of this size feels negligible, and I haven’t had experience yet working on a large project that gives me a look into how small changes can make big performance changes.

One change that I was told would help performance was moving my list of all possible items out of the PlayerInventory script and into a separate class, and I am sure I have put repetitive functions in places throughout my code (which I am too afraid to change as I don’t want to break the project on the day it is due), but beyond that I do not know what would help or hinder the performance of my system.

# Changes Required

* I initially submitted my project by just zipping up my project and its scripts, but was told through reviews that users didn’t want to have to download Naughty Attributes, which I am using in some scripts. The same goes for the new input system having to be implemented for the scripts to function. I have since started submitting this project as a Unity Package.
* My required inventory system scripts used to reference (and thus, require) my UI and player controller scripts, which I have stated aren’t required to make the system work. This has been fixed.
* My PickUpItem script didn’t work unless my PlayerController was being used, as the PalyerController keeps track of when colliders are triggered, and subsequently determining which slot an item can be added to. This has been changed by having the PickUpItem script check for the slot to add to when the function is called.
* Originally, two sets of arrays were used for each inventory slot: one for the item, and one for the quantity of that item. I have since set up a struct that can hold both, and use it in place of all of the aforementioned arrays (except for the UI, as I feared breaking my UI the day before the project was due).
* I used to have a list of all of the possible items in game attached to the player, but that didn’t make much sense as the PlayerInventory script has no need to know about every possible item in the world. I have created an ItemDatabase class to hold these items instead.