Modular Complex System Brief – Inventory Manager System

Documentation for the system by Joseph Huskey

# System Overview

I am designing an inventory system that will allow a designer or programmer to make an inventory for a roguelike or rpg game with the purpose of sorting, removing, adding, and stacking items in an adjustable list.

# Packaging

* Unity
  + A Unity package (.unitypackage)

# Quick Start – Using the Modular Complex System

## Integrating the System

Import the unity package into your Unity 2020.1.10f1 and you should be able to right click in your project view and able to access the items/inventory scriptable objects.  
  
You can make use of these objects simply by attaching them to the player character or whatever other object you need an inventory on (chest etc), design your specified objects by setting their parameters and either using the provided display method or designing your own.

## Building Upon the System

To build onto this system you may want to design more item types with their own abilities this can be done by inheriting from the ItemType class which provides some basic variables. Note when creating a new item type to add to the ItemType Enum as well as to use the asset menu code to be able to generate the object.   
  
(Example of asset menu)  
CreateAssetMenu(filename = “New [type] Object”, menuName = “Inventory System/Items/[type])

# Underlying Mathematical Operations and Algorithms

## Mathematical Operations

This inventory system does not currently make use of any mathematical operations removing and adding items is entirely based on the index/ID of the items.

## Advanced Algorithms

Currently no advanced algorithms are in the base version.

# Additional Third-Party Libraries

No third-party libraries.

# Research Material

<https://medium.com/@yonem9/create-an-unity-inventory-part-1-basic-data-model-3b54451e25ec>  
<https://docs.unity3d.com/Manual/class-ScriptableObject.html>

# Conventions

**Optionally**, list the conventions that you will adhere to over the course of the development of your project.

* C# Style
* Files will be named using pascal casing
* Base Folder > Subtype folder > Content

# Technical Risks

No risks obviously it’s perfect

# Credits

Joseph Huskey © 2022