Inventory System

An inventory and crafting system I built for my Complex Game Systems assessment at AIE.

# Quick Start

To see an expected setup for the scripts included in this system, open the UnityExample package and open the Survival scene.

# API

This Inventory System consists of 8 classes.

* 5 are required to make the system work
  + The **Item** class contains variables that each item that can be collected and/or interacted with in the system should have associated with them. NOTE: Currently, for readability, this script relies on the [**NaughtyAttributes**](https://github.com/dbrizov/NaughtyAttributes/tree/master/Assets/NaughtyAttributes) extension, which will be explained in further detail later and is included in both packages in my submission.

Ex (picture only includes variables that can be edited from Unity):

Graphical user interface

Description automatically generated with medium confidence

* + The **PlayerInventory** class stores information about all item prefabs that can/do exist in the scene(s). It also controls the player’s current inventory, as well as the ability to sort said inventory.

No picture included as only one variable is modifiable from Unity.

* + The **Recipe** class contains two arrays. It should be applied to any prefabs with the **Item** class that can be crafted by the player. Though it holds so little information, it is required for the **Crafting** class to work as intended.

Ex:

Graphical user interface, application

Description automatically generated

* + The **Crafting** class contains the logic to craft any items that have the **Recipe** class attached.

Ex:

Graphical user interface, application

Description automatically generated

* + The **ItemDatabase** class contains logic to create a dictionary of items from an array populated within the engine. This is meant to hold one prefab for each item that can be obtained in the game by the player.
* 3 are used only in the Survival scene for the purpose of showing how the inventory works
  + The **PlayerController** class controls the player’s movements and tracks input. Also uses NaughtyAttributes.
  + The **UI** classis in charge of updating the UI in the scene appropriately.
  + The **FollowCam** class makes the camera in the scene follow the player.

When it comes to the first 4 classes, the values which need to be adjusted within the engine are noted in the comments beside the variable in question. I have not listed which variables need to be assigned to objects in the scene, but Unity should tell you if it needs a reference.

# Diagram of References

Diagram

Description automatically generated

# Third Party

I used the Naughty Attributes extension for Unity in this project. It is used in the Item and PlayerController scripts, and although it is not required to make the script work, some adjustments may have to be made to the script to avoid errors if you do not use the extension. NaughtyAttributes is included in both of the Unity packages in my submission, but below are some links in case you want more information on the extension. The Naughty Attributes extension uses the MIT license.

The github with relevant scripts and links to documentation can be found here: [dbrizov/NaughtyAttributes: Attribute Extensions for Unity (github.com)](https://github.com/dbrizov/NaughtyAttributes).

The extension can be downloaded for free from the Unity asset store here: [NaughtyAttributes | Utilities Tools | Unity Asset Store](https://assetstore.unity.com/packages/tools/utilities/naughtyattributes-129996).