Inventory.cs

# Overview

Inventory is a **scriptable object** that can be added to any scene a referenced to get access to its methods of storing and getting items. Items hold objects of the class Item. Loading items into your inventory will check for an inventory JSON object - see DB documentation for more info.

# Setup

1. Drag Inventory scriptable object into your scene
2. Any scripts within the scene can now use the “Inventory” object

# Methods

## Inventory:

### LoadFromDb()

* + Returns: **void**
  + Looks for JSON object in root folder. If it doesn’t exit, it makes a blank one. If it does exist, it converts the JSON into a dictionary of items.

### Save()

* + Returns: **void**
  + Takes all current items in Inventory and adds them to JSON database

### RemoveAllItems()

* + Returns: **void**
  + Removes each item from the inventory (does not save to DB)

### Add(String itemName, int num)

* + Returns: **void**
  + Creates and Item object based off the name you pass in and how much of that item should be added. The item object is added into the items dictionary

### GetTotalItems()

* + Returns: **integer**
  + Counts how many types of items you have (not quantity)

### GetItemQuantity(String itemName)

* + Returns: **integer**
  + Checks the quantity of the item name you pass in

### GetTotalInventoryQuantity()

* + Returns: **integer**
  + Goes through all items and adds up the quantity of all of them

### GetNewId(String itemName)

* + Returns: **integer**
  + Creates a new ID number for the item name you pass in
  + **Private** because this should only be used within the Add function for now

### ReduceItemCount(String itemName, int reduce)

* + Returns: **integer**
  + Decrements the quantity of item name passed in by the integer passed in

### GetItemValue(String itemName)

* + Returns: **decimal**
  + Calculates the total value of the item based off price and quantity

### GetInventoryValue()

* + Returns: **decimal**
  + Calculates total value of entire inventory
  + This could be useful to check how much currency user has if there are multiple levels like gold, silver, etc.

## Items:

### get; set for following properties

* + Id - unique **integer** to identify an item
  + Name - **string** holding name of item
  + Price - **decimal** for how much item is worth to buy/sell
  + Quantity - **integer** on how much of that item there is