

InventoryManager

Inventory Manager takes care of most of the inventory related functionality. It uses almost exclusively public static functions to make it easy to work with.

Public Static Methods:

```
public static bool IsOpen()
```

Returns true if inventoryMain is active.

```
public static void OpenInventory()
```

Enables inventoryMain GameObject.

If inventoryPause is set to true, pauses the game by making Cursor visible and sets Time.timeScale to 0.

```
public static void CloseInventory()
```

Disables inventoryMain GameObject.

If inventoryPause is set to true, unpauses the game by making Cursor invisible and sets Time.timeScale to 1.

```
public static int AddItemToInventory(ItemSO item, int amount)
```

Called when the player makes contact with an ItemPickupable, or when using the cheat menu.

Goes through each slot in the inventory and adds the specified amount of a single item. Checks for slots of matching item type and checks for stack limits. Returns 0 if done successfully, otherwise the inventory was too full and it returns the remaining amount.

```
public static void DropItem(ItemSO item, int amount, bool removeCurrentItem = true)
```

Called when the user selects an item in their inventory and clicks outside of any UI, or when closing various menus.

Physically drops specified item and places it in front of the player. Does not actually remove anything from the inventory.

```
public static void SwapItemWithSlot(InventorySlot slot)
```

Called when the user clicks on an inventory slot.

Replaces currently selected item with the item in specified slot, or adds currently selected item onto the slot's stack if they're the same item.

```
public static void GrabItemFromSlot(InventorySlot slot, int amount)
```

Called when the user right clicks on an inventory slot.

Sets currently selected item to item in slot with specified amount, or adds to current stack if items are the same.

Also works with negative numbers to add items to the specified slot.

```
public static bool CheckItem(ItemSO item, int amount)
```

Called during simple crafting to check ingredients.

Returns true if the inventory contains at least [amount] of [item] among all slots. Useful for checking for stuff like keys or gold.

```
public static void RemoveItemFromInventory(ItemSO item, int amount)
```

Called when crafting.

Removes [amount] of [item] from all of inventory.

```
public static void RemoveItem()
```

Called from the BinItem() function.

Removes the currently held item completely.

Public Methods:

```
public void BinItem()
```

Called when Bin Button in inventory is pressed.
Calls public static void RemoveItem().