Important Information

Support and Full Documentation

- Full documentation and examples can be found at our website
- Support for this asset can be found at our Github
- Contact us at support@quantumtekhub.com for problems with the asset

Installation

1. Package Requirements

Before installation of Quantum Inventory, you need to have the Text Mesh Pro package from the Package Manager (Under Window > Package Manager) installed. After installation of Text Mesh Pro, you need to import the files necessary for Text Mesh Pro from Window > Text Mesh Pro > Import TMP Essential Resources. **However, you don't need Text Mesh Pro if you don't import the demo folder.**

2. Install

After you meet the package requirements you can install Quantum Inventory.

3. Demo Scenes

There are three demo scenes: the Inventory Demo, the Crafting Demo and the Vendor Demo, located under the Demo folder. The Inventory Demo shows how to add and remove items from a character's inventory. The Crafting Demo shows how to craft items. The Vendor Demo shows how to make transactions between two players.

Getting Started

Basics

Quantum Inventory is a small but flexible and customizable inventory system made from three separate parts: the actual inventory system with items, the crafting system, and the vendor/shop system. The inventory system handles the items that each character has. The crafting system is responsible for crafting items into a new item. The vendor system takes care of transactions made between two characters. The idea is that each character that can hold items has an inventory. Each character that can craft items has a crafting handler. Each character that can buy or sell items has a vendor script.

A character ideally could have multiple inventories, such as a main inventory a backpack, etc. Each inventory has many stacks of items potentially, with each stack representing an item and how many of them there are. This could be used to have a drag and drop slot inventory system with many stacks of each item, or one stack per item with unlimited stack capacity (programmed by having a max of 0). Inventories can also have a maximum number of stacks, or 0 for no max. Additionally, the amount of each item in an inventory is kept track of by dictionary, making it easy to find how many of an item an inventory holds.

Add/remove items to/from an inventory using AddItem/RemoveItem. Get how much of an item is in an inventory using GetStock.

Characters that can craft should have a crafting handler. This handler is responsible for putting crafting recipes into queues and crafting items, either in order or all at once. When an item is done crafting, it is automatically put into the character's inventory. A queue consists of an item and how much of it is being crafted.

Craft items using the Craft function.

The vendor system is also simple, in that each character with a vendor script has currency that can be used for trading goods. A Transaction function is used for the buyer to buy and the seller to sell the item(s).

Script Terminology

Quantum Inventory contains many custom scripts, with each outlined in the table below.

QI_Inventory - QI_Inventory stores a list of item stacks and represents an inventory, such as a backpack.
QI_ItemData - QI_ItemData stores data about an item.
QI_ItemDatabase - QI_ItemDatabase stores a list of item data.
QI_ItemStack - QI_ItemStack represents one stack of items in an inventory.
QI_Item - QI_Item represents a single item, such as a sword or healing potion.
QI_CraftingHandler - QI_CraftingHandler manages crafting queues of items waiting to be crafted.
QI_CraftingRecipe - QI_CraftingRecipe represents a list of crafting ingredients for making an item.
QI_CraftingRecipeDatabase - QI_CraftingRecipeDatabase stores a list of crafting recipes.

QI_CraftingIngredient - QI_CraftingIngredient represents a certain amount of an item in a crafting recipe.
QI_CraftingQueue - QI_CraftingQueue represent how long is left on crafting an item.
QI_Vendor - QI_Vendor handles everything that a vendor would need: currency, items, and the ability to exchange between them.
QI_Currency - QI_Currency represents any form of currency in the game.
QI_CurrencyDatabase - QI_CurrencyDatabase stores a list of currencies.
QI_CurrencyStash - QI_CurrencyStash represents a certain amount of currency.