Q INVENTORY SYSTEM

1.1.0

USER GUIDE

1 Table of Contents

Other

Q INVENTORY SYSTEM

Hawk Quan

2 About Q Inventory System

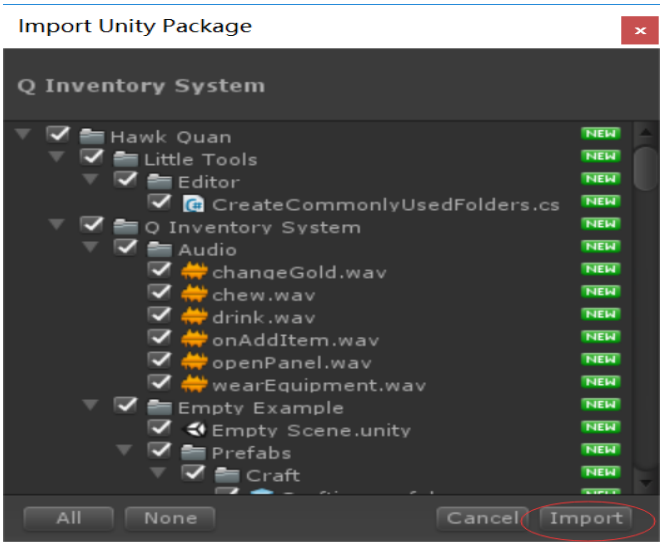
Q Inventory System is a useful and flexible inventory system which is mainly designed for 2D games.

You can easily manage **items, currencies, blueprints, attributes and other inventory things** with a quite convenient **editor extension**.

3 First Steps

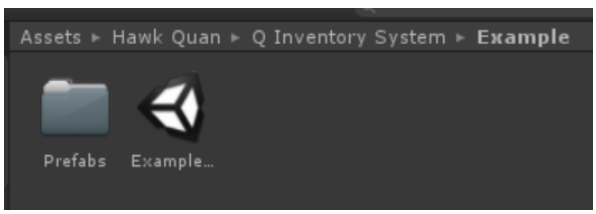
3.1 Step 1 - DownLoad Q Inventory System From Asset Store.

3.2 Step 2 - Import Q Inventory System Unity Package



Before Step 3, Make sure your tags have already contained these : **Inventory, Vendor, Craft, Equipment, Storage, SkillBar** (Weapon and Enemy are not necessary, but if you want the example scene to run well, just make sure it has.)

3.3 Step 3 – Open the Example Scene and Take a look at what it has.



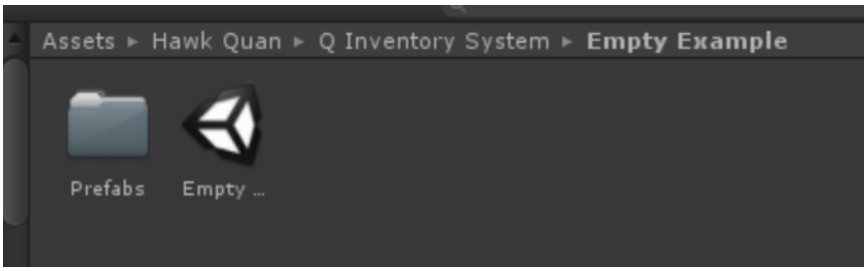
4 How to ……

4.1 Set up Inventory UIs

There are "**empty" scene** and **"empty" prefabs** which you can drag into your game scene immediately. (Just choose one of the following two)

4.1.1 Empty Scene

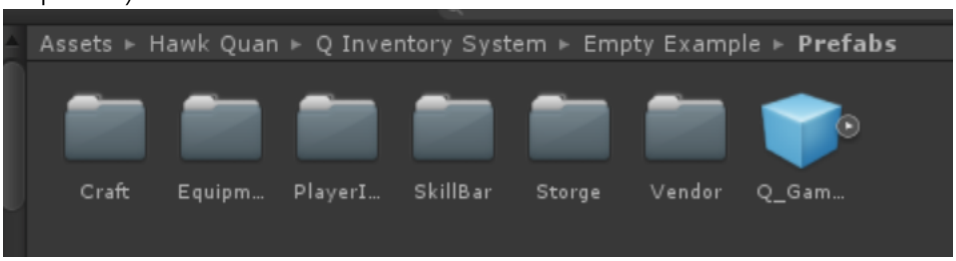
In the Empty Scene, there are only basic Q Inventory System UIs. So you can just duplicate this scene (select it and press Ctrl and D) and change its name, then you can move this one to wherever you want and start your new game!



4.1.2 Empty Prefabs

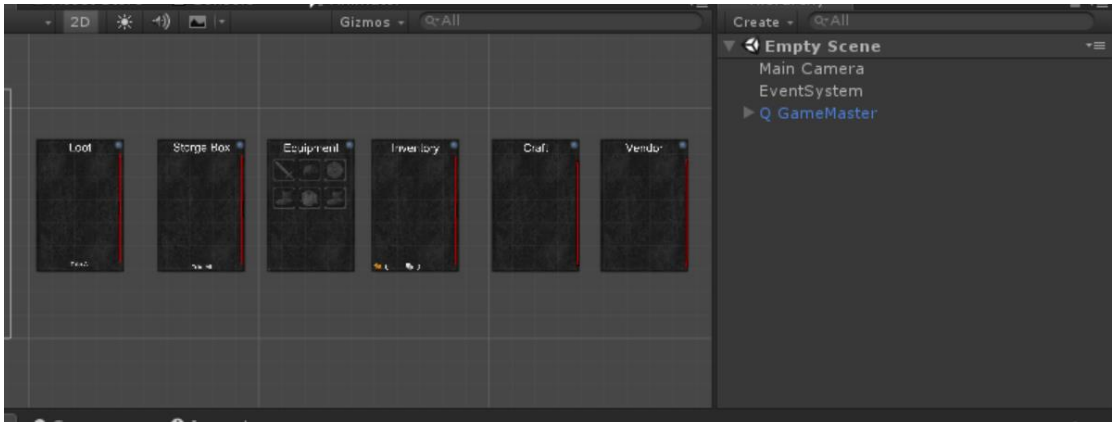
In this prefabs folder, all these prefabs are just base Q Inventory System UIs' prefabs. They don't have any Item Database, items, blueprints, attribute or currencies. There are only basic script references and some resource references (sounds, prefabs, etc) which are quite boring if you assign all of these by yourself.

So just drag the Q GameMaster into your scene. Then you can see it has an Inventory Manager, an Audio Source and an InventoryCanvas. But don't forget to create an Event System if your scene doesn't have one (or you can't interact with these panels).



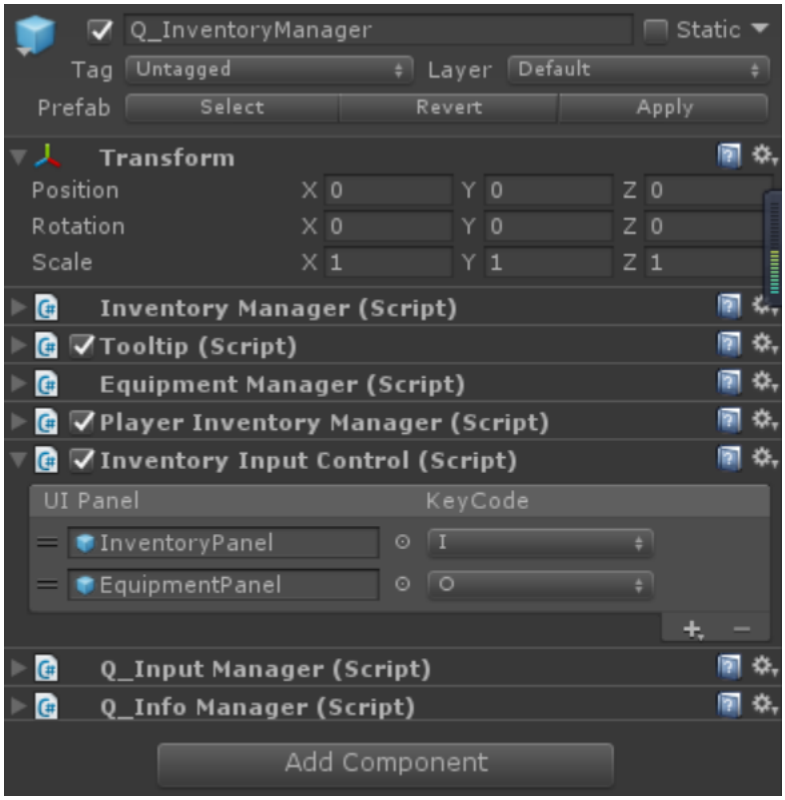
4.2 Inventory Panels Controling

Up to now, your scene has already haven the basic Inventory UI just looks like the following picture.

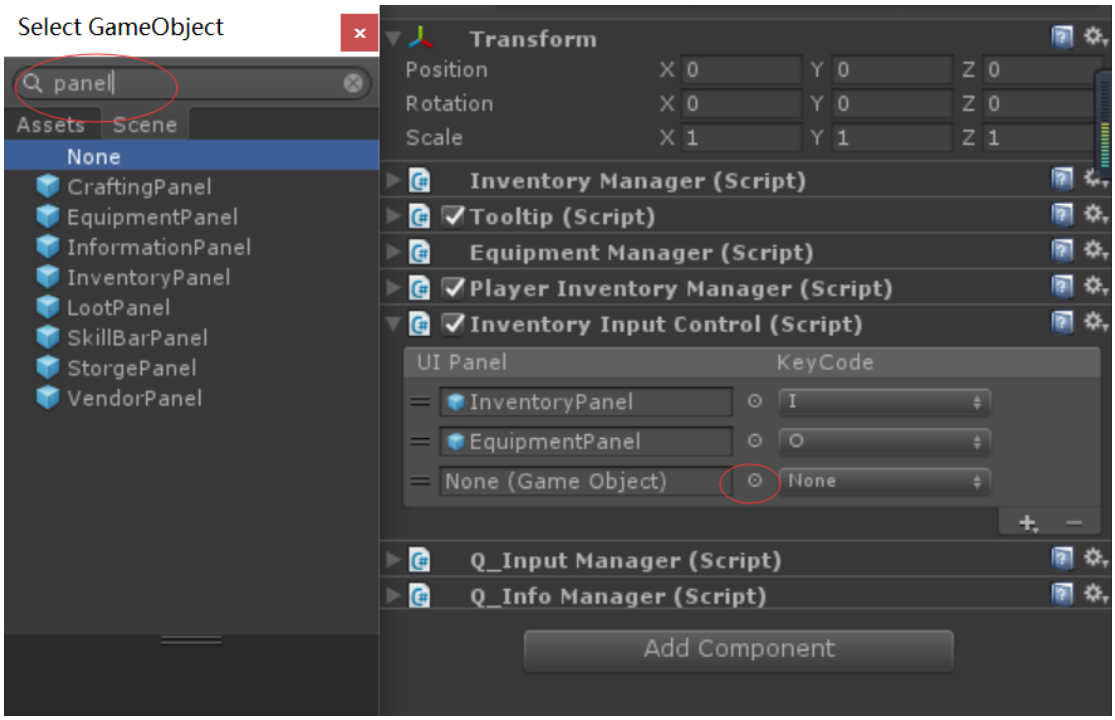


You can try playing the scene, but I only assigned two panels' controlling keys (**I** for player Inventory panel and **O** for equipment panel).

You can assign it by yourself. (**Remember to exit the playing mode**) Just find the script on Q GameMaster / Q_InventoryManager.



And click the plus. Select a new panel (panel not its parent) and choose the key.



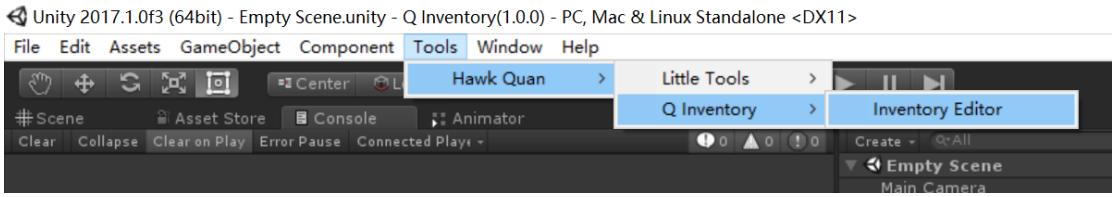
Then play the game again, see if you can control the panel you assigned. (Usually the Craft, Vendor, Storage and Loot panels are triggered by something, so I didn't assign them, but things are different from different games, so do what you want)

4.3 Create A New Database to Manage Items, Blueprints Attributes, Currencies

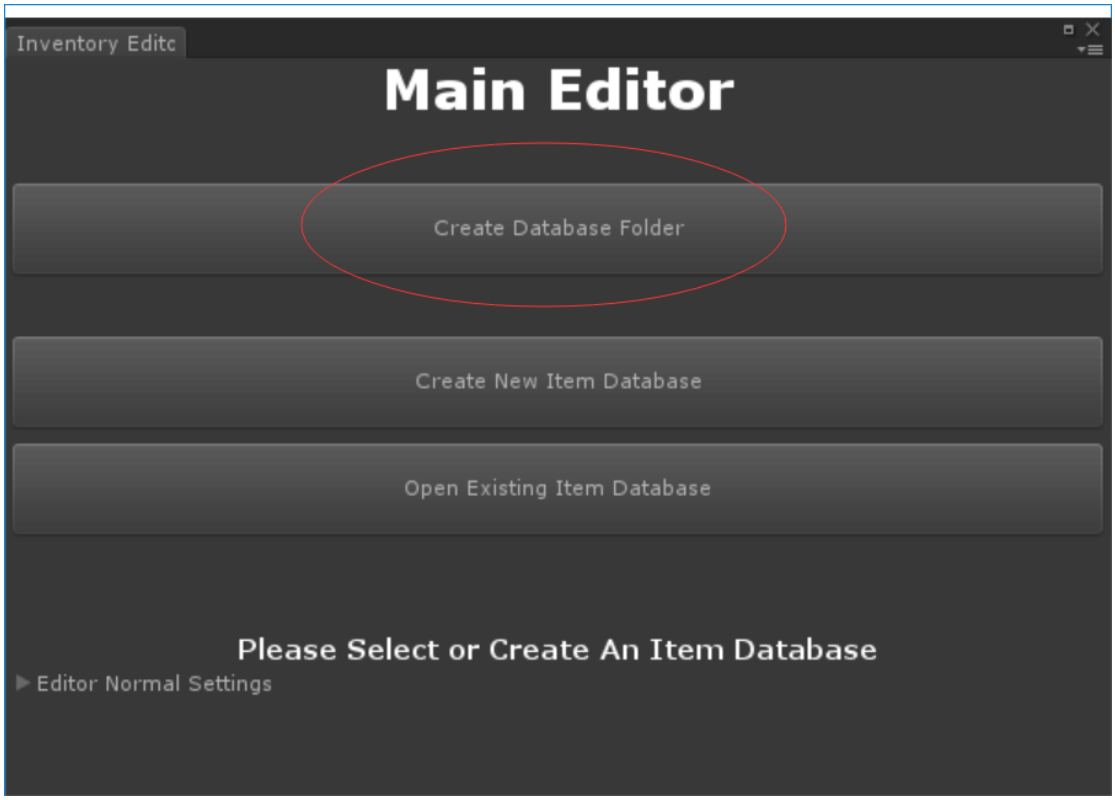
After the UIs, we need to create a database to store all the inventory data. So let's begin.

4.3.1 Create the Inventory Folders

At the top of Unity, you can find the tools, then open the Inventory Editor



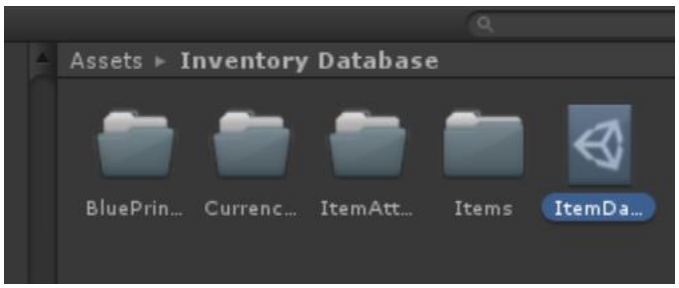
Now you can see the Main Editor like this.



Click the "**Create Database Folder**" Button and select a path in your assets to place the folders. Wait a while and then you can see the debug messages.

4.3.2 Create the Inventory DataBase

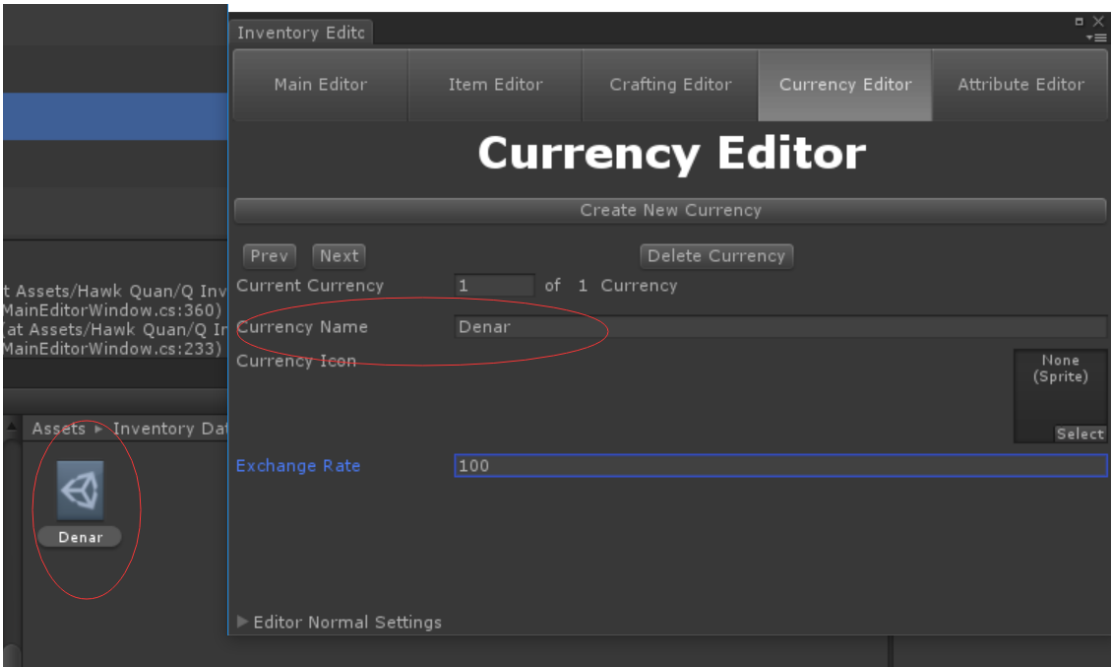
After that, click the "**Create New Item Database**" button and you can find a new **.asset file** named **ItemDataBase** was created in the folder. And the Editor Window changed as well. Now we can create items.



4.3.3 Create Our First Currency and Currency System introduction

Click the **Currency Editor**, you will see the label "Don't have any Currency", so just click the **create** button, then we have our first currency, name it whatever you want, and you can find it in the "**Inventory Database/Currencies**" folder. Once you change its **name** in the editor, the **file name** is changed as well. (**Attention ! ! ! Better**

not change the .asset file directly in the folders, just editor them in the awesome editors.)



4.3.3.1 Currency Icon

It doesn't matter if you don't assign an icon for a currency, cause I didn't write any functions about the icon, but if you assigned it, and want to use it in somewhere in your own code, you can find the icon here.

```
public class Currency : ScriptableObject {
    public string currencyName;
    public Sprite icon;
    public float exchangeRate;
}
```

4.3.3.2 Exchange Rate

If you created **multi currencies** and used them in your game, there might need a relation between two different currencies. So that is what **Exchange Rate** is.

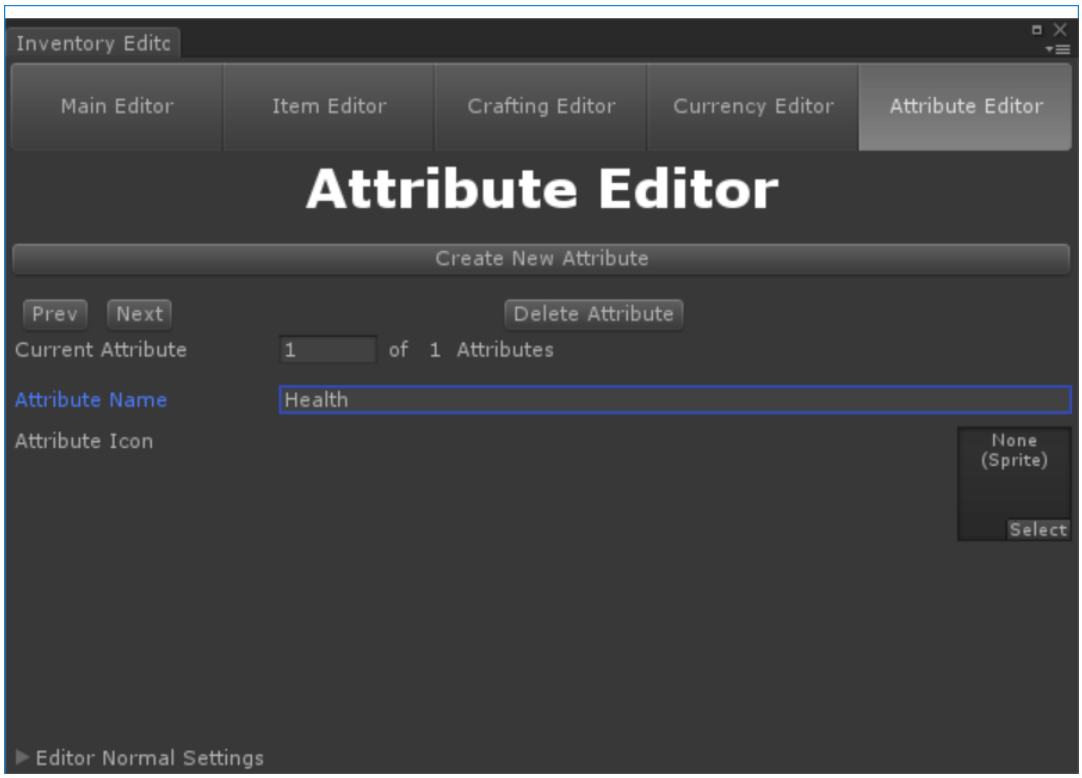
The default calculating base value is one. (Better set all the rates to integers)

For example, if I create two currencies named "Gold" and "Silver". The former's rate is 10 and the latter's rate is 1. So if I have 9 silvers, that's 9. But if I have **more than 10 silvers** like **16**, it will automatically convert into **one gold and 6 silvers**. So just **take care** of the rates when you manager the currencies.

- **make sure the currency of the highest value will be in the first place. Just like a descending order.**

4.4 Create Our First Attribute

The same as currency, just create it.



4.4.1 Attribute Icon

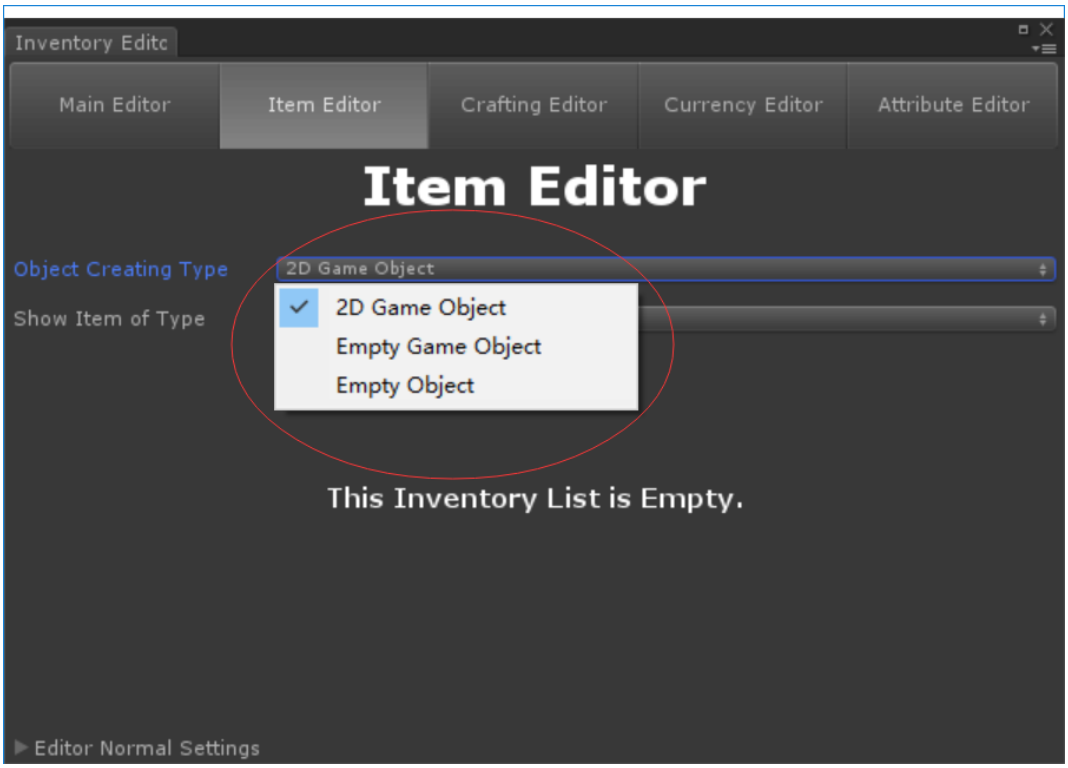
Same as currency icon

```
4
5       [System.Serializable]
6     public class ItemAttribute : ScriptableObject{
7         public string attributeName;
8         public Sprite icon;
9     }
10
11
```

4.5 Cteate Our First Item

After these, we can create an **item** which we can use, equip or just regard as a material.

So, first let's come to Item Editor and **before clicking the create button**, you can choose the **object type** which will create item as a **prefab** in the item folder. (Default choice is 2D Game Object. And actually I have already written some 3D Game Object creating codes, but to be honest, I'm not quite familiar with 3D games in Unity and don't know if these codes are useful. So I decided to add them later)



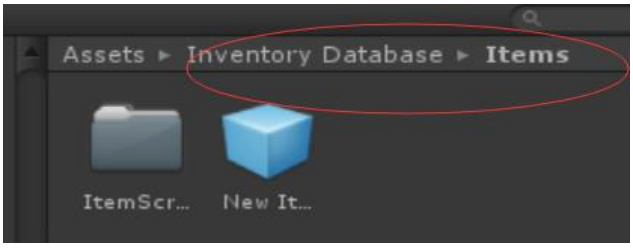
Then you can assign the items' information.

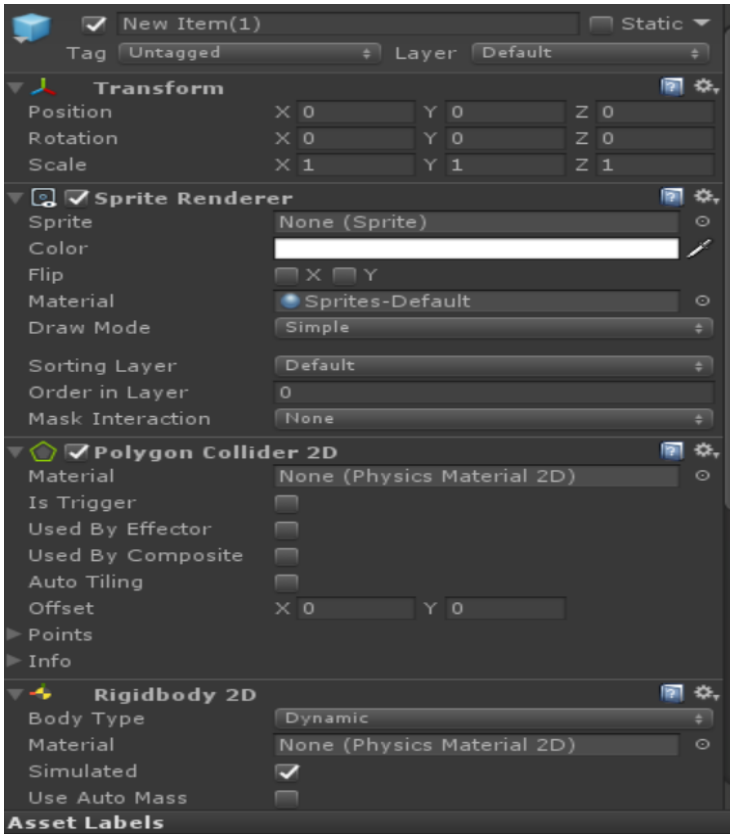
4.5.1 Item ID

Each item has a **uniqe ID** after its creating, but you can change its order with the **reorder items** function (I'll introduce it a moment later).

4.5.2 Item Icon

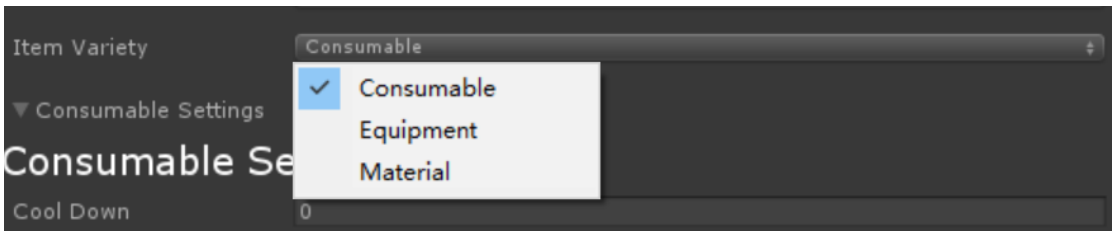
When you create the 2D gameobject, editor will add a **Polygon Collider 2D** to the its prefab(You can choose whether add the Rigidbody 2D). And once you **change the icon**, editor will **remove the old Collider and add a new one** to suit the new icon. But sometimes you may think Polygon Collider 2D is not good enough, you can find the prefab in the **item folder** or press the **show item button** and add what you want to the game object.



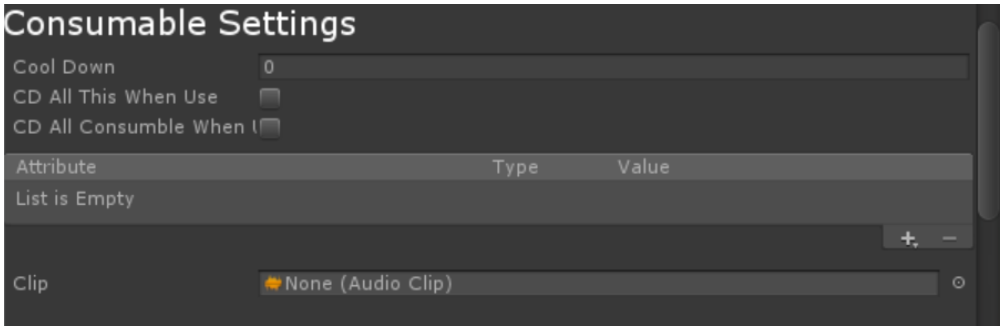


4.5.3 Item Varity

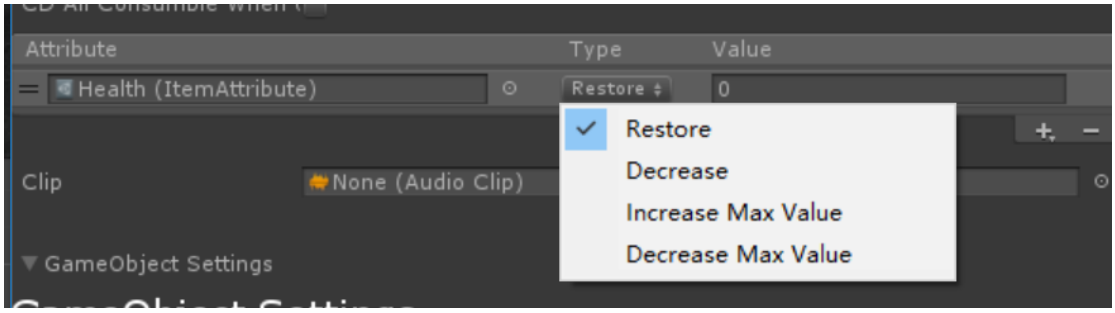
You will have three kinds of varities to choose: Consumble, Equipment and Material. (If you want to add more type, you can just open the Q_InventoryEnum.cs file and there you can find all the enums)



4.5.3.1 Consumble

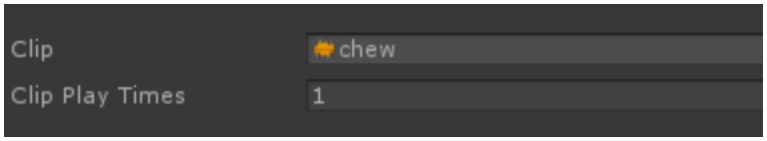


- **Cool Down : float**
-
- **CD All This When Use** : this means once you use an item, the same items in your Inventory and skillbar will come into CD state as well.
-
- **CD All Consumble When Use** : this means once you use an item, all of the consumble items in your Inventory and skillbar will come into CD state (CD time is as long as the used item's CD Time)
-
- **Attribute :** Just click the plus and you can choose the attribute you create in the Attribute Editor

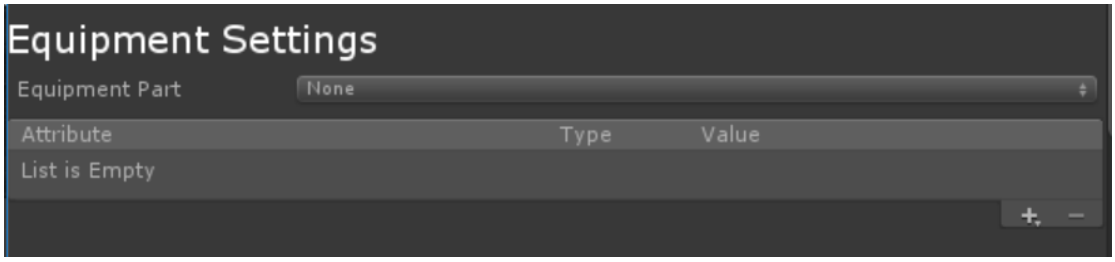


Then you have four types to choose and set the value to what you want

- **Clip :** The clip you assign here will be played when you use the item.

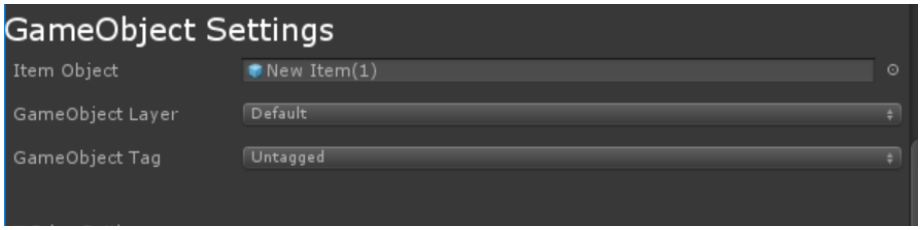


4.5.3.2 Equipment



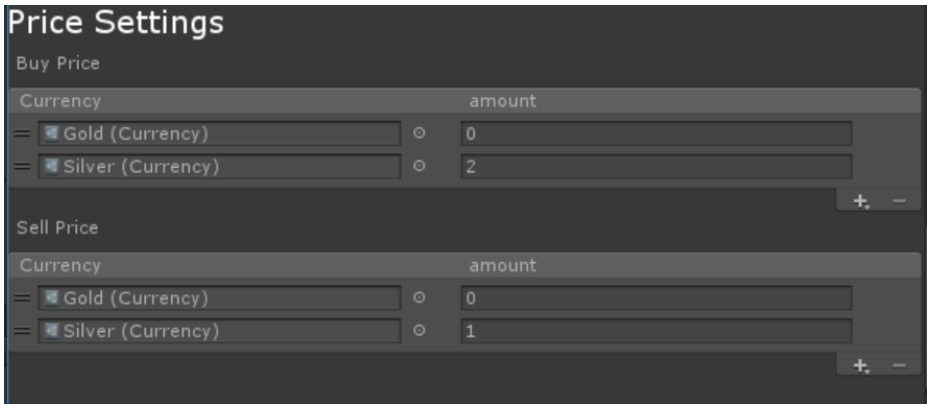
- **Equipment Part** : you can add more in the **enum file** mentioned in the Consumble.
- **Attribute** : Same as Comsumble's attribute, but better not set its type to restore.

4.5.4 GameObject Settings



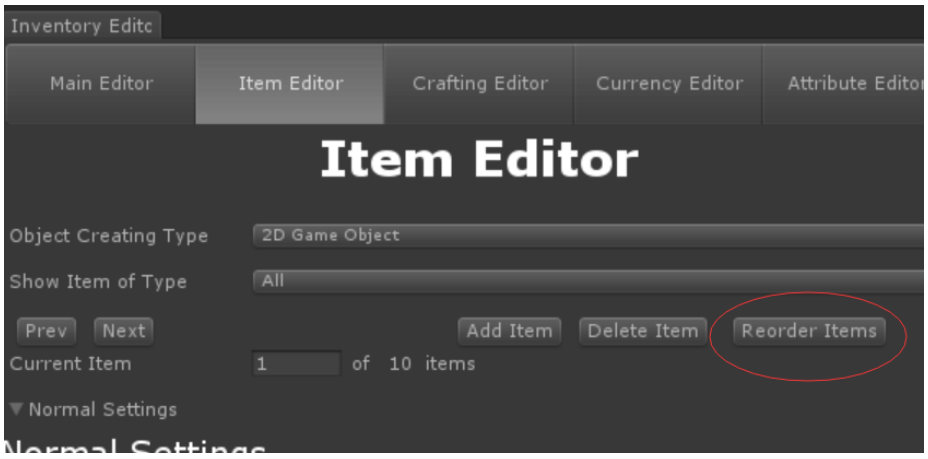
- **Item Object :** The **prefab** created when you create a new item. If your creating type is **Empty Object**, this will not be added automatically. So **remember to assign its reference** after you drag something to the object and make it a gameobject.

4.5.5 Price Settings

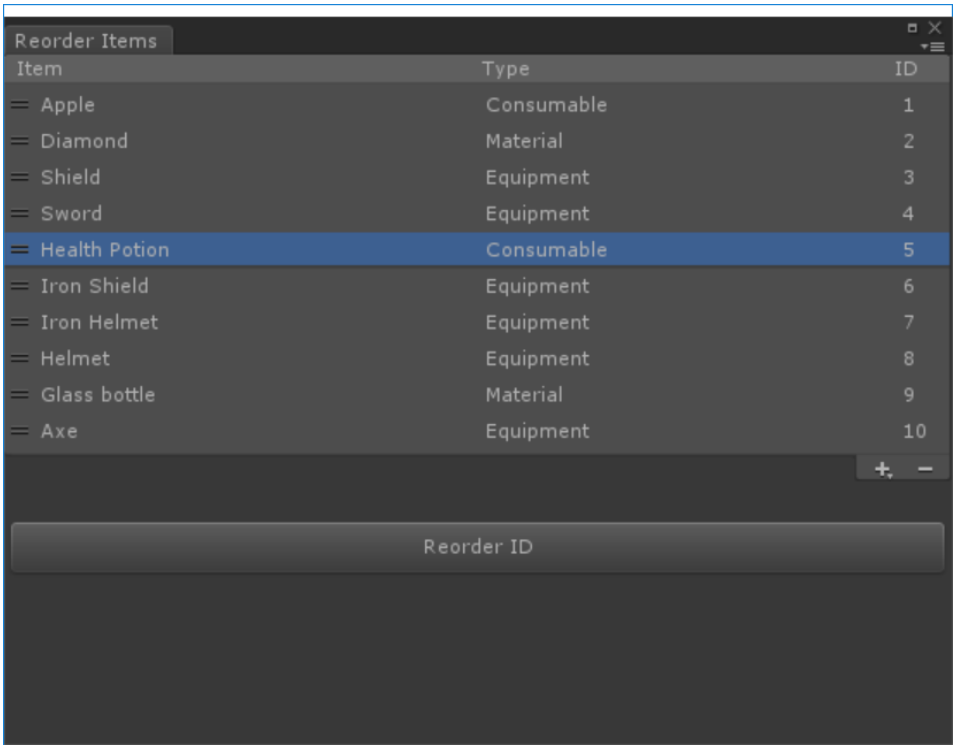


- Just remember when you click the plus, it will add **all the currencies** into the list, so when you create currencies, make sure the currency of the highest value will be the in first place. Just like a **descending order**.
- And even if you don't the currency with a higher value, just **set its amount as 0, don't delete it**. Just like the Gold in the picture.

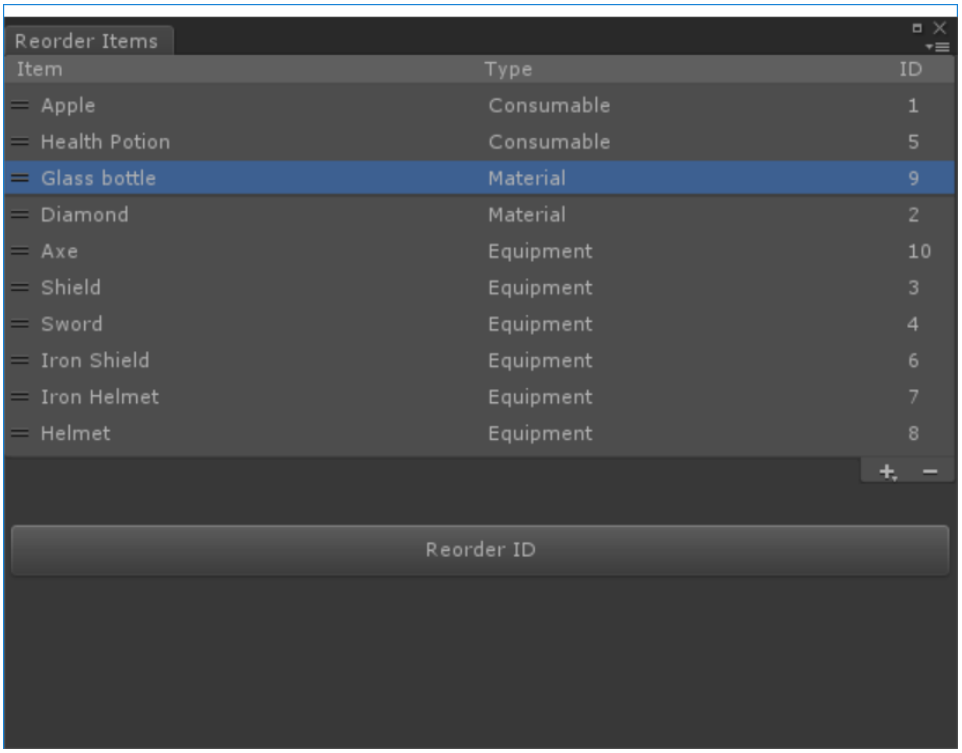
4.5.6 Reorder Items



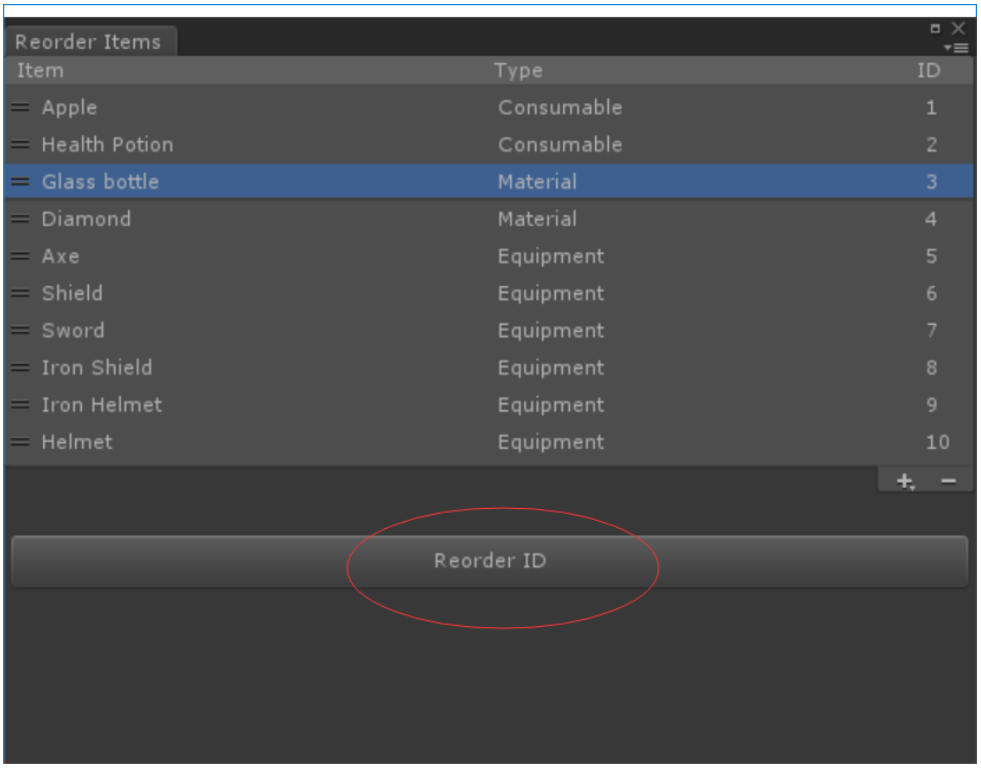
When you create tons of items and find the Item List becomes a big mess, just **press the “Reorder Items” Button.**



Here you can **drag your item** to the position you want to make them in the order you want.



But then you will find the IDs become a mess instead. Don’t worry, just **press the “Reorder ID”**, and everything will be in a good order now.

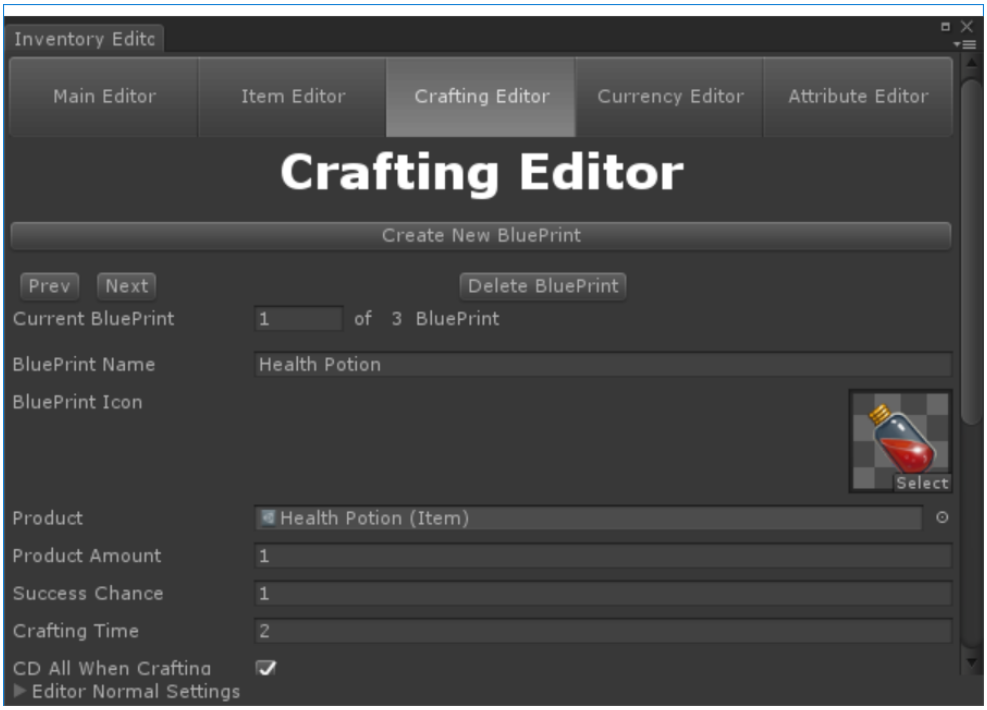


Here you can also use as a place to **look all the items** with brief information.

(The plus and minus buttons' functions **are banned** by me, so don't worry if you press them by accident.)

4.6 Create Our First BluePrint

Just create it and then you can find it in the BluePrints folder.



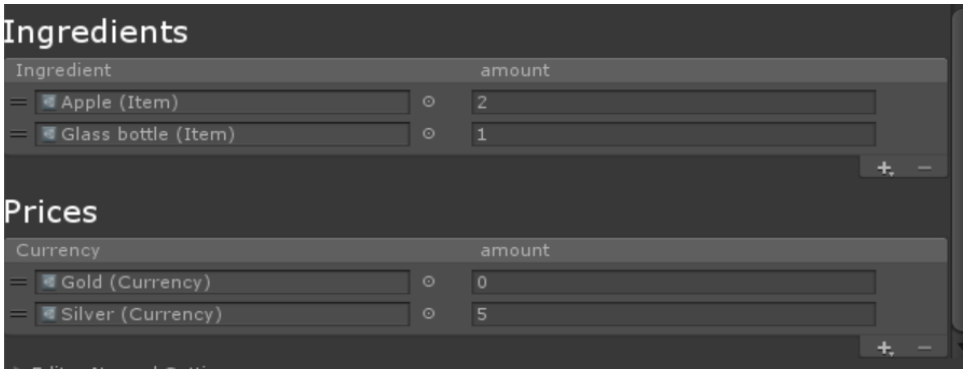
4.6.1 BulePrint Icon

If this icon is null, the craft panel will display the **Product item's icon**.

4.6.2 Product

- **Product :** The item you want to craft.
- **Product Amount :** The amount you want to get after crafting successful.
- **Success Chance :** Float.
- **Crating Time :** Float.
- **CD All When Crafting :** This means when you are crafting this item, you can't craft another item at the same time.

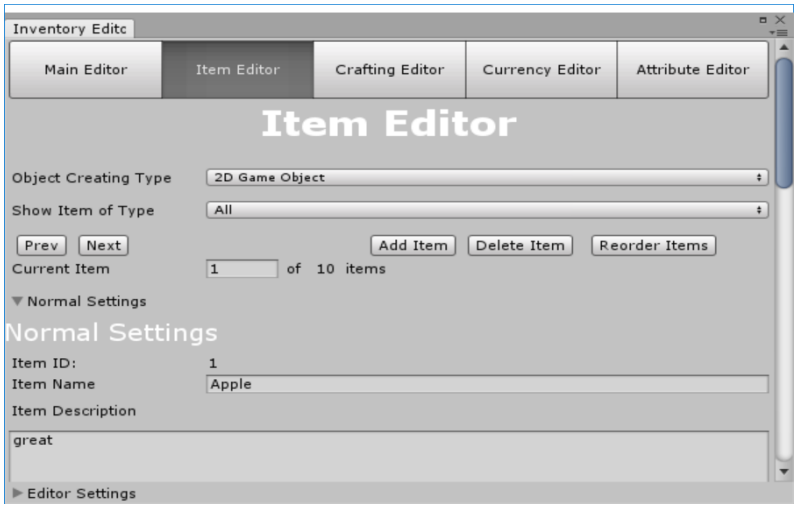
4.6.3 Ingredients and Prices



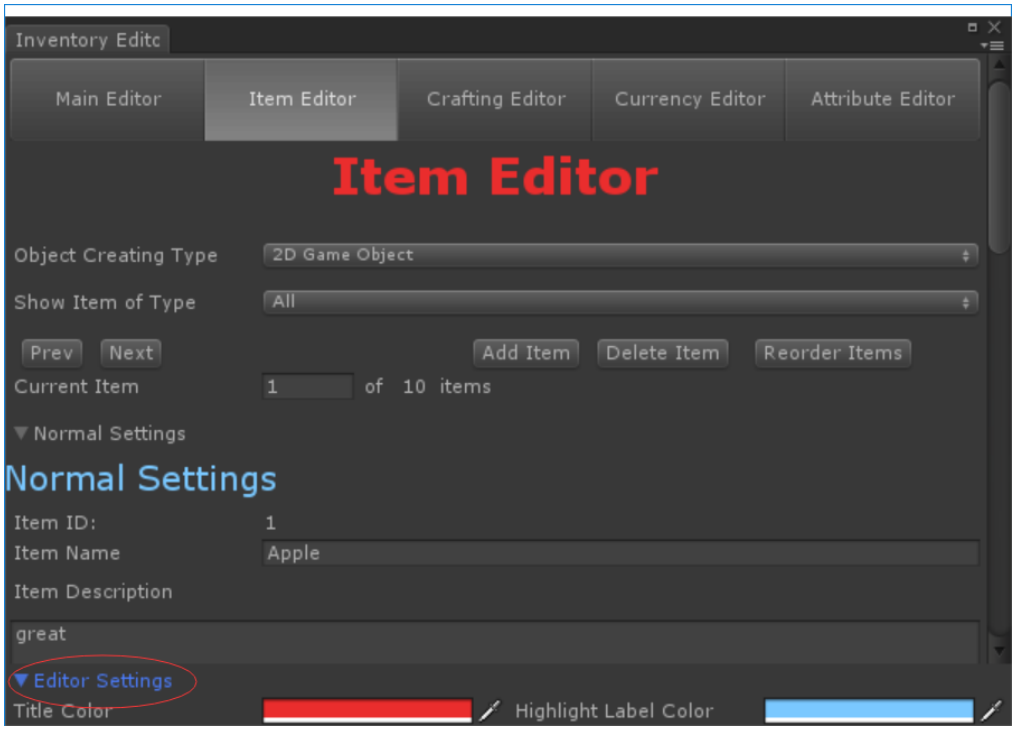
- **Prices :** When you craft something in the a smith's shop, you might need to pay some service charge, not only the ingredients, and that's it.

4.7 Editor Settings

If your Unity Editor Skin are not personal, you may find it's **hard to see the title and highlight label with the bright background.** Or just for some reason, you **get tired of the boring white title.**



Then you can go to the editor settings and change the title color and highlight label color to whatever you like.

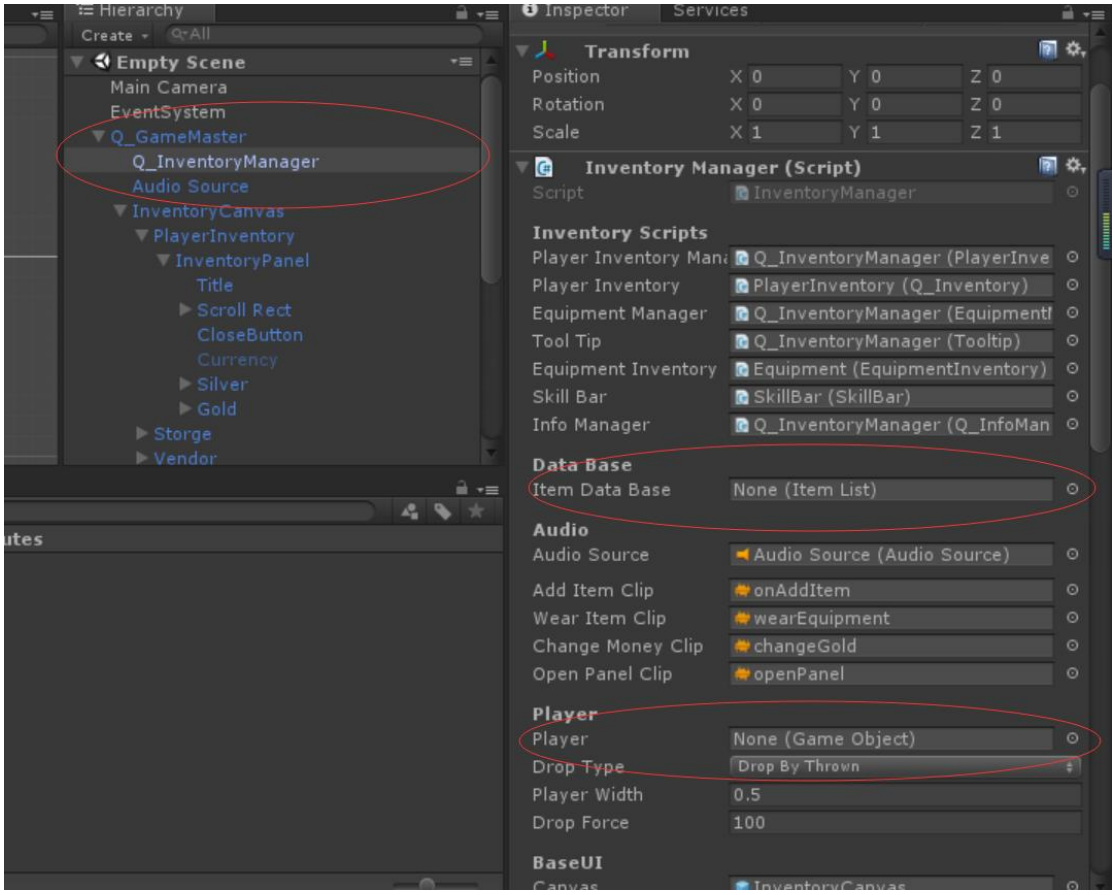


(The Editor setting data is saved in the Inventory Database/Editor folder)

5. Assign Some References

5.1 Database and player

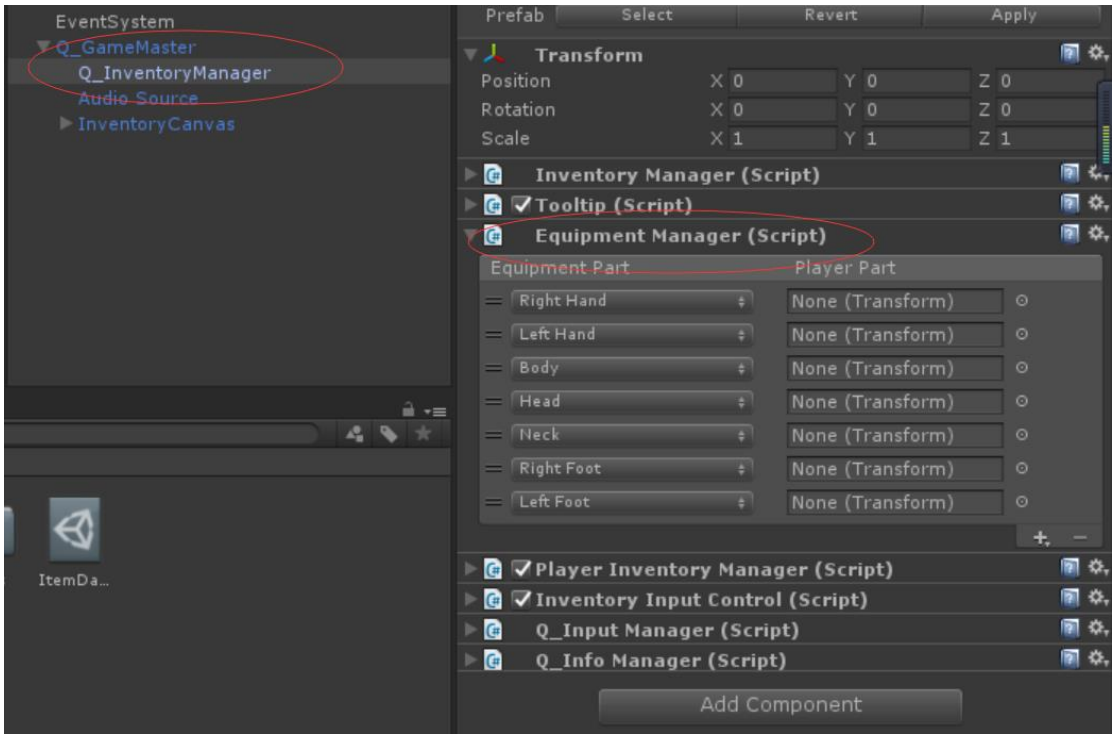
First of all, we need to assign some references to our scripts. Find the **Inventory Manager** and drag or choose the **Item Data Base** and **player** to the script. (A trigger is needed on player)



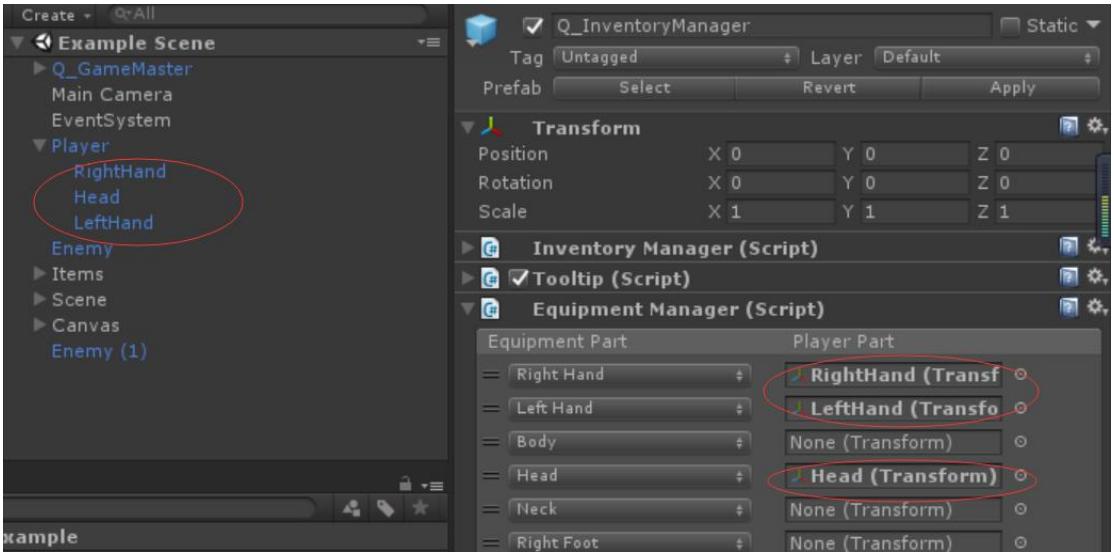
Did you notice the settings under the player? That's the **Item Drop Settings**, I'll talk about it moment later.

5.2 Equipment Part

If you want your player really to **wear some equipment items** instead of just changing some boring attributes' values in the character panel. Just assign the references in the **Equipment Manager.**

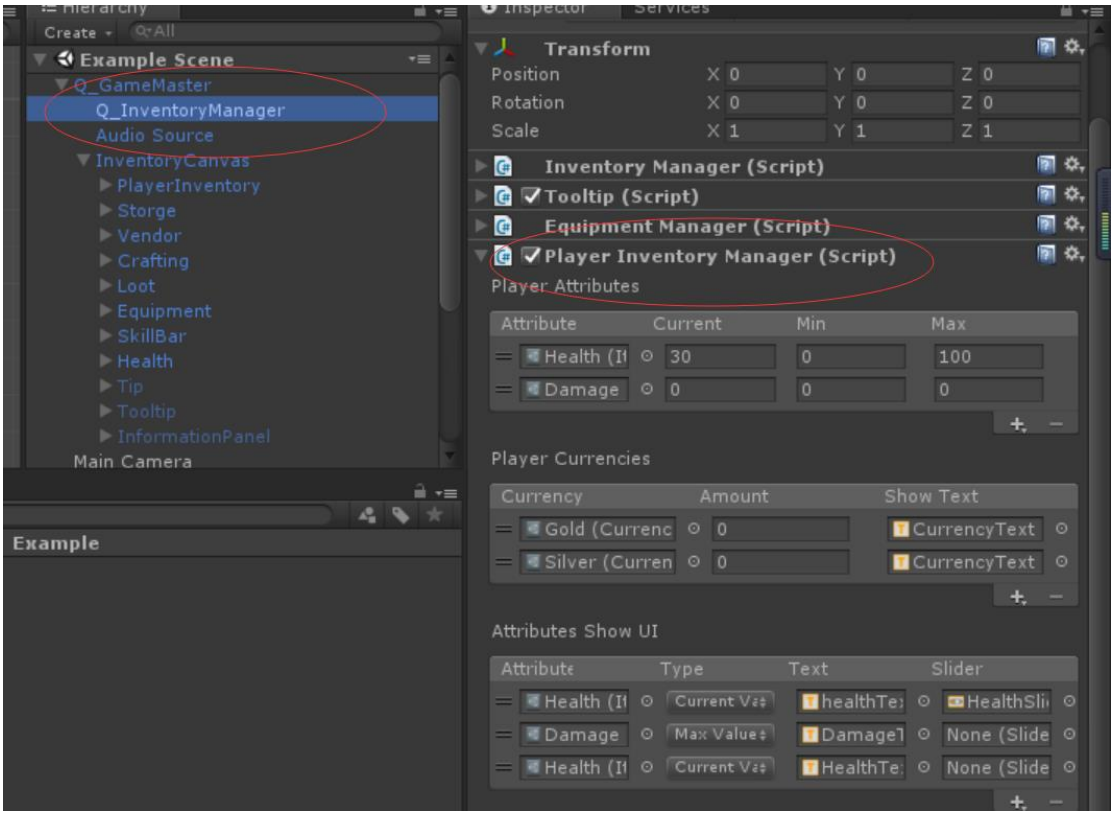


The player part is the place you want the equipment item to spawn on the player when wear the item. (So I highly recommend you to create some empty gameobjects as your player's children to place the equipment items. Just like the player in the example scene!)



5.3 Player Inventory Manager

First, just find the Player Inventory Manager.



5.3.1 player Attributes

This setting means the **attributes** you want your player to have, like **health, strength, damage**, etc.

- **Current(Float):** This means the attribute your character actually has. For example, Health : 30/100, and current value is the number 30.
-
- **Min(Float):** This is the lower limit of the current value.
-
- **Max(Float):** This is the higher limit of the current value. For example, if you use a health potion when your state is Health : 100 / 100**,** your health will not restore any more.

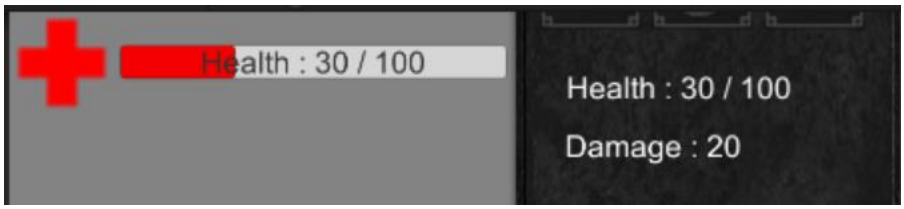
5.3.2 Player Currency

This is how much money your player has, even if your player owns nothing, just **remain the currency** and **set their amounts** to 0. The **the currency order** must be in **descending order** as I mentioned before.

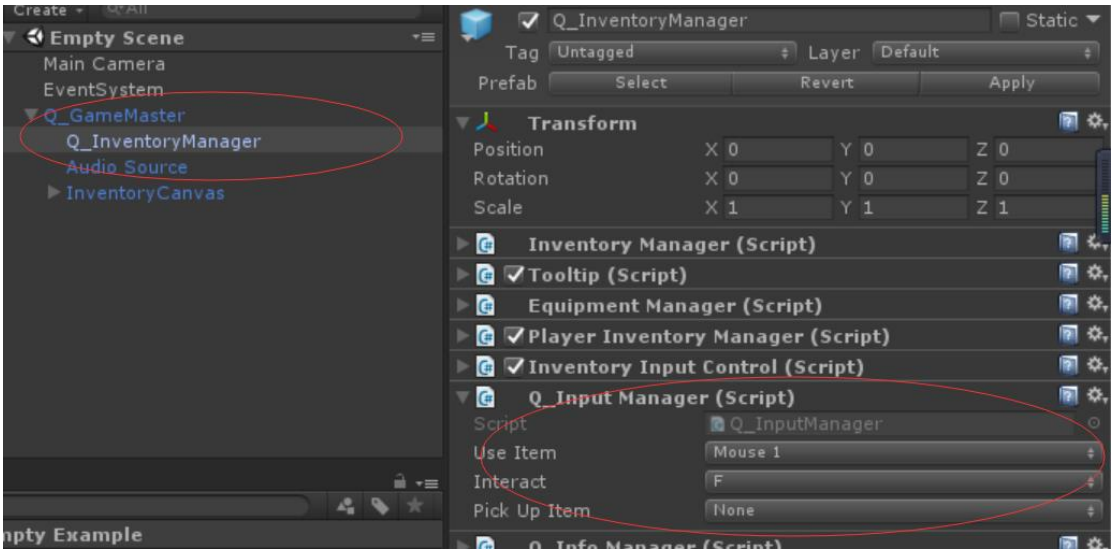
- **Show Text:** If you assign a text here, when you start the game, the text will show the amount of the currency.

5.3.3 Attribute Show UI

Here you can assign the **attribute value** you want to show in your game using a **text** or a **slider** or **both**.



5.4 Q_Input Manager



Here are some keys' settings.

- **Interact:** open the panel and etc.
- **Pick Up Item:** If it's none, the player must pick up the item on trigger enter. Or player will pick items up when you press the key while on trigger stay.

6. Add Item to Game

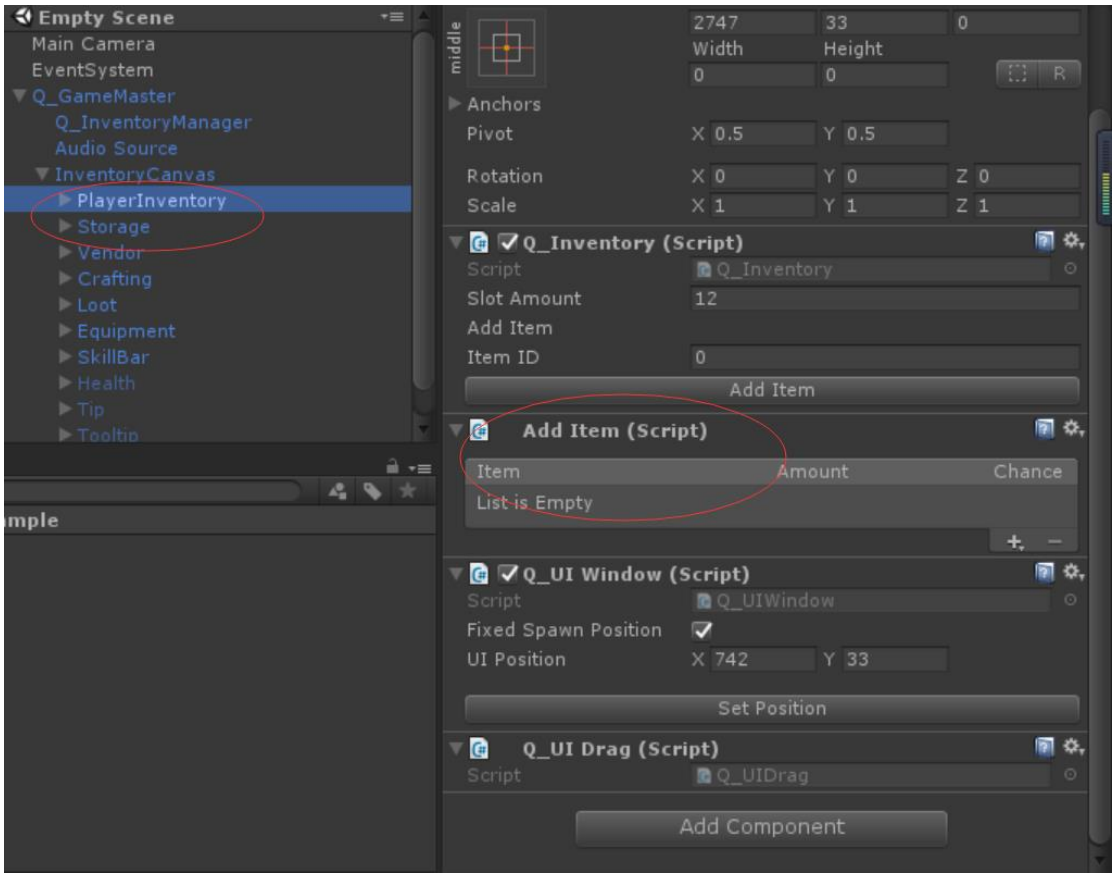
6.1 Drag Items into scene immediately

If you want to add items to your game scene directly, just drag the item into the scene.

6.2 Add Items to Inventory & Storage & Loot

(Better take a glance at the **Q_inventory, Crafting, Vendor, Storage components** first)

Find the **PlayerInventory** or **Storage** or **Loot** in the **InventoryCanvas**. All of them have the **Add Item** components.

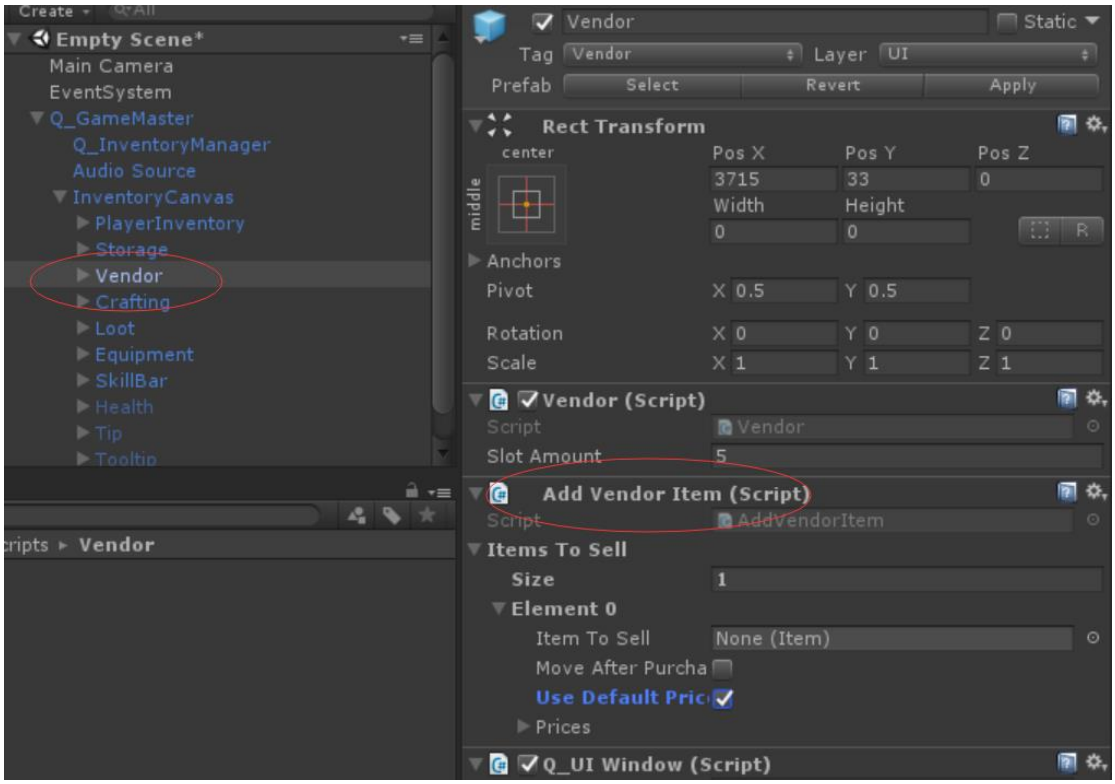


Click the plus and add your items here. (Items will be added at the start of playing mode)

- **Chance(Float):** This one means the chance it can be added to the inventory. (between 0 and 1)

6.3 Add Items to Vendor

Just find this.

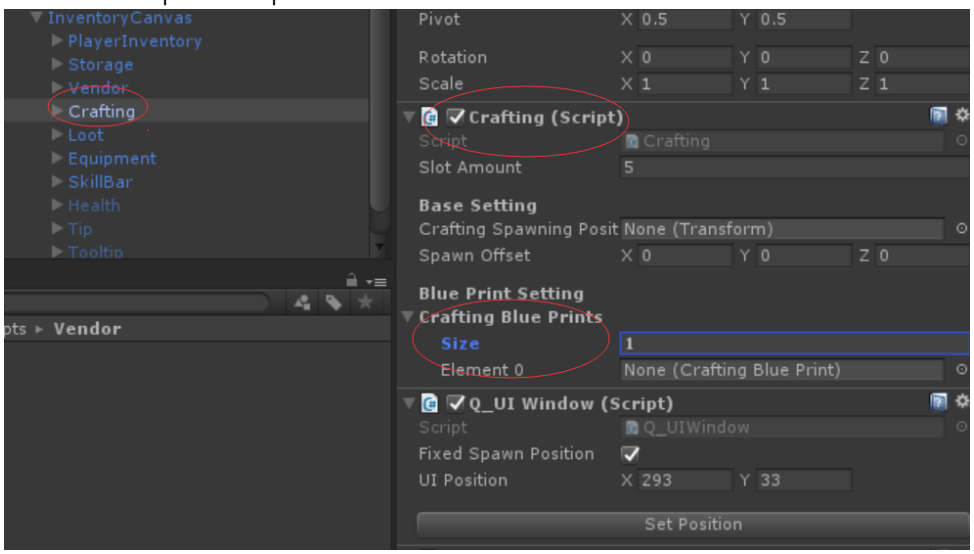


- **Use Default Price :** If this is true, it will use the buy price you assigned in the Item Editor, or you can set it as false and assign the prices.

- **Prices :** Just remember to keep the order.

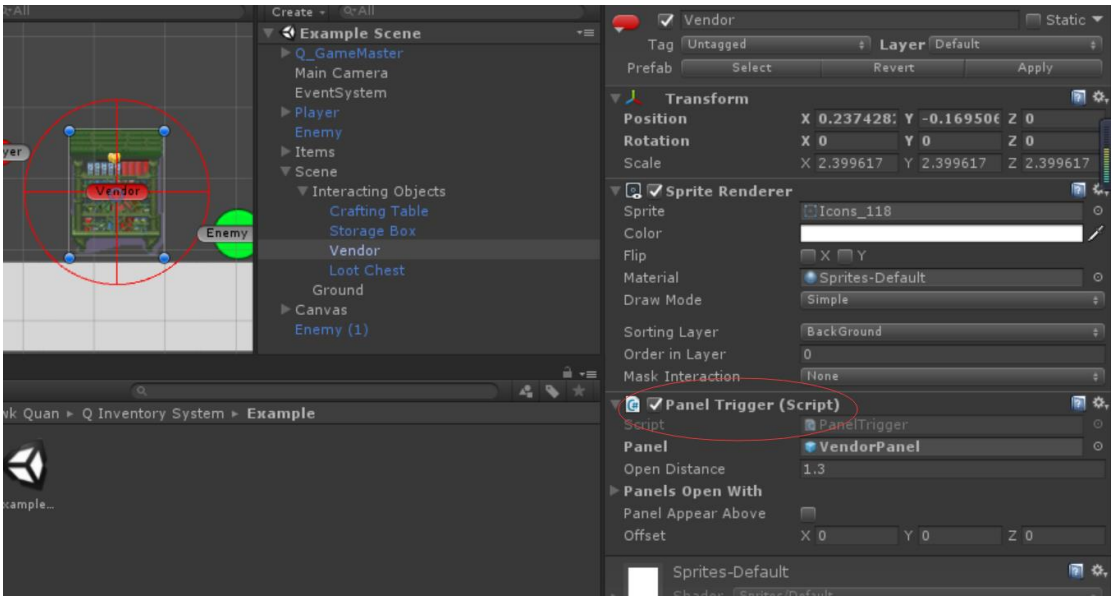
6.4 Add Blueprints to Craft Panel

This one is quite simple



6.5 Assign the Trigger to open these panels

Choose the gameobject you want and add the **Panel Trigger** on it.
For example, the vendor



The gameobject actually don't need a real collider, the player trigger it by the distance and the interact key.

- **Panel :** The panel you want to open when player enter the trigger range.(Panel not its parent !)
- **Open Distance :** It decides the trigger range (A sphere and its radius is this, here is gizmos to help you assign it)
- **Panels Open With :** This one is used, for example, when you open the vendor and you want to open the player's inventory as well to see if you have enough money.
- **Panel Appear Above :** This one can make the panel opened right above the gameobject, but sometimes right above is not good enough, so here comes the offset.

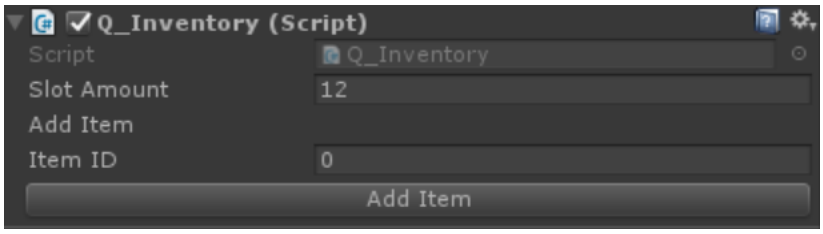
6.6 Play the Scene to See If It works

If you haven't write the move script for player yet, you can use the **Q_SimpleMove** to simply test the scene, but remember that this move script is **not good enough** to use in your game, so write your own script later.

7 Components

1.Q_Inventory

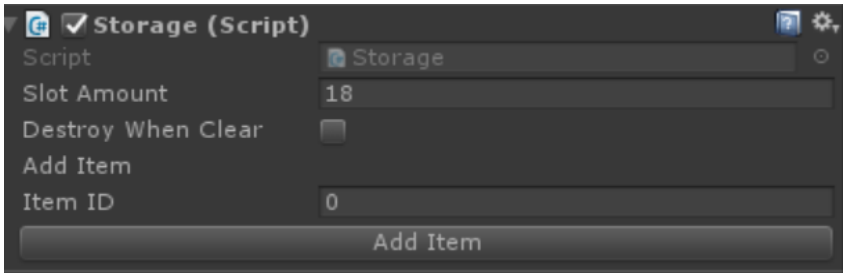
This one is used on playerInventory.



- **Slot Amount :** Set the amount of the slot.(Better larger than the amount of items to add)
- **Add Item :** When in the play mode, input the id and press the button to add items to test the inventory.

2.Storage

This one is used on Storage or Loot

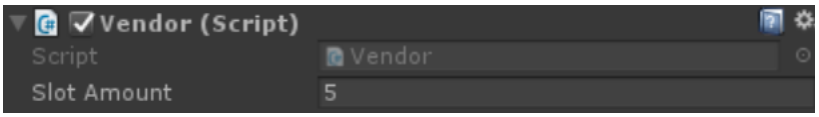


- **Slot Amount :** Set the amount of the slot. .(Better larger than the amount of items to add)

- **Destroy When Clear :** Destroy the panel when it's clear, then player can't open it again. (often used on loot)
- **Add Item :** When in the play mode, input the id and press the button to add items to test the Storage or Loot.

3.Vendor

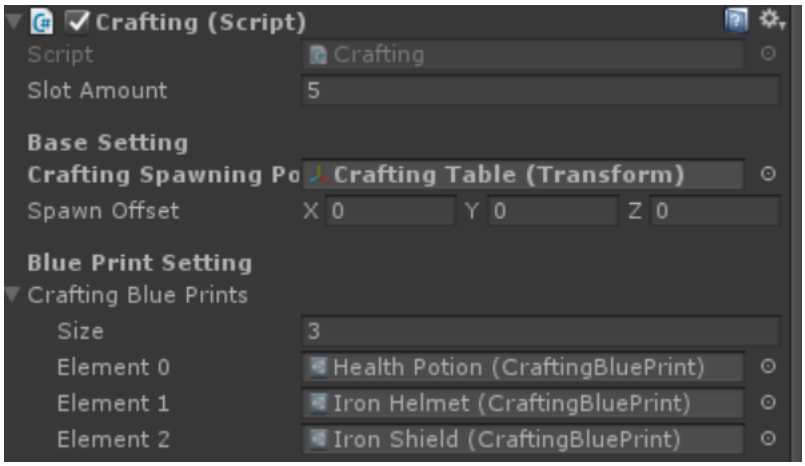
This one is used on the Vendor



- **Slot Amount :** Set the amount of the slot. .(Better larger than the amount of items to add)

4.Crafting

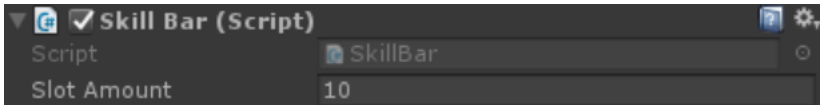
This one is used on the Craft



- **Slot Amount :** Set the amount of the slot. .(Better larger than the amount of items to add)
- **Crafting Spawning Position and Spawn Offset :** The position the products spawned when crafting successfully.
- **CraftingBluePrints :** Add new blueprints.

5.SkillBar

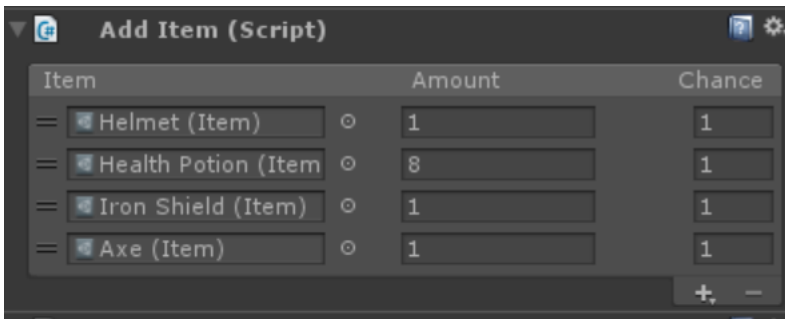
This one is used on the Skillbar



- **Slot Amount :** Set the amount of the slot. .(Better larger than the amount of items to add).

6.Add Item

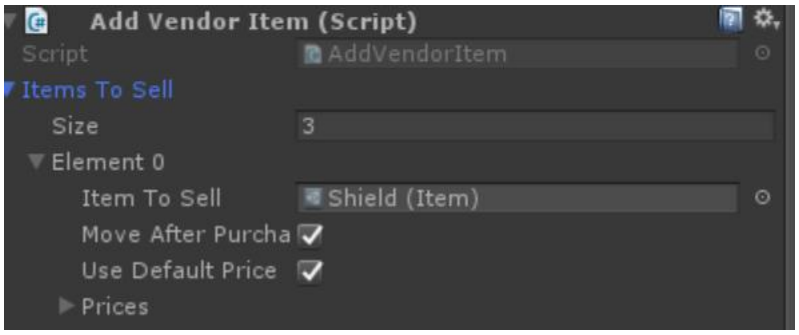
Used on the playerinventory, storage or loot to Add new Items.



- **Chance(Float):** This one means the chance it can be added to the inventory. (between 0 and 1)

7.Add Vendor Item

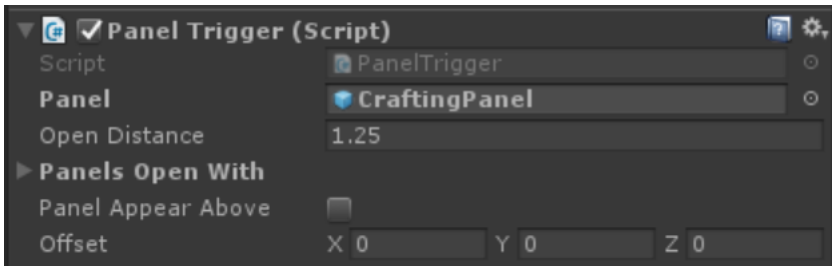
Used on the vendor.



- **MoveAfterPurchase :** Remove the Item on the vendor After purchasing.
- **Use Default Price :** If this is true, it will use the sell price you assigned in the Item Editor, or you can set it false and assign the prices.
- **Prices :** Just remember to keep the order.

8.Panel Trigger

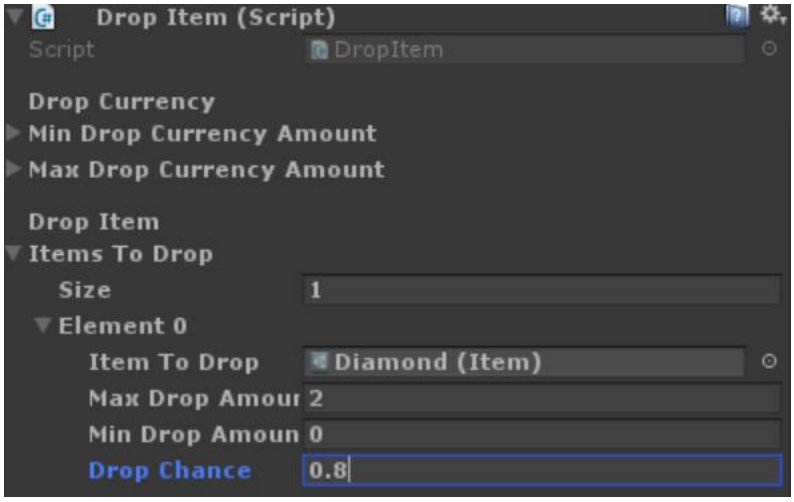
The gameobject actually don't need a real collider, the player trigger it by the distance and the interact key.



- **Panel :** The panel you want to open when player enter the trigger range.(Panel not its parent !)
- **Open Distance :** It decides the trigger range (A sphere and its radius is this the distance, here is gizmos to help you assign it)
- **Panels Open With :** This one is used, for example, when you open the vendor and you want to open the player's inventory as well to see if you have enough money.
- **Panel Appear Above :** This one can make the panel opened right above the gameobject, but sometimes right above is not good enough, so here comes the offset.

9.Drop Item

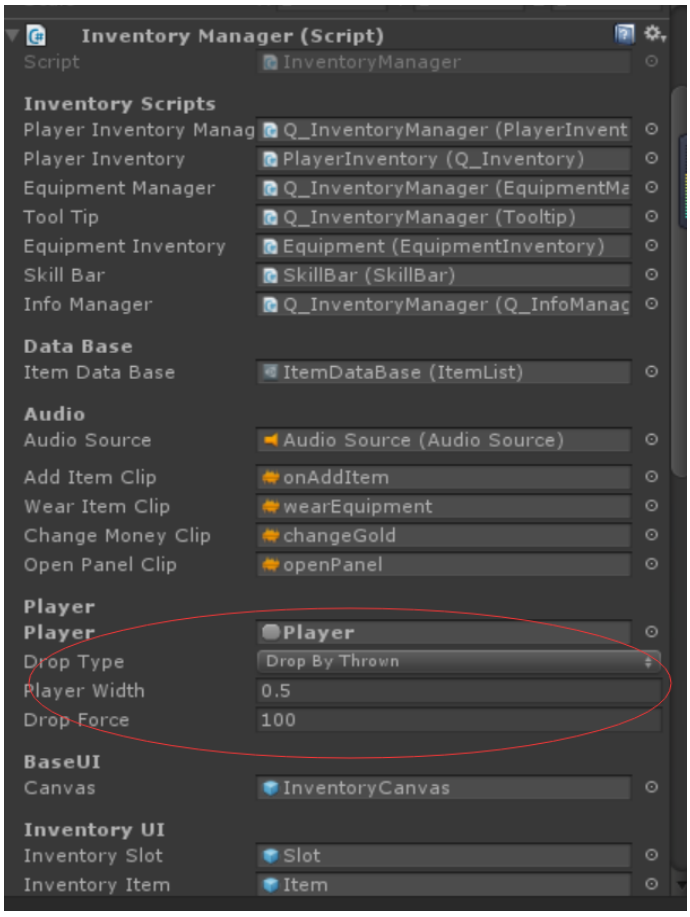
Used on the gameobject like monsters as loots.



- **Drop Currency :** The money you want to drop.
- **Drop amount :** The amount you want the item to drop
- **Drop Chance :** The **c**hance you want the item to drop

10.Inventory Manager

The script handle most of the prefabs, clips and other inventory things.



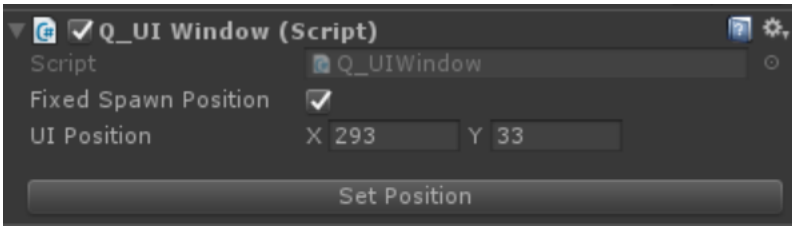
- **Drop Type :** The way player drop items
- **PlayerWidth :** Need to set when drop type is Drop By Thrown (To prevent picking up item while dropping)
- **DropForce :** Need to set when drop type is Drop By Thrown

11.Q_UI Drag

Used on the UI you want to drag around

12.Q_UI Window

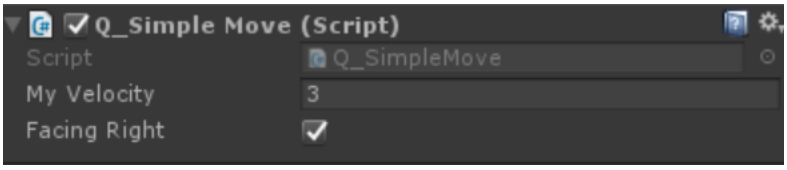
Used on the panel you want to fix its position (**Don't use with panel trigger's panel appear above**)



Just set the Fixed Spawn Position to true and drag to panel to the position you want or just and press the **Set Position button,** or just input the UI position (**No need to press button**)

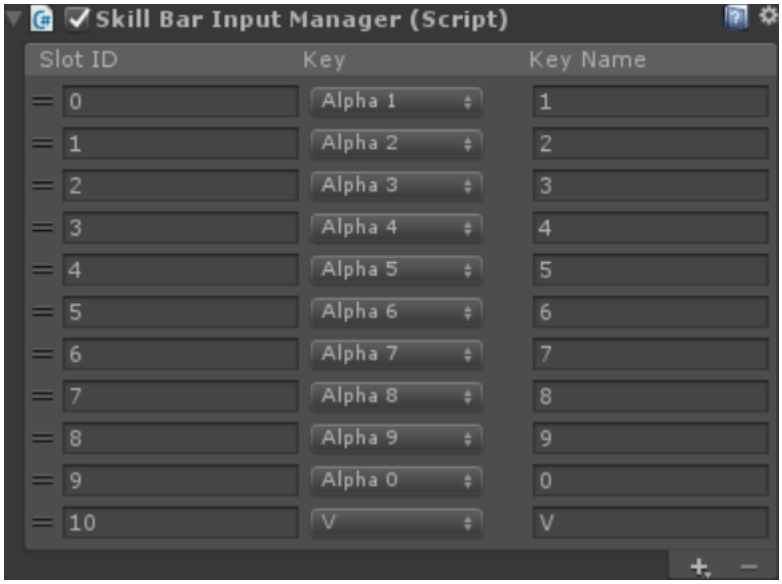
13.Q_Simple Move

A temporary script to control the player's moving. (Collider 2D and Rigidbody 2D needed)



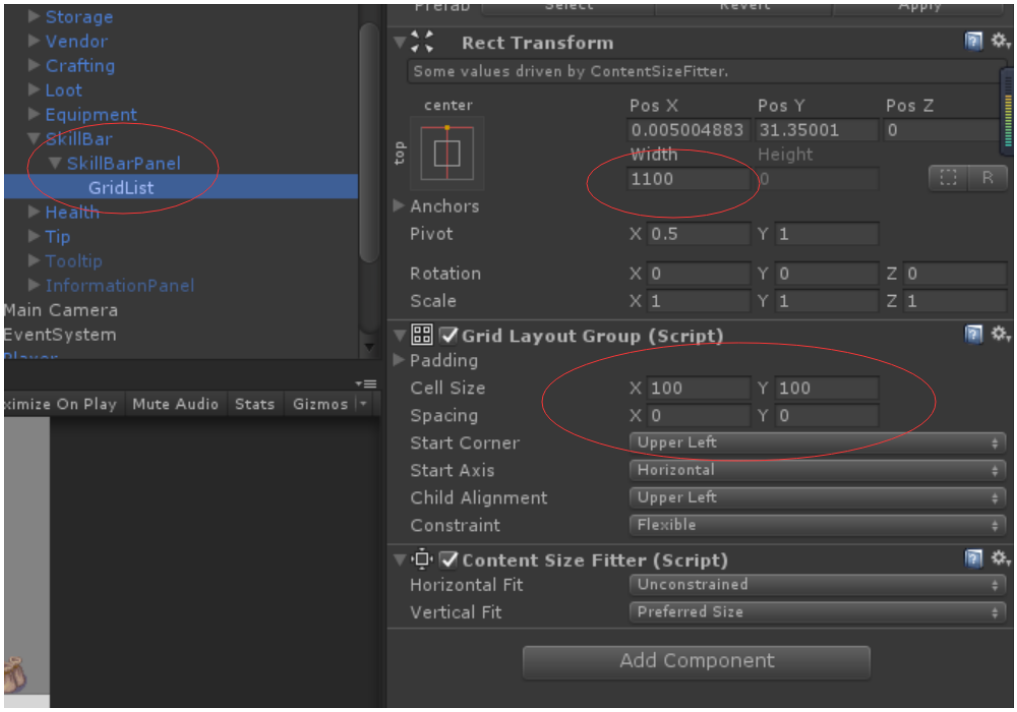
14.Skill Bar Input Manager

Used on the SkillBar



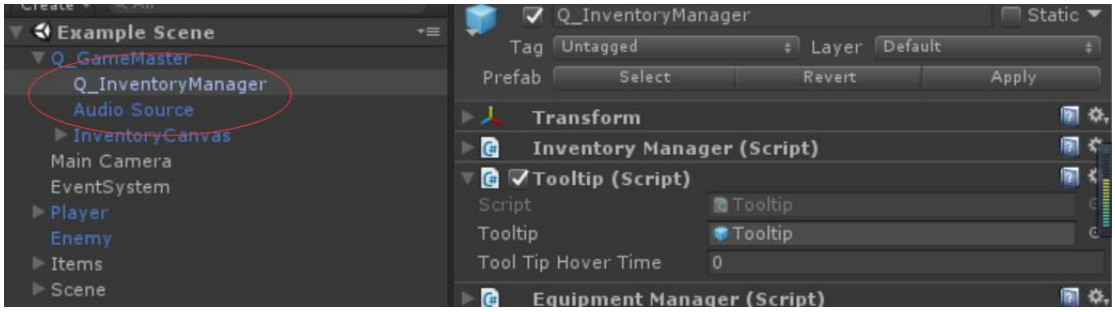
- **SlotID :** The slot you want the key to control
- **KeyName :** The name of key displayed in the skillbar slot, if it is null, the text will display the key's name like **Alpha 1** in the picture.

If you want to add more keys, you must only add keys here.



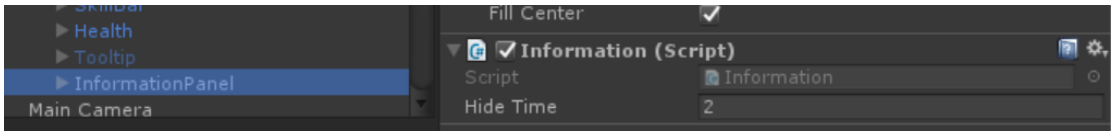
Go here to set these settings to suit the amount of the key (for example, if the key amount is 10 and cell size.x is 100, the GridList's Width must be bigger than 1000, or some keys won't be displayed on the screen)

15.Tooltip



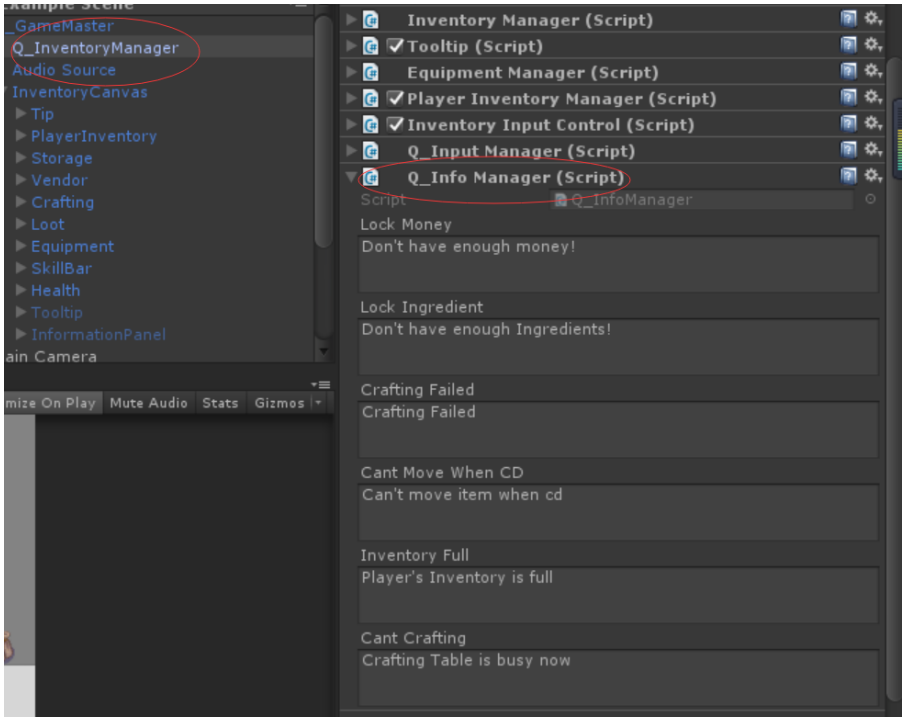
- **Tooltip :** The gameobject of tooltip
- **ToolTipHoverTime :** The delay time tooptip appears after you put your mouse on the item.
- Tooltip has an auto pivot adjustment to make sure itself are always in the screen, you can have a try.

16 Information



- **HideTime :** The dalay time information panel disappears after its activation.

17 InformationManager



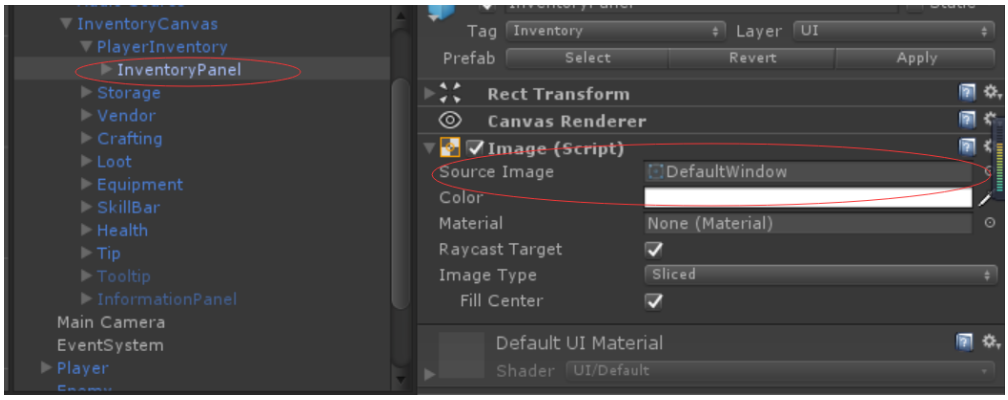
- The reminding information when player can't do something

8 Custom

Warning ! Don't Change the Structure, Order or Name of these prefabs, just change their position, size, source image and etc. Because I used some transform.find("") and getchild(0) in the code.

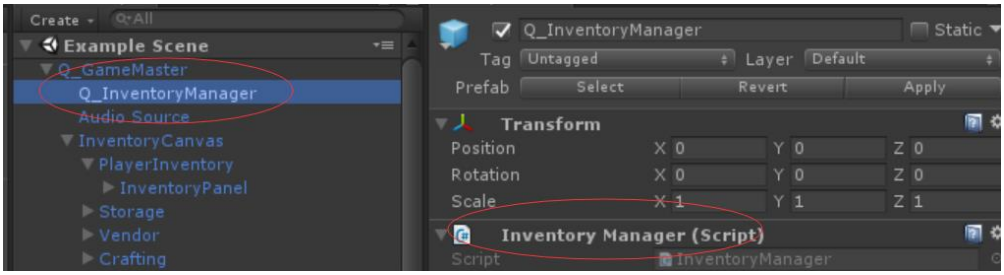
8.1 Custom Panel

Just find the ui whose name contains panel like this, and change its **Source Image**.



8.2 Custom Slot

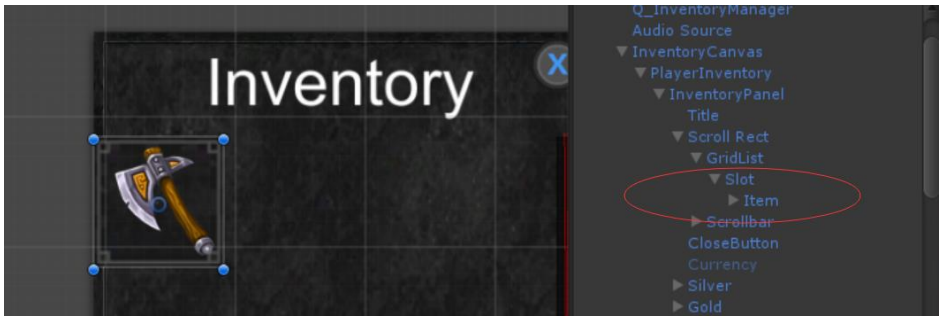
Find the slot you want custom in the inventoryManager



Double click on it and the prefab will be selected in the Project and then you can change its **Source Image**. (The size is setted in the GridList's component named Content Size Fitter)

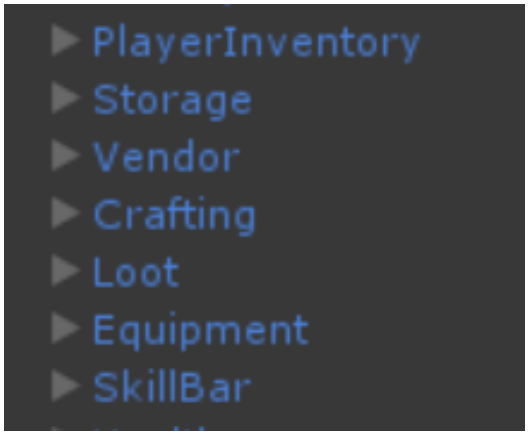
8.3 Custom Item(Not the item we create in the editor)

The same way as cumtom slot to find item, and I recommend drag the slot and item to the scene like this. Then you can custom it very conveniently. The Num and CD are Item's children, so you can find it and custom them there.



8.4 Add More Panels

Just **duplicate** these things in the picture.



8.5 Custom Tooltip

To be honest, the original tooltip is really really ugly, so just design yours.
Find the Tooltip.cs file and open it and design it here

```
public void ConstructDataString(string tag,ItemData itemdata)
{
```

```
void AddSellPriceInformation()
{
```

```
void AddVendorInformation()
{
```

```
void AddConsumableInfomation()
{
```

```
void AddEquipmentInfomation()
{
```

```
void AddCraftingInfomation(ItemData itemdata)
{
```

9 Other

If you find any bug or meet any problem or have any suggestion.

Just Contact me!

- **Email hawk_quan@163.com .**
- **Twitter https://twitter.com/Hawk_Quan**
- **Leave your comment in the asset store**
- **YouTube for more tutorials**

https://www.youtube.com/channel/UCCpifMR6O_DC--pZ4QxKOlQ?disable_polymer=true