

Q INVENTORY SYSTEM

1.1.0

用户手册

1 目录

关于 Q Inventory System

第一步

如何...

... 建立 Inventory UI 界面

... 物品面板的控制

... 创建一个新的数据库

... 创建我们的第一个属性

... 创建我们的第一种货币

... 创建我们的第一个物品

... 创建我们的第一个蓝图

... 编辑器设置

手动添加一些引用

将创建的东西添加到游戏场景中

... 直接将物品拖进场景中

... 添加物品到玩家背包&存储箱&宝箱

... 给商店添加售卖的物品

... 给合成面板添加蓝图

... 设置打开这些面板的触发器

... 运行场景看看能不能用了

脚本组件

... Q_Inventory

- [... Storage](#)
- [... Vendor](#)
- [... Crafting](#)
- [... SkillBar](#)
- [... Add Item](#)
- [... Add Vendor Item](#)
- [... Panel Trigger](#)
- [... Drop Item](#)
- [... Inventory Manager](#)
- [... Q_UI Drag](#)
- [... Q_UI Window](#)
- [... Q_Simple Move](#)
- [... Skill Bar Input Manager](#)
- [... Tooltip](#)
- [... Information](#)
- [... InformationManager](#)

[自定义](#)

- [... 自定义面板](#)
- [... 自定义物品格](#)
- [... 自定义物品](#)
- [... 添加更多的面板](#)
- [... 自定义提示框](#)

[其他](#)

Q INVENTORY SYSTEM

Hawk Quan

2 关于 Q Inventory System

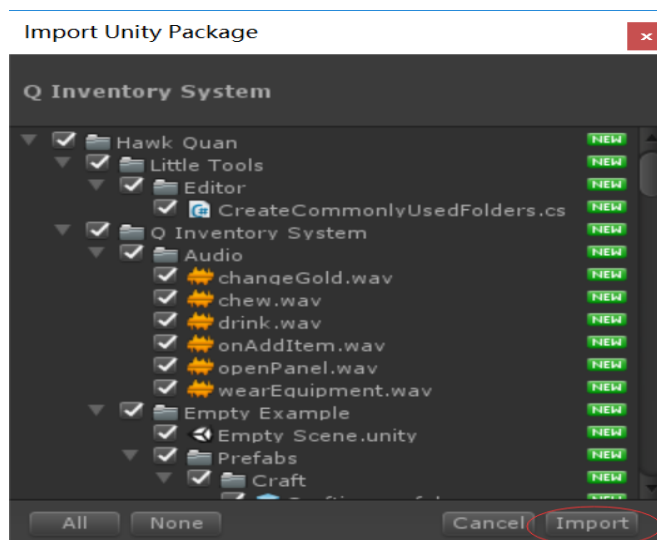
Q Inventory System 是一个十分有用和灵活的物品系统插件（就是装备啊，消耗品啊，材料之类的），它一开始是主要针对 2D 游戏设计的。

通过便捷的编辑器系统 (Editor Extension)，你可以很轻松地创建和管理物品、蓝图、货币、属性等元素。

3 第一步

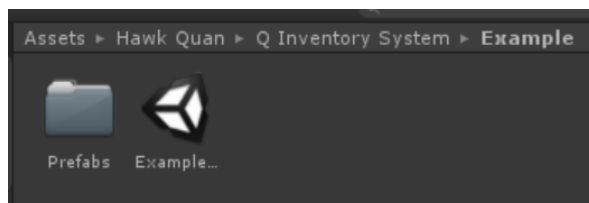
3.1 Step 1 – 从资源商店下载 Q Inventory System

3.2 Step 2 – 将下载的资源包导入



在开始 Step 3 之前，首先确保你的 tags 已经包含以下 tag : Inventory, Vendor, Craft, Equipment, Storage, SkillBar (Weapon 和 Enemy 并不是必须的，但是如果你想完美地运行示例场景的话，最好还是加上)

3.3 Step 3 – 打开示例场景看看里面的东西



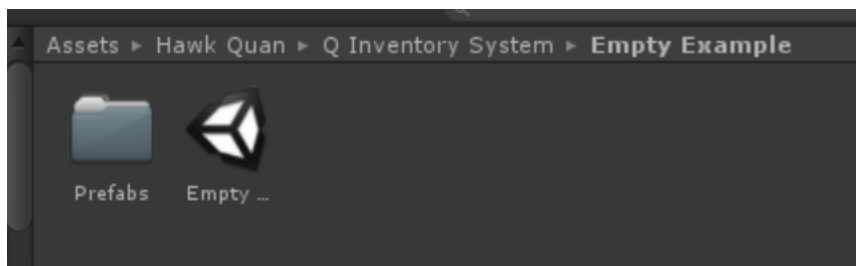
4 如何

4.1 创建 Inventory UI 界面

我已经预设好了“空”场景和“空”预制体，你可以直接把它们拖进你的场景中。
(下面两种任选其一即可)

4.1.1 “空”场景

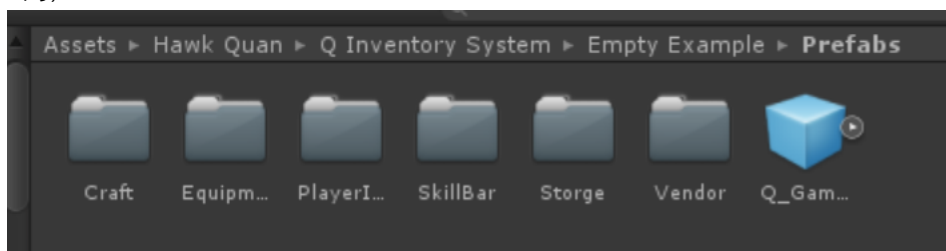
空场景中只有最基本的 Inventory UI 元素。所以你可以直接复制这个场景 (选中按 Ctrl 加 D)，然后重命名，之后你就可以把它随便放在哪并且在里面开始创建你的新游戏了。



4.1.2 “空”预制体

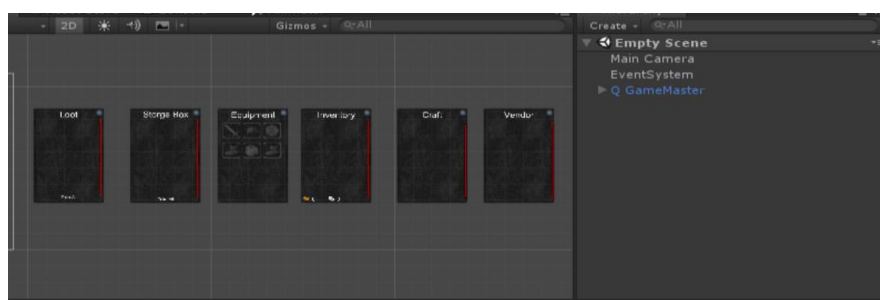
在上图的 prefabs folder 中，所有的预制体都只是最基本的 Inventory 预制体。它们没有任何数据库、物品、蓝图、属性和货币。它们只是有最基本的一些脚本和资源引用 (如果这些让你自己说的话，会很麻烦)。

因此，只需将 Q GameMaster 拖进你的场景中。然后你就可以看到一堆面板，但是请别忘了创建一个 Event System 如果你的场景里之前还没有的话。(不然你是不能与这些面板交互的)



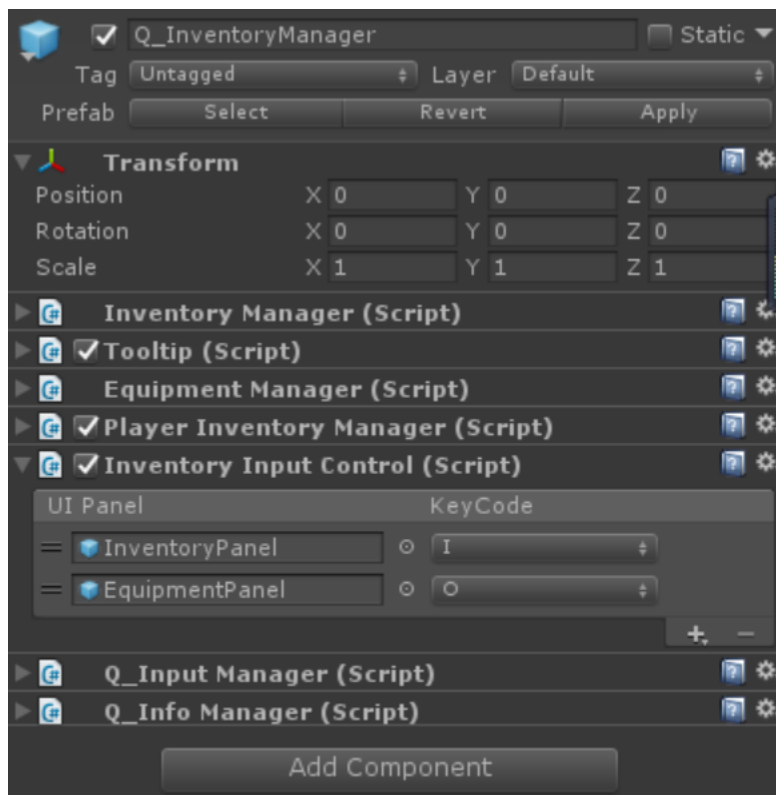
4.2 物品面板控制

至此，你的场景已经有了最基本的 Inventory UI，看起来就像下图那样。

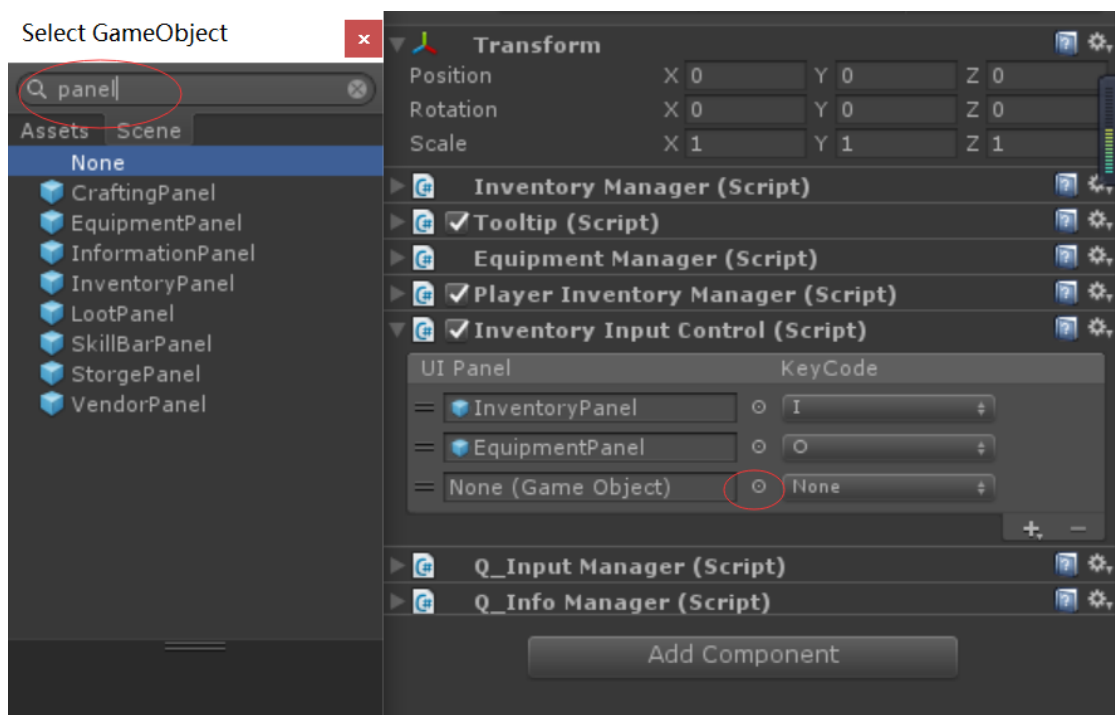


你可以尝试着运行一下场景，但是我事先只预设了两个面板的开关键（I 是玩家背包，O 是玩家信息面板）

你可以自己去设置一些开关键。（记得退出运行模式先）找到 Q_GameMaster / Q_InventoryManager.



然后点击那个加号。选择一个 panel（是名字带 panel 的不是它们的父物体）并选择键位。



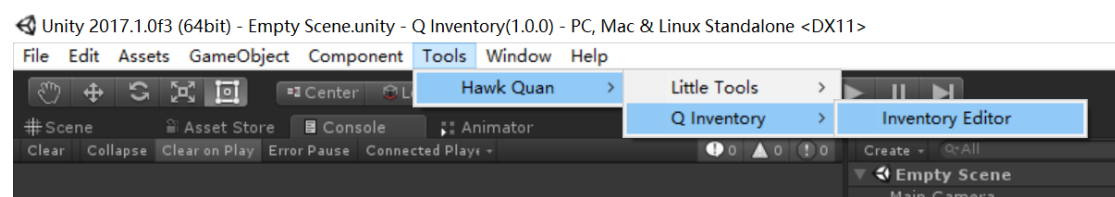
在设置好后，再次运行场景，看你现在是否能够控制你之前设置的面板的开关了。（通常情况下合成栏、商店栏、储存栏和宝箱栏是被一些其他事物触发的，所以我没有事先预设好这些键位。但是每个人想法不同，所以想怎么设就怎么设吧）

4.3 创建一个新的数据库来管理物品、蓝图、属性和货币

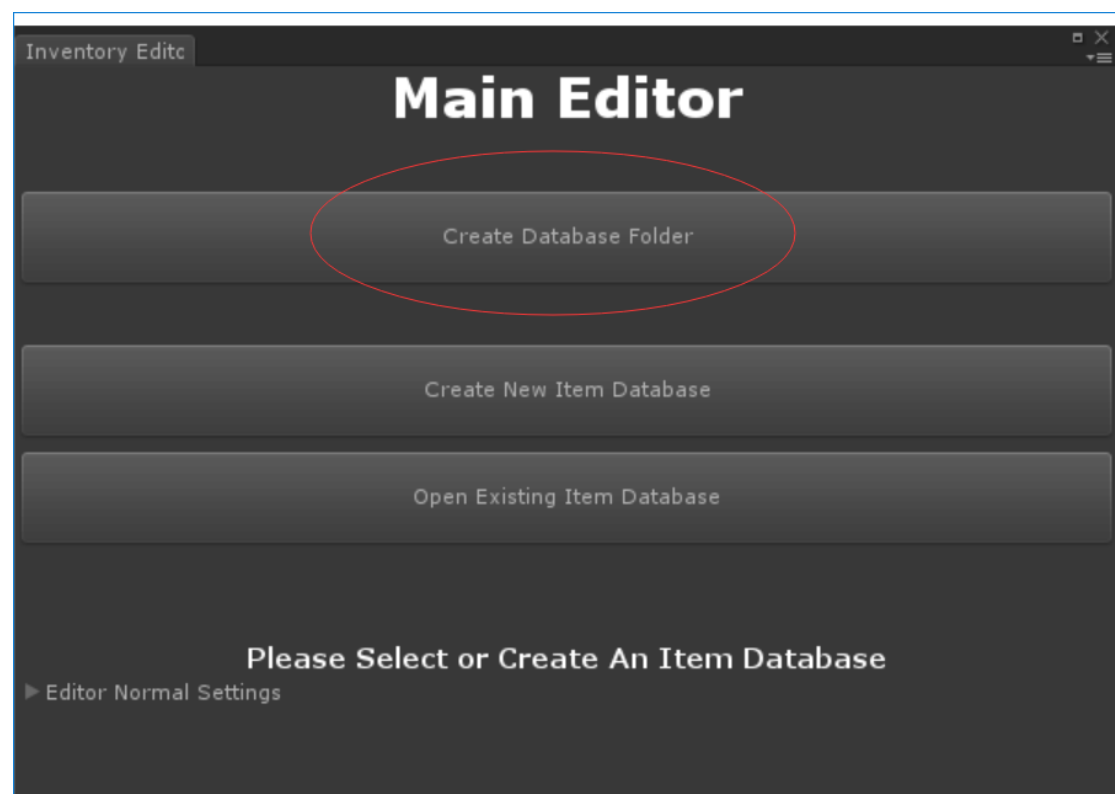
在 UI 创建完后，我们现在需要一个数据库来储存数据。

4.3.1 创建 Inventory Database 文件夹

在 Unity 上方你可以找到 Tools/Hawk Quan/Q Inventory/Inventory Editor，打开它。



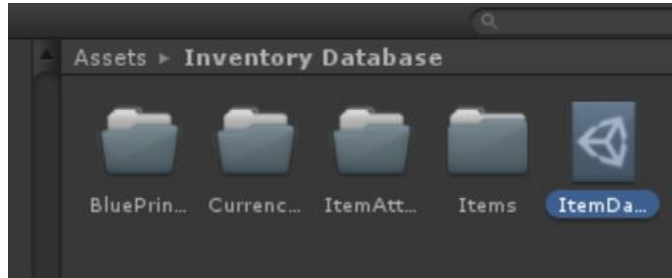
现在你就可以看到主编辑器了



点击 **“Create Database Folder”** 按钮并且在你的 assets 中选择一个地方来放这些文件夹，稍等片刻你就可以看见创建成功的 debug 消息了。

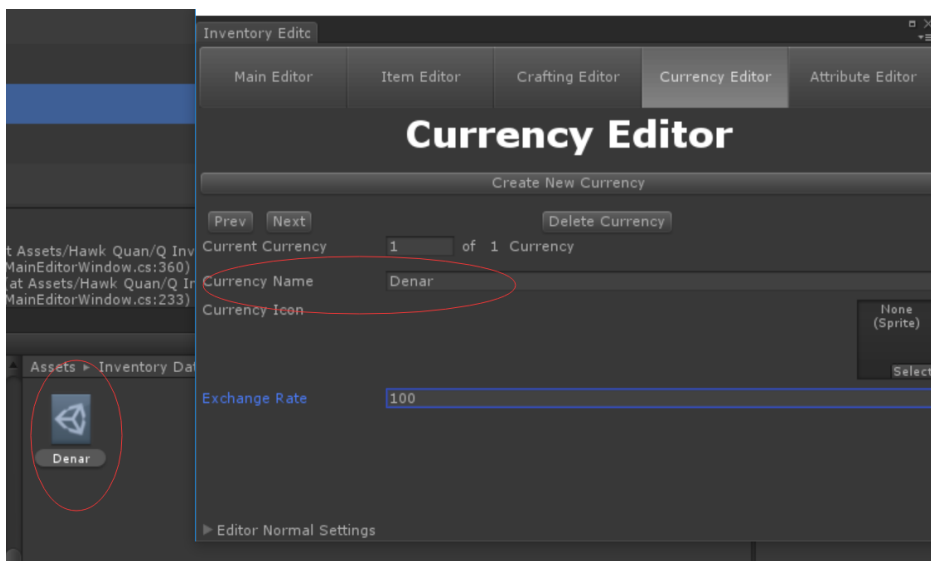
4.3.2 创建 Inventory 数据库

点击“Create New Item Database”按钮你就可以在刚刚创建的文件夹里看到一个新的数据库文件了。并且编辑器窗口的界面也会有一些变化。现在我们可以开始新建东西了。



4.3.3 创建我们的第一种货币

点击 Currency Editor, 然后你会看到提示信息“Don't have any Currency”。点击 Create 按钮, 我们的第一种货币就被创建了, 给它重命名。一旦你在编辑器里重命名, 文件名也会自动被更新, 所以会有一下小卡顿。(注意! 最好不要直接在.asset 文件中更改数据, 在编辑器中编辑数据即可)



4.3.3.1 货币图标

不管你加不加这个图标都是没有关系的, 因为我并没有写关于这个图标的任何功能, 但是如果你想自己在一些脚本中用到它的, 你可以在这里找到它。

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

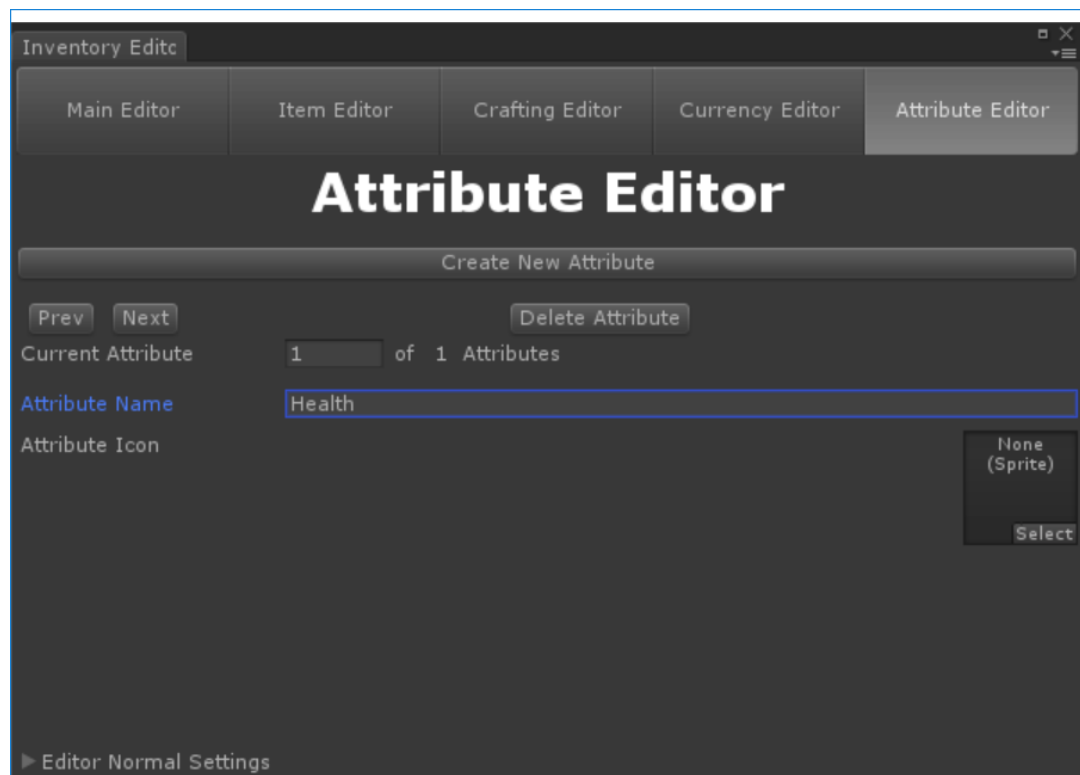
4.3.3.2 汇率

如果你创建了多种货币的话,不同货币之间肯定会有一个汇率关系,这就是这个的作用。默认的货币汇率是 1 (最好把汇率设为整形的)

举个栗子,如果我创建了“金币”和“银币”这两种货币。前者的汇率是 10, 后者的是 1。如果我有 9 枚银币,那就是 9 枚银币,但是如果我有超过 10 枚银币比如说 16 枚,它会自动转换成 1 金 6 银。所以弄清这个关系再去设定(注意,汇率越大的货币放在越前面,让货币顺序保持一个降序)

4.4 创建我们的第一个属性

和货币一样,直接建就好了



4.4.1 属性图标

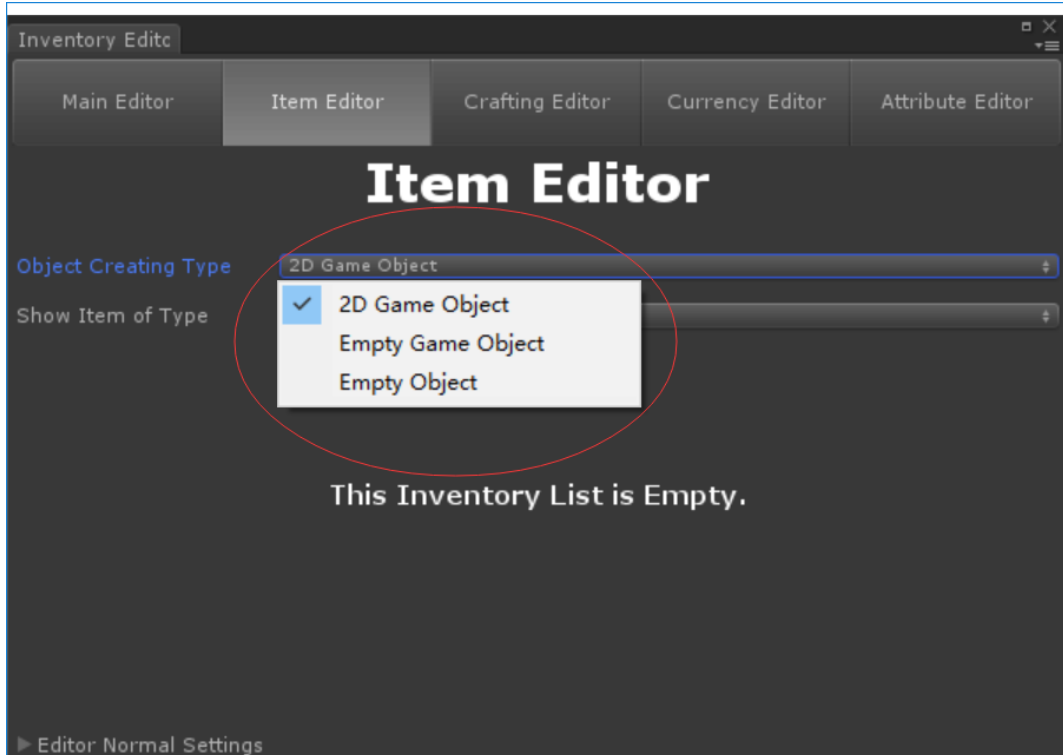
和货币的一样

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

4.5 创建我们的第一个物品

终于，我们可以创建第一个物品了。

首先来到 Item Editor，在点击 create 按钮之前，你可以在 **object type** 选择你想创建的物品类型（会在 Item 文件夹中自动创建一个预制体，默认的选择是 2D Game Object）



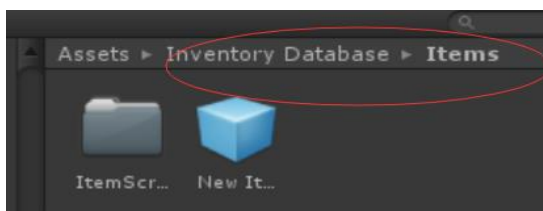
然后你就可以修改物品的参数了

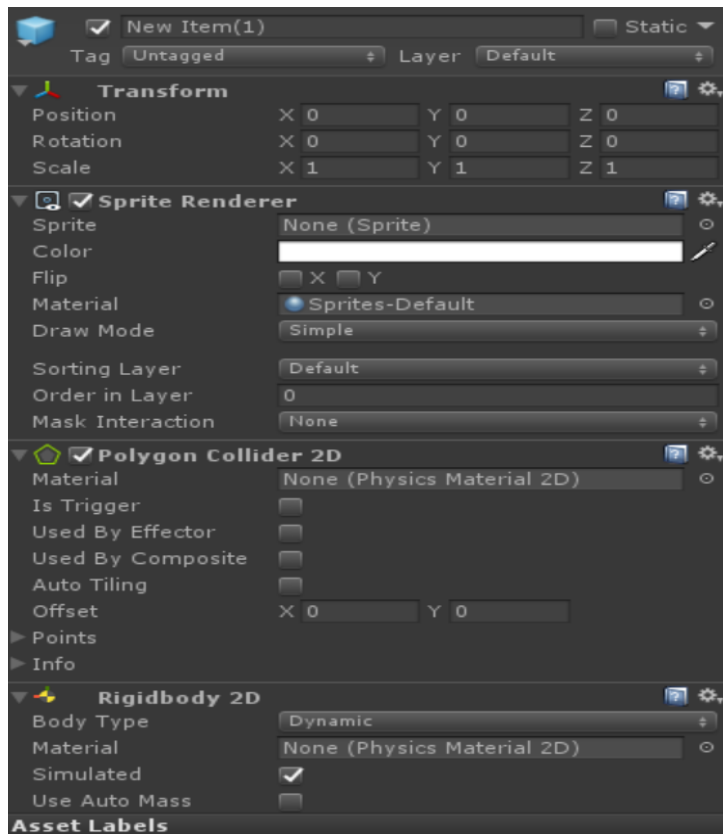
4.5.1 物品 ID

每一个物品都有一个独一无二的 ID，但是你修改它们的顺序和 ID 在 **reorder items** 中（我稍后会讲到）

4.5.2 物品图标

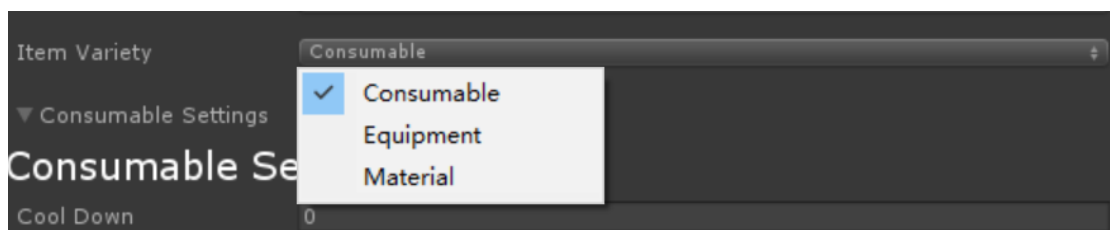
当你的创建的物品类型为 2D 的时候，编辑器会自动给你创建的物品加上 Polygon Collider 2D（你可以选择加不加 Rigidbody 2D），并且当你更换物品图标时编辑器会自动移除就得 Polygon Collider 2D 并重新加一个 Polygon Collider 2D 来适应图标的变更。但是有时也许你会觉得这个 collider 不够好，你可以自己去 Item 文件夹或者是点击 show item 按钮找到预制体来自己手动加。



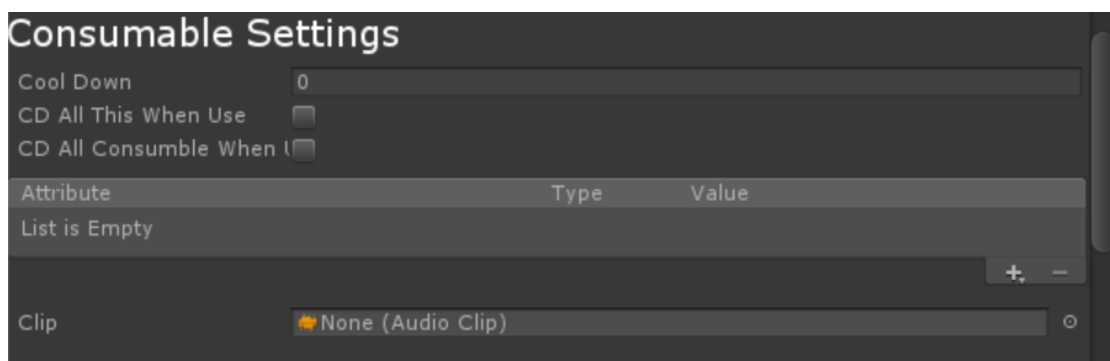


4.5.3 物品种类

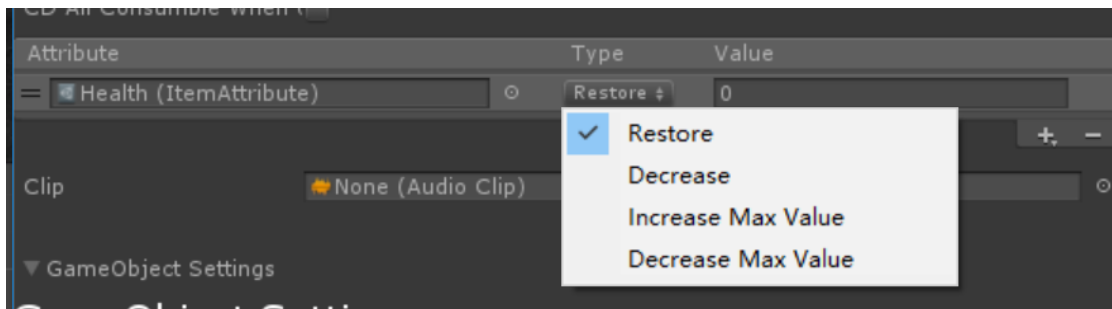
你有三种种类选择：消耗品、装备、材料（如果你想自己加一些种类的话，到 Q_InventoryEnum.cs 文件中去找到对应的枚举类型加在后面就好了）



4.5.3.1 消耗品

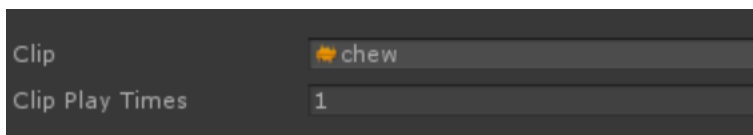


- **Cool Down（冷却时间）** : float
-
- **CD All This When Use** :这个意味着当你使用这个物品，快捷栏和玩家背包里的这种物品都会进入 CD 状态
- **CD All Consumable When Use** : 这个意味着当你使用这个物品后，快捷栏和玩家背包里的所有消耗品都会进入 CD 状态（CD 时间是这个物品的 CD）
- **Attribute（属性）**：点击加号添加属性

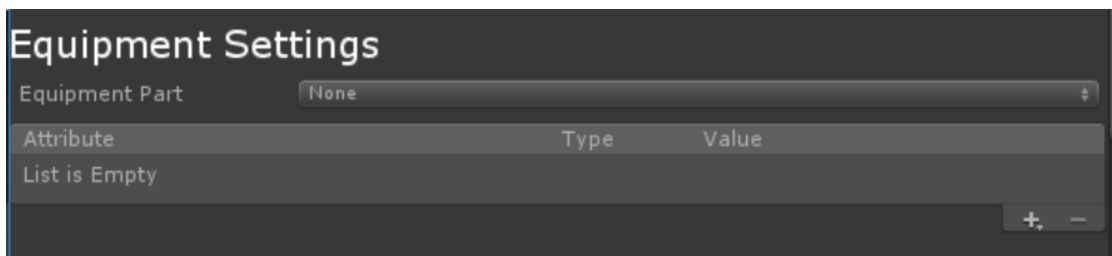


你可以在四种效果中选择

- **Clip（音效）**：在这里设定的音效会在使用物品时播放。



4.5.3.2 装备



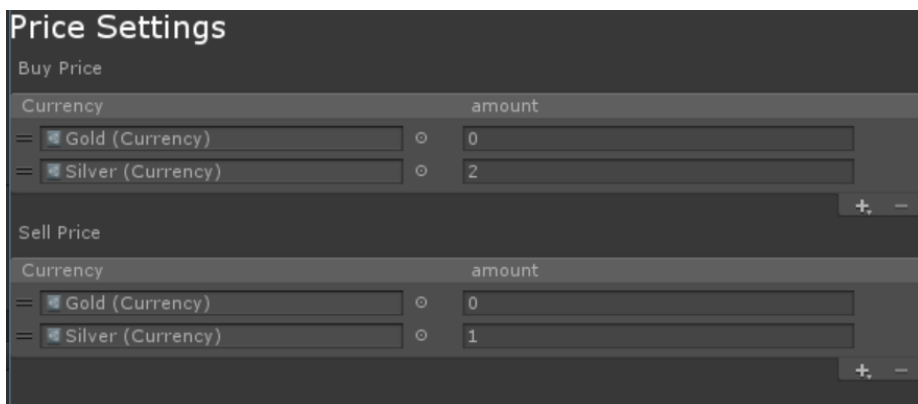
- **Equipment Part（穿戴位置）**：你可以在上面说的枚举文件中自己添加种类
- **Attribute（属性）**：和上面消耗品的属性一样设置，但是最好不要设置为 restore。

4.5.4 游戏物体设定



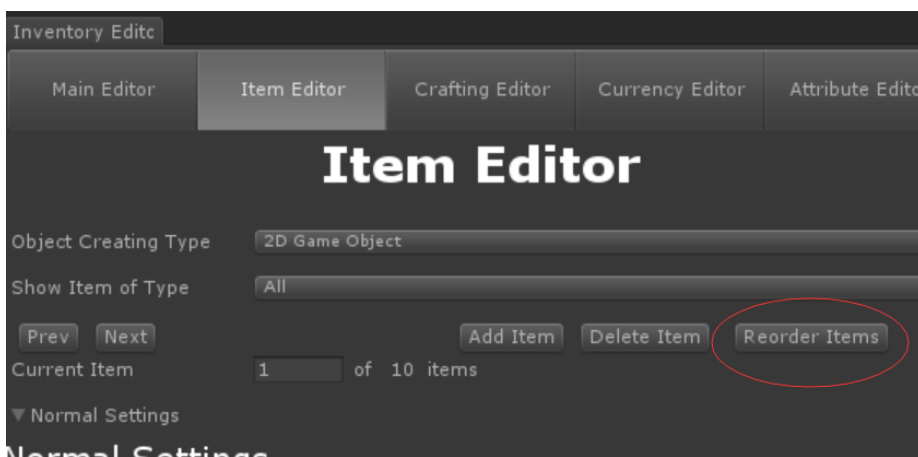
- **Item Object (预制体)**：在你创建物品时自动添加的预制体，但是如果你创建的种类是 Empty Object 的话，这个是不会自动添加的。所以记得在让空物体变为预制体后来这手动添加引用。

4.5.5 价格设定

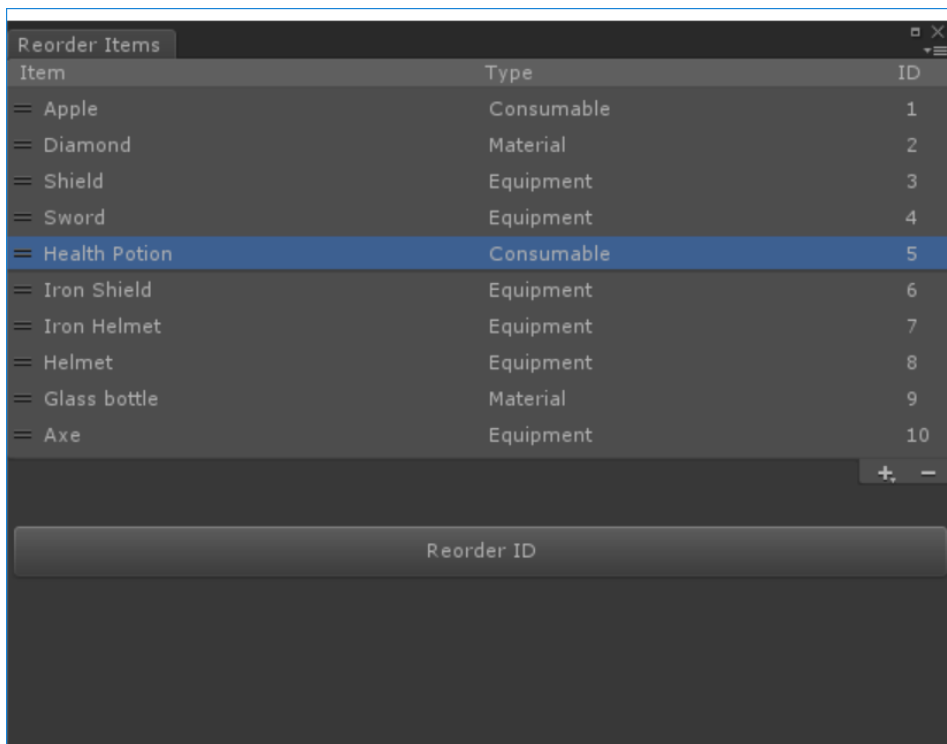


- 当你按下加号后，编辑器会自动地添加所有货币，（再次强调，当你创建货币时，确保汇率是降序的）
- 就算价格很小，用不到前面几种大货币，就让它 amount 为 0 即可，不要删除。

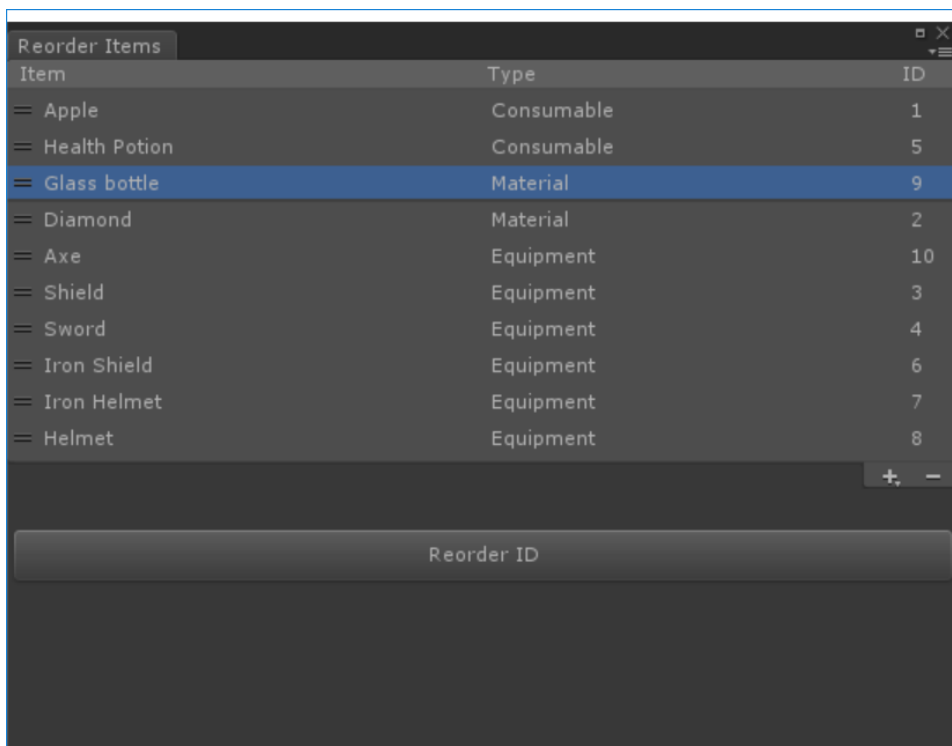
4.5.6 重新排列物品



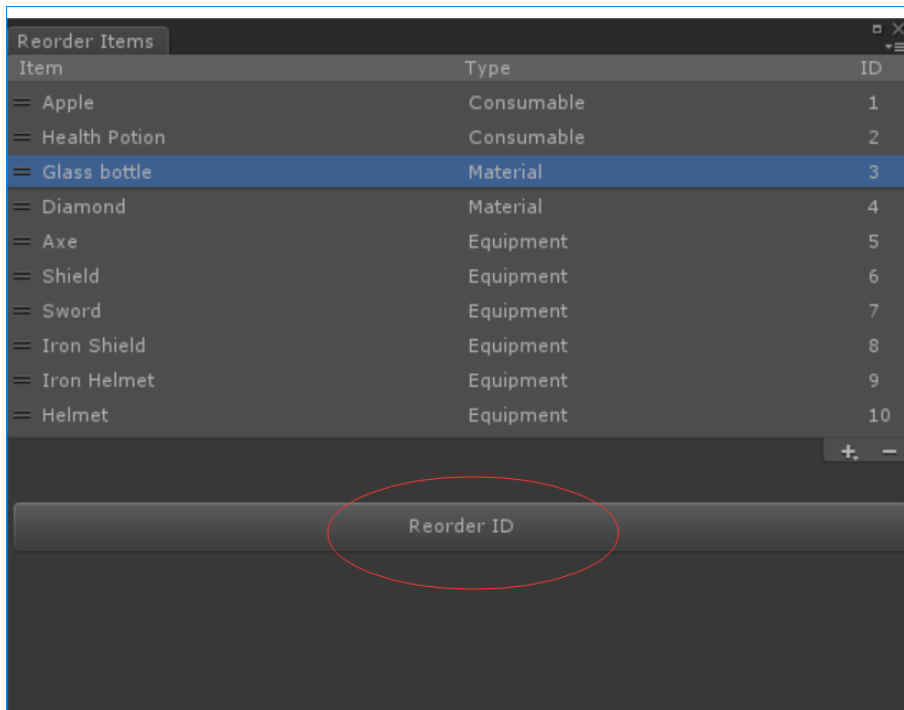
当你创建的物品很多时，你可能会发现物品的顺序之类的变得一团糟，这时候就需要点下 Reorder Items 按钮进入排序界面了。



在这里你可以随意移动物品顺序来重新排序



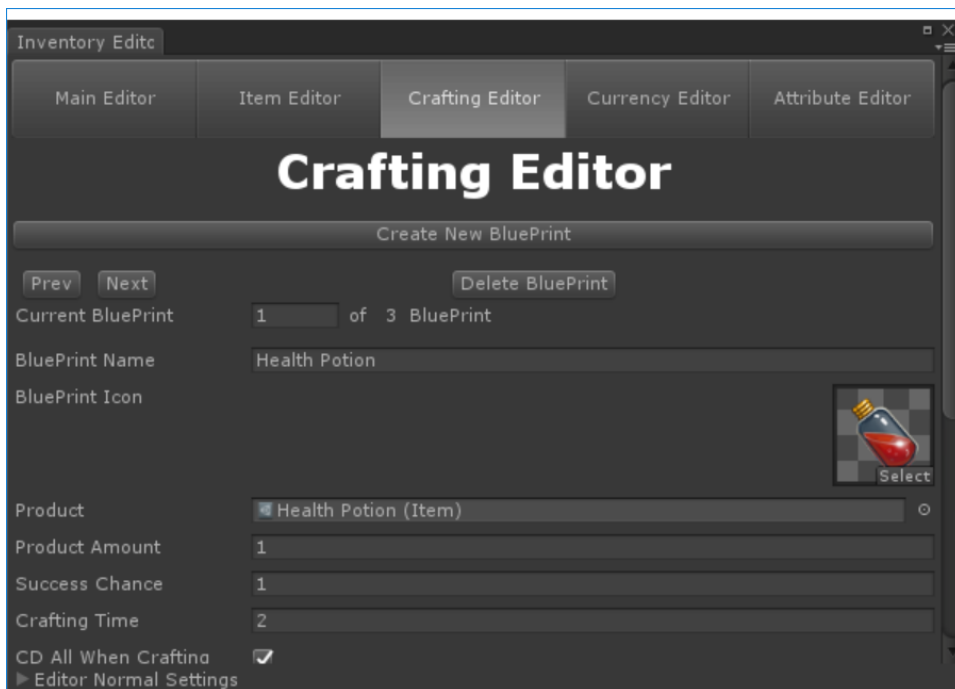
然后排好顺序后，按下 Reorder ID 按钮，物品被打乱的 ID 也就被重新设定了。



在这里你也可以当做是一个快速浏览所有物品的地方。
(加减号的功能都被我 ban 掉了，所以不用担心当你不小心点了一下那个会搞出什么大新闻)

4.6 创建我们的第一个蓝图

就像前面一样创建就好



4.6.1 蓝图图标

如果图标是空的话，系统会自动展示要制造的物品的图标。

4.6.2 产品

- **Product (产品)** : 就是造出来最终得到的东西.
- **Product Amount** : 当一次建造成功时得到的数量.
- **Success Chance (成功几率)** : Float.
- **Crating Time (合成时间)** : Float.
- **CD All When Crafting** : 这个意味着当你建造这个时其他蓝图也会进入 CD (就和上面消耗品一样)

4.6.3 原料和价格

The screenshot shows two configuration panels. The top panel, titled 'Ingredients', has a table with two columns: 'Ingredient' and 'amount'. It lists 'Apple (Item)' with an amount of 2 and 'Glass bottle (Item)' with an amount of 1. The bottom panel, titled 'Prices', also has a table with 'Currency' and 'amount' columns. It lists 'Gold (Currency)' with an amount of 0 and 'Silver (Currency)' with an amount of 5. Both panels have a '+' and '-' button at the bottom right.

Ingredient	amount
Apple (Item)	2
Glass bottle (Item)	1

Currency	amount
Gold (Currency)	0
Silver (Currency)	5

- **Prices (手续费)** : 当你去铁匠铺之类的地方造东西时, 有时候铁匠会收你点手续费, 这个就是了。

4.7 编辑器设定

当你的Unity皮肤是个人版时, 白色的标题可能就会比太好看, 或者因为其他什么原因, 你就是想换个颜色, 那么你就可以在下方 Editor Settings 里面来自定义标题的颜色。

The screenshot shows the 'Item Editor' window. It has a tabbed interface with 'Main Editor', 'Item Editor', 'Crafting Editor', 'Currency Editor', and 'Attribute Editor'. The 'Item Editor' tab is active. It displays 'Object Creating Type' as '2D Game Object' and 'Show Item of Type' as 'All'. There are buttons for 'Prev', 'Next', 'Add Item', 'Delete Item', and 'Reorder Items'. The 'Current Item' is '1 of 10 items'. Under 'Normal Settings', 'Item ID' is 1, 'Item Name' is 'Apple', and 'Item Description' is 'great'. There is an 'Editor Settings' button at the bottom left.

Inventory Editc

Main Editor Item Editor Crafting Editor Currency Editor Attribute Editor

Item Editor

Object Creating Type: 2D Game Object

Show Item of Type: All

Prev Next Add Item Delete Item Reorder Items

Current Item: 1 of 10 items

▼ Normal Settings

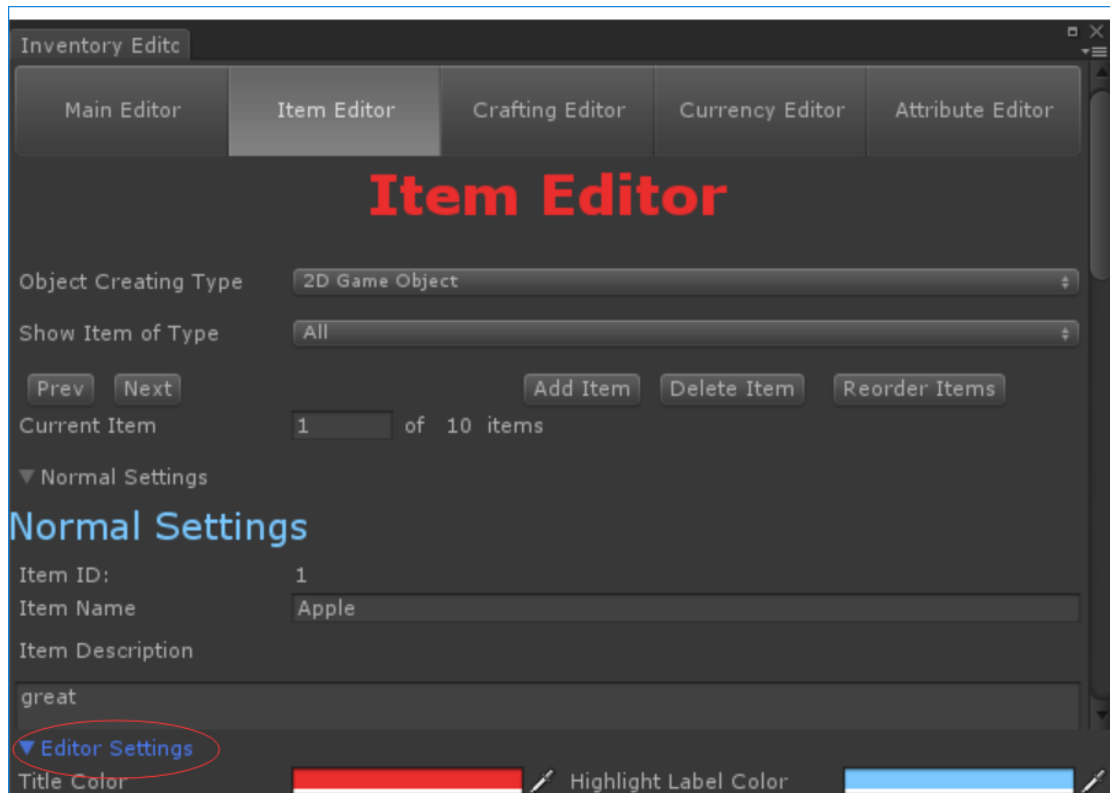
Normal Settings

Item ID: 1

Item Name: Apple

Item Description: great

► Editor Settings

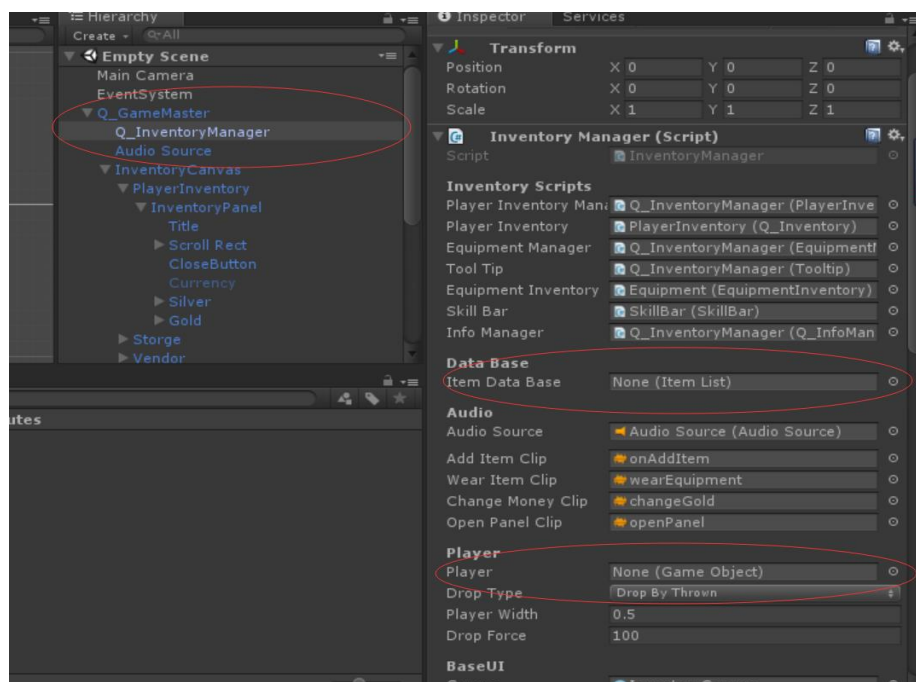


(这个数据存储在 Inventory Database 文件夹里的 Editor 里面)

5. 手动添加一些引用

5.1 Database and player

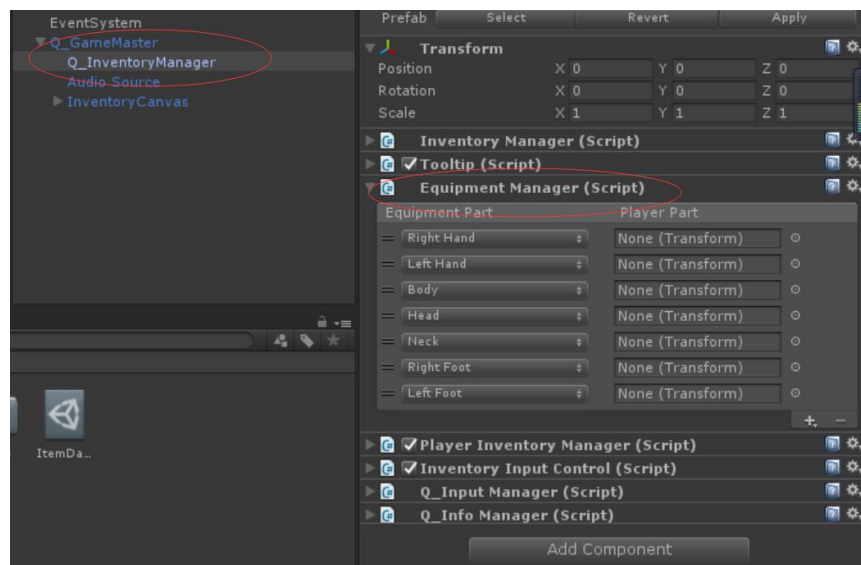
找到下图中所标识的东西，给它们添加上引用。(player 需要一个 trigger 来捡物品的，Database 就是刚刚创建的那个)



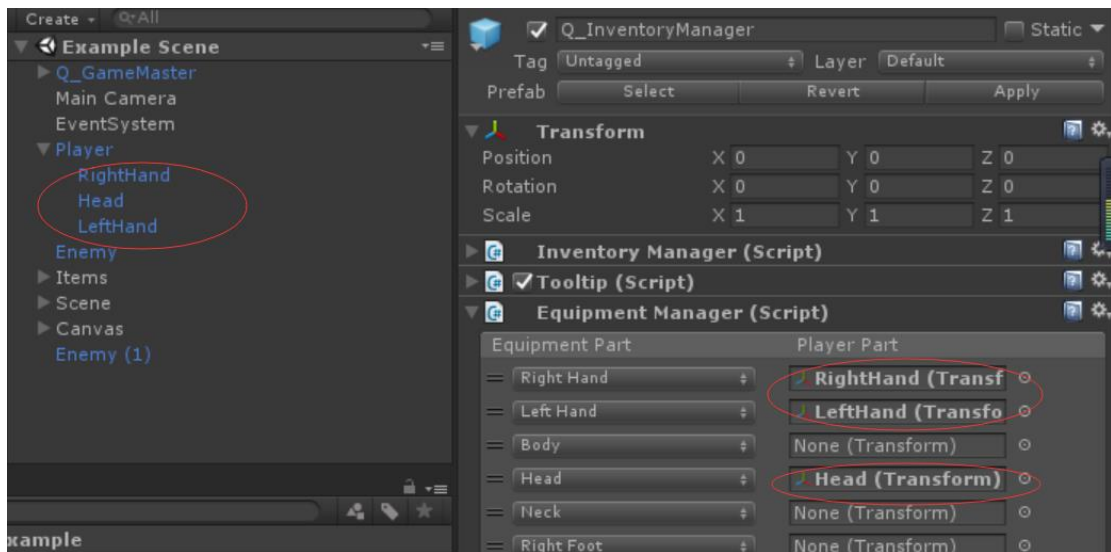
注意到 Player 下面那几个设定了么?那些是玩家丢弃物品时的设定，我稍后会讲到这些。

5.2 穿戴位置

如果你想你的 player 真正地穿戴上一些装备，而不是只是单纯的在属性面板里改变几个参数，那就在 Equipment Manager 里面设定穿戴部分的引用吧。

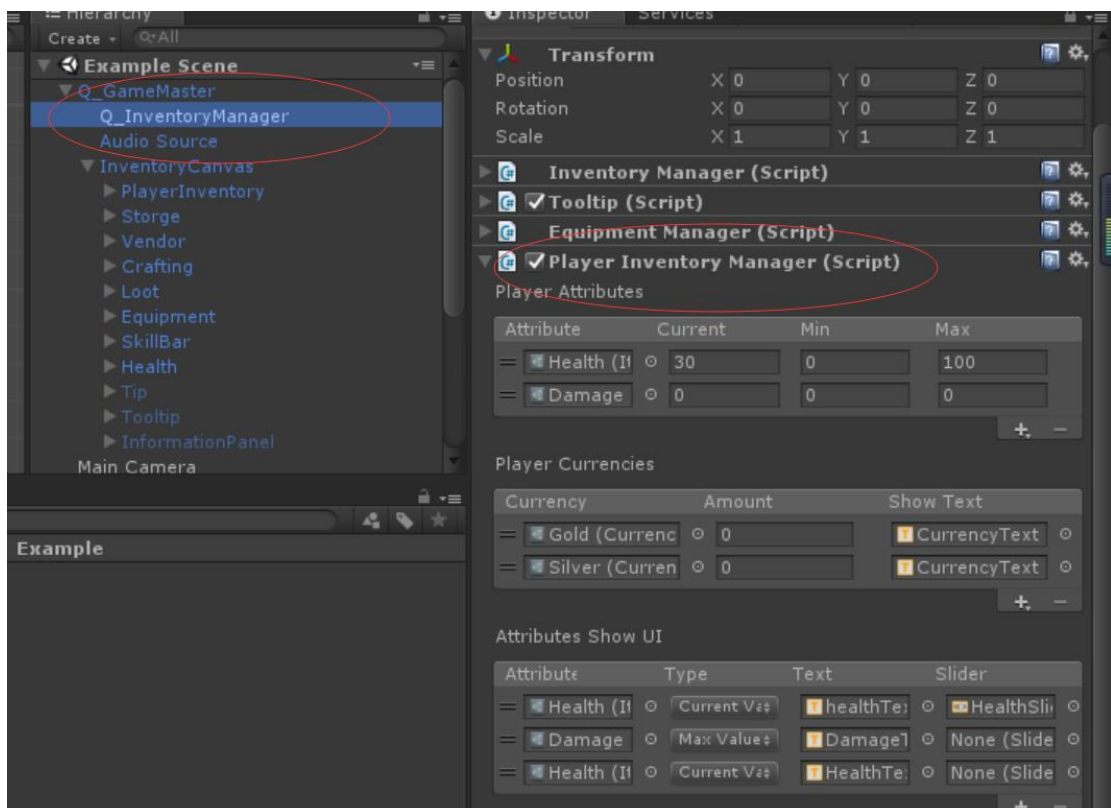


Player part 是当你使用装备时装备物品出现的位置（我的建议是在 player 下面放几个空的子物体来作为穿戴的位置，就像 example scene 里面的 player 一样）



5.3 Player Inventory Manager

首先在下图位置中找到这个



5.3.1 player Attributes（玩家属性）

这个设置是你想 player 拥有的属性（想生命值啊，魔法值啊，力量，敏捷之类的）

- **Current (当前值) (Float):** 这个就是你的 player 确切的属性值, 比如 Health : 30/100, 30 就是这个值。
- **Min (最小值) (Float):** 属性值的下限。
- **Max (最大值) (Float):** 属性值的上限, 比如 Health : 100/100, 这时你如果使用的生命药水, 血量也不会增加。

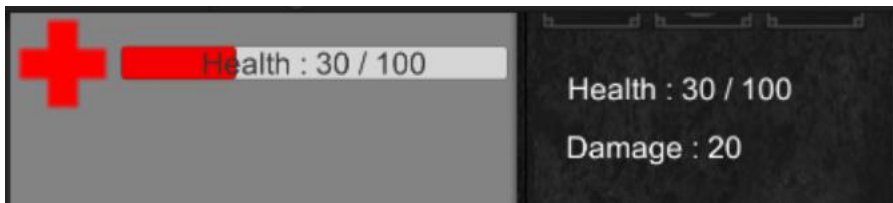
5.3.2 Player Currency (玩家货币)

这个就是玩家有多少钱了, 记住就算一开始你想让玩家一无所有, 你也需要向上面提岛的把所有货币添加上 (降序), 然后把 amount 设置为 0 即可。

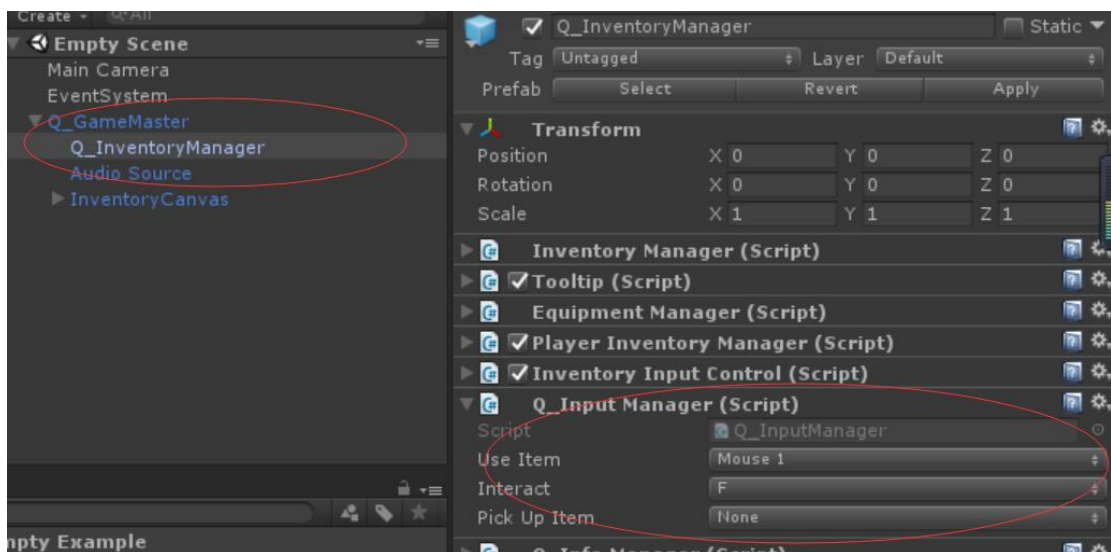
- **Show Text (展示的文字):** 如果你添加了 text 的引用, 那么这个 text 会显示对应的货币数量。

5.3.3 Attribute Show UI (属性展示 UI)

这里你可以设置你想要展示属性, 展示方式, 有 text 和 silder 两种方式。如下图



5.4 Q_Input Manager



这里是一些键位的管理

- **Interact (互动键):** 和场景中的物品互动, 比如在商店打开交易面板之类的
- **Pick Up Item (捡东西键):** 如果为 none, 玩家会在 ontriggerEnter 时捡起物品, 否则会在 ontriggerStay 时检测到这个键的按下而捡起物品。

6. 将创建的物品添加到场景中

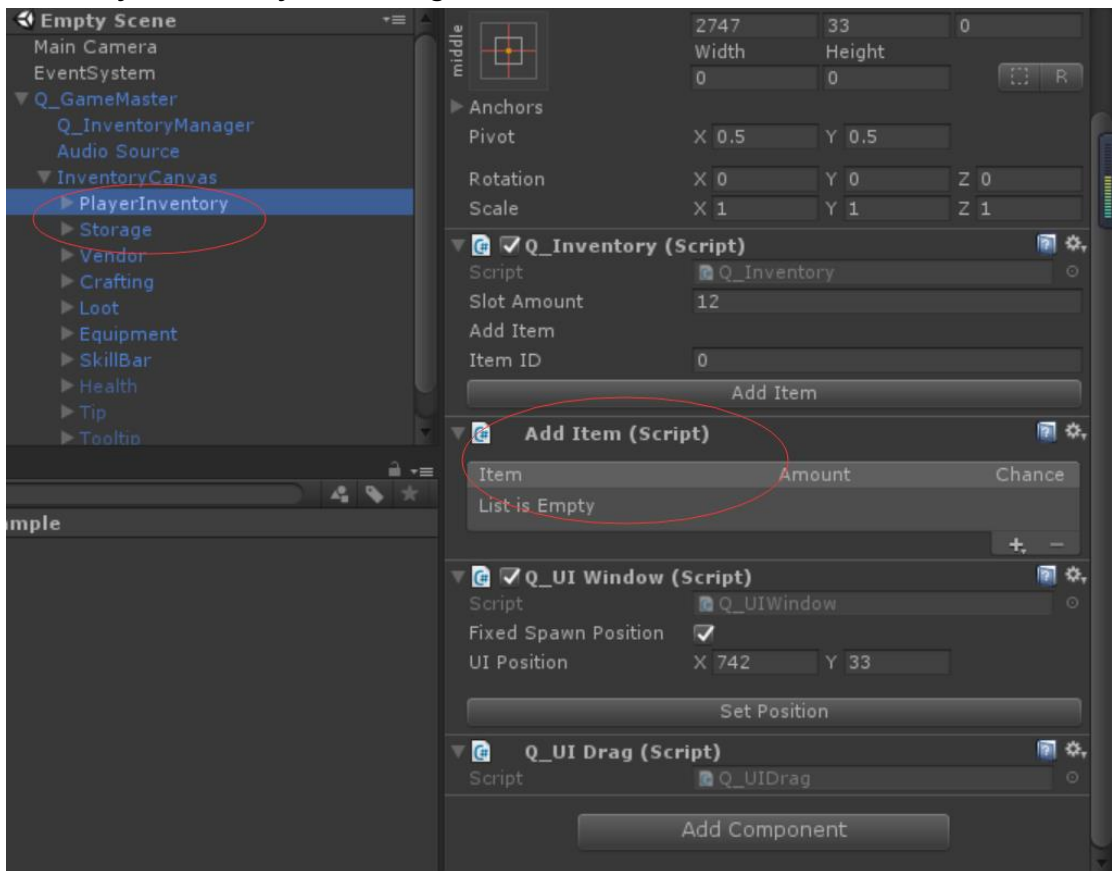
6.1 直接将物品拖进场景中

如果你想直接在场景里面布置一些物品，找到它们的预制体拖进场景中即可。

6.2 添加物品到玩家背包&储物箱&宝箱

(最好先看一下这些脚本组件 **Q_inventory**, **Crafting**, **Vendor**, **Storage components**)

找到 **PlayerInventory** 或 **Storage** 或 **Loot**。它们都会有一个 Add Item 组件。

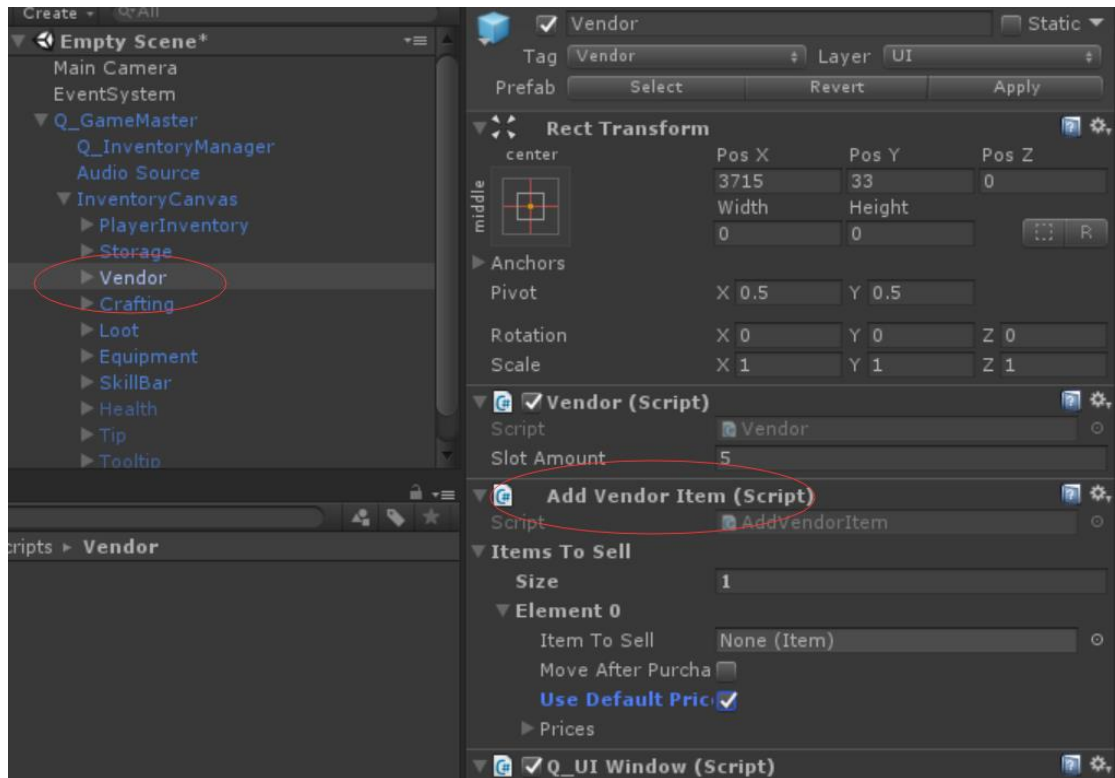


点击加号，添加你想要的物品，物品就会在场景运行开始时添加。

- **Chance(几率)(Float):** 成功加入到对应面板中的几率（可以用在宝箱随机生成物品上）。(between 0 and 1)

6.3 给商店添加售卖的物品

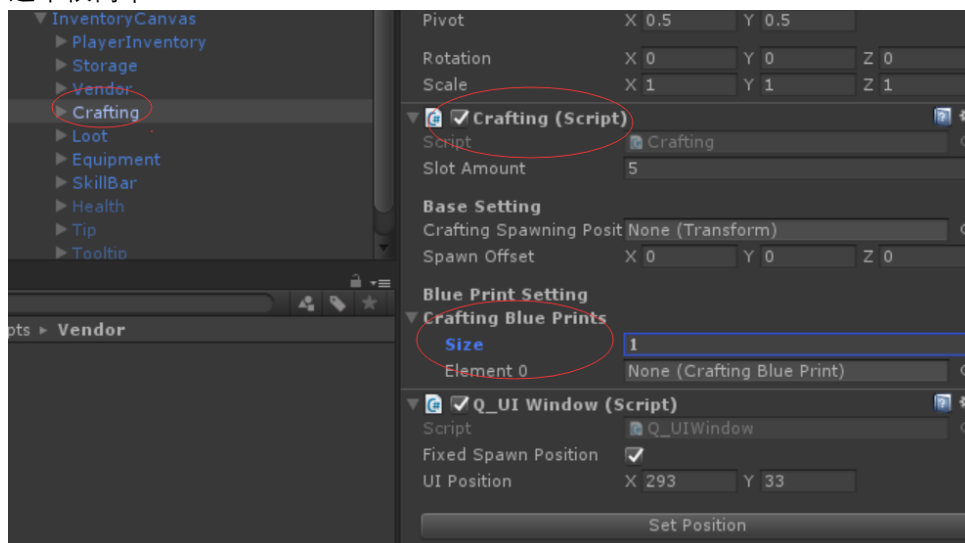
先找到这个



- **Use Default Price (使用默认的价格)** : 如果勾上的话, 会使用在 Item Editor 里面设置的 buy price。
- **Prices** : 自定义价格。(记得降序)

6.4 给合成面板添加蓝图

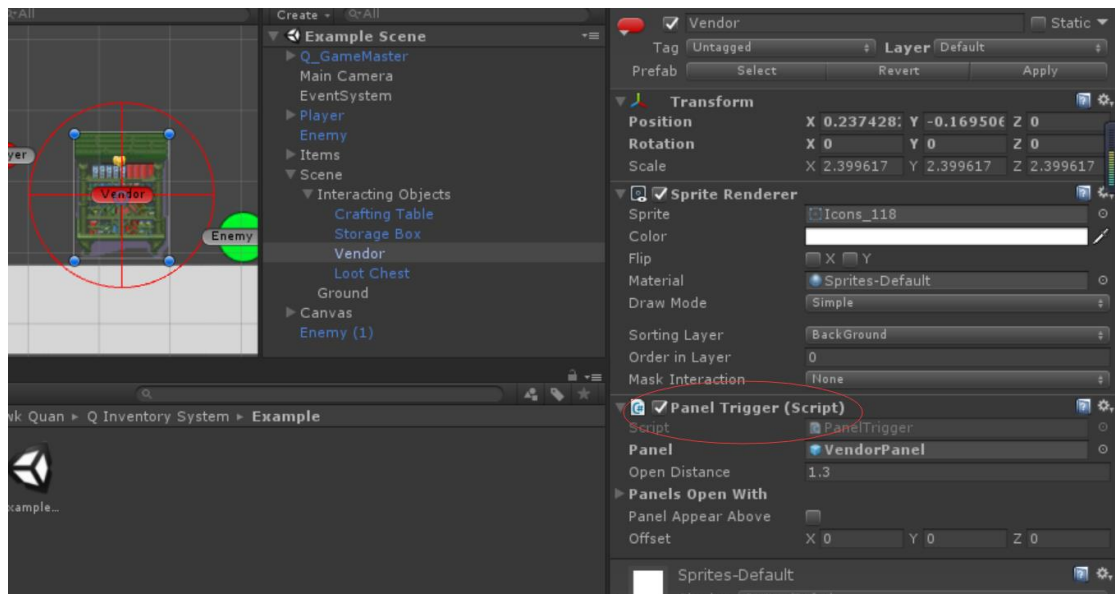
这个很简单



6.5 设置打开这些面板的触发器

选择你想与之互动的物体，然后添加 Panel Trigger 组件。

比如，商店。



这个商店物体其实并不真正需要 trigger，我是通过距离来判断触发的。

- **Panel（面板）**:就是你想与这个物体互动时打开的面板.(记住是名字带 panel 的那个不是它的父物体！)
- **Open Distance**: 决定了触发的范围（有一个简单的球型 gizmos 帮助设置）
- **Panels Open With**: 比如当你打开交易面板时，同时也想打开其他面板，比如玩家背包，就是在这里设置。
- **Panel Appear Above**: 这个会让面板在物体正上方出现，但是你大多时候需要调一下 offset 的值来确保好看。

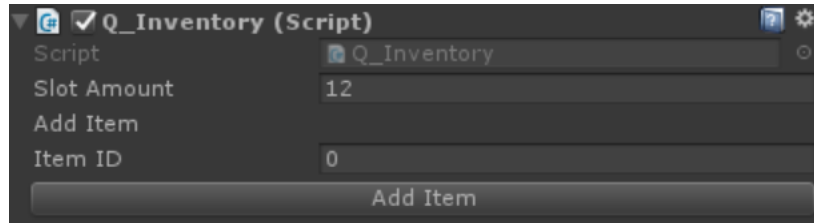
6.6 运行一下场景看现在能不能用了

如果你还没有写玩家移动脚本的话，你可以暂时使用 Q_SimpleMove 脚本来让玩家移动，但是这个脚本写的很简单，你最好还是之后有时间自己写一个好的。

7 脚本组件

1.Q_Inventory

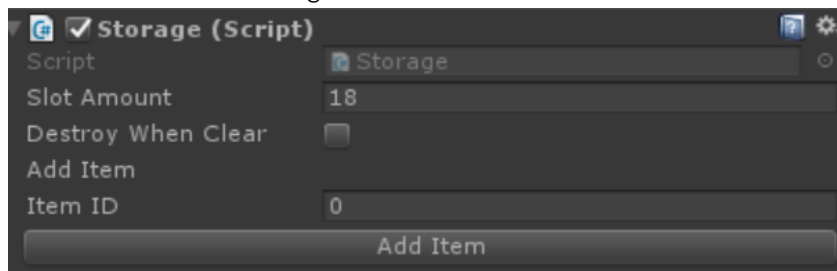
This one is used on playerInventory.



- **Slot Amount**：设置对应面板的格子数量（最好比你一开始要添加的物品数量多）
- **Add Item**：在场景运行时使用用来添加物品的，输入 ID 号，点击 button 即可。

2.Storage

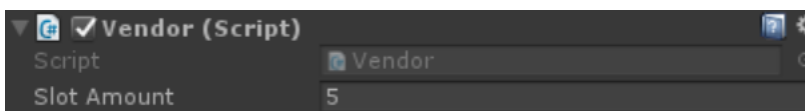
This one is used on Storage or Loot



- **Slot Amount**：设置对应面板的格子数量（最好比你一开始要添加的物品数量多）
- **Destroy When Clear**：在这个面板的所有物品被拿走后摧毁这个面板. (一般用在宝箱之类的上面)
- **Add Item**：同上

3.Vendor

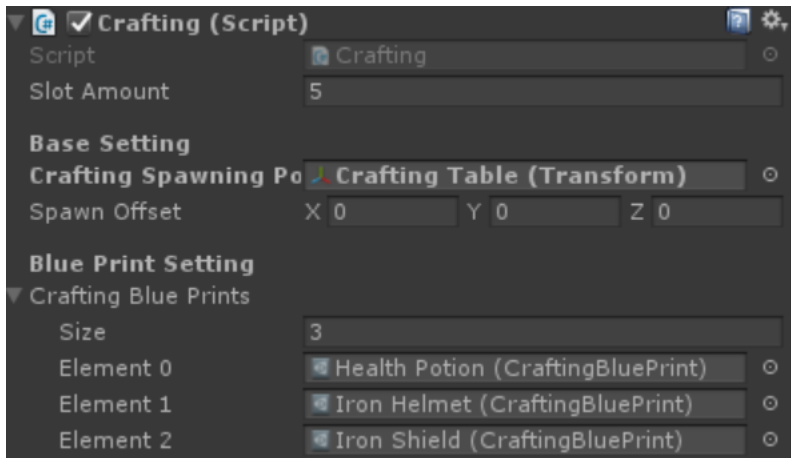
This one is used on the Vendor



- **Slot Amount**：设置对应面板的格子数量（最好比你一开始要添加的物品数量多）

4.Crafting

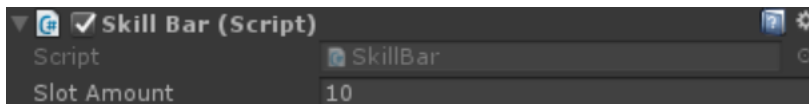
This one is used on the Craft



- **Slot Amount** : 设置对应面板的格子数量 (最好比你一开始要添加的物品数量多)
- **Crafting Spawning Position and Spawn Offset** : 当成功合成物品时物品掉落的位置
- **CraftingBlueprints** : 添加蓝图.

5.SkillBar

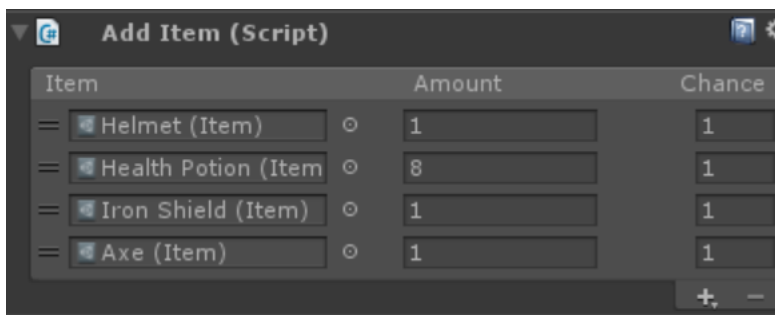
This one is used on the Skillbar



- **Slot Amount** : 设置对应面板的格子数量 (最好比你一开始要添加的物品数量多)

6.Add Item

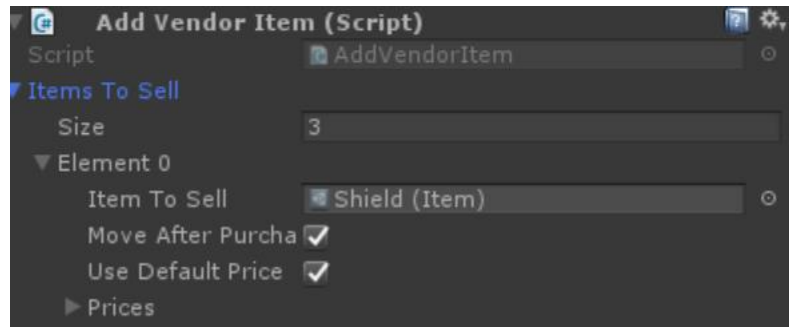
Used on the playerinventory, storage or loot to Add new Items.



- **Chance(Float)**: 成功添加物品到面板的几率, 每个物品单独计算 (between 0 and 1)

7.Add Vendor Item

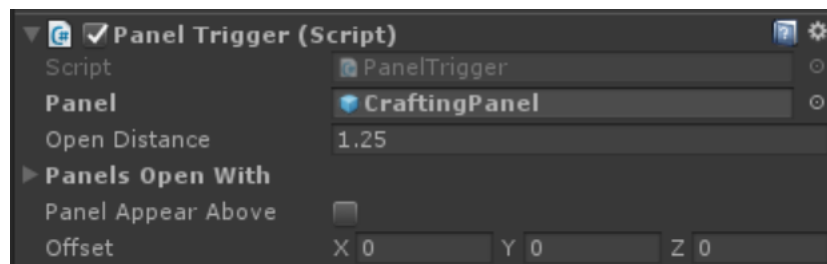
Used on the vendor.



- **MoveAfterPurchase** : 在玩家购买一次后从商店移除（就是只能买一次）
- **Use Default Price** : 使用默认价格，即为你在 Item Editor 设定的 buy price。
- **Prices** : 记住降序

8.Panel Trigger

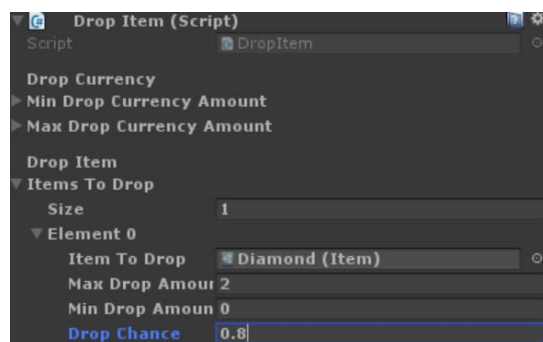
使用这个脚本的物体不需要真正地 trigger。



- **Panel（面板）**:就是你想与这个物体互动时打开的面板.(记住是名字带 panel 的那个不是它的父物体！)
- **Open Distance** : 决定了触发的范围（有一个简单的球型 gizmos 帮助设置）
- **Panels Open With** : 比如当你打开交易面板时，同时也想打开其他面板，比如玩家背包，就是在这里设置。
- **Panel Appear Above** : 这个会让面板在物体正上方出现，但是你大多时候需要调一下 offset 的值来确保好看。

9.Drop Item

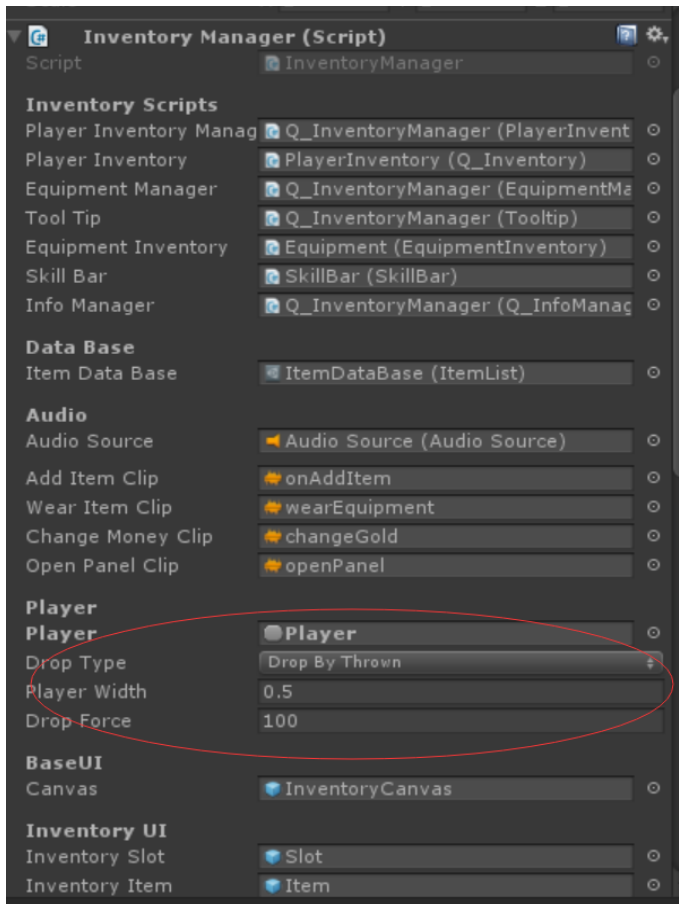
Used on the gameobject like monsters as loots.



- **Drop Currency (掉落货币)** : 就是你想掉落的钱的数量, 在这个区间随机.
- **Drop amount (掉落数量)** : 掉落的数量
- **Drop Chance (掉落几率)** : Drop Amount 配合使用

10.Inventory Manager

管理绝大部分脚本、预制体、资源的脚本



- **Drop Type** : 物品丢弃方式
- **PlayerWidth** : 在方式为 Drop By Thrown 时需要设置 (防止物品丢弃时立马捡起)
- **DropForce** : 在方式为 Drop By Thrown 时需要设置 (物品扔多远)

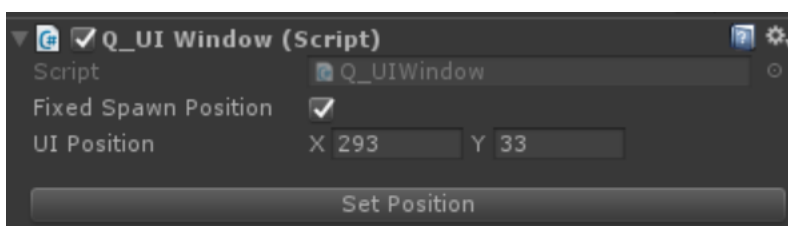
11.Q_UI Drag

用在你想要实现 UI 拖拽效果的 UI 上



12.Q_UI Window

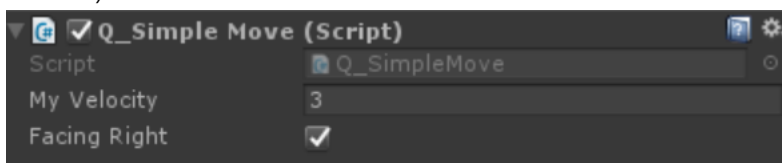
用在你想要固定出现位置的 UI 上 (不要与 Panel Trigger 的 open above 一起使用)



将第一个值勾上，然后将面板拽到你想要位置，按下那个 button 即可。你也可可是直接输入数据。（这个就不用按 button 了）

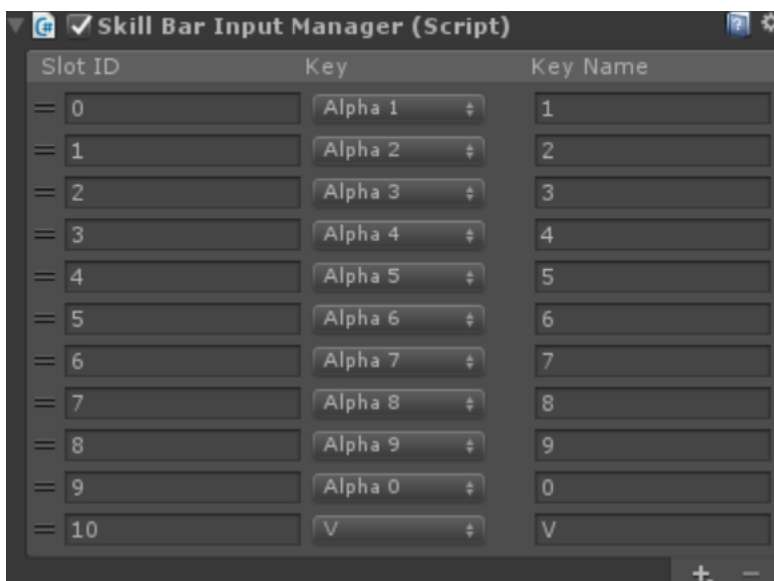
13.Q_Simple Move

一个用于解决移动脚本还没写好的尴尬的临时解决方案. (Collider 2D and Rigidbody 2D needed)



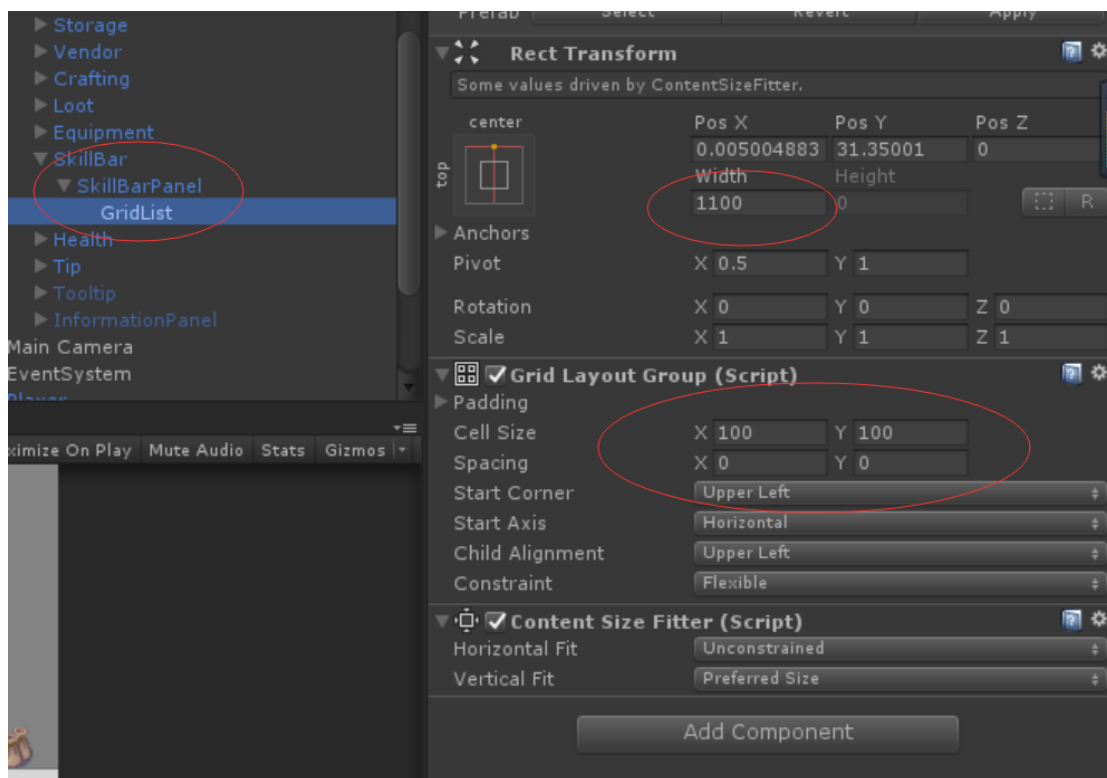
14.Skill Bar Input Manager

Used on the SkillBar



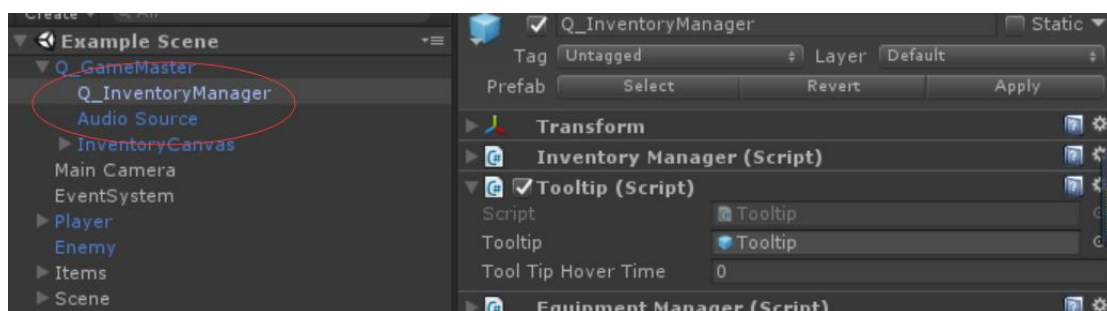
- **SlotID**：就是你想要这个键位去控制的格子
- **KeyName**：格子 text 展示的内容，如果为空，则展示设置的键位的名字，比如图中的 Alpha 1。

如果你想要增加更多的键，看下面



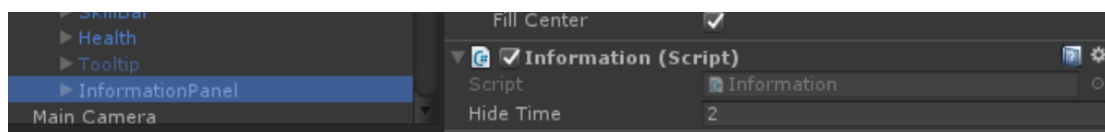
到这里来设定这个东西的宽度 (Width) 和格子的大小 (Cell Size)，比如格子数为 10，每个格子宽 100 (cellsize.x)，那么宽度就必须大于等于 1000，不然有些格子在屏幕上就看不见了。

15.Tooltip



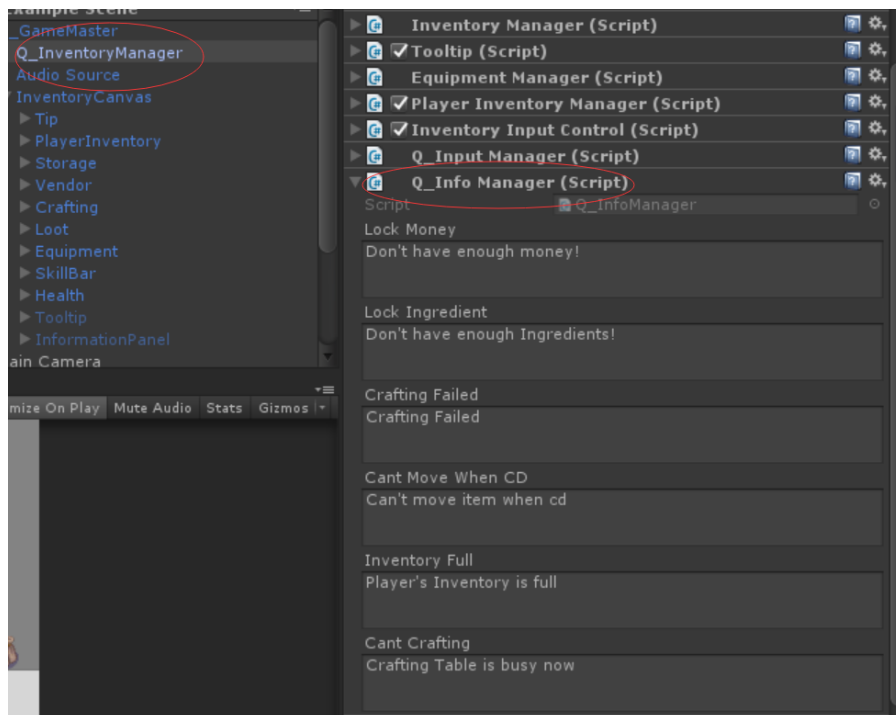
- **Tooltip**：就是提示框的 UI 物体
- **ToolTipHoverTime**：当鼠标放到物体上时，提示框延时出现的时间
- 提示框会跟位置自动调整 Pivot 来防止其超出屏幕范围。

16 Information



- **HideTime**：当信息框出现后又自动消失的时间。

17 InformationManager



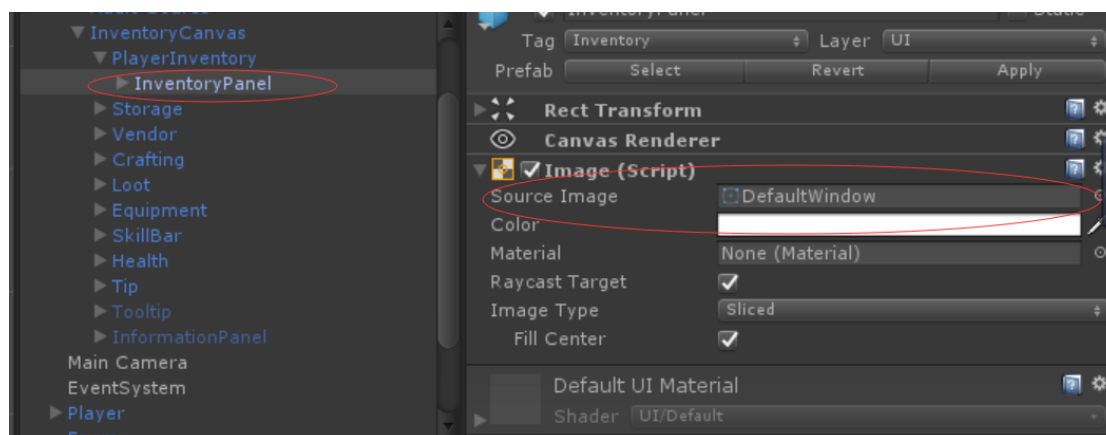
- 一些提示信息

8 自定义

警告！不要改变预制体的结构、顺序和名字。自定义时改变位置、大小、图片等即可。因为有些地方我用的是 `transform.find("")` 和 `getchild(0)` 代码。

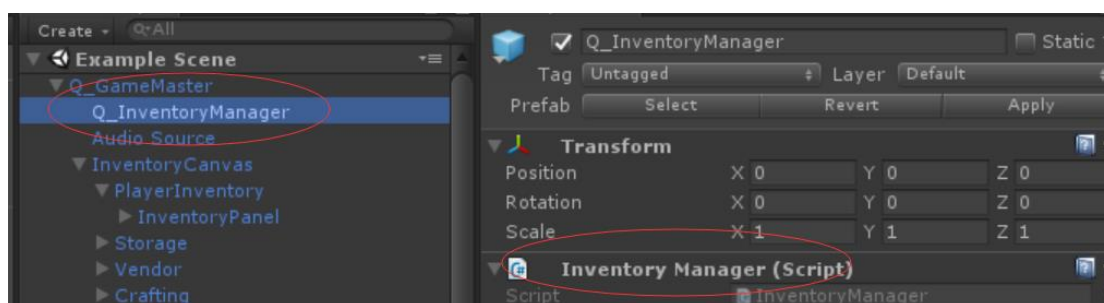
8.1 自定义面板

找到名字带有 panel 的 UI 物体就像图里的一样，改变它们的 Source Image 即可。



8.2 自定义物品格

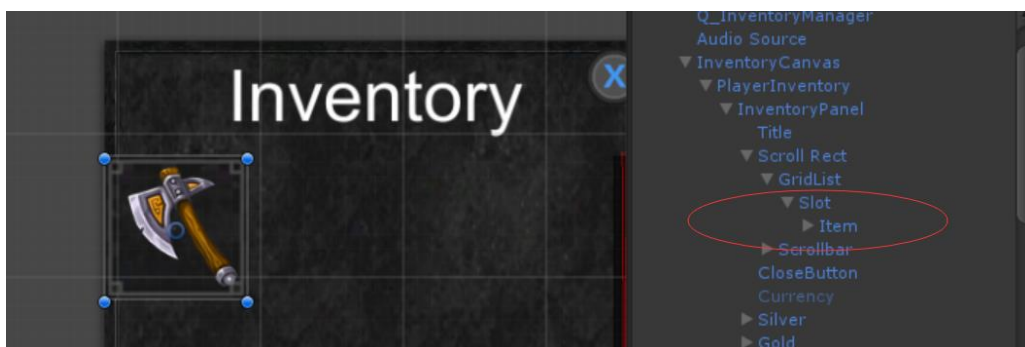
在 InventoryManager 中找到你想要自定义的格子



双击它就会在 Project 中自动选中这个预制体，改变 Source Image 即可。（大小是在 GridList 的组件 Content Size Fitter 中设置）

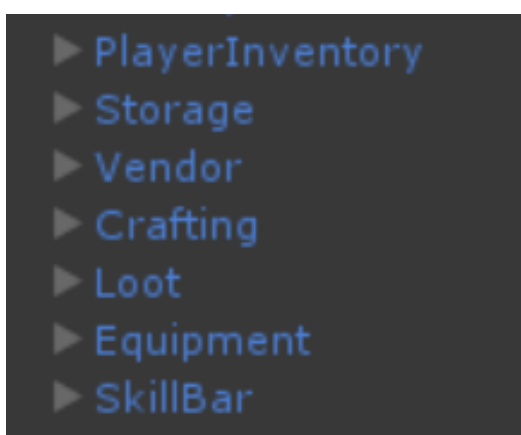
8.3 自定义物品(不是我们在 Item Editor 中的那个物品)

和上面格子一样，（我建议是和格子一起拖到场景中的 GridList 下设置，这样方便看效果，CD 的 Source Image 和 Num 的位置也就可以顺便设置了，如下图）



8.4 添加更多的面板

只需要复制之前有的面板即可



8.5 自定义提示框

不得不承认我自己做的是有点丑_(:°3」∠), 所以设计你自己的吧
找到并打开 Tooltip.cs file, 在这些方法中设计

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

```
void AddSellPriceInformation()
{
```

```
void AddVendorInformation()
{
```

```
void AddConsumableInformation()
{
```

```
void AddEquipmentInformation()
{
```

```
void AddCraftingInformation(ItemData itemdata)
{
```

9 其他

如果你有遇到任何 bug 或者是有任何问题或是任何建议。

联系我！

- Email hawk_quan@163.com .
- Twitter https://twitter.com/Hawk_Quan
- 也可以在评论区给我留言即可
- 在 B 站也会上传一些视频教程 <https://space.bilibili.com/28260941/#/>