

ULTIMATE INVENTORY PLUS

With :

Crafting, Equipment System, Furnace System
Item Spawner System, Building System (+ UI)
Door Mechanism, Plant Growing System

Welcome to UInventory System

This is the official documentation for UInventory Plus System based on Version IV Beta. In this documentation you can find all the features that this system has with examples. Remember that if you have any question after reading the documentation you can email at : paraskevlos@yahoo.gr and you will receive the answer within 7 days at max.

This is a Beta version, it is stable but a few features does not included.

ULTIMATE INVENTORY

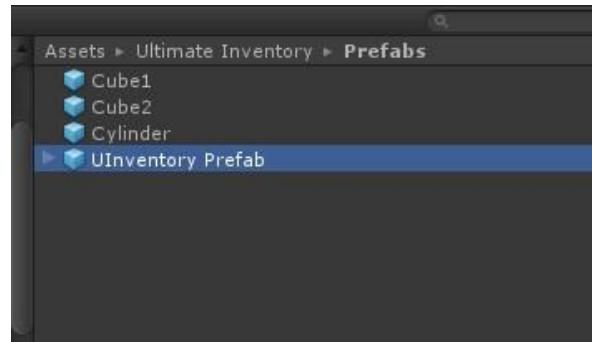
Contents

- 1.) Adding Inventory Into Your Scene [Page : 3]
- 2.) Getting started with Ultimate Inventory inspector [Page : 4]
- 3.) System Settings [Page : 5 - 6]
- 4.) All Items Tab [Page : 7 - 8]
- 5.) Equipped Items Tab [Page : 9]
- 6.) GUI Settings [Page : 10]
- 7.) Audio Settings [Page : 11]
- 8.) Custom Activation [Page : 12 - 13]
- 9.) Check For Equipped Item [Page : 14]
- 10.) Save And Load Inventory [Page : 15]
- 11.) Crafting Settings [Page : 16]
- 12.) Drag & Drop [Page 17]
- 13.) Player Equipment Settings [Page 18]
- 14.) Stacking Settings [Page 19]
- 15.) Item Spawner [Page 20]
- 16.) Door System [Page 21]
- 17.) Documentation's End [Page : 22]

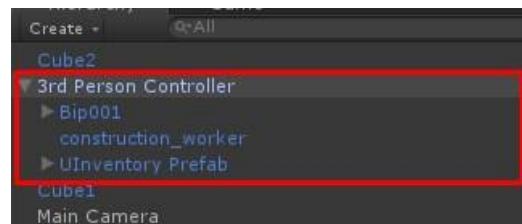
ULTIMATE INVENTORY

1.) Adding UIInventory into your scene

It's very easy to add UIInventory into your project, all you have to do is to drag and drop the UIInventory Prefab which is located at Ultimate Inventory >> Prefabs.



Then child the prefab into your player game object.



That's all you have to do, UIInventory System is now implemented into your scene.

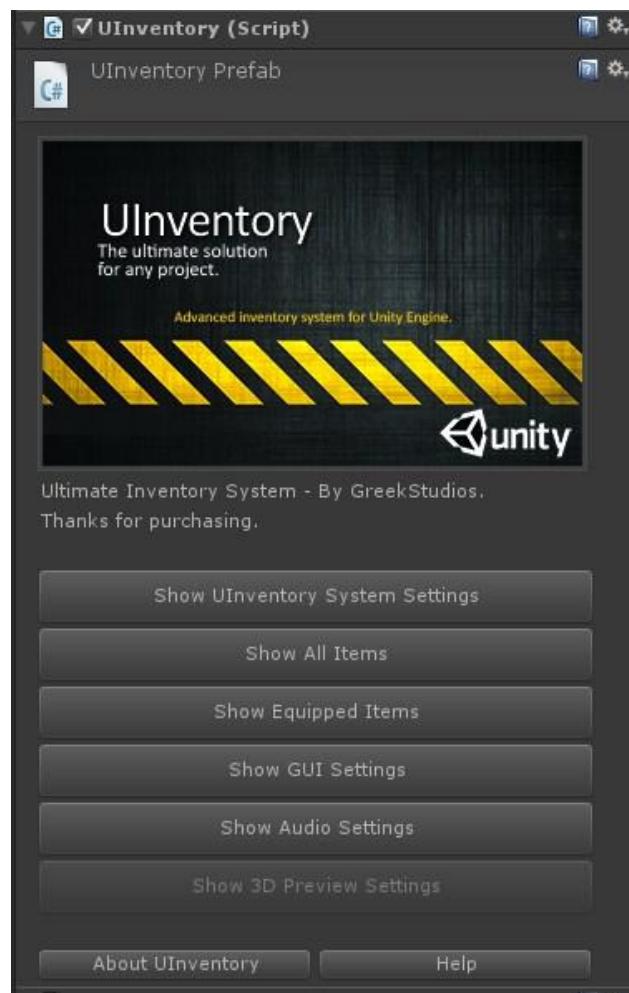
ULTIMATE INVENTORY

2.) Getting started with Ultimate Inventory inspector.

As you can see the settings of Ultimate Inventory are separated on five categories.

- 1.) System Settings
- 2.) All Items
- 3.) Equipped Items
- 4.) GUI Settings
- 5.) Audio Setting

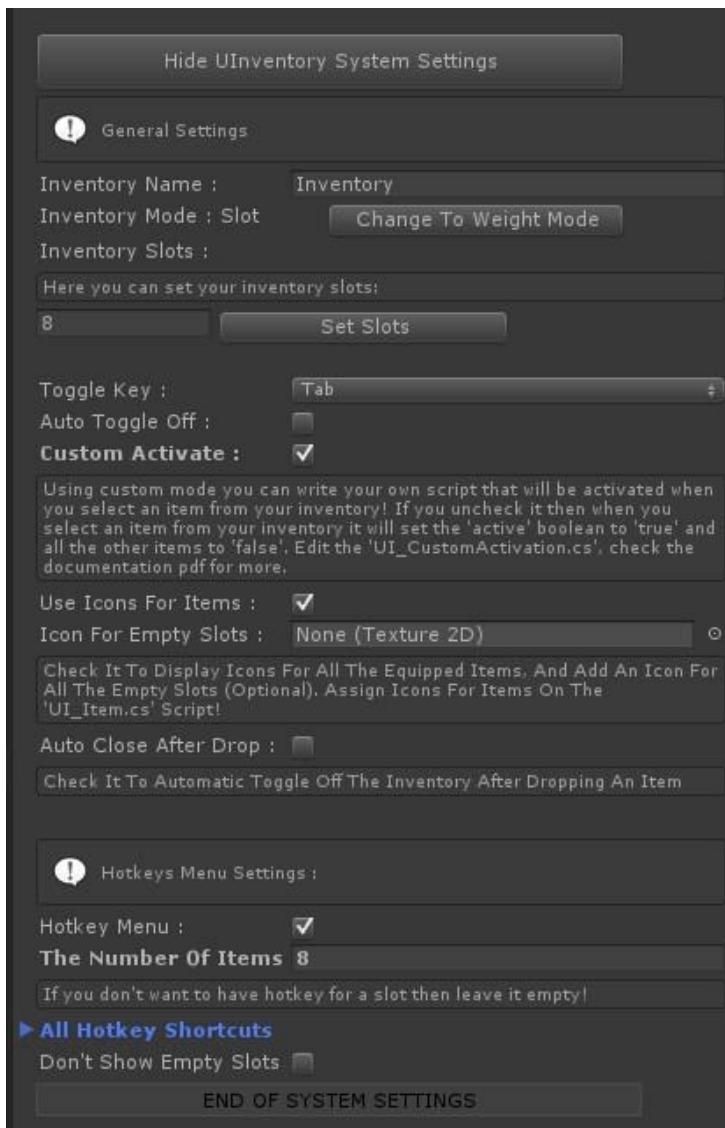
We will cover all of them separately.



3.) System Settings

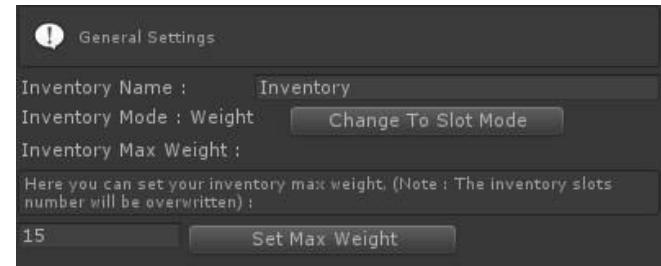
On system settings you will find the common settings of the Ultimate Inventory.
On this chapter we will cover each one separately.

Lets expand the System Settings tab,



Inventory Name is a string that will be displayed at the inventory box if you don't select a background image.

Inventory Mode is the way the inventory will work, there are two supported modes, the Slot mode and the Weight mode. You can change the mode by clicking the 'Change To Weight/Slot Mode' button. Using the **Slot Mode** you can set up how many slots your inventory will have, (note that the number can change at runtime). There is not limit on the number of slots so you can use as many as you want, set the slots by typing the number at the textbox which is left from the 'Set Slots' button, then click it to apply the new number. Using the **Weight Mode** the inspector will change as you can see bellow :



As you can see now you can set the maximum weight that the inventory can hold. Use the textbox which is left from the 'Set Max Weight' button.(The weight of the items is located at the UI_Item.cs script, we will cover it later).

Toggle Key is a KeyCode variable, that key will toggle your inventory on / off .

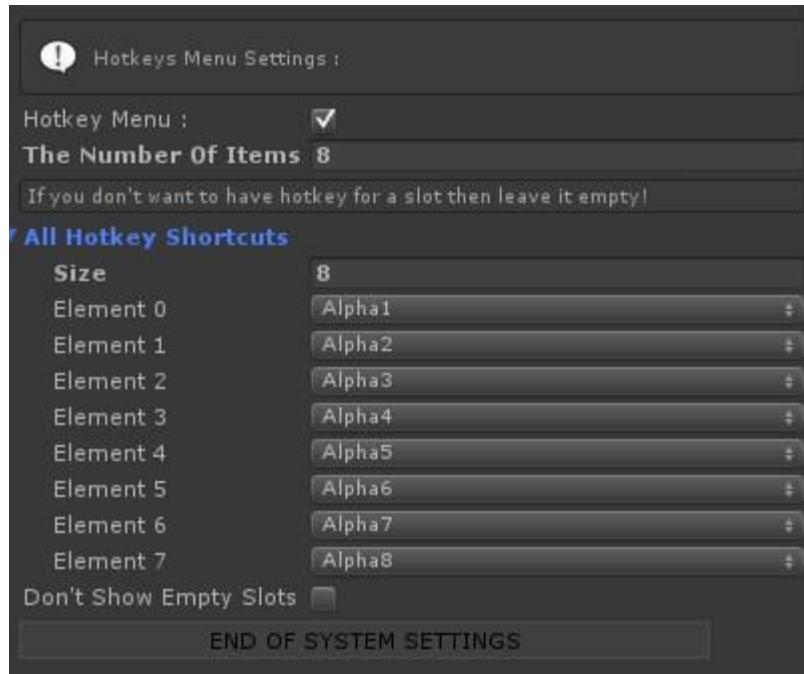
Auto Toggle Off is a boolean, if it's checked then a textbox will appear, insert the time you want (in seconds). Then the inventory will automatically close after that time.

Custom Activate is a boolean, if you set it to false then whenever you use an item it will just say the name of it, so make sure you have check it. When checked whenever you use an item a custom code that you will write will be executed.(Go to page x)

Use Icons For Items is a boolean, if checked then instead of the item's name it will draw the item's image(which is assigned into the item script).

Icon For Empty Slot is a 2D Texture, if null then the empty buttons will have "Empty" (or the selected text), if you assign an image then it will draw the image.

Auto Close After Drop is a boolean, if it's checked then when ever you use an item or you drop it the inventory will close automatically.



Hotkey Menu is a boolean, if checked then a quick inventory will be enabled in game.

Number Of Items is an integer, you can set the number of slots you want to have on your quick inventory.

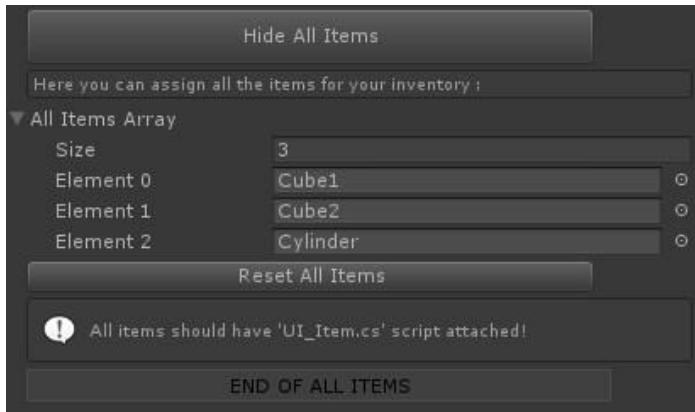
All Hotkey Shortcuts, the size of this array will be the same with the number of items, so you can't change the size direct. On this array you can set the shortcuts for each slots that exists in the quick inventory.

Don't Show Empty Slots is a boolean, if checked then the quick inventory will only draw the slots which contains an item.

We have just covered all system settings.

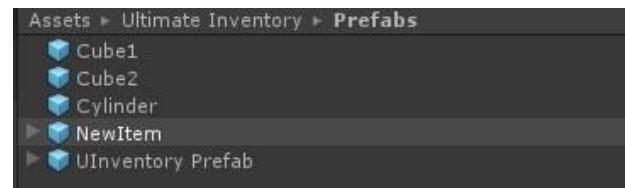
ULTIMATE INVENTORY

4.) Lets now expand the All Items tab



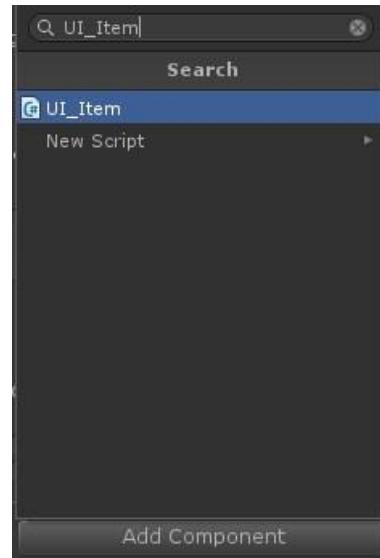
This is an array of gameobjects. Here you will add all the pickable items. Make sure to add the script UI_Item.cs on each item.

Lets add one more item here, first locate the prefab of the item you like,

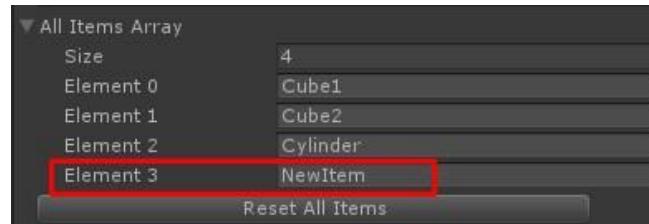


Select the prefab and click **Add Component** button and type **UI_Item**

and now add it.

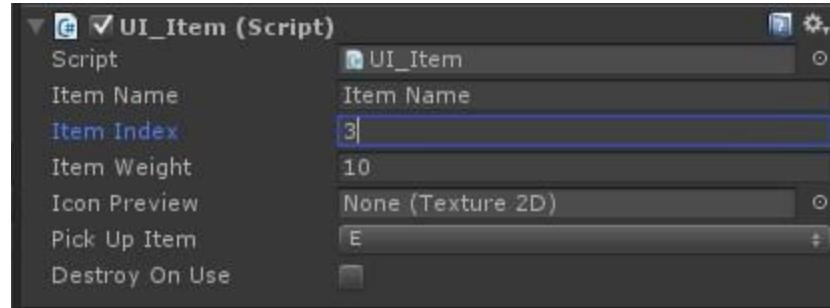


One very important step is to change the item index on your new item, but first let's add it into the array.



Let's now take a look on the inspector of our new item.

Here is the inspector of our new prefab.



Item Name is a string that represents the item's name.

Item Index is an integer which represents the index of the item into the All Items Array

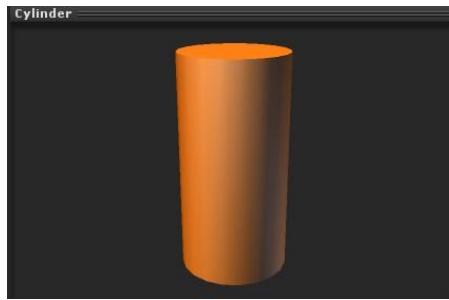


As you can see the index is the number 3, so insert it into the Item Index textbox.

Item Weight is a float which represents the weight of the item and it used only if the inventory mode is the "Weight Mode".

Icon Preview is a 2D Texture image. If you have checked the Use Icons For Items on System Settings then you have to assign an image of the current item.

Note that you can easily take one screenshot from the preview window



Pick Up Item is a keycode, select the key you want the user to press in order to pick up the key (Default is E)

Destroy On Use is a boolean, if checked it would remove the item from your inventory upon use. [Suitable for food/water or another one use items.]

We just finished this chapter.

5.) Let's now expand the Equipped Items tab

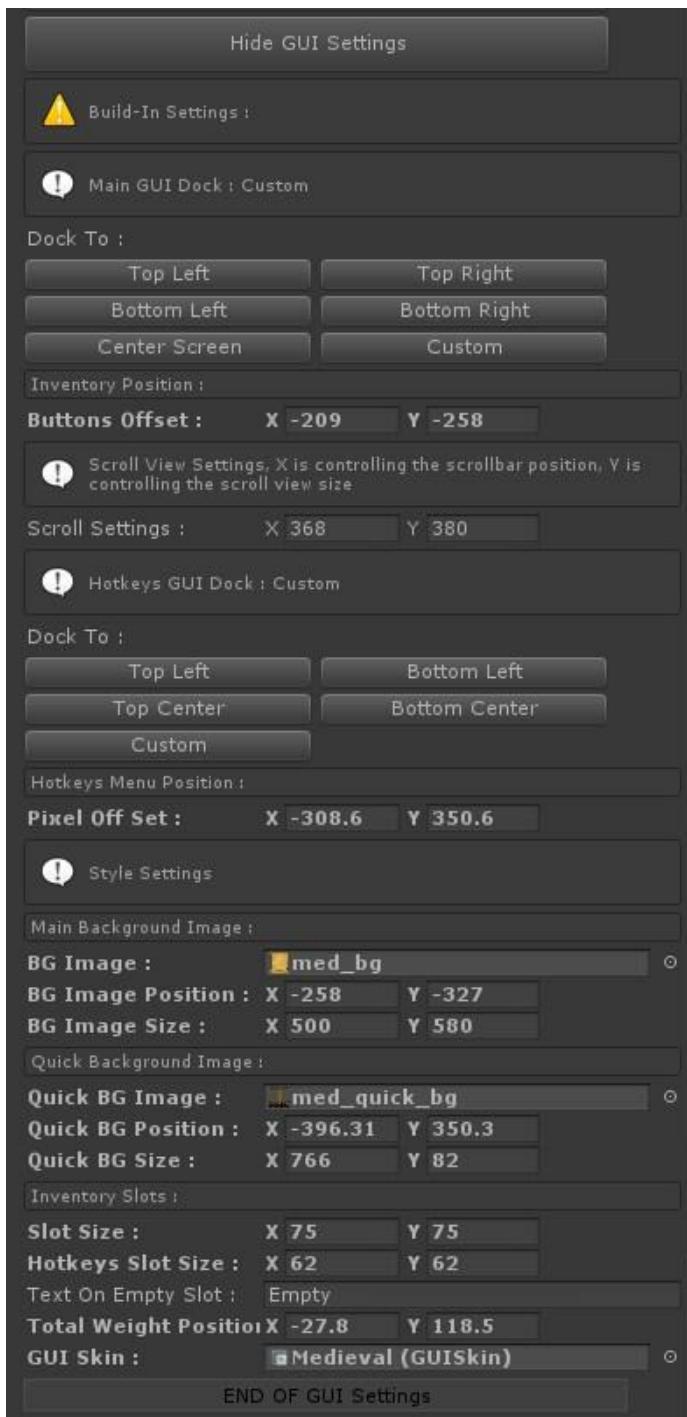


That's the array with all the equipped items. From default the array is empty, however you can assign the items you want to start with. [Make sure that the items are on the All Items List.]

The size is changed automatically when you change the slots from the system settings, so you can modify it directly.

ULTIMATE INVENTORY

6.) Let's now expand the GUI Settings tab



For the main inventory, you can dock the slots at Top Left or Right, bottom left or right and center screen, or you can use custom rect position (based on Screen width and height).

The Buttons Offset (in pixels) is the position of the slots.

Scroll Settings X is the horizontal position of the scrollbar and Y is the size of the scroll view.

The quick GUI can be docked at top,bottom left and right or top,bottom center. You can also use custom position.

The Pixel Offset is the position of the quick slots.

BG Image is the background of the main inventory.

BG Image Position and BG Image size are for the main inventory.

Quick BG Image is the background of the quick inventory, Quick BG Position and size again for quick inventory.

Slot size is the size of the slots on the main inventory.

Hotkeys slot size is the size of the slots on the quick inventory.

Text On Empty Slot is the text that would be appeared on empty slots (if the empty texture is empty).

Total weight position is the position of the label which displays the weight info (if is in weight mode).

GUI skin is the main skin for the gui.

7.) Audio Settings



Pickup item sound is the sound which will be played when you pick up an item.

Drop item sound is the sound which will be played when you drop an item.

Use item sound is the sound which will be played when you use an item.

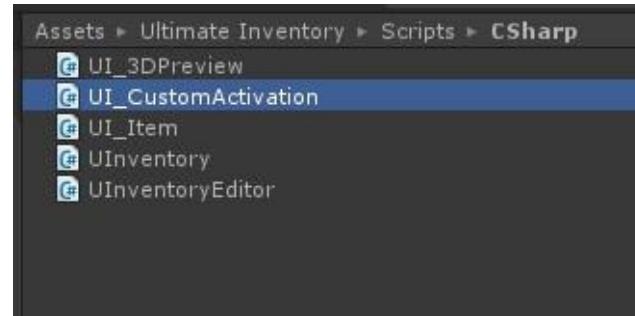
Full sound is the sound which will be played when you are full and you try to pick up an item.

Volume is a float which set the volume for all sounds from the inventory.

ULTIMATE INVENTORY

8.) Custom Activation

On this chapter we will make the inventory to execute custom code when you select an item. First, locate the UI_CustomActivation.cs script which is located at Ultimate Inventory >> Scripts > CSharp



Open it with your editor, ignore all the code and just focus on the first void which is called "CustomActivate"

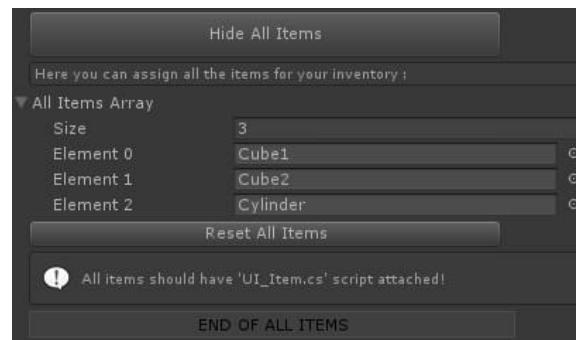
```
public void CustomActivate(int itemIndex, bool destroyAfterUse)
{
    /*
     * Type your custom activation code here
     * use :
     * GameObject.FindGameObjectWithTag("UIInventory").GetComponent<UIInventory>().currentItems[itemIndex]
     * To Access The GameObject With Index the itemIndex integer
     */

    //Do Not Remove This Code
    if (destroyAfterUse == true)
    {
        GameObject.FindGameObjectWithTag("UIInventory").GetComponent<UIInventory>().currentItems[itemIndex] = null;
    }
    //End Of Not Removable Code
}
```

As you can see the code is already commented. So if you want to access the object which activated then use :

`GameObject.FindGameObjectWithTag("UIInventory").GetComponent<UIInventory>().currentItems[itemIndex]`

The itemIndex is the index of the object which was activated. So if the itemIndex is equals to 0 then the Cube1 was used.



So, let's write an example code :

In order to get the name of the activated object you can use this :

```
GameObject.FindGameObjectWithTag("UIInventory").GetComponent<UIInventory>().currentItems[itemIndex].GetComponent<UI_Item>().itemName
```

So here is the example which will print the name of the activated object

```
public void CustomActivate(int itemIndex, bool destroyAfterUse)
{
    /*
     * Type your custom activation code here
     * use :
     * GameObject.FindGameObjectWithTag("UIInventory").GetComponent<UIInventory>().currentItems[itemIndex]
     * To Access The GameObject With Index the itemIndex integer
     */
    Debug.Log(GameObject.FindGameObjectWithTag("UIInventory").GetComponent<UIInventory>().currentItems[itemIndex].GetComponent<UI_Item>().itemName);

    //Do Not Remove This Code
    if (destroyAfterUse == true)
    {
        GameObject.FindGameObjectWithTag("UIInventory").GetComponent<UIInventory>().currentItems[itemIndex] = null;
    }
    //End Of Not Removable Code
}
```



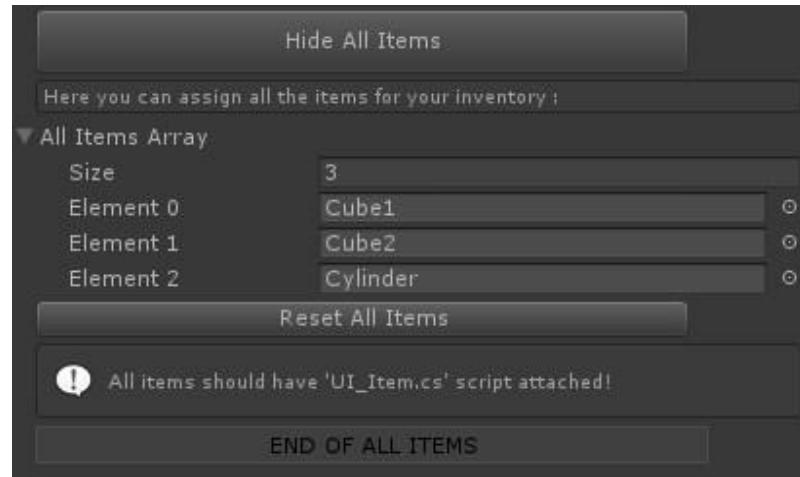
ULTIMATE INVENTORY

9.) Check For Equipped Item

There is a simple way to check for a specific item. All you need to know is the item's index on the All Items Array.

```
CheckForItem(itemIndex as int, deleteOnUse as bool);
```

itemIndex is the index of the item you want, deleteOnUse is a boolean, true means that the item will be removed from your inventory upon use. An example for the Cube1 would be :



```
if (CheckForItem(0, true) == true)
{
    //You just ate cube1
}
```

ULTIMATE INVENTORY

10.) Save and Load the Inventory

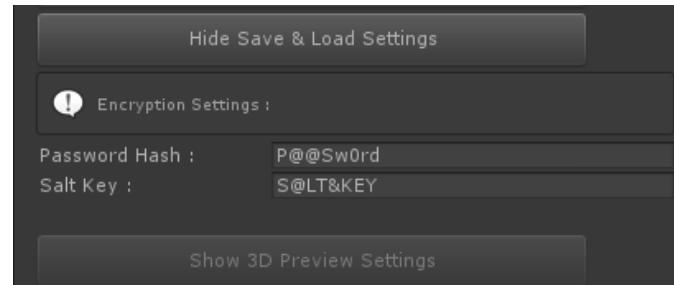
On version 1.1 a new feature was added, you are now able to save and load your inventory with one line of code, just call :

```
SaveInventory();
```

to save your inventory and :

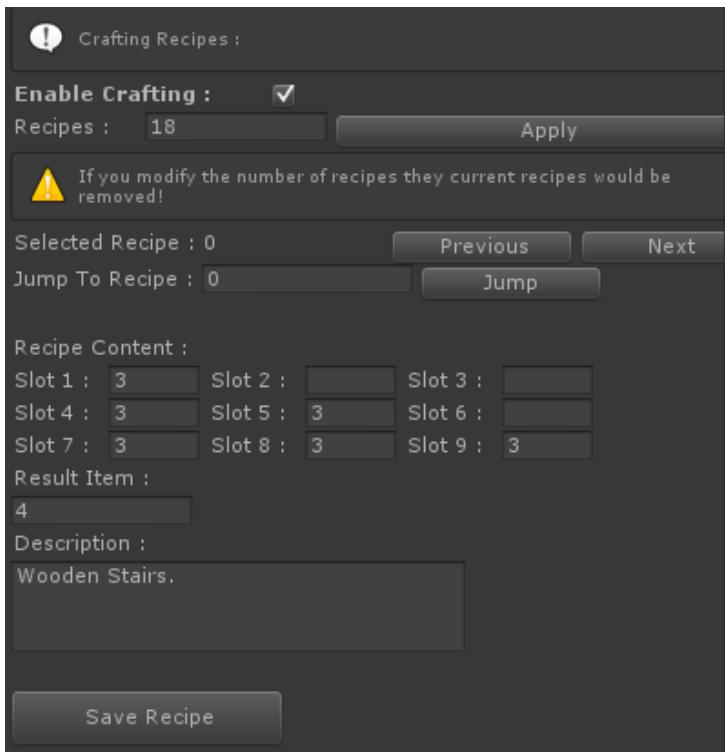
```
LoadInventory();
```

to load your inventory. Data is encrypted and you can change the password hash and the salt key.



ULTIMATE INVENTORY

11.) Crafting Settings



Toggle key is the key which toggles on/off your crafting menu, select the same key with the inventory if you want to open/close together.

Craft Slots Position is the position (x,y) of the crafting slots.

Scroll Size - Let it 250,250.

Result Slot Position : Is the position of the slot which has the crafted item.

Crafting Background is the background image of the crafting menu.

Background position and Background size are the position and the size of the background image.

Item Description Position and Item Description Size is the textarea's position and size which displays the information of the crafted item.

New Feature : You can now Import & Export your recipes.

By pressing "Export Recipes" a txt file named : UltimateInventory_Crafting_Recipes would be created into your project directory.

If you want to import a recipe list, then you have to place the file (UltimateInventory_Crafting_Recipes.txt) into your project directory and press 'Import Recipes' --> 'Import'. If there is not file into your directory then a message will appear.

Enable Crafting, check it if you want to allow the player to craft.

Recipes, is the number of the recipes, IF YOU MODIFY the number the current recipes would BE REMOVED. Selected Recipe is showing you the recipe that you are editing. Navigate through recipes using Previews , Next buttons or even jump to a specific one.

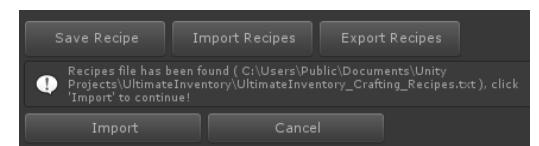
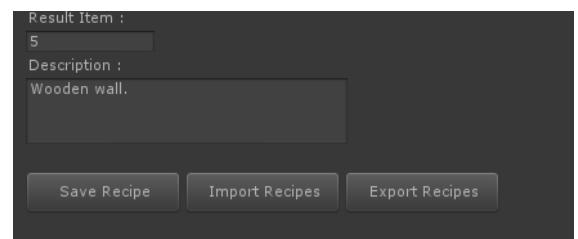
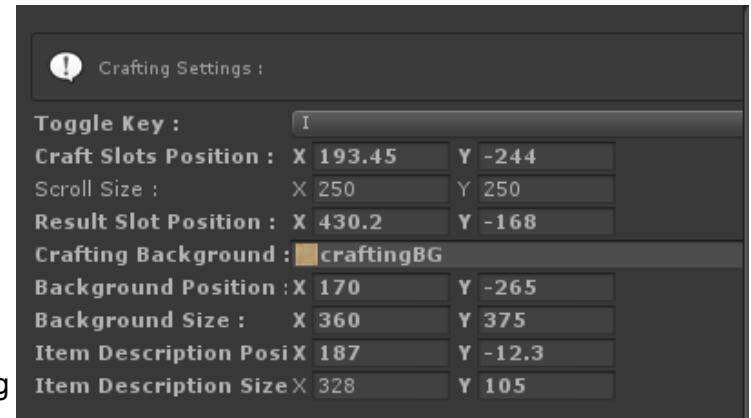
The recipe content has 9 fields, which represents the 9 slots in-game, type the element of the 'All Items' array which you want the recipe to have on each slot.

On this example Element 3 is a wooden block, and 4 is wooden stairs. On the 'Result Item' field type the id of the crafted item on the current recipe.

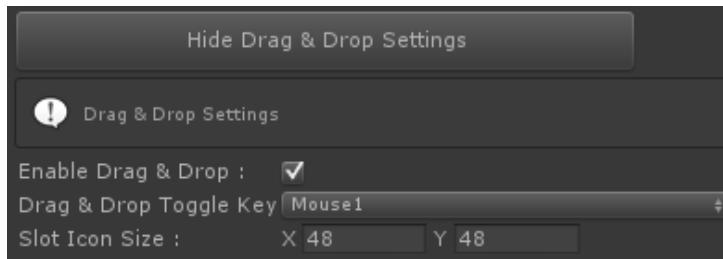
Description IS NOT MULTILINE (YET) so changing line would not be saved.

Make sure to click "[Save Recipe](#)" before you leave.

Note : If you save and you can't see the saved recipe just select an other object and then select back the UIInventory prefab.



12.) Drag & Drop



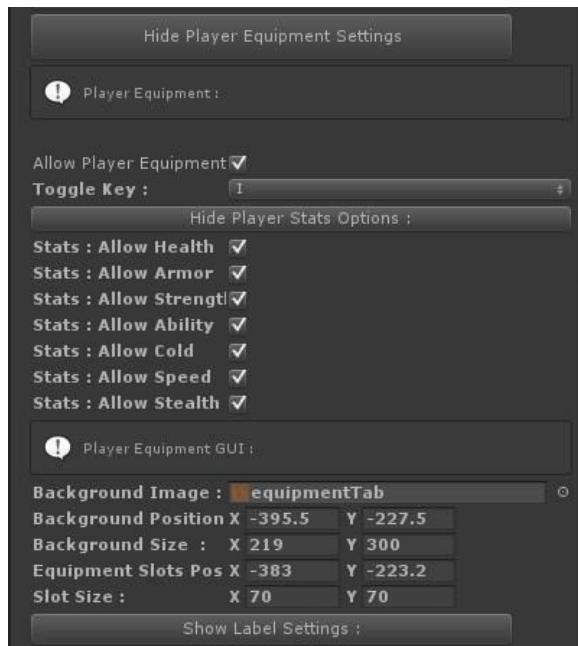
Enable drag & drop , check it if you like to have drag and drop functionality. Note that the Crafting System IS NOT WORKING **WITHOUT DRAG & DROP**.

Drag & Drop Toggle key is the key which you have to press in order to be able to drag and drop slots, when you press it then left click an item and it would be selected, move your mouse on the slot you want and left click again, if you want to switch back to the use mode (left click = use item) then press the toggle key again.

Slot icon size is the size of the icon of the dragged item.

ULTIMATE INVENTORY

13.) Player Equipment System

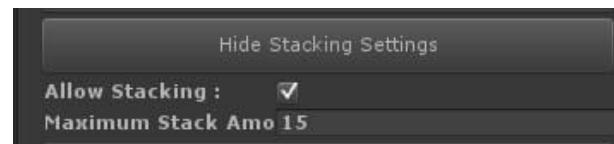


Check the Allow Player Equipment to enable the equipment system. Toggle Key is the key which will toggle the equipment system. If you choose the same key with the main inventory then the inventory and the equipment system will be toggled together. Player stats are the available stats for player equipment system, check each box for each stat you want. Background image for the player equipment system, background position and size for the same system. Equipment Slots Pos is the slot's position and size.

Label settings, you can rename and change the size and the position of each label.

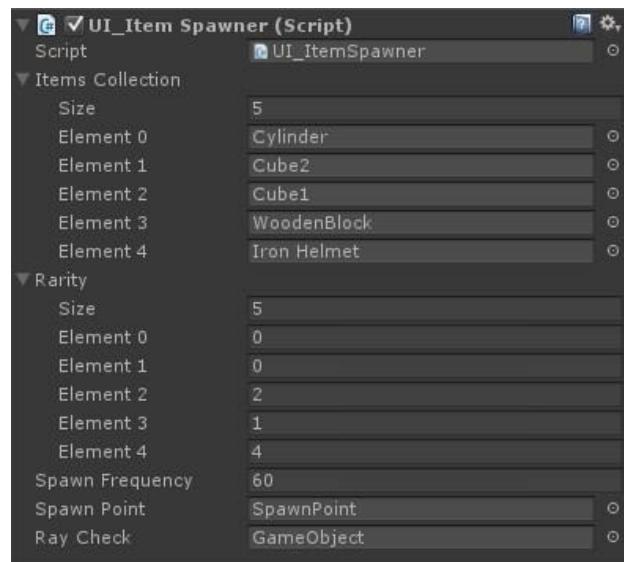


14.) Stacking System



Stacking system is currently on BETA stage, not many settings and available **ONLY FOR SLOTS** mode

15.) Item Spawner



Items Collection is the array of the items you want the system to spawn, make sure to add items which are included on the 'All Items' array. Also, Rarity is an array of integers and the size of this array MUST be the same with the Items Collection array.

How does the Rarity works : Before explaining how does it work you have to know how the system choose the items, firstly, the system chose one of the items on the 'Items Collection' randomly (each item has the same possibilites to be chosen [on our example 1/5]), then the larger the number of rarity the most rare the item will be spawned, so if you enter 0 the item will be spawned 100% if it is selected, if you place 2 then the item has 50% (1/2) to be spawned else the system will go back and do all the process again.

Spawn Frequency is a float and determines the time the system needs in order to spawn an other item when the currently spawned is gone (picked up).

Spawn Point is where the items will be spawned.

Ray Check should always be beneath the spawn point. (Check whether the item is picked up).

In order to add a spawn system just go to prefabs folder and drag into your scene the 'ItemSpawner' prefab.

16.) Door System

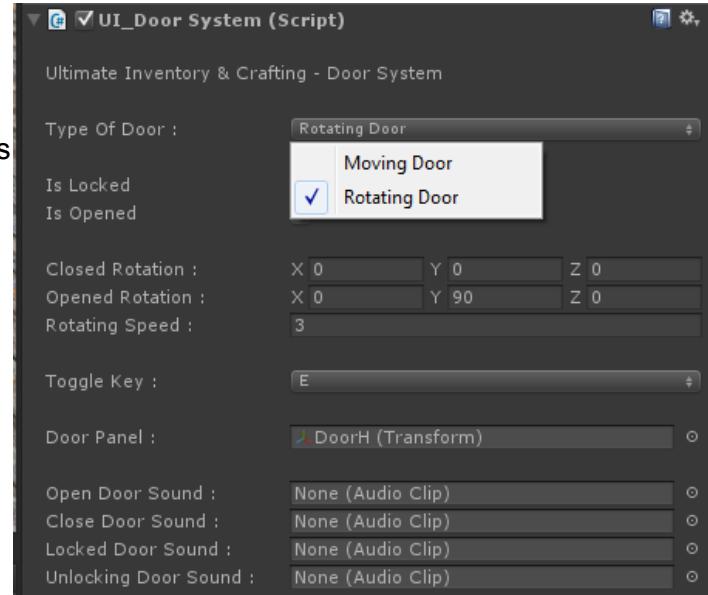
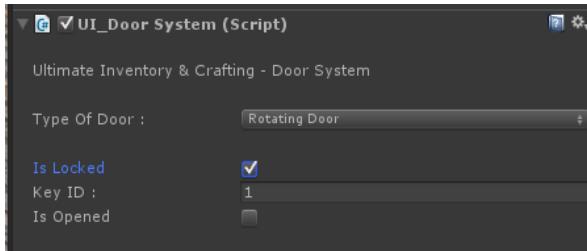
You can now add doors into your level which are compatible with Ultimate Inventory. This allows you to create locked doors which will unlock only if you have on your inventory the key with the same id with the door.

There are two types of doors, the moving door and the rotating door. Each one has closed and opened positions as well as speed variable.

There are two checkboxes, Is Locked determines whether the door is locked or not and CAN be changed through scripts. Is Opened determines the state of the door (opened/closed).

Toggle Key is the key which user has to press when he is looking on the door in order to open / close or unlock it (if locked).

If you make the door locked a new field would appear,



This door will be opened only with keys marked with id 1. In order to make a prefab to work as a key you have to add the 'UI_Key.cs' script on it and then select the key id.

Door Panel is the gameobject which will be moved / rotated when you open/close the door.

Then all the rest are the sounds the door will make when you : Open/Close/Unlock or try to open a locked door without a key.

17.) Documentation's End

That's all the documentation for Ultimate Inventory. If you have any questions please email at : paraskevlos@yahoo.gr or visit <http://greekstudios.url.ph/> .

Thanks for purchasing

Your Team :

GreekStudios



ULTIMATE INVENTORY