

Contents

[Introduction to Ultimate Inventory 5 3](#_Toc440814092)

[Downloading and Importing UInventory 5 3](#_Toc440814093)

[Downloading UInventory 5 3](#_Toc440814094)

[Importing UInventory 5 3](#_Toc440814095)

[Using Ultimate Inventory 5 4](#_Toc440814096)

[Opening the example scene 4](#_Toc440814097)

[Adding the inventory at your own scene 4](#_Toc440814098)

[Ultimate Inventory 5 – Custom Editor 5](#_Toc440814099)

[General Settings 5](#_Toc440814100)

[Setup Settings 6](#_Toc440814101)

[Inventory Settings 7](#_Toc440814102)

[UI Settings 9](#_Toc440814103)

[Editor Settings 10](#_Toc440814104)

[Adding Items On The Inventory 11](#_Toc440814105)

[Creating a consumable 16](#_Toc440814106)

[Creating an equipment 18](#_Toc440814107)

[Modifying The Inventory Slots 19](#_Toc440814108)

[Crafting System 20](#_Toc440814109)

[Adding crafting item 22](#_Toc440814110)

[Adding crafting category 24](#_Toc440814111)

[Creating blueprints 27](#_Toc440814112)

[Looting System 28](#_Toc440814113)

[Non-animated looting 28](#_Toc440814114)

[Animated looting 31](#_Toc440814115)

[Container slots 31](#_Toc440814116)

[Player Equipment 32](#_Toc440814117)

[Save System 33](#_Toc440814118)

[UInventory Namespace 33](#_Toc440814119)

[Item Spawner 34](#_Toc440814120)

[Inventory Functions 35](#_Toc440814121)

THIS DOCUMENTATION IS WRITTEN FOR THE INITIAL RELEASE OF ULTIMATE INVENTORY 5 AND THE SCREENSHOTS USED FOR THE EXPLANATION OF EACH SUB-SYSTEM MAY DIFFER FROM THE CURRENT VERSION. THE DOCUMENTATION WILL BE UPDATED ONLY FOR A SIGNIFICANT UPDATE OF THE SYSTEM.

# Introduction to Ultimate Inventory 5

Ultimate Inventory 5 (aka: UInventory 5) is an update to UInventory 4 but it is completely re-written from scratch based on Unity 5 game engine. This asset is the ultimate inventory solution for any project and it is very easy to be modified and extended with some basic knowledge and understanding of C# and Unity.

This document is the official documentation for Ultimate Inventory 5 which will cover almost all the features of UInventory and it will help you to understand how it works. Studying this documentation and experimenting on the asset will help you to understand how a complex drag & drop inventory system which includes crafting, player equipment and a lot of more features is created. *This document is written for UInventory 5.0*

# Downloading and Importing UInventory 5

## Downloading UInventory 5

You can download Ultimate Inventory 5 from Unity asset store after purchasing it for $15. If you already own UInventory 4 then you will have this new version for free despite the fact that you paid $10.

A link on the asset can be found here: [https://www.assetstore.unity3d.com/en/#!/content/19900](https://www.assetstore.unity3d.com/en/%23!/content/19900)

## Importing UInventory 5

You may import UInventory 5 on your current project or on a new one, make sure to import every each file from the asset. This version is no longer compatible with UInventory 4 and it’s recommended to not use both versions at the same project.

Last but not least Unity 4 is **not** supported, so if you intend to use Unity 4 then only UInventory 4 is compatible with it. To get UInventory 4 you have to contact at ‘paraskevlos@yahoo.gr’ and send your request as well as your invoice number of your purchase.

# Using Ultimate Inventory 5

## Opening the example scene

Into the main folder ‘Ultimate Inventory Pro’ there is a scene named ‘exampleFPS’, open this scene and you see the example scene of our asset. On this scene you can get a preview of almost every feature included at UInventory 5. Play with the system and see how it interacts with the player.

## Adding the inventory at your own scene

In order to add the UInventory 5 at your own scene all you have to do is to open the ‘exampleFPS’ scene and copy the following game objects:

* FPS Controller
* EventSystem
* Canvas

If you use a custom player game object then instead of coping ‘FPS Controller’ then copy the ‘Ultimate Inventory Pro’ game object which is located as a child in the FPS Controller. If you do this then make sure to parent the game object with your player and select Ultimate Inventory 5, then at the custom editor click at the ‘Setup Settings’ and then assign the following variables:

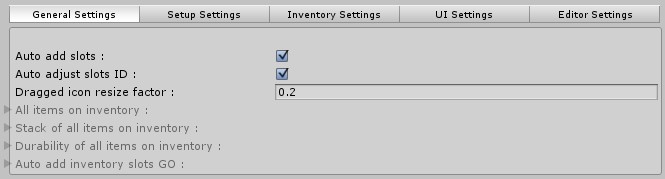
* Player game object
* Main camera

Make sure to use the correct game objects and our main camera. If your scene already contains a ‘Canvas’ then make sure to get the child of the canvas (‘Ultimate Inventory UI – Drag’) and assign it at your own canvas.

# Ultimate Inventory 5 – Custom Editor

Select Ultimate Inventory Pro game object and you will see this custom editor. There are five different tabs/categories of settings. We will take a short look on each one and we will comment each available option.

## General Settings



***Auto add slots*** is a variable (type of Boolean) which when it is checked disables the following 4 options:

* All items on inventory
* Stack of all items on inventory
* Durability of all items on inventory
* Auto add inventory slots GO

What it actually does it’s that you can simply add slots by dragging and dropping them into the canvas (view: Modifying the inventory slots) and takes care of all the code & settings stuff. It is very useful and it’s not recommended to turn it off.

***Auto adjust slots ID*** is a similar option which sets up the id of each slot automatically, it’s also not recommended to turn it off.

***Dragged icon resize factor*** is a variable (type of Float) which determines the scale of the dragged item’s icon, 1 means the full scale. The optimal value for the current inventory skin is about 0.2, however you can modify it to fit your needs.

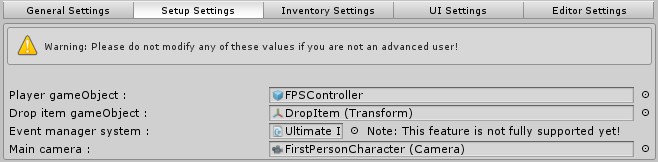
***All items on inventory*** is a list of game objects which holds all the currently items on the inventory. It uses System.Collections.Generic namespace and it can be modified dynamically. If you choose to add and adjust the inventory slots manually then you have to set up the correct size of the list.

***Stack of all items on inventory*** is also a list but of integers this time. It holds the stack of each slot in the inventory. Needs to set up the correct size if you chose to manual set up the inventory.

***Durability of all items on inventory*** is also a list of floats this time and stores the durability value of each object if it has. Manual set up also means that you need to manually set up the size of the list.

***Auto add inventory slots GO*** is also a list of game objects and holds the actual inventory slot game objects (the GUI). This is automatically set up if you don’t chose the manual way.

## Setup Settings



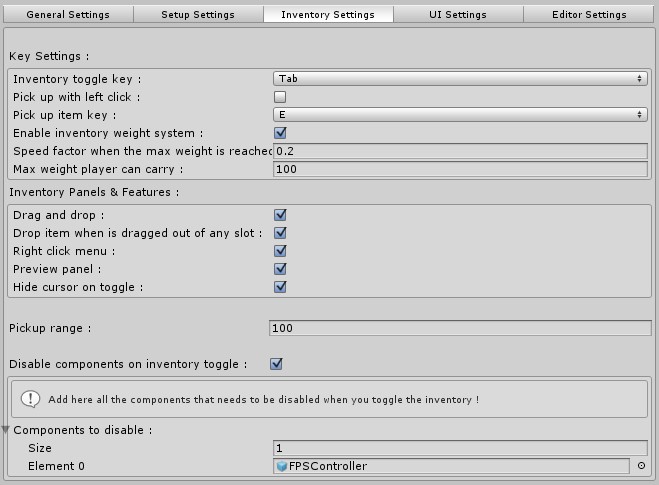
***Player gameObject*** is a variable (type GameObject) and represents the actual player. On the example scene the player is the default FPSController which is include into the standard assets however if you add the inventory on your own scene and you use any kind of player you need to parent the Ultimate Inventory Pro at your player and then assign this variable with your player.

***Drop item gameObject*** is a variable (type of Transform) and it is just a placeholder for the position of the items you may drop from your inventory and it is not visible.

***Event manager system*** is not supported yet.

***Main camera*** is a variable (type of Camera) and it should be your main camera. It is used for ray casting proposes in order to pick up items.

## Inventory Settings



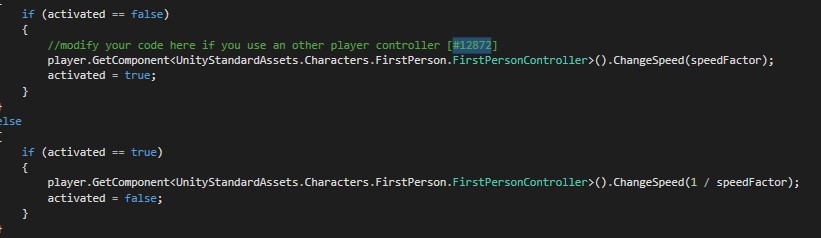
***Inventory toggle key*** is a variable (type of KeyCode) and it is the key which will toggle your inventory.

***Pick up with left click*** is a variable (type of Boolean), when it is checked then the player can pick up the items by left clicking on them, when it’s not checked then the player picks up the items using a key.

***Pick up item key*** is a variable (type of KeyCode) and represents the key which the player should use in order to pick up an item.

***Enable inventory weight system*** is a variable (type of Boolean), when it is checked then the player can carry up to a weight limit of items, carrying more than this weight makes the player move slower. If you don’t use the FPSController then you need to open the ‘UI\_Inventory.cs’ script, press CTRL+F and type ‘#12872‘ and search for it.

Then as you can see on the image there is a simple line of code which changes the speed of the player, comment this line and add your own line which affects your player.



***Speed factor when the max weight is reached*** is a variable (type of Float) which determines the speed of the player when he carries more than the max allowed weight.

***Max weight player can carry*** is a variable (type of Float) which determines the maximum weight allowed to be carried.

***Drag & Drop*** is a variable (type of Boolean), when enabled then the player is allowed to use the drag and drop feature on the inventory.

***Drop item when dragged out of any slot*** is a variable (type of Boolean), when enabled items which are dragged out of any slot will automatically be dropped out of the inventory.

***Right click menu*** is a variable (type of Boolean), when enabled the right click menu will be enabled for the inventory items.

***Preview panel*** is a variable (type of Boolean), when enabled the user can select an item from the inventory via left click and it will display a panel of information about this item.

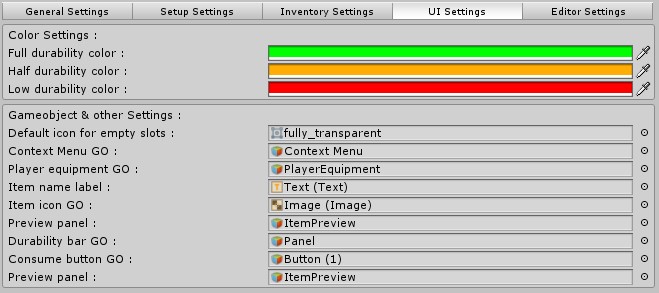
***Hide cursor on toggle*** is a variable (type of Boolean), when enabled the mouse cursor will be hidden when the inventory is not toggled, and it will be automatically re-appeared when the inventory is opened.

***Pick up range*** is a variable (type of Float) which determines the maximum pick up range for the items.

***Disable components on inventory toggle*** is a variable (type of Boolean), when enabled you can add as many components as you want into the following list, these components will be disabled when you toggle the inventory and they will be enabled again when you close it. It is very useful for disabling the mouse look script or any other component.

***Components to disable*** is a list of game objects which are going to be disabled on inventory toggle.

## UI Settings



***Full durability color*** is a variable (type of Color). This will be the color of the durability bar when the durability of the items is from half to full.

***Half durability color*** is a variable (type of Color). This will be the color of the durability bar when the durability of the items is from low to half.

***Low durability color*** is a variable (type of Color). This will be the color of the durability bar when the durability of the items is from 0 to low.

***Default icon for empty slots*** is a variable (type of Sprite). This will be the icon for all the empty inventory slots. If you don’t want to have an icon then assign the ‘fully\_transparent’ file which is included.

***Context Menu GO*** is the context (right click) menu’s UI which is located at the canvas.

***Player equipment GO*** is the player’s equipment UI which is located at the canvas.

***Item name label*** is the label for the selected item’s preview.

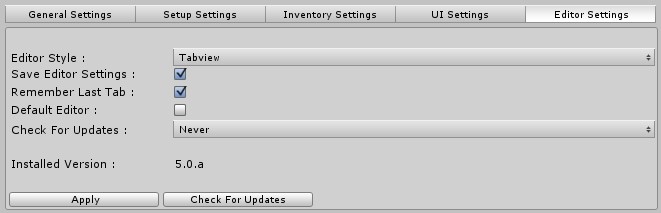
***Item icon GO*** is the image for the selected item’s preview.

***Preview panel*** is the whole item’s preview panel.

***Durability bar GO*** is the durability bar for the item’s preview.

***Consume button GO*** is the button on the preview panel which is visible only at consumable items.

## Editor Settings



Editor settings is a new feature which will allow you to have the ultimate experience with UInventory. A new feature on the editor is also the ‘Check For Updates’ function which will not download the new update since it would be a huge security exploit but it will inform you that a new version is available.

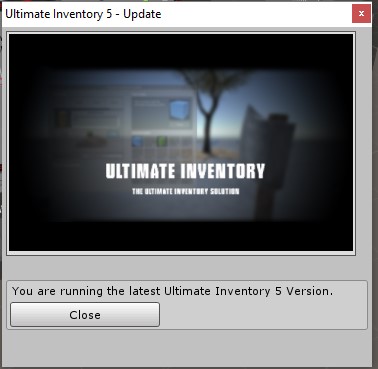
***Editor Style*** gives you the option to choose how the editor looks.

***Save Editor Settings*** is a Boolean, when checked UInventory will automatically save the changes on the editor such as the Editor Style and the other variables. It uses the PlayerPrefs function.

***Remember Last Tab*** is a Boolean, when checked UInventory editor will remember the last tab/category you had opened (General Settings, Setup Settings, etc.) and it will be opened again the next time you select Ultimate Inventory Pro prefab.

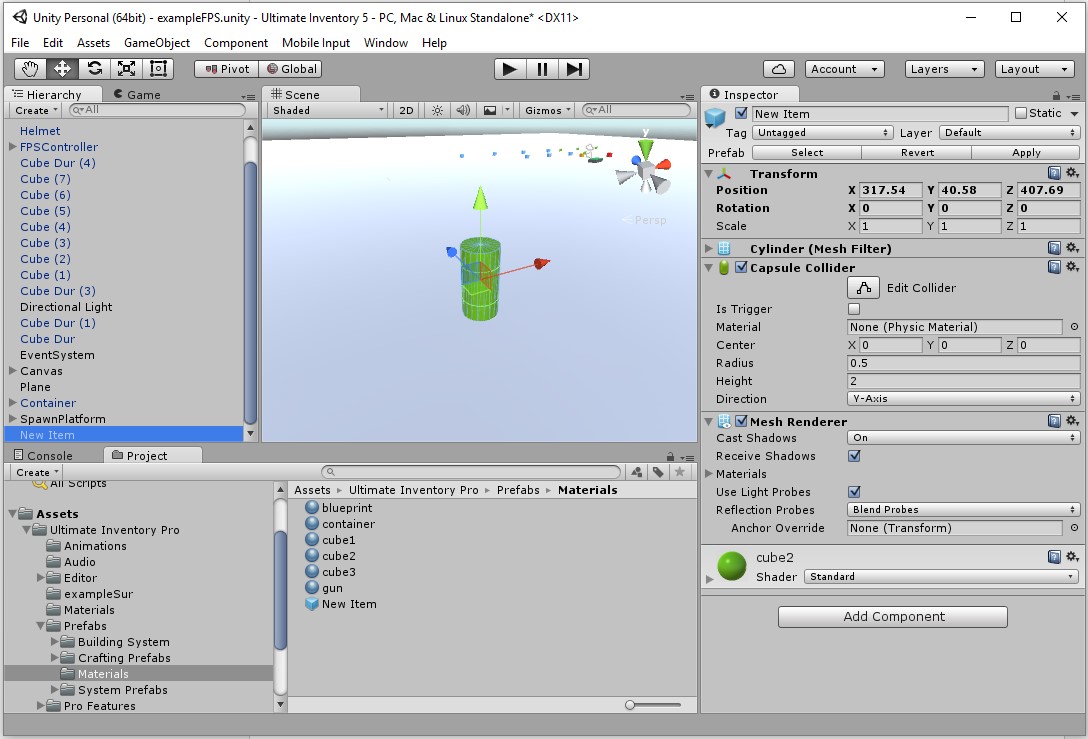
***Check For Updates*** has two options, the On Enable and the Never. On Enable means that each time you select the UInventory prefab it will check for updates which is not recommended. You may choose Never for now and check manually until the new update where it will check once a day.

To check for updates you have to click the ‘Check For Updates’ button and a dialog will pop-up informing you whether there is a new version or not.

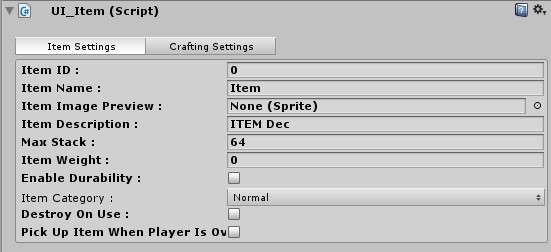


# Adding Items On The Inventory

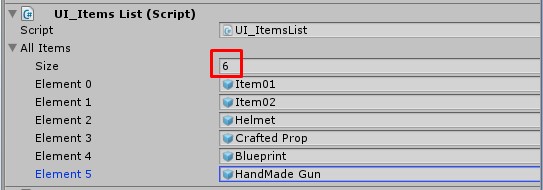
You can add new items on the inventory very easily. Select the prefab you want to add and drag it on the scene.



Now you have to add the ‘UI\_Item’ into your prefab.

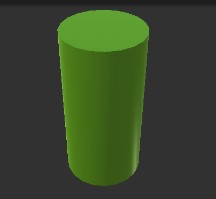


On the ‘Item ID’ variable you have to insert the ID of the item we will create. To get this just go at ‘Ultimate Inventory Pro’ prefab and you will see the ‘UI\_Items List’ script on it, if the size of the array is ‘6’ for example then you should type on your ***Item ID: 6***

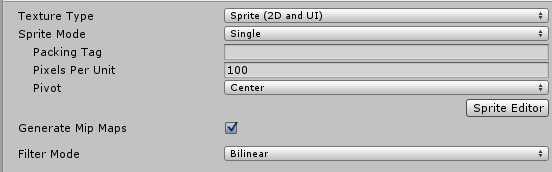


Do not assign your prefab here yet. Now you can edit the ‘***Item Name***’ variable and name your new item as you wish.

On the ‘***Item Image Preview***’ you should assign a sprite with a preview of your item. For example in our case



After importing it make sure to check the image as ‘Sprite (2D and UI)’

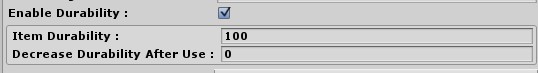


Now edit the ‘***Item Description***’ with the description you like. This text will be displayed on the preview panel of the item.

***Max Stack*** is the maximum amount of this item which can be hold on a single slot.

***Item Weight*** is the weight of the current item, 0 means no weight.

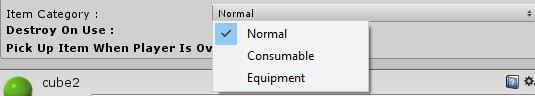
***Enable Durability*** is a Boolean, when checked the item will have durability and a new panel of options will appear on the editor.



***Item Durability*** is the maximum durability which the item can have. 0 means that the item will be destroyed.

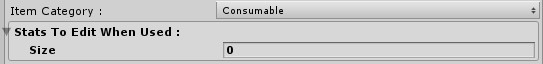
***Decrease Durability After Use*** is how much durability will be removed when the player use the item.

***Item Category*** is a pop-up which allows you to choose what kind of item you want to create.



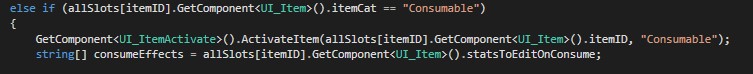
***Normal*** is an item with no extra features.

***Consumable*** is an item which can be consumed by the player and there is a button for this action on these items automatically. Consumables can also be combined with durability.

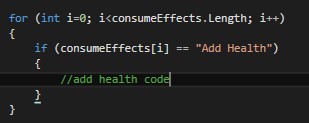


When you select Consumable a new array is appeared.

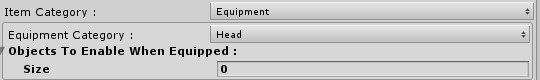
***Stats To Edit When Used*** is a simple array of strings, it does not have a special way to add things and it does nothing from default. This option allows you to add some strings for each item and when you consume this item you can edit the code and apply effects. The code can be found at ‘UI\_Inventory.cs’ by searching for (CTRL+F) the code ‘#29549‘.



With some basic experience you can use a ‘for’ or a ‘foreach’ statement and access all the consumeEffects. Here is an example code:



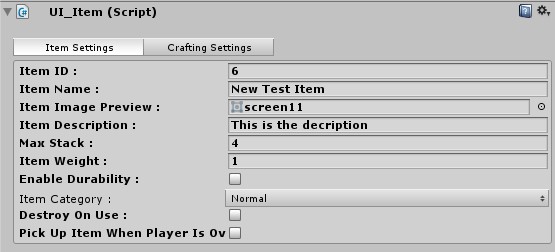
***Equipment*** is an item which can be equipped by the player using the built-in equipment system. This also has a sub-category for which kind of equipment the item is.



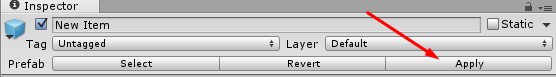
Each equipment category has a special slot on the inventory which allows only this type of items to be equipped on it.

***Objects To Enable When Equipped*** is a simple array of GameObjects which will be enabled when you equip this item. Useful for equipping armor, helmets etc.

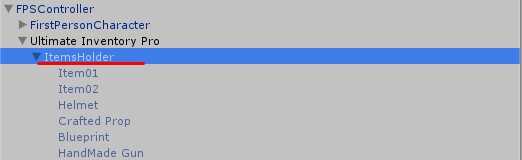
We are not going to explain the ‘Crafting Settings’ for now since it will be covered later. Your item settings should be similar to the following



Now we are ready to add this item into the system but before that make sure to hit the ‘Apply To Prefab’ button if the item is already a prefab or create a prefab of the item.

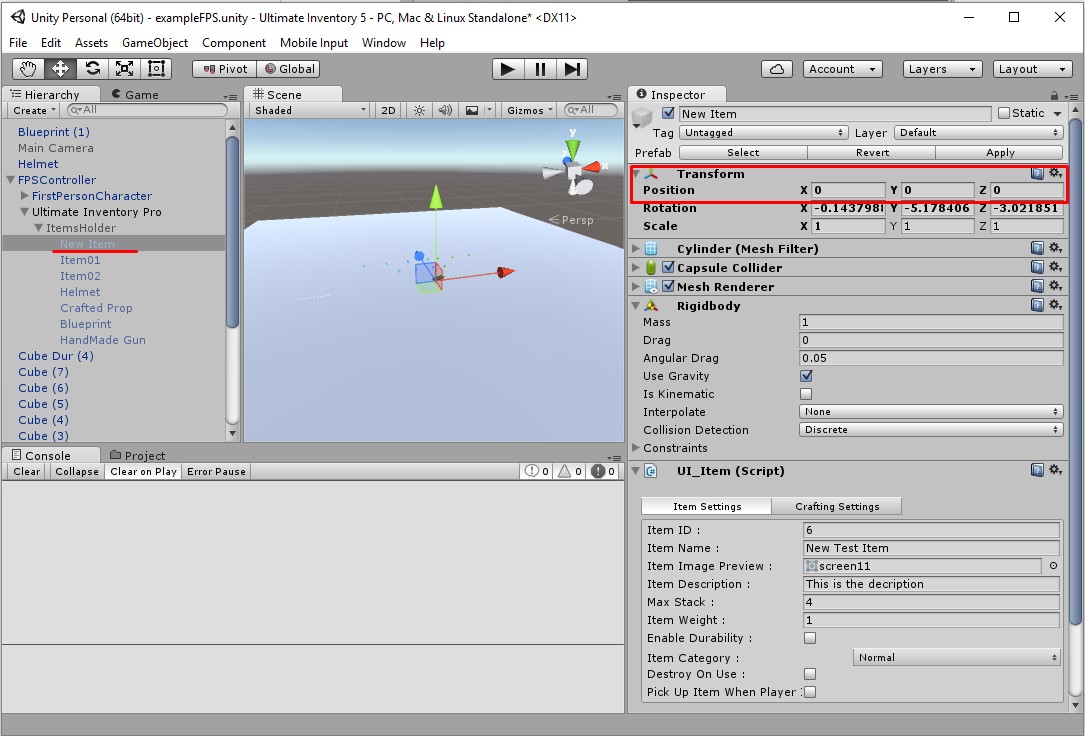


Select the ‘***Ultimate Inventory Pro***’ prefab and then click on the child named ‘ItemsHolder’ and extend it.

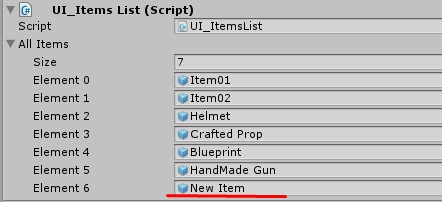


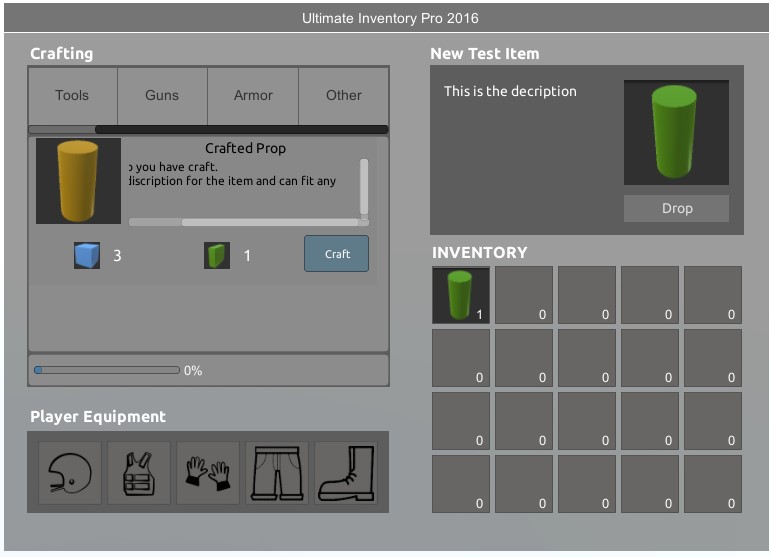
Drag the prefab from the scene and make it child to the ‘***ItemsHolder***’ game object and you may set the position to 0, 0, 0.

It should look like this:



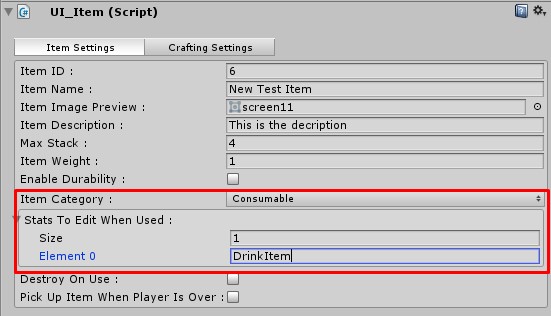
Now you should select the ‘***Ultimate Inventory Pro***’ prefab and drag your new item which is child of the ‘ItemsHolder’ into the ‘UI\_Items List’. And your item is now ready.

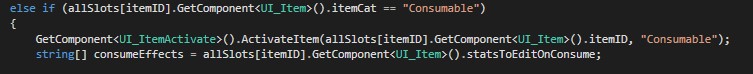




## Creating a consumable

In order to create a consumable all the process is exact the same but you have to select the ‘***Item Category***’ to consumable, and we will set the item to give us a simple message when it is consumed.



In order to make the message to work just open the ‘UI\_Inventory.cs’ script and search for the code ‘#29549‘.

Now below this code just add the following code:

for (int i=0; i<consumeEffects.Length; i++)

{

if (consumeEffects[i] == "DrinkItem")

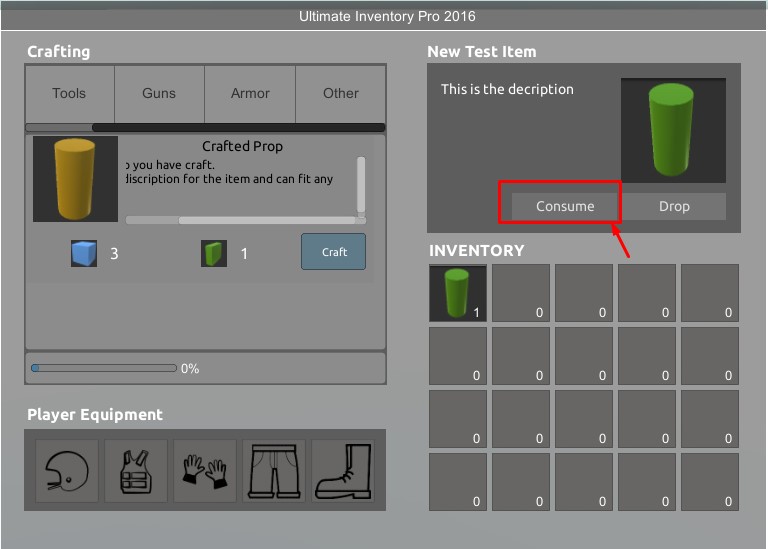
{

Debug.Log("You are drunk now!");

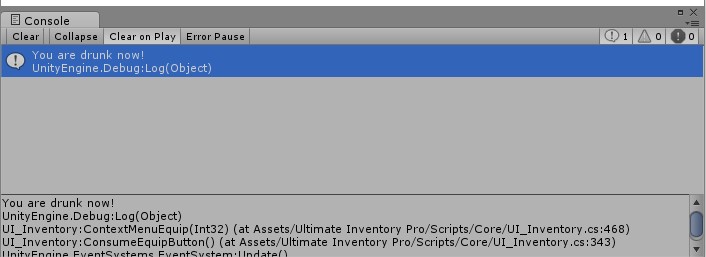
}

}

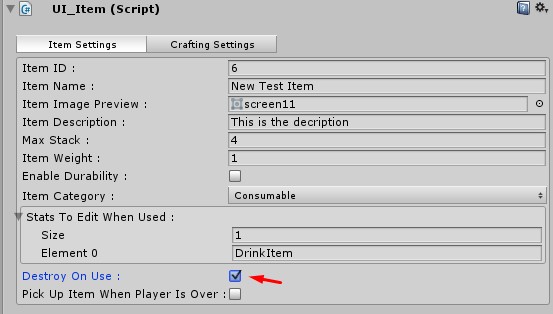
Now when you pick up this item you can consume it.



When you consume it you will notice that the message appears but the item does not disappear from the inventory.

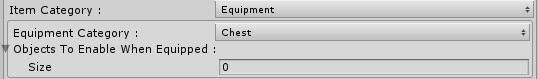


In order to fix this you either, add durability to this item and allow the user to consume it a specific amount of times before it disappears, or you select the ‘***Destroy On Use***’ option and it will be consumed only once.



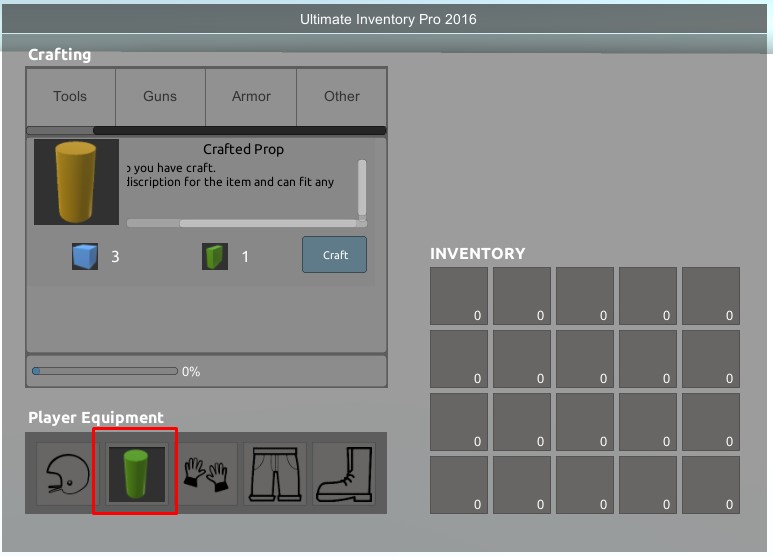
## Creating an equipment

In order to create an equipment item the process is the same and you just need to select the ‘***Item Category***’ to ‘Equipment’ and then choose a type for the equipment.



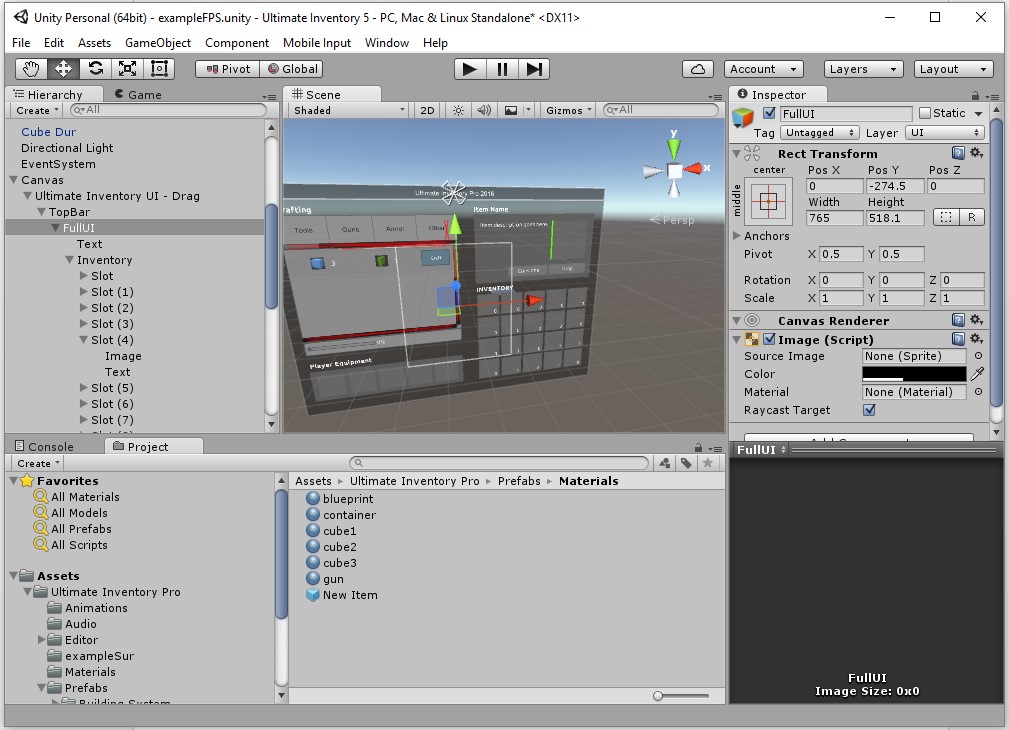
We will not demonstrate how the ‘***Objects To Enable When Equipped***’ array will work since it’s very easy, you just drag the game object on the array.

You can test it and see that the item we created can only be dragged on the ‘Chest’ equipment slot.



# Modifying The Inventory Slots

You can modify the inventory slots by adding, removing or changing the UI, the size and more. It is really easy since Ultimate Inventory 5 uses the new built-in Unity 5 UI. There are literally no limitations on how you can modify the inventory in order to fit your game style.



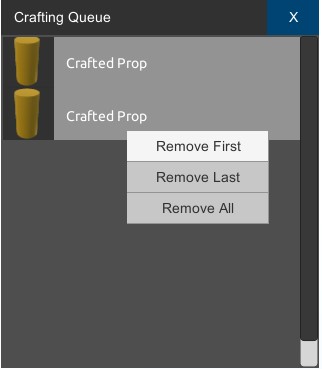
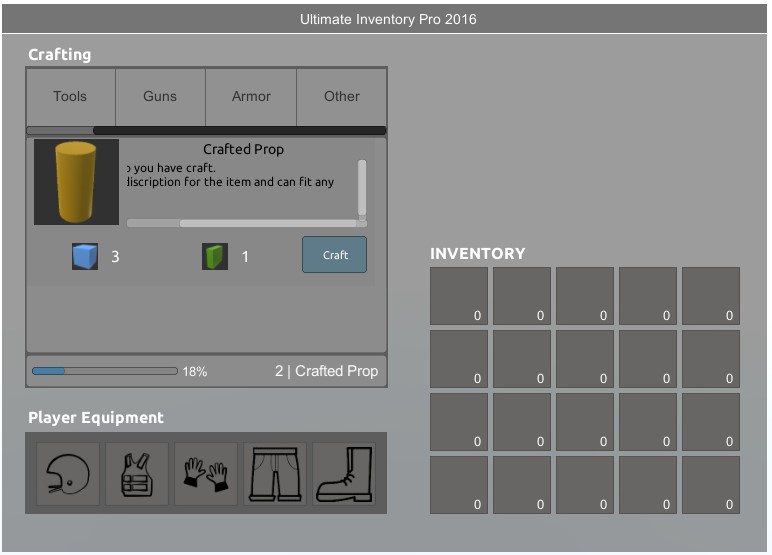
You can add slots by duplicating the ‘Slot’ game object located at the Inventory as you can see on the picture above. You don’t have to modify the slots script as long as you don’t use the manual inventory set up (view at: General Settings). The slots are automatically get arranged into the inventory so you don’t have to worry about the position of the inventory slots.

# Crafting System

Ultimate Inventory 5 has a built-in crafting system which is totally different from the previous crafting system from the UInventory 4. This time crafting is a little more dynamic and fits better on the survival aspect.

The player has a crafting menu where he can choose between the categories of what he wants to craft, then on each category there are items which can be crafted using resources from the inventory. The whole process of reading if the resources are available and then removing the resources is included.

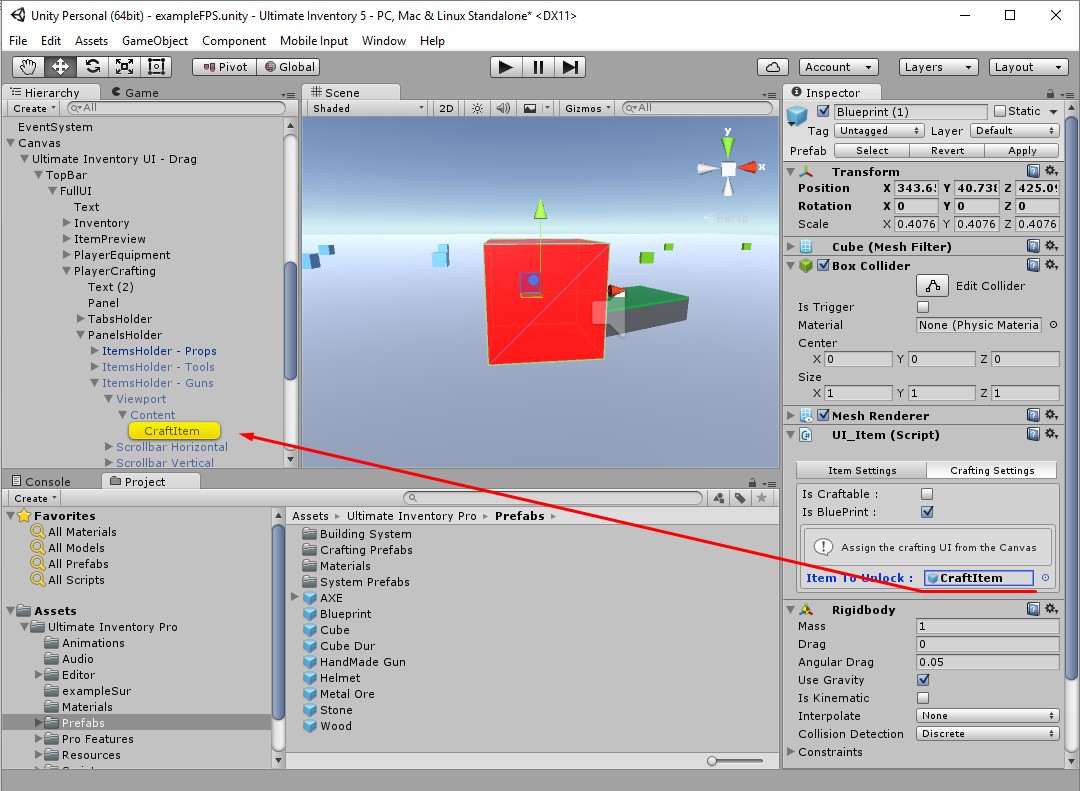
There is also a crafting queue available which supports unlimited number of crafting items waiting in the queue or you can set a limit. The queue is displayed with a simple label giving the player some basic information about how many items are crafted and also what is the items which is being crafted now. Clicking that label pop-ups a window with detailed information about the crafting queue as you can see on the picture below.



The crafting process is being displayed on the user with a progress bar and a label with the percentage of the crafting.

The crafting system supports also a method of enabling items for the user in order to craft them later during the game using the ‘Blueprint’ technology similar to the ‘Rust’ game. Creating a blueprint is rather simple, you just need to check the ‘Is Blueprint’ variable on your item and then assign the Crafting Item (UI) which is located on the canvas on the variable which appears.

You can see a preview of a blueprint on the following screenshot.

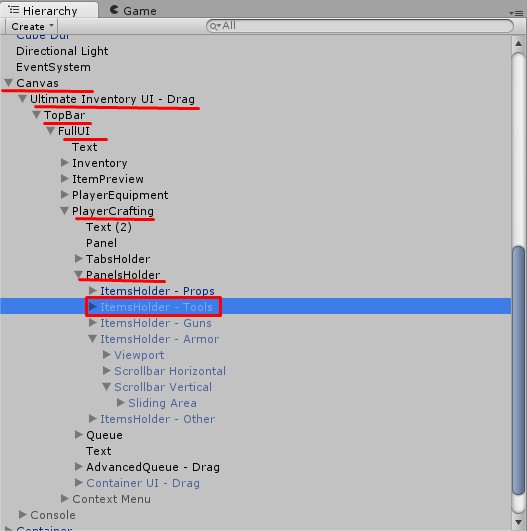


It is very important to assign the correct item on the ‘***Item To Unlock***’ variable.

## Adding crafting item

You can easily add items on the crafting system following this simple process. First you have to choose on which category you want to add your new item, you can see how to create more categories on the next step.

For this example we will choose to add a new item on the ‘Tools’ category. Go on the canvas and locate the ‘Tools’ category as you can see on the following screenshot.

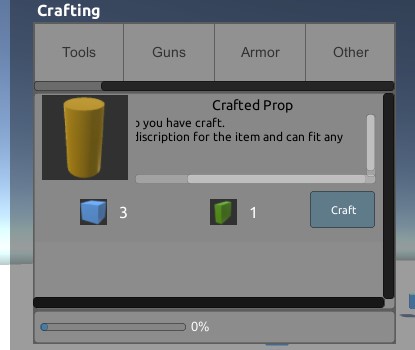


After locating the category extend it and select the ‘Content’.



You may enable this category by setting the ‘ItemsHolder – Tools’ game object visible so you can see what is going on. Make sure to disable all the other categories since only one can be visible at a time.

Now locate the ‘CraftItem’ on your project directory, Ultimate Inventory Pro, Prefabs, Crafting Prefabs and drag it into the ‘Content’ game object. The result should look like this

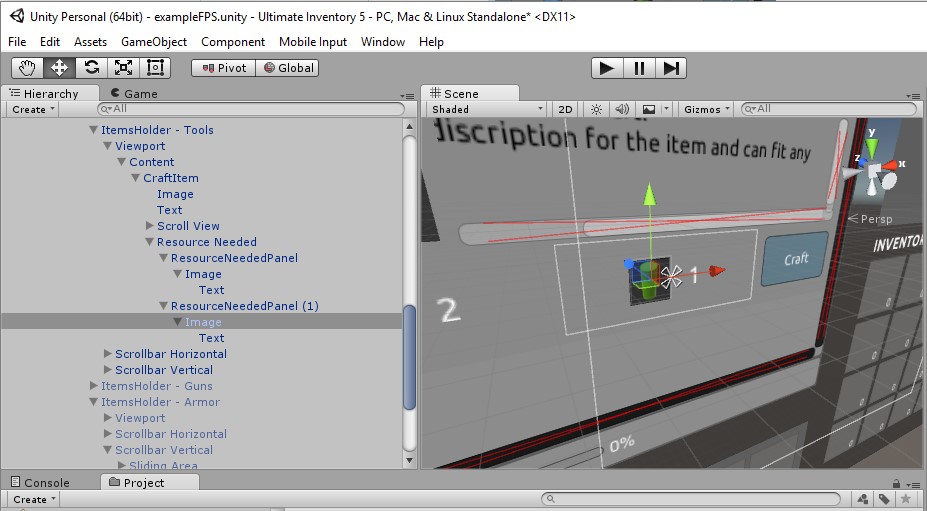


Now you can edit the UI to fit your needs by modifying the name of the crafting item, the icon, the description as well as the resources needed to be crafted. Keep in mind that this is only the UI part, the actual amount of resources is stored into the item script.

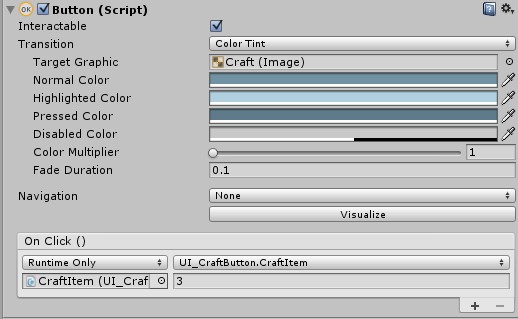


You can easily modify the resources needed by duplicating the ‘ResourcesNeededPanel’ and then changing the icon and the amount of the item.

Let’s say we will require 2 of the ‘Item 01’ and 1 of the new item we just made.



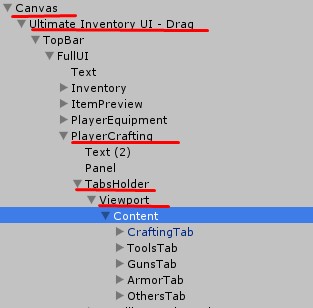
Now the last step is to determine what item is going to be crafted when the user press the ‘Craft’ button. To do this just click the button and type the ID of the item you want to be crafted based on the ‘UI\_Items List’.



That’s it, you just added your new crafting item.

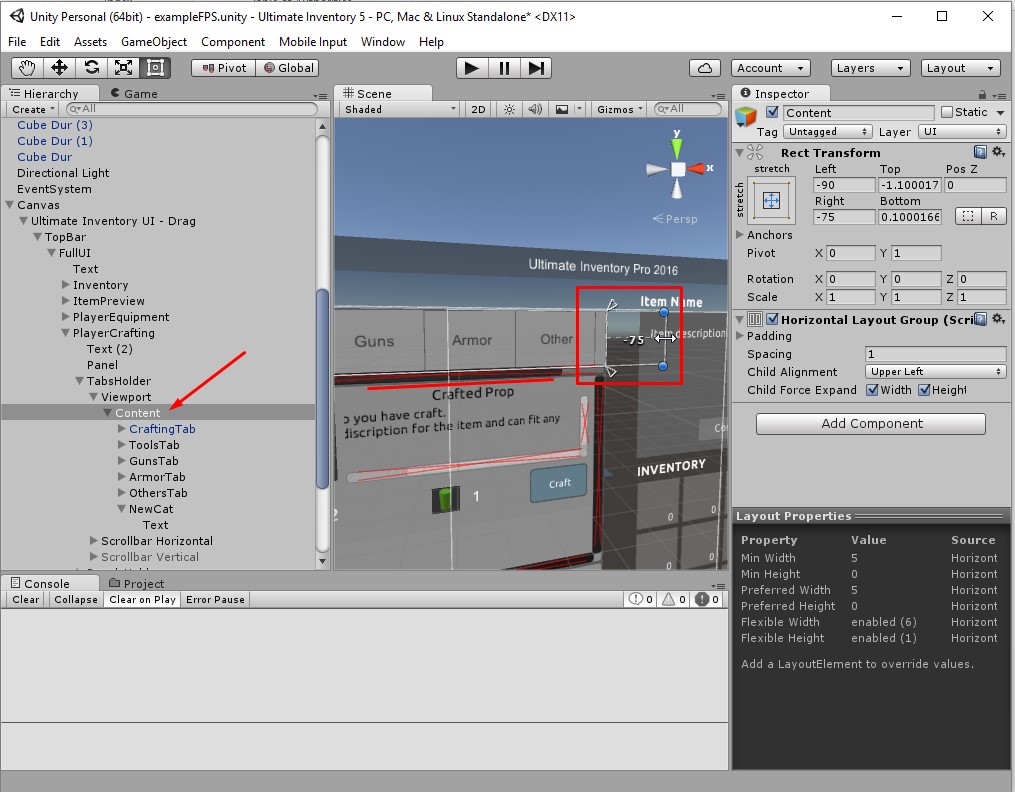
## Adding crafting category

Adding a new crafting category is also very simple and just needs to follow a couple of easy steps. First of all, you have to locate the category holder game object which is located at the Canvas, Ultimate Inventory UI – Drag, PlayerCrafting, TabsHolder, Viewport, and Content.



To create a new category just duplicate one of them and then the system will automatically modify the position and make it fit correctly to the UI. Let’s add a category named: ‘New Cat’.

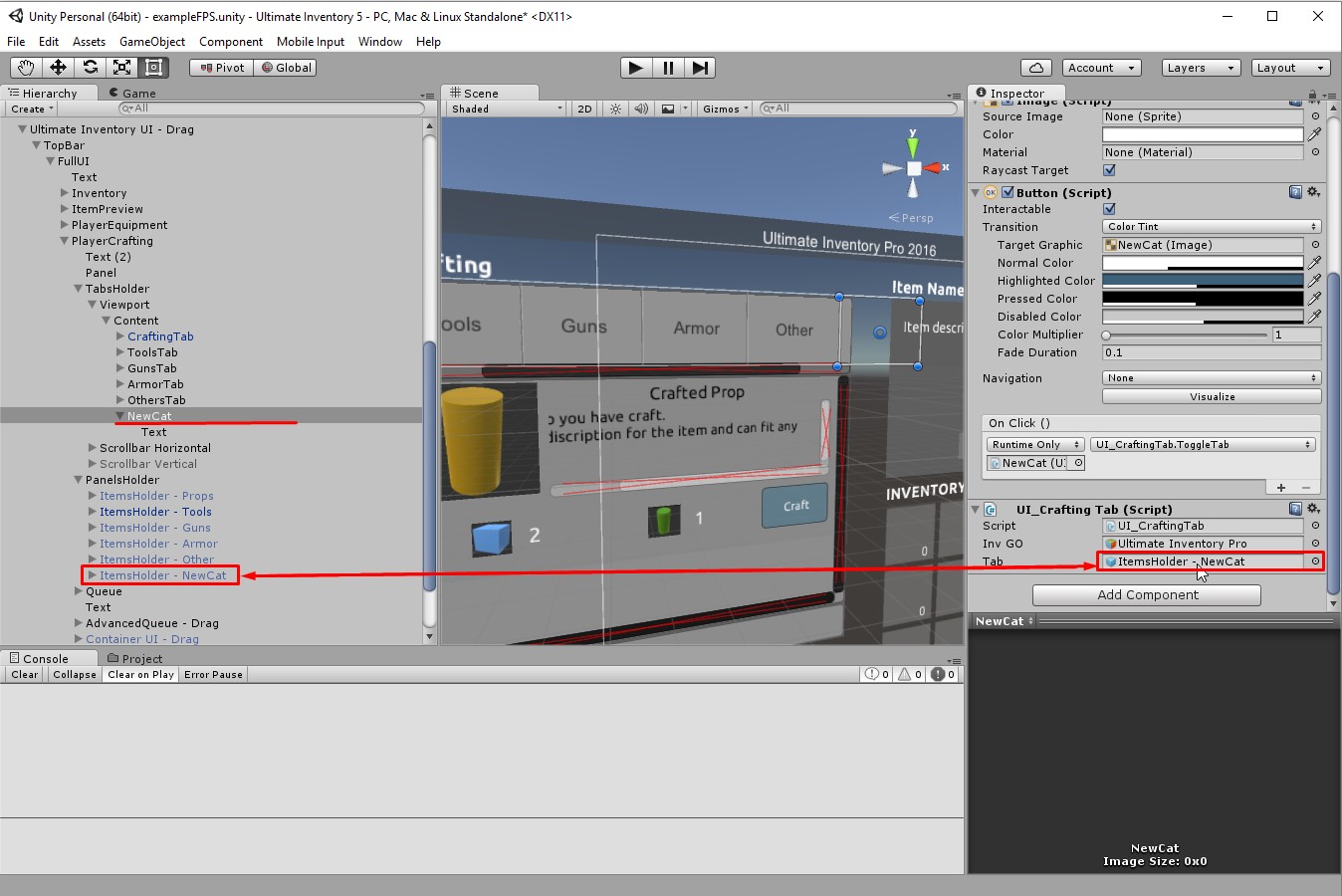
You can either let the system make the categories little smaller so they can fit or resize the container so the scrollbar will be automatically adjust. We will resize the container on this example.



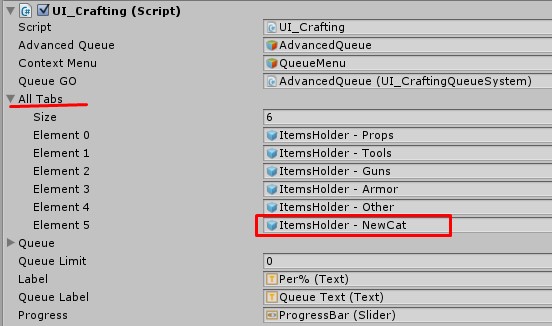
Now we need to create a panel which will hold the items for our new category. Select the ‘Panels Holder’ and with a similar process just duplicate one of the items and rename it properly.



Now go back to your category tab and select it, then assign the ‘ItemsHolder – NewCat’ we just created as you can see on the image below.



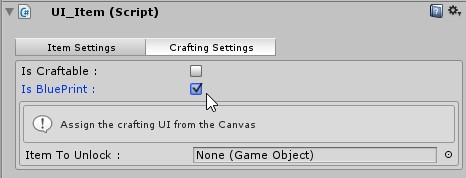
And now there is the final step, go at the ‘Ultimate Inventory Pro’ game object, and locate the ‘UI\_Crafting component.’



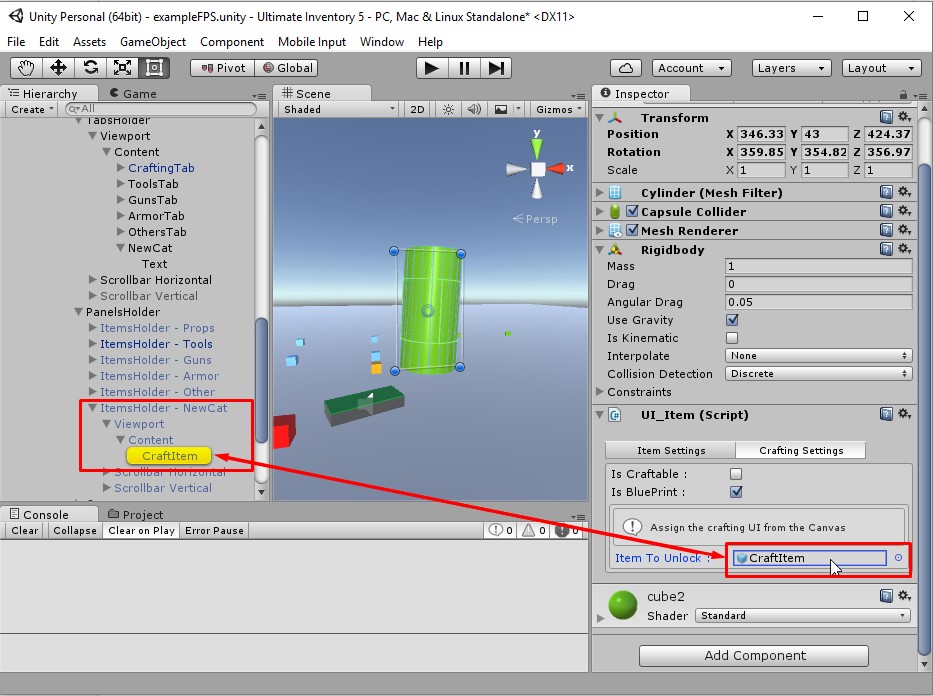
Now on the ‘All Tabs’ array just add the panel we created named ‘ItemsHolder – NewCat’ and your new category is now working.

## Creating blueprints

In order to create a blueprint\* all you have to do is create a new item just like we did on the ‘Adding Items On The Inventory’ above and then just go into the ‘Crafting Tab’ and check the Boolean ‘Is Blueprint’



Now, the most important part is to assign the correct Game Object on the ‘Item To Unlock’ variable. This variable must be the CraftItem prefab we added on the UI just like the screenshot below.



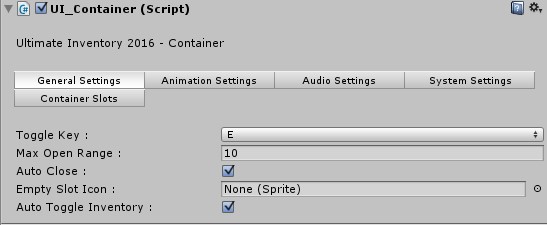
And then make sure to disable the ‘CraftItem’ we assigned there so it would be visible only when the player unlocks this blueprint.

# Looting System

Ultimate Inventory 5 contains a looting system quite similar with the old one of the UInventory 4, remade completely from scratch for Unity 5. This looting system includes a very easy to use custom editor which will allow you to create as many looting points (either containers or bodies) as you want within few minutes or less.

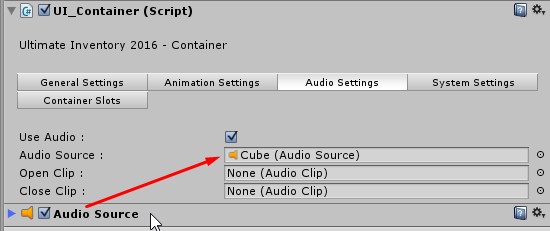
## Non-animated looting

A non-animated looting system can be a dead body, a trunk etc. Just select your prefab and add the ‘UI\_Container’ script on it. This is what it should looks like:



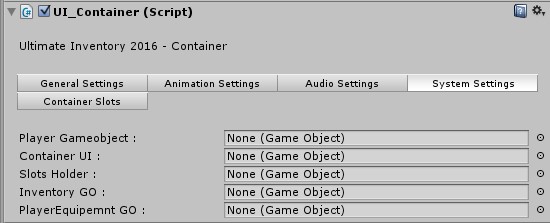
As you can see the settings are very easy and you may modify them as you like in order to fit your needs.

If you want to allow audio when the container is opened or closed then make sure to add a ‘Audio Source’ and assign it on the ‘Audio Settings’ tab.



There are two examples of ‘Open’ and ‘Close’ audio for your container, you can add your own by importing them into Unity.

Here is the most important part of the Looting system, click ok the ‘System Settings’ and you will see that you have to assign some variables.



***Player GameObject*** should be your player controller, in our case the FPSController.

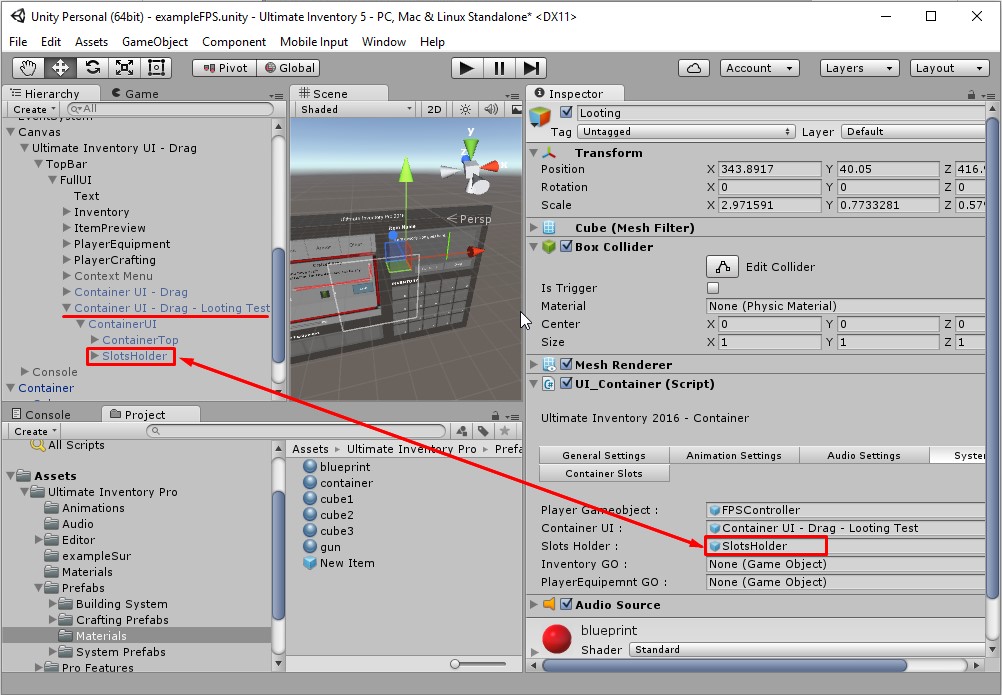
Now it’s time to create the container UI for this looting container. Go into your canvas and locate the ‘***Container UI – Drag***’.



This is the default one and it should not be deleted, just duplicate and rename it properly. Now go back to your container prefab and assign this game object we just created to the variable named ‘***Container UI***’.

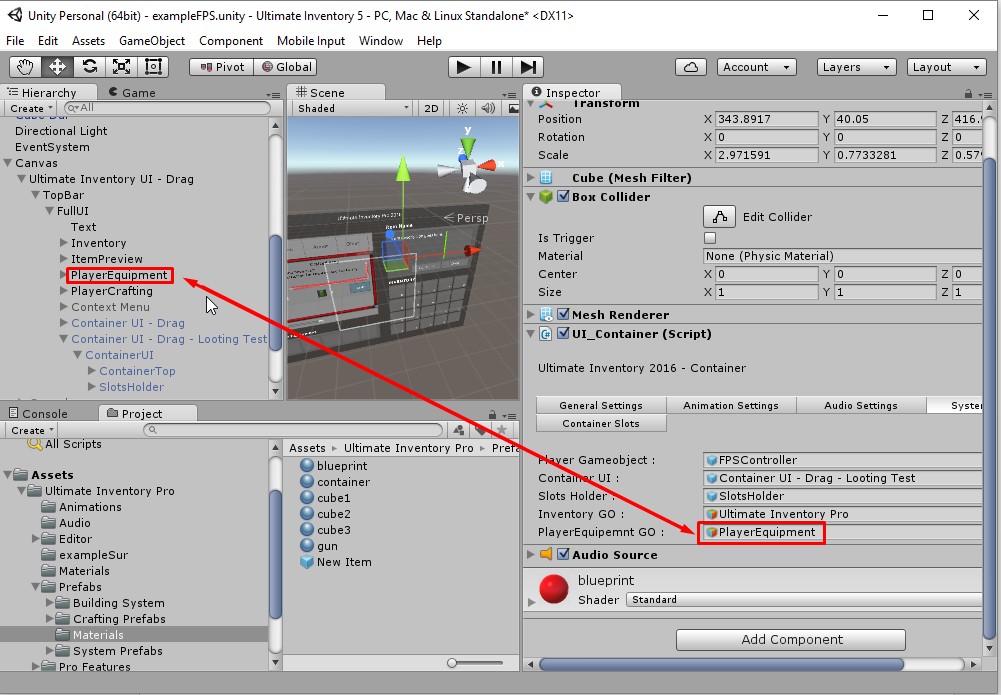
Now go at the container UI we just created, extend it and locate the ‘SlotsHolder’ game object. Then go back to your container and assign it to the ‘Slots Holder’ variable.

After following all of the steps your container should look similar to this:



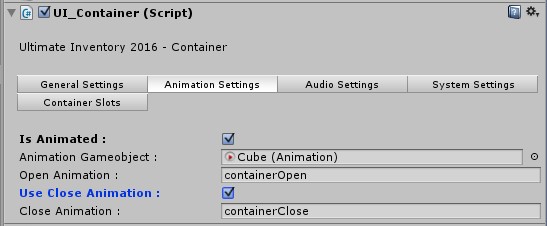
We are almost ready, just assign to the ‘***Inventory GO***’ the ‘Ultimate Inventory Pro’ game object located at your character controller.

Finally, locate the ‘PlayerEquipment’ game object at the Canvas and assign it to the ‘PlayerEquipment GO’.



## Animated looting

In order to create an animated looting system like a container which can open and close you have to follow exactly the same process like before, and when you are done, click at the ‘Animation Settings’.



In order to make this work you have to add an ‘Animation’ component into your container and assign it at the ‘Animation GameObject. You can notice that you can make animated a child of the container and not necessary the container prefab itself.

***Open Animation*** should be the name of the open animation for your container.

You have the option to choose whether you want your container to open and close or just open once and then it will be open for the rest of the game. If you choose to allow it to close then you should add the name of the animation on the ‘***Close Animation***’ variable.

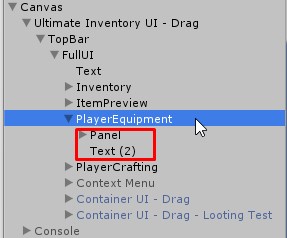
## Container slots

You can add and remove slots for the container exactly like you can do it with the inventory. If you choose to set the slot settings automatically then you have nothing to worry about.



# Player Equipment

Ultimate Inventory comes with a built-in player equipment system which is simple to use and extend. You can find that system at your Canvas, Ultimate Inventory UI – Drag, Player Equipment. You can disable in by disabling the two children of this game object and not the game object itself because it would cause errors.



With a basic experience and understanding of Unity and C# you can extend this system and add more equipment slots. You just need to duplicate the equipment slots and then make sure to name your equipment category here:



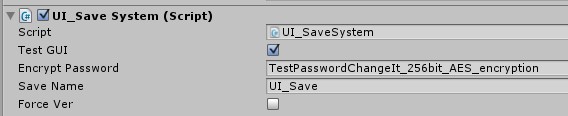
Then you have to edit the scripts and add your own category into the code in order to work. On a later version we might add a custom editor which will do this process automatically but it’s not on our priorities now.

You can modify the empty icon of each equipment slot by changing the ‘***Empty Icon***’ variable.

# Save System

Ultimate Inventory 5 has a built-in save system for the inventory which uses a powerful 256bit AES encryption. The encryption functions are included on the namespace ‘UInventory’ which is described later. It uses the ‘PlayerPrefs’ function which is built-in at Unity.

The save system script is assigned to the ‘Ultimate Inventory Pro’ game object and you can modify the encryption key there.



If you check the ‘Test GUI’ then two simple buttons will appear on your screen allowing you to save and load the inventory.

You can modify the ‘***Save Name***’ at runtime in order to save at multiply slots for example for different players of your game.

If you check the ‘***Force Ver***’ variable then when you load data UInventory 5 will not verify if the version of the load file is the same with the current version. This might crash the save system.

You can save or load your game by calling the two simple functions ‘Save Inventory’ and the ‘Load Inventory’ from the ‘UI\_Save System’ script.

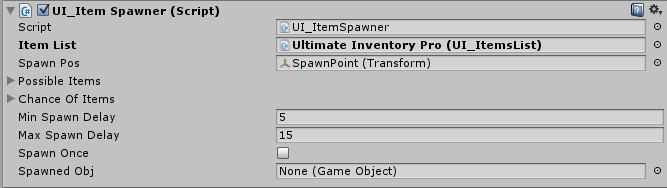
# UInventory Namespace

Ultimate Inventory includes a simple namespace with some basic functions used by the save system. Currently it does only contain one class the ‘Security’ class and this class includes an Encrypt function, a Decrypt function and a function which extracts string between tags. You can import this namespace with: ‘Using UInventory’.

More function might be added on later versions.

# Item Spawner

Ultimate Inventory 5 contains a random item spawner system. There is an example prefab located at the ‘Prefabs’ directory which you can use and modify it. It does not require any mesh renderer but we used one just for the demonstration.



‘***Item List***’ is the list of all the inventory items located at the ‘Ultimate Inventory Pro’ game object.

***‘Spawn Pos***’ is a variable (type of Transform) which is used just for the spawn position.

‘***Possible Items***’ is an array of integers. Add the id’s of what items you want your spawner to be able to spawn.

‘***Chance of Items***’ is an array of floats. It determines the chance of being spawned for each item. It ranges from 0 to 100 and it should be used like the following example:

Chance for Item with ID=1 🡪 30

*This means that the item with id = 1 has 30% chance of being spawned.*

Chance for item with ID=4 🡪 50

*This means that the item with id = 4 has 20% (50-30) chance of being spawned.*

Chance for item with ID=7 🡪 100

*This means that the item with id = 7 has 50% (100-50) chance of being spawned.*

*Note: The last item must always finish with 100.*

‘***Min Spawn Delay’*** is the minimum amount of time that the spawner will wait until it does spawn an item.

‘***Max Spawn Delay’*** is the maximum amount of time that the spawner will wait until it does spawn an item.

*The system gets a random value between the Min and the Max.*

‘***Spawn Once***’ determines whether the spawner will spawn only one item and never reset.

‘***Spawned Obj***’ is the latest spawned object. Do not modify it, just access it via script if needed.

# Inventory Functions

Ultimate Inventory 5 has some built-in functions which are going to make your life easier. You can remove an item by the slot of the inventory, you can count the amount of a specific item and you can add an item via script.

In order to use then you need to access the ‘UI\_Inventory.cs’ script which is located at the ‘Ultimate Inventory Pro’ game object.

public void DecreaseItem(int index, int amount)

*This function decrease (removes) a specific amount of a specific item in the inventory. If the amount you demand is more than the amount the inventory has then it removes the all of them.*

public int CountItem(int index)

*This function count the amount of a given item from the inventory and returns the value as integer.*

public float CountWeight()

*This function counts the weight of all the items and return the value as float.*

public void PickupItem(GameObject item)

*This function force the inventory to pick up the given item (as GameObject) without the input of the player. If the inventory does not have enough space then the function is canceled.*

public void UpdateInventoryUI()

*In case you modify the inventory manually then you might end up with a problem of not visualizing the actual change. If you ever need to force a refresh of the inventory’s UI then call this function.*