



Assets

Motion Controller (v 2.62): <https://www.assetstore.unity3d.com/en/#!/content/15672>

Archery Motion Pack (v 0.77): <https://www.assetstore.unity3d.com/en/#!/content/70487>

Sword & Shield Motion Pack (v 0.28): <https://www.assetstore.unity3d.com/en/#!/content/77196>

Inventory Pro (v 2.5.8): <https://www.assetstore.unity3d.com/en/#!/content/66801>

MC Inventory Pro Integration: www.ootii.com/UnityMotionVault.cshtml#InventoryPro

Overview

While the Motion Controller includes a “Basic Inventory” solution, Inventory Pro (IP) is a popular and powerful inventory solution that you can use with the Motion Controller (MC) and its motion packs.

Following Inventory Pro standards, we can allow our character to equip items using animations and motions. I am not an Inventory Pro expert and this guide isn't meant to cover all aspects of Inventory Pro, but it will show you how you can use Inventory Pro with the motion packs.

Special Thanks

A massive 'thank you' to TeagansDad and Mustafa for the work they did. They really helped me understand Inventory Pro.



MC Update 2.6

In MC 2.6, I changed how the motion packs work with common actions like “idle” and “equip item”. The new “basic” motions like “Basic Idle”, “Basic Item Equip”, “Basic Item Store”, and others represent actions that are similar across multiple packs or in multiple situations. The only real difference is which animation plays.

So, there’s no need for an “Unarmed Idle”, “Sword & Shield Idle”, “Archery Idle”, etc. All of these are wrapped inside of one “Basic Idle”. We then use a “form” property to determine which animation will play within that one basic motion.

For example, “Basic Item Equip” is now the motion we use for equipping the sword, bow, shield, etc. However, each of these has a different animation. So, we need to add extra stats to Inventory Pro that not only tell us which motion to play, but which animation or form to use.

The result is that we have some additional items stats to setup in Inventory Pro. Then, we can fill in those additional stats for each of the items as needed.

Please see [Inventory Pro Item Stats](#) and [Inventory Pro Item Editor](#).



Integration

For this guide, I'm going to assume you have the MC, IP, and the two motion packs listed above imported. I'll also assume you've done the basic character setup with both the MC and IP.

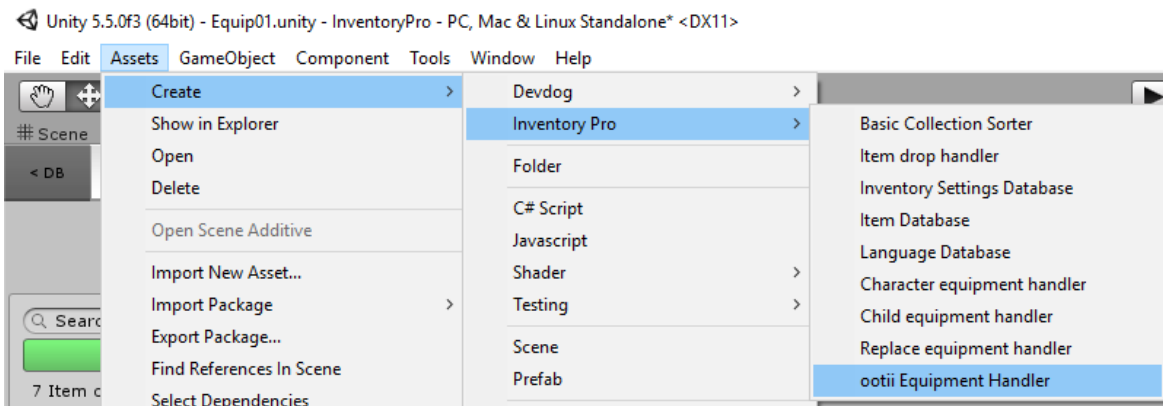
Finally, you'll want to have imported the MC Inventory Pro integration package from above.

I'll also assume that you understand how to work with Inventory Pro's item editor and database.

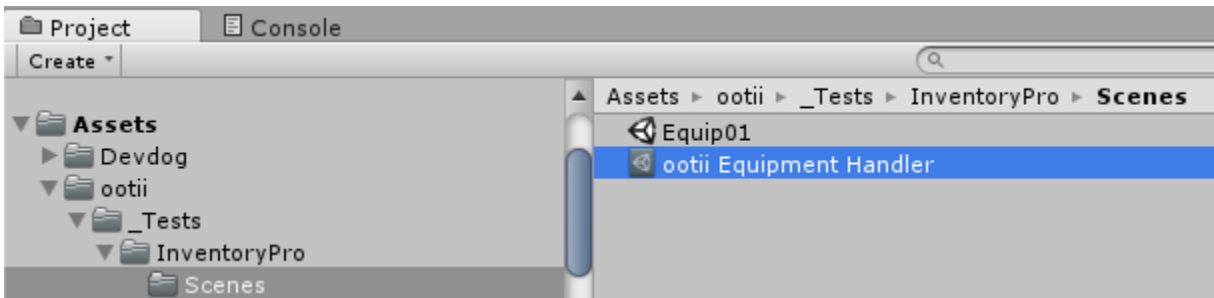
Inventory Pro Equipment Handler

IP allows us to intercept the equipping and un-equipping of items. To do this, I've created an "ootii Equipment Handler". You'll need to create an asset out of this class for IP to use.

1. Create the ootii Equipment Handler from Unity's Asset | Create | Inventory Pro | ootii Equipment Handler menu.

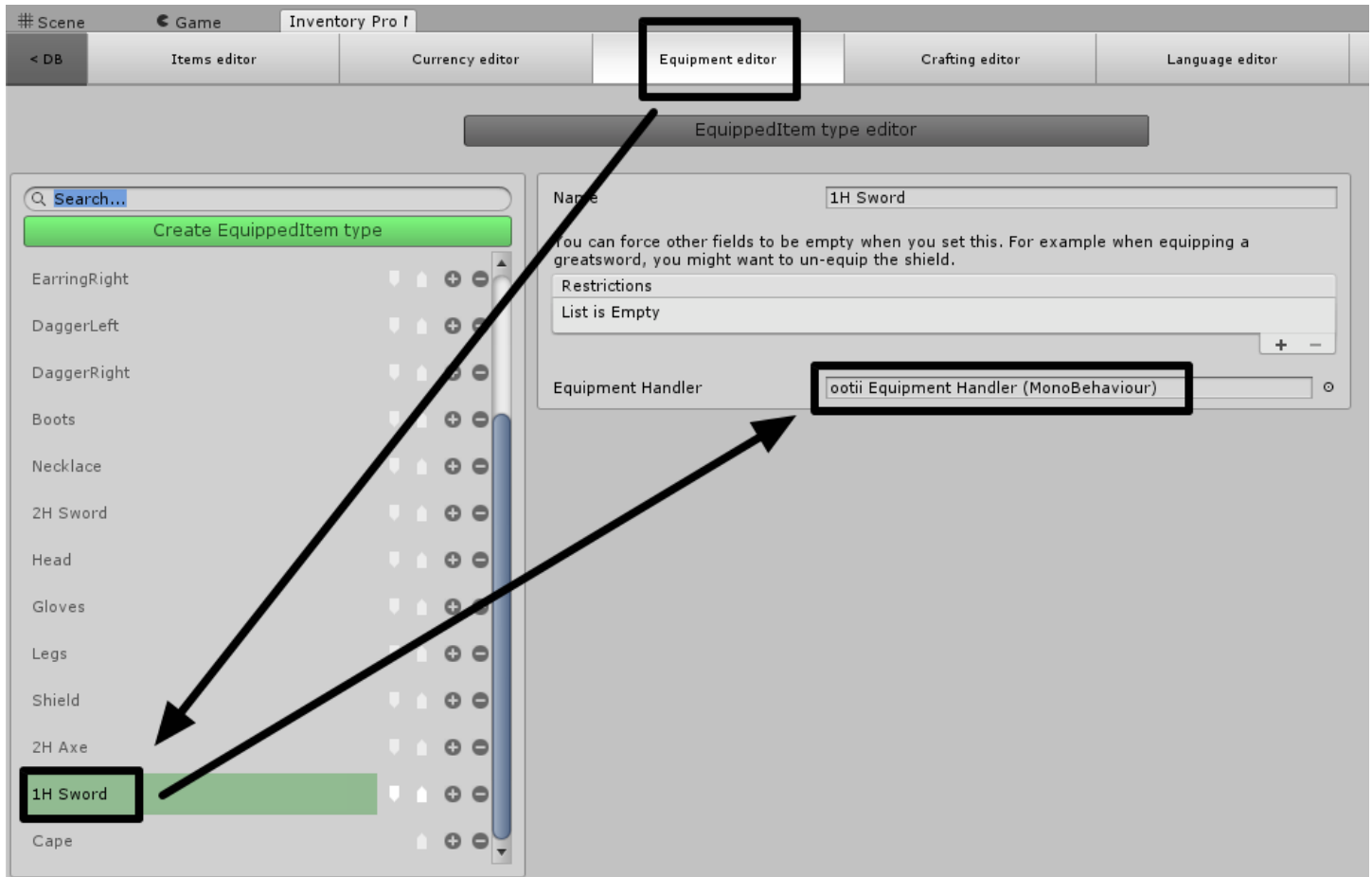


2. A new "Equipment Handler" asset will be created (the name and location may be different).





3. Each of your IP Equipped Item Types you want to use with the MC, needs to have this set as the "Equipment Handler".





Inventory Pro Item Stats

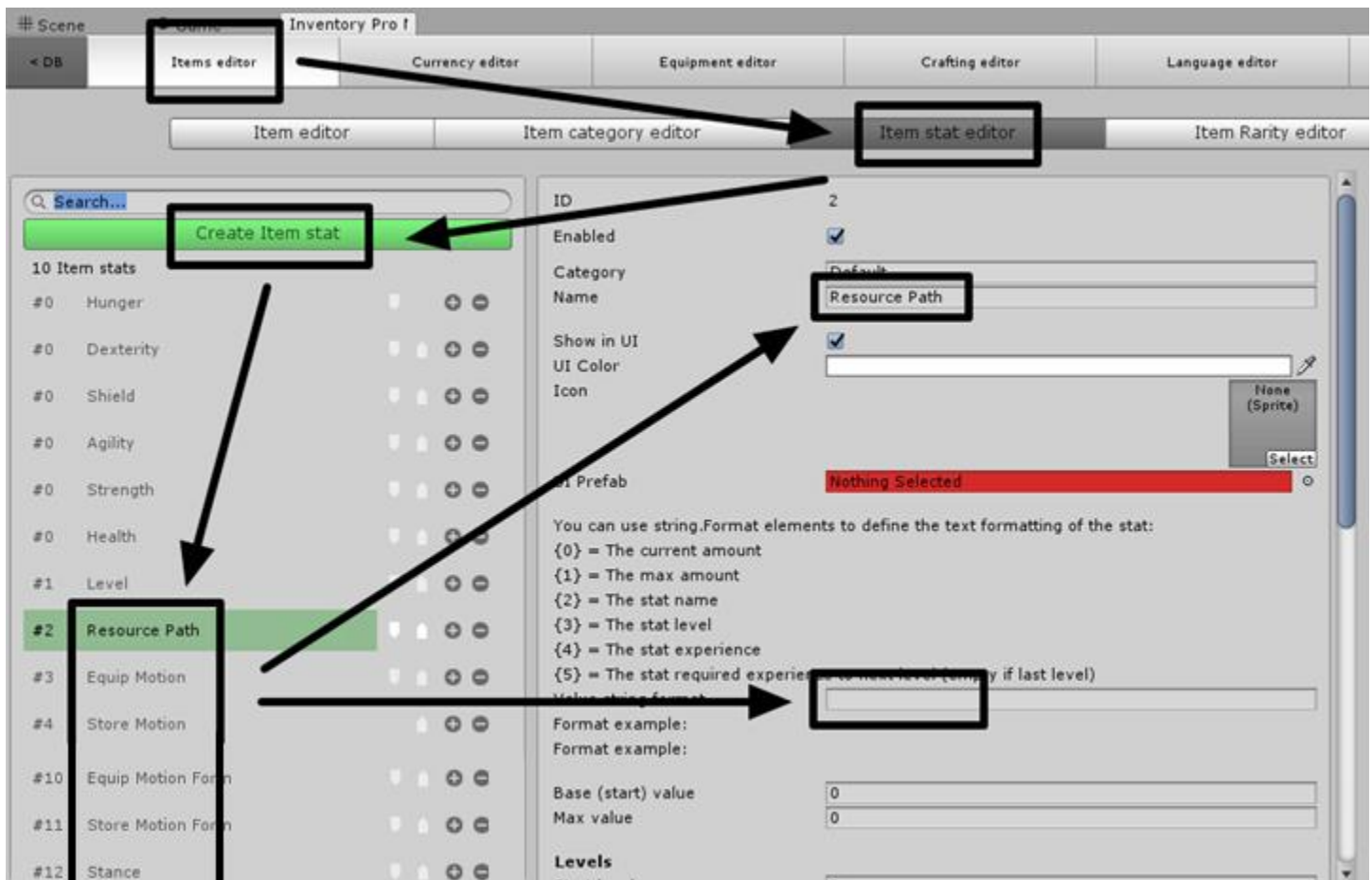
The MC needs to get information specific to each item. This includes the prefab to use, the equip motion, etc. We'll define this information for each item our motions care about.

However, first we need tell Inventory Pro about these properties. We'll create an Item Stat for:

- “Resource Path” – Used to find the prefab of the model we'll create to represent the item
- “Equip Motion” – Used to define the motion used to equip the item
- “Store Motion” – Used to define the motion used to store the item

New with MC 2.6:

- “Equip Motion Form” – Used to specify which animation is used with the Equip Motion
- “Store Motion Form” – Used to specify which animation is used with the Store Motion
- “Stance” – Used to specify the stance the character is in when the item is equipped
- “Default Form” – Used to specify the default animation to use in other motions when the item is equipped

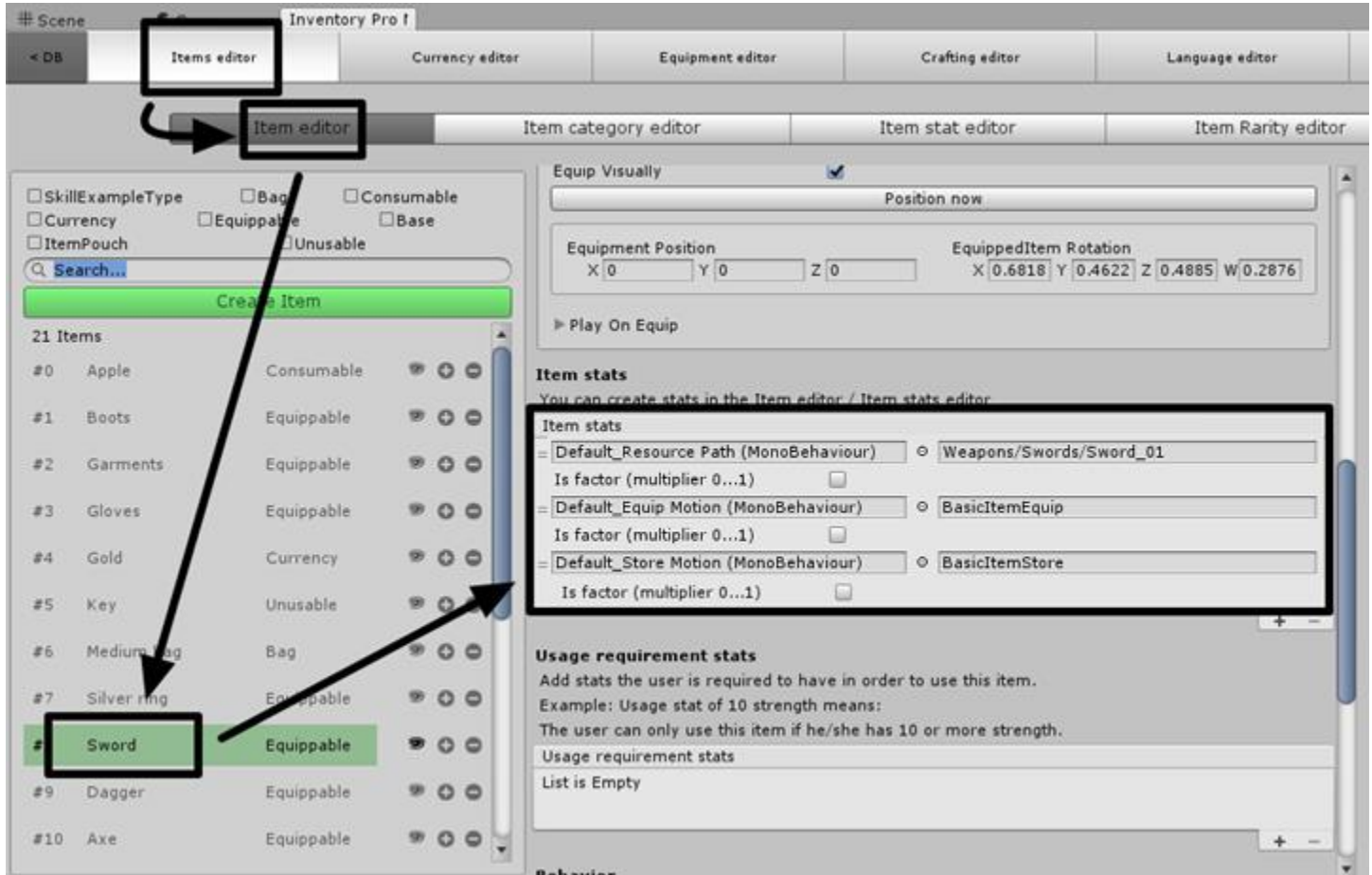


With the properties defined, we can now set the item specific values.



Inventory Pro Item Editor

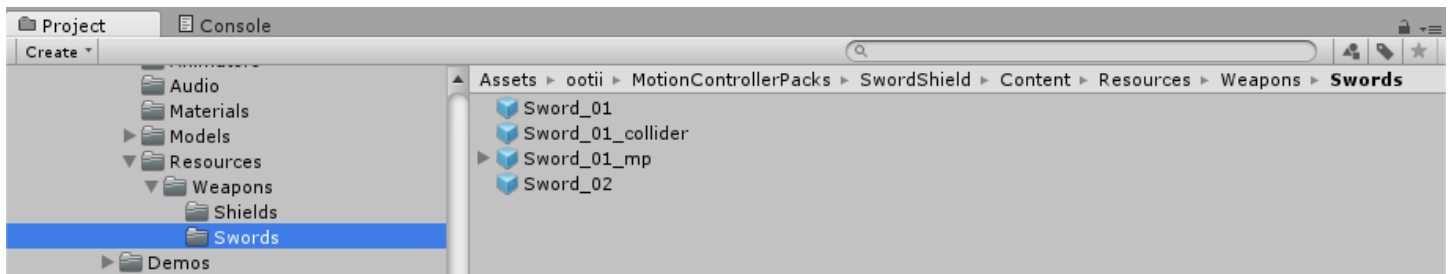
Each item will probably have its own prefab and/or equip motions. They don't have to, but if they do we will need to tell Inventory Pro about them. We do that in the Item Editor.



If you look at the Item Stats values, you'll see they match values that the motions expect.

Resource Path

This is the path to the prefab in a "Resources" that represents the item. In the example above, the prefab is found here:



Remember that you only need the path after the "Resources" folder.



Equip Motion & Store Motion

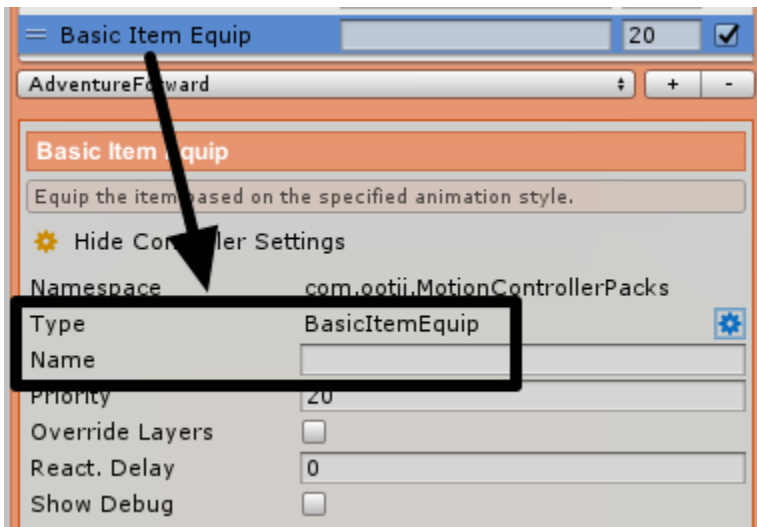
These are the motion names that will be triggered to equip and store the items. You can leave these blank if you don't want to play an animation and just have the item pop-in and pop-out.

Typically the names are:

BasicItemEquip

BasicItemStore

The names should match the class type or motion name in your motion list:



New with MC 2.6:

Equip Motion Form & Store Motion Form

These properties are based on the Any State transitions that exist in the motions you defined above. For example, by default the "form condition" for the following exist:

Sword = 100
Bow = 200
Spell = 300
Shield = 400

You may use custom animation have changed these, but these are what I use by default.

Stance

When an item is equipped, this is the stance the character goes into. Typical stances include:

Traversal (default) = 0
Combat Melee = 1
Combat Ranged = 2
Sneak = 4
Spell Casting = 8



There are others and you can customize them as needed. See EnumControllerStance.cs for more

Default Form

Provides a default animation select for idle and movement motions when items are equipped. For example, by default the motions use the following “default form values” to determine which movement animation to play:

Sword & Shield = 100

Archery = 200

Spell Casting = 300

As you add new animations, you may add “Polearm” and give it a unique default form value.

Optional

Equip Motion, Equip Motion Form, Store Motion, and Store Motion Form are optional. They are only needed if you want to have an animation play to equip and store an item.

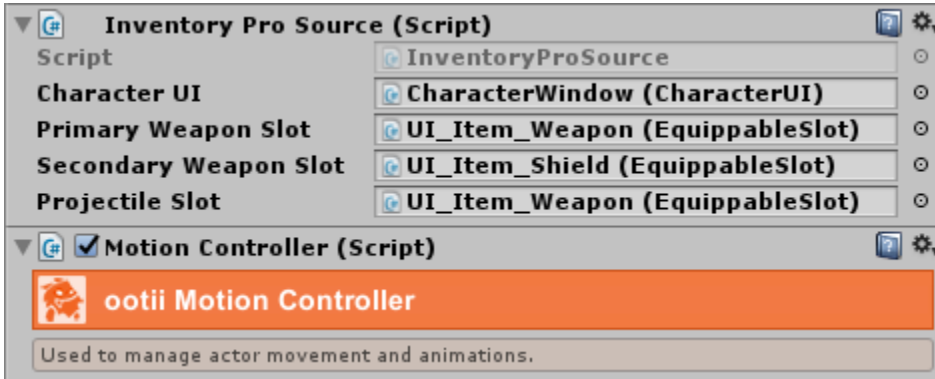
Stance and Default Form are optional. They are only needed if you want the act of equipping an item to change how the character behaves. For example, equipping a sword would change the character’s stance from “Traversal (default)” to “Combat Melee”. We then use this stance to provide logic and determine what the attack button does. Default Form would change the default idle and movement animations.



Inventory Source

With the Inventory Pro work done, we can add the Inventory Pro inventory source to the character.

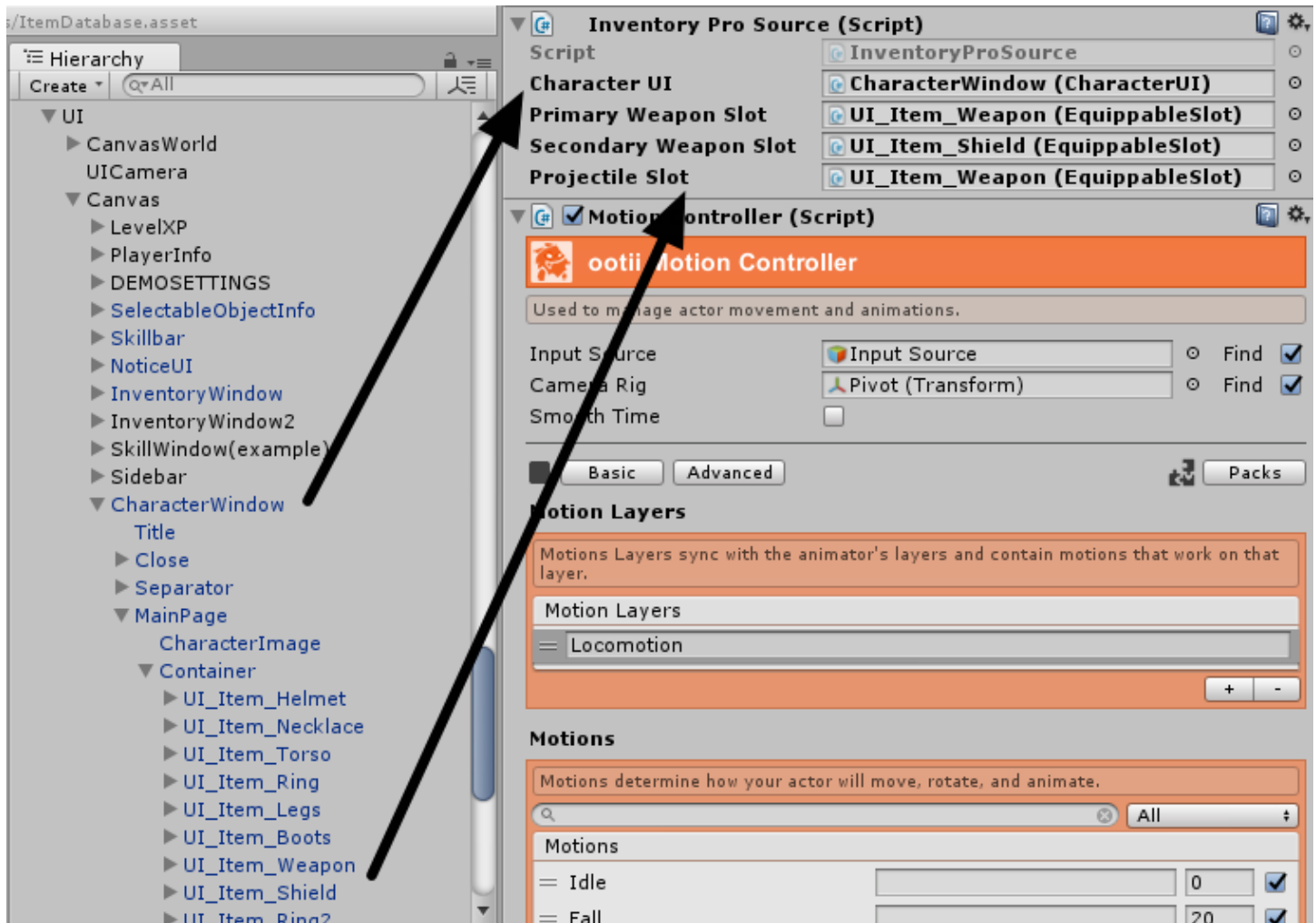
We're simply going to add the "Inventory Pro Source" component to the character. This is the same character that has the Motion Controller component and the Inventory Player component.





Inventory Pro Slots

The Inventory Pro Source needs to know what item slots you use. So, they need to be assigned.



The “Primary Weapon Slot” is typically the right hand (sword, ax, etc.)

The “Secondary Weapon Slot” is typically the left hand (shield, bow, etc.)

The “Projectile Slot” is what holds the arrow we’re using. Depending on your Inventory Pro setup, it may just be the primary slot.

Done (almost)

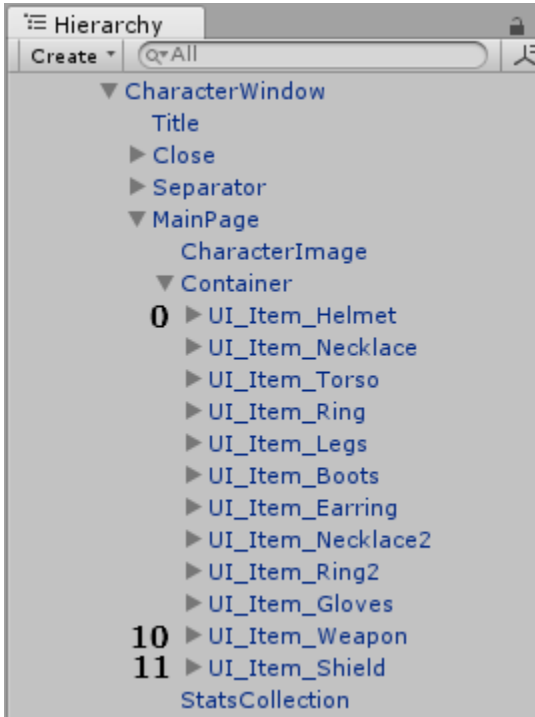
With that done, you’re good to use Inventory Pro with the Sword & Shield Motion Pack or any other equip/store motions. However, with the Archery Motion Pack, there’s one other thing we need to do...



Archery Motion Pack

Some of the motions in the Archery Motion Pack need to look at the character slots to determine what arrow to use, grab the bow, etc.

So, we need to tell those motions which slots should be used. In this case, we use the index of the slot as the identifier:



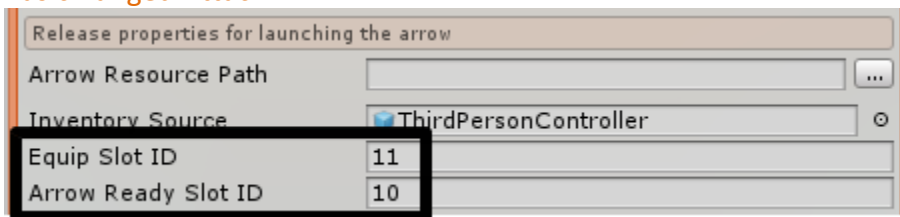
For the bow, think about what we just set as the Inventory Pro Slots above:

Secondary Weapon Slot = **11**

Projectile Slot = **10**

So, we'll use these values with the motions.

Basic Ranged Attack





Support

Learning Inventory Pro isn't a trivial task and the approach I've taken isn't the only way to do things. However, this work was done for free and you're welcome to use it with the Motion Controller.

There's no guarantee this will continue to work if Inventory Pro changes its API.

Feel free to change the code to fit your specific needs.