Third Person Controller – Melee Combat Template

(v2.2b 05/06/2017)

Thank you for support this asset, we develop this template because a lot of developers have good ideas for a Third Person Game, but build a Controller is really hard and takes too much time.

The goal on this project was always to deliver a top quality controller that can help those who wants to make a Third Person Game but are stuck trying to make a controller.

With this template, you can setup a 3D Model in just a few seconds, without the need of knowing hardcore code or wasting time dragging and drop gameobjects to the inspector, instead you can just focus on making your game.

--- Invector Team ---

Ps This Documentation is for the **MeleeCombat** Features, there is another for the Basic Locomotion and Melee Combat in their respective folders.*

Summary

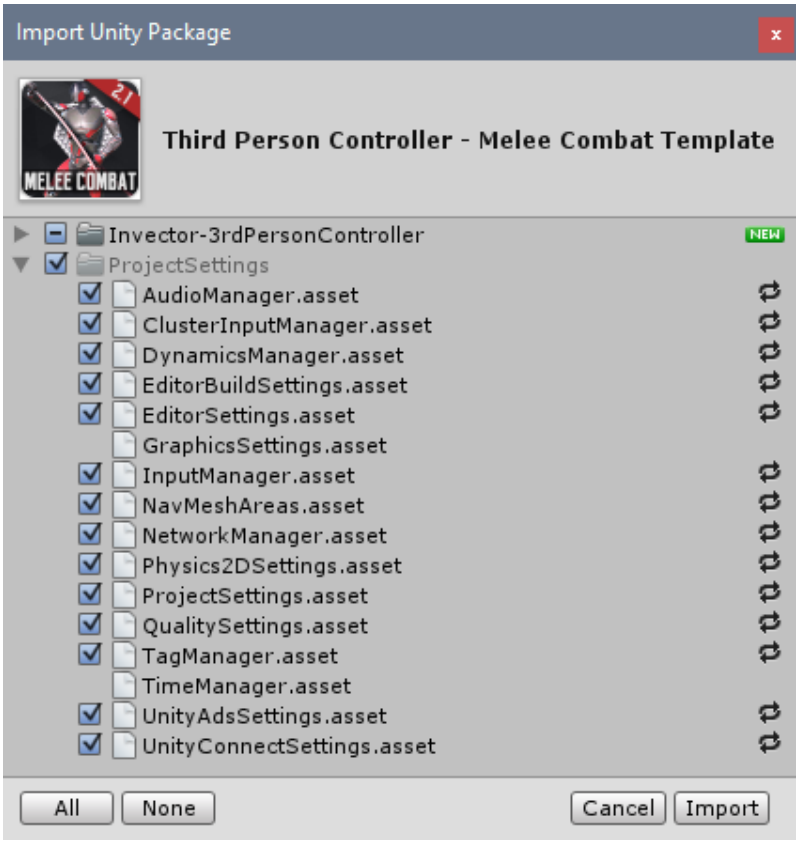
FIRST RUN

IMPORTANT

This is a **Complete Project**, and as every complete project it includes a custom **InputManager**, **Tags**, **Layers**, etc… **Make** sure that you import **on** a **Clean Project**.

- *Importing on an existent project*

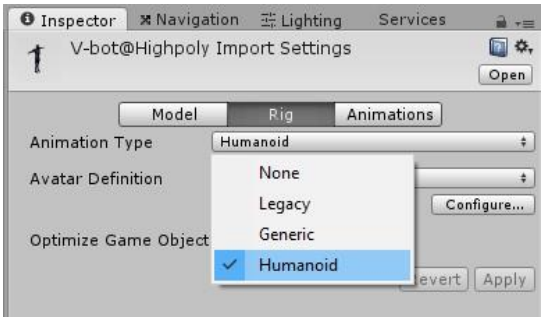
If you want to import into another project, you can UNCHECK some project settings to avoid conflicts or replace your project settings like the TagManager (which includes all the Layers), and add later the tags and layers that we use. We recommend to import the InputManager because it's kind of painful to add manually later (lots of inputs).



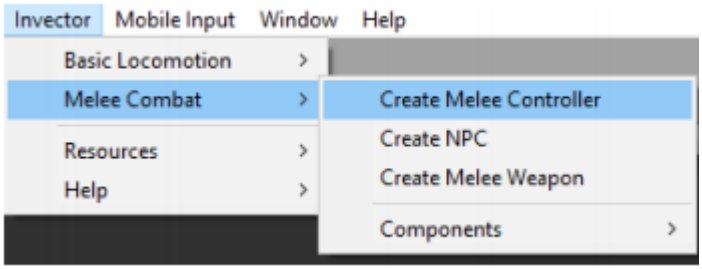
***Updates also need to be imported into a Clean Project, so MAKE SURE TO BACKUP your previous project and transfer the necessary files to your new project. ***

CREATING A NEW MELEE CONTROLLER

Make sure that your fbx character is set up as **Humanoid**



To setup a new character, go to the tab *Invector* > *Melee Combat* > *Create Melee Controller*



Make sure your Character is **Fully Rigged** and set up the FBX as a **Humanoid**, then assign the FBX to field "Humanoid" and click on the button "Create".

Done.

The **Character Creator** window will take care of all the hard work automatically and set up components such as capsule collider, layers, tags, rigibody, etc... It will create the **ThirdPersonController**, **ThirdPersonCamera** and a UI Canvas with a **HUD** to display health, stamina and other information's.

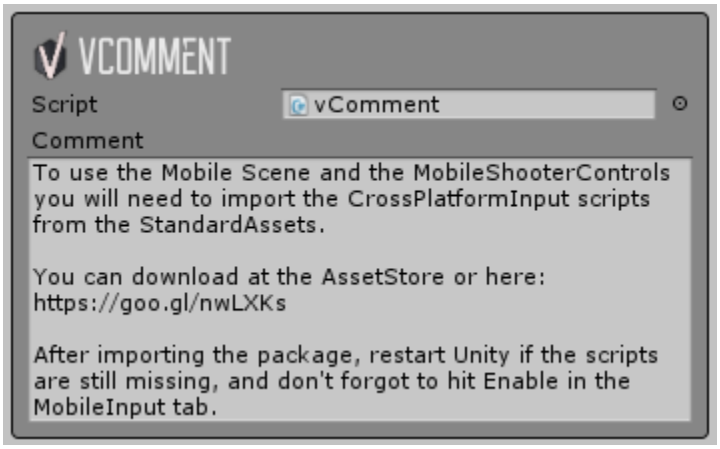
Your Capsule Collider settings will be based on your model proportions, if the capsule gets the wrong size, make sure that you rig is correct, and that your **model is using the correct Scale Factor** the same goes if the ragdoll **gets** weird.

Hit Play and enjoy ☺

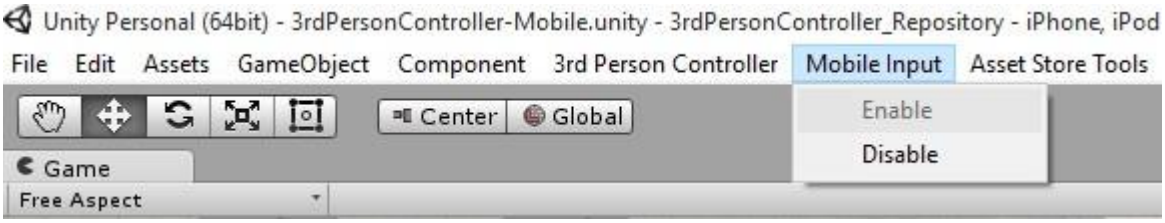
MOBILE CONTROLS

Since the release of the Shooter Template, we have to remove all content of the **StandardAssets** from our project, and since we need some files from the **CrossPlatformInput** in order to the Mobile Controls work, we have to separated those files into a package, you can [DOWNLOAD HERE]

This information is also available in the Mobile Demo Scene, in the hierarchy we add the gameObject "**___README FIRST!!!**"



After importing the package, change your platform to **Android** or **iOS** on the **Build Settings** and make sure you have the **SDK** installed and don't forget to **Enable** the Mobile Input after change the platform, it should work right on the Editor.



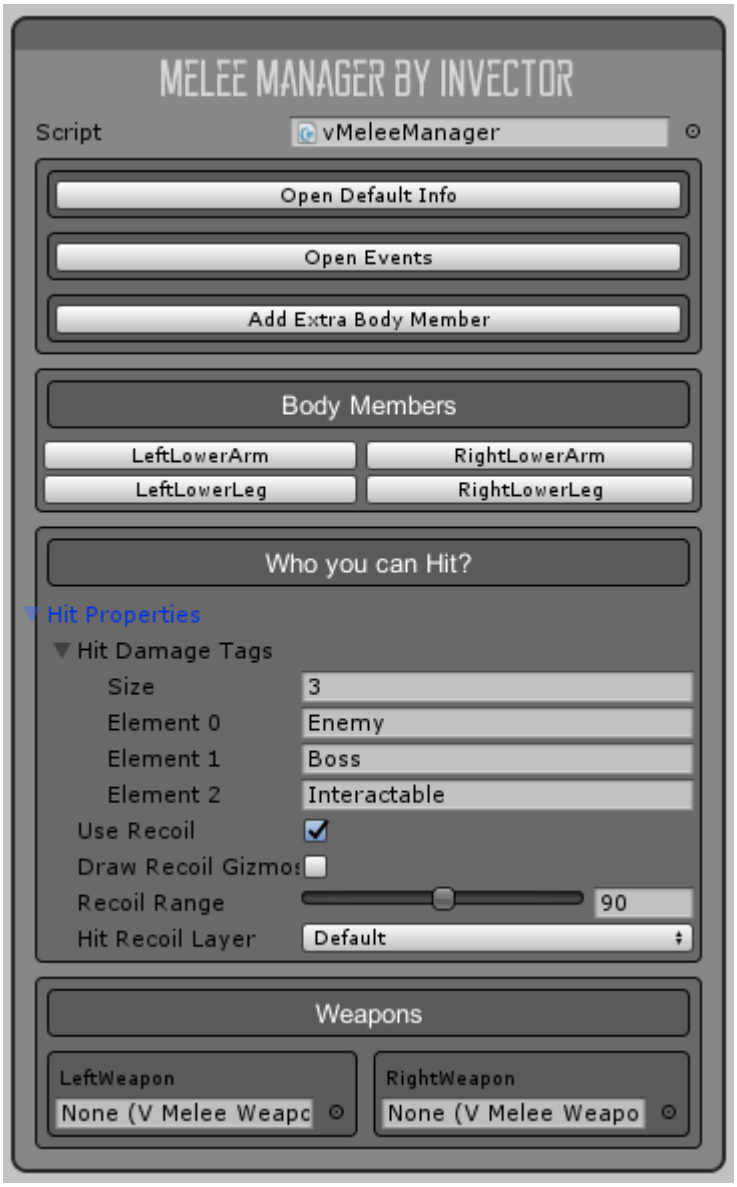
In order to have a **stable performance** on mobile devices, we recommend **compress all your textures**, set the **Quality Settings to Good or Simple**, and remove any **Camera Effects.**

If you don't want the HeadTrack in a specific animation, you can add the Tag CustomAction into the animationState and the headtrack will turn off while this animation is playing.

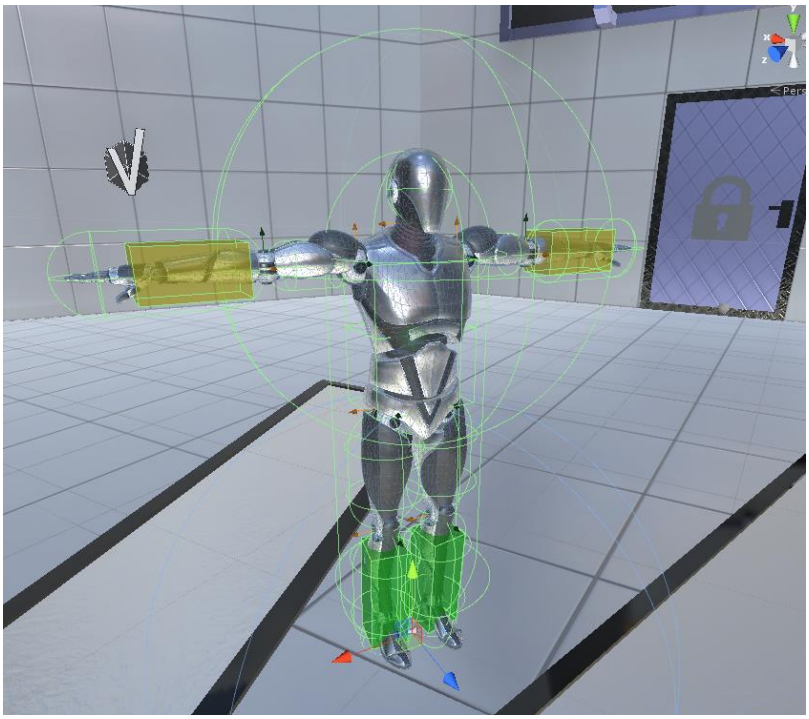
MELEE MANAGER

V2.0 - You can add a **Melee Manager** Component by opening the Invector tab > Melee Combat > Component

Open Default Info: here you can setup the default values for Hand to Hand Combat
Open Events: here you can add generic events like trigger something when you make damage
Add Extra Body Member: If you need an extra hitbox for example a Head Hitbox for a zombie, you can add
Who you can Hit > **Important** this is the tag that will receive Damage, so if you are using this component on the Player, assign the Tags of the gameObjects that you want to apply damage (the receiver need to have the method TakeDamage).
Use Recoil > Check if you want the character to trigger a recoil animation when hit a wall
Recoil Range > max angle to allow trigger the recoil animation
Hit Recoil Layer > the layer that will affect the recoil (usually it's the Default layer)



When you assign the **MeleeManager** component into your character, it will automatically create default hitboxs for both hands and legs, you can add an extra hitbox if you need.

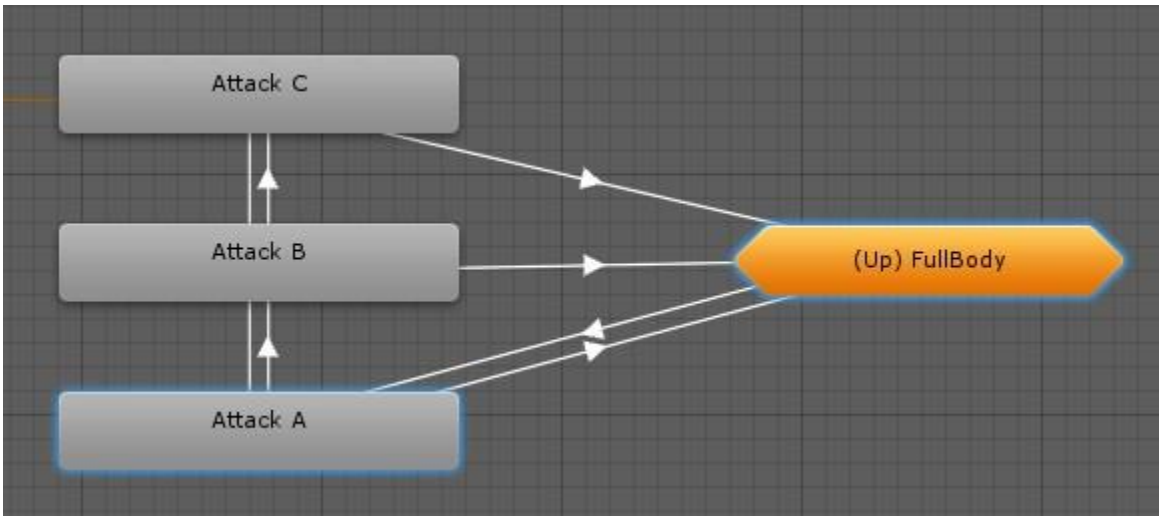


The animations for the hand to hand combat can be set up in the **Unarmed** state machine, trigger by the **ATK_ID 0** and the defense **DEF_ID 1** on the UpperBody Layer**, Default Defense.**

The **Basic Attack** State Machine is just an example, you can have as many State Machines you need, just remember to set up the **ID** to the corresponding weapon.

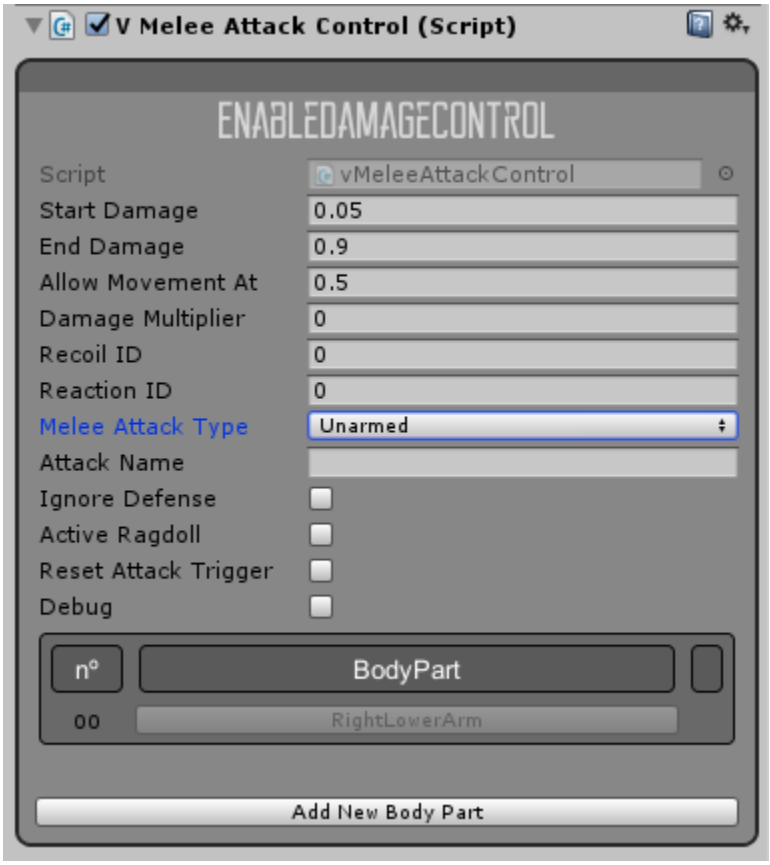
You can use **UpperBody** to attack as well, this way you can move the character and attack at the same time.

You can set up as many combo as you want, just put the attack animation and apply a transition.



Every Attack State need to have a **vMeleeAttackBehaviour** script attached.

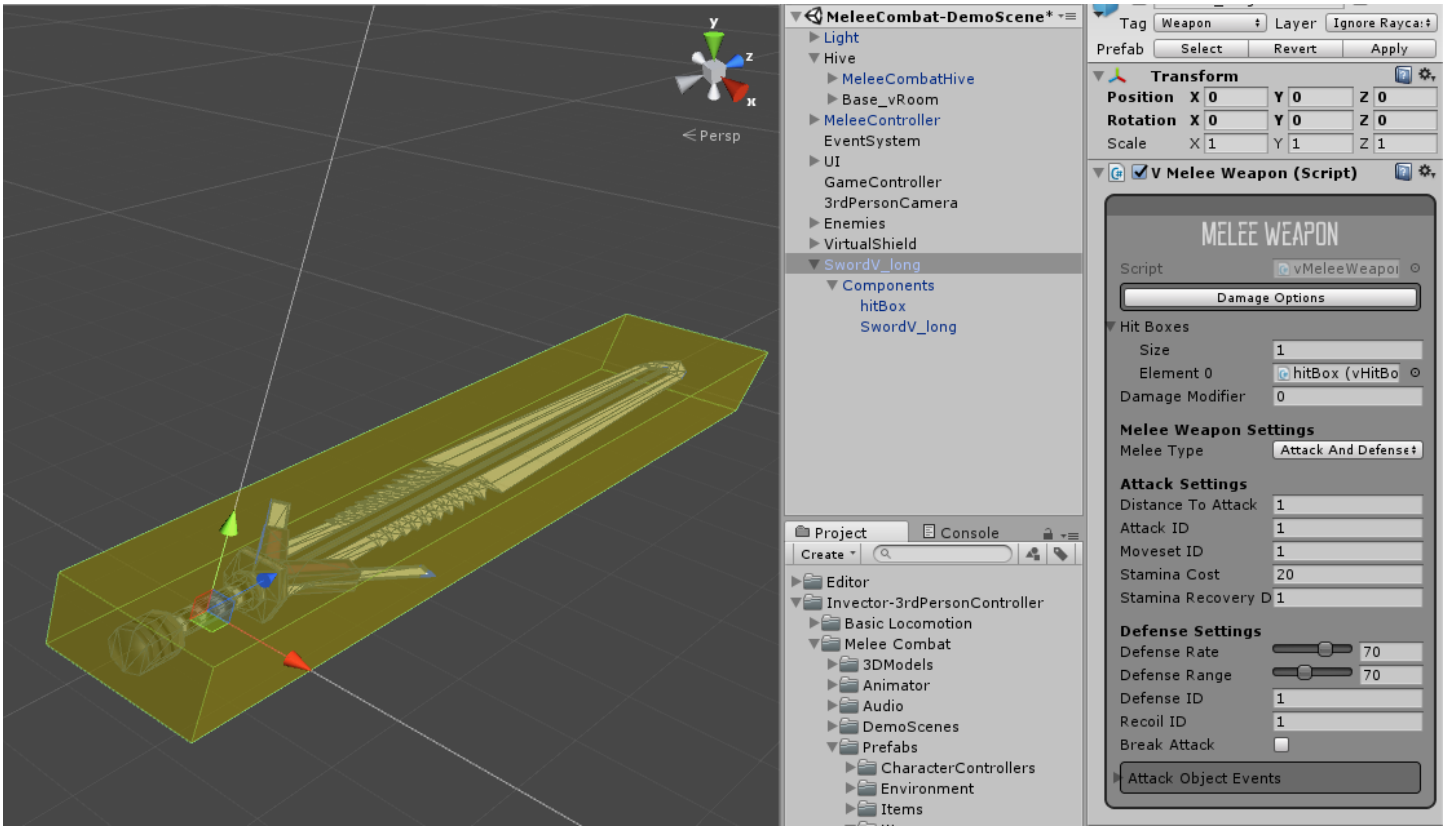
StartDamage > Time of the animation that you will apply damage
End Damage > Time of the animation that will stop trying to apply damage
Allow Movement At: free your character rotation during the attack animation
Recoil ID > Trigger a Recoil animation if you hit a wall or an object
Reaction ID > Trigger a Reaction animation when you take damage
Melee Attack Type > Select Unarmed or Melee Weapon
Reset Trigger > Check this bool for the last attack, to reset the combo
Attack Name > You can write an Attack Name to trigger different HitDamage Particles on the Target, Ex: If your weapon has electric damage, you can match the Attack Name with the HitDamage Particle and instantiate a different particle for this specific weapon.
Ignore Defense: it will apply damage even if the target is blocking
Active Ragdoll: activate the target ragdoll



Ps* Don't forget to assign the limb member of your BodyPart to match the animation, this will trigger the correct HitBox, you can add new BodyParts if your attack use more than one member.

CREATING A MELEE WEAPON

To create a new weapon, you just need to select your weapon Mesh and go to the menu Invector > Melee Combat > Create Melee Weapon.



After that your mesh will be transfer inside the Components gameobject, and the parent will have a vMeleeWeapon attached where you can set up your weapons settings.

A single hitbox will be created and if you need more you can just duplicate the first and assign into the Hitbox List into the MeleeWeapon component.

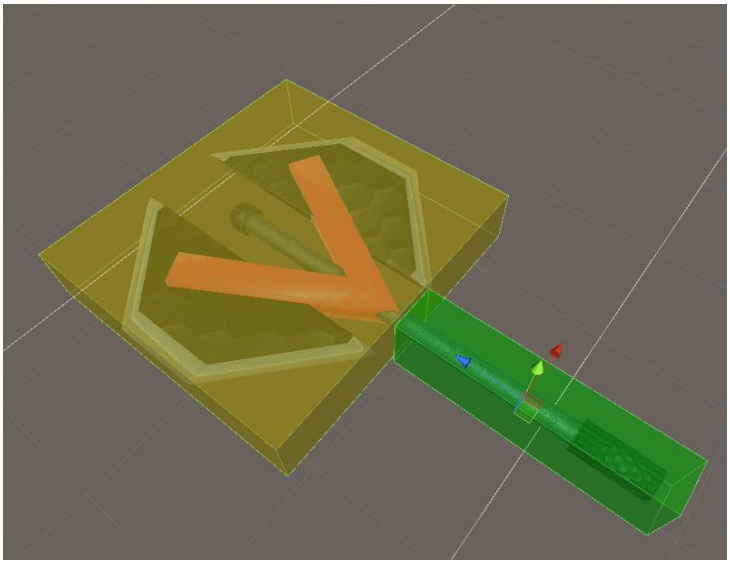
IMPORTANT – don't forget to set your **Weapon Layer to Ignore Raycast and the Tag to Weapon**, if you put a weapon into an Enemy or Player and change the Layer and children's, you need to set the weapon layer to Triggers again

After create your weapon, you can just drag and drop inside a hand Bone of your character and hit Play, the MeleeManager will auto assign into the Weapon slot.

To change weapons ingame you will need a ItemManager assign into your Character.

Attack Settings:

[Damage Options]
Value: Total damage of your weapon
Stamina Block Cost: How much stamina the target will lost when receive this attack while blocking
Stamina Recovery Delay: How much time will wait to start recover the stamina
Ignore Defense > Check if this weapon can pass through shield
Active Ragdoll > Check to make this weapon activate the Ragdoll of the target



Example of a weapon with 2 hitbox

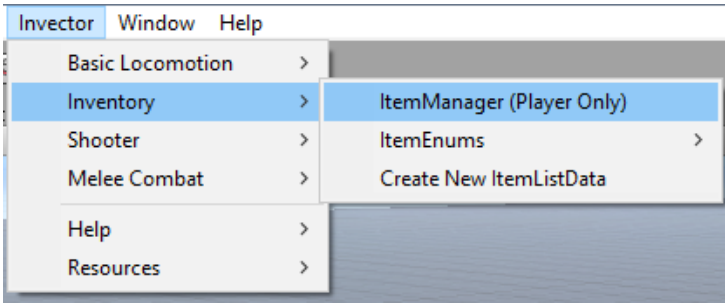
HitBoxes List: Assign your hitboxs here
Damage Modifier: Extra Damage
Melee Type: Just Attack, Just Defense or Both;
(SOON) Use Two Hand > Check this if your weapon uses two hands (the left weapon will drop)
Distance to Attack > Used for AI only, to know the distance to attack if this weapon
ATK_ID > correspond to the Attack Animation State that will trigger
MoveSet_ID > it's the correct move set that the character will move when using this weapon
Stamina Cost > how much stamina the attack will cost
Stamina Recovery Delay > how much time will take to the stamina start recovery

Defense Settings:

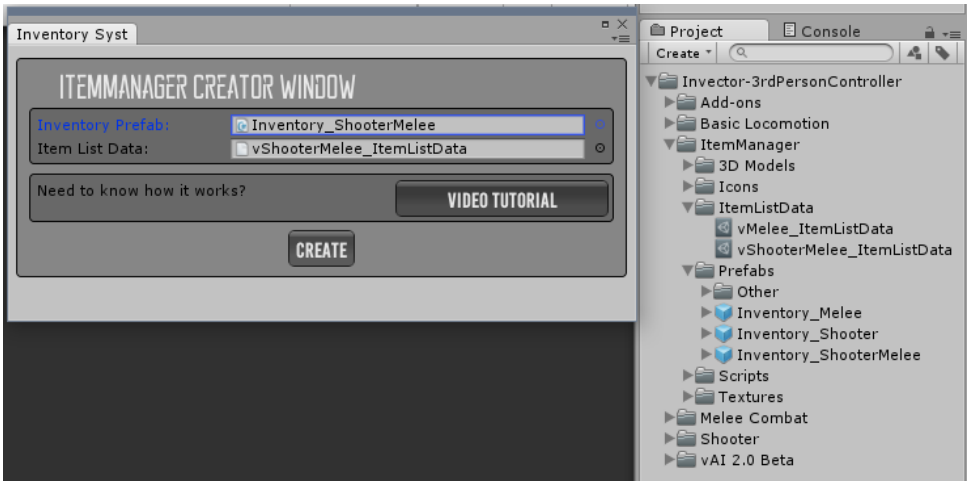
DEF ID > ID of your defense animation
Recoil_ID > Trigger a recoil animation
Defense Rate > how much damage you can defend from an attack
Defense Range > When you select the shield, a Gizmos will appear to help you see how much of Defense Range you need.
Break Attack > Trigger a Recoil Animation on the Attacker

ITEM MANAGER (INVENTORY)

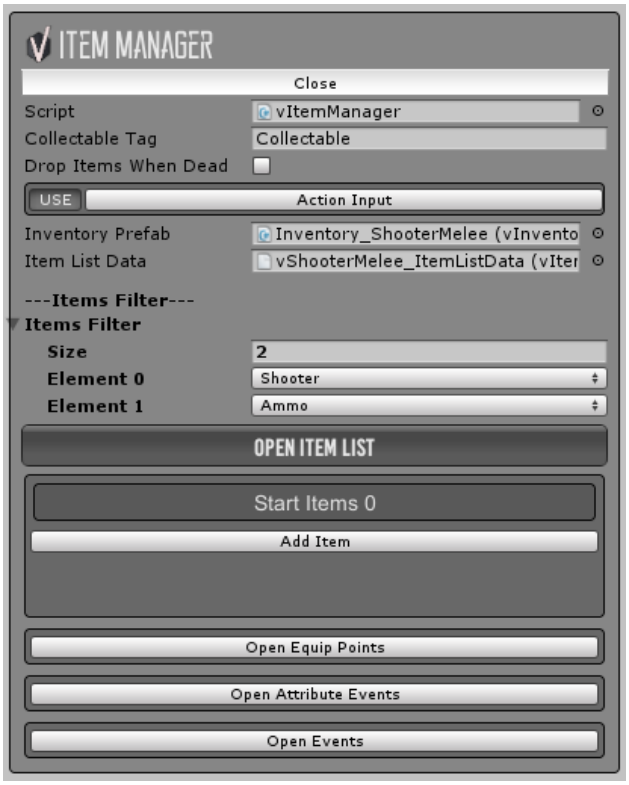
- Add the ItemManager into your Player from the menu **Invector > Inventory > ItemManager**



- Select the Inventory Prefab from the **Project > ItemManager > Prefabs**
- and a **ItemListData** > **vMelee_ItemListData**



- You can use the **ItemFilter** to filter the items that you want to show



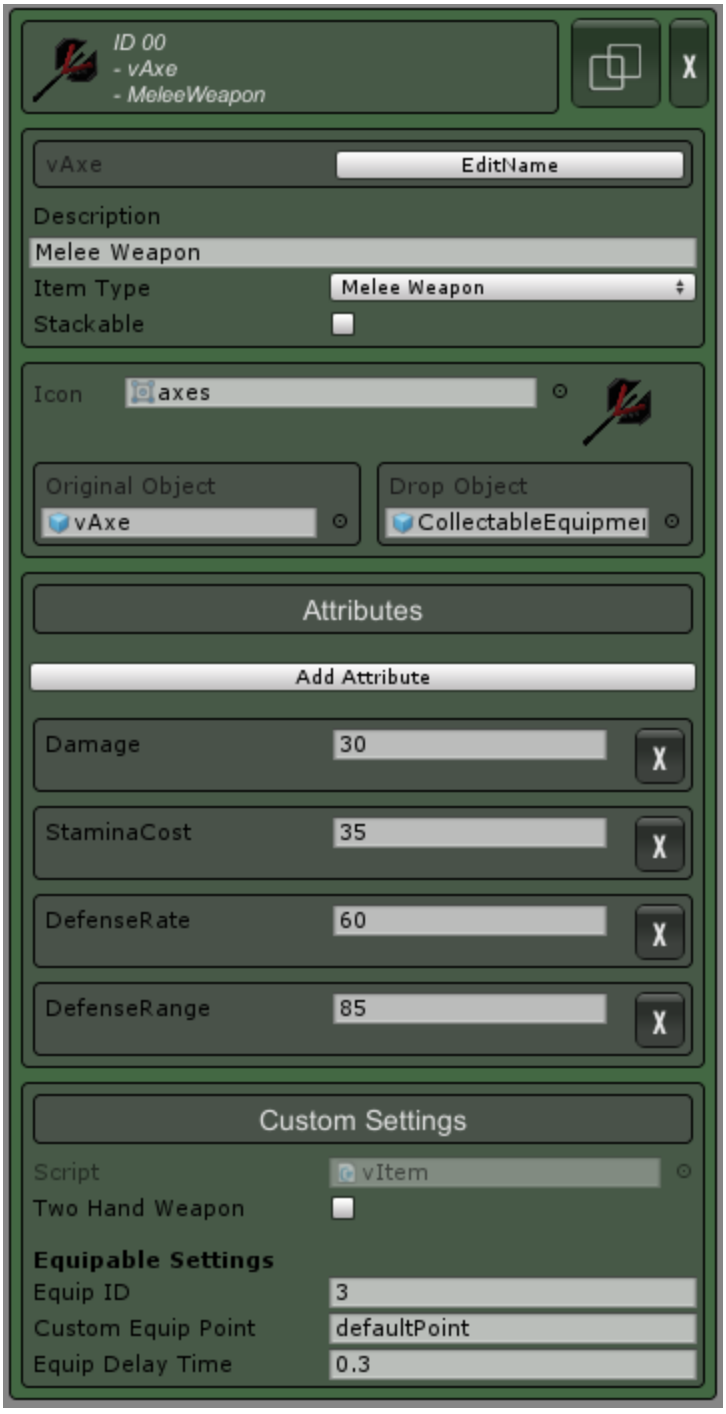
We have added a window for **Events** that are very useful, for example if you want to lock the input of the character while the Inventory is Open, just assign the Character and call the method.

You also have the option to Drop all your items when you die, it will instantiate the Prefab that you select to be the Collectable of your item.

By clicking in **Open Item List**, a new window will open and you can create new items there.

You can create new items or duplicate a current one, keep in mind that each item has a unique ID.

When creating a Weapon Item, you need to assign the **Original Object** (that instantiate into the Player with a vMeleeWeapon or vShooterWeapon) and a **DropObject** which we have a prefab called "**CollectableEquipment**" that you can use and it will automatically drop the item you assign or create a unique collectable with a mesh that matchs your item.



Don't forget to add the attributes of your weapon, this will allow you to drop and collect your weapon with the same attributes, making it into a unique weapon.

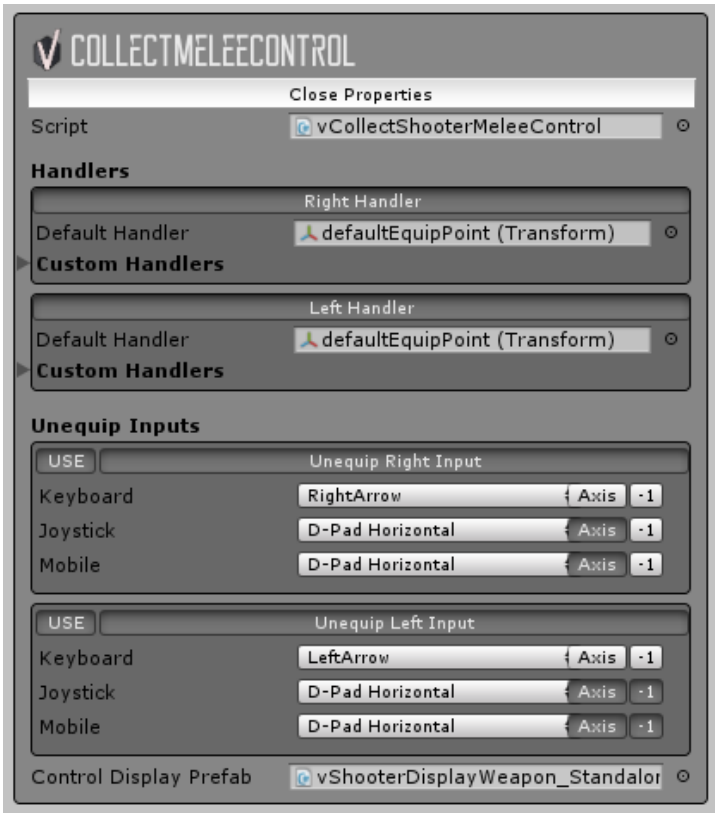
This Inventory Example goes further and further into options to customize, like consumable itens, if is stackable or not, and much more that is better explained on video tutorials that you can watch on our Youtube Channel.

COLLECTABLE STANDALONE (NO INVENTORY)

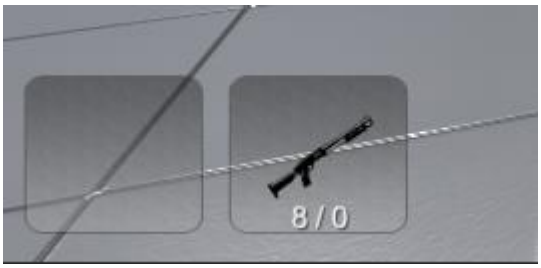
If you don't want to use the ItemManager to manage your items, we have another solution for 'on demand' collecatables, notice that you can only equip 1 item, once you try to equip another the current item will drop.

Take a look into the Demo Scene call "vShooterMelee_NOInventory", instead of adding the ItemManager component, now you will add the "vCollectShooterMeleeControl" component to automatically collect and equip weapons.

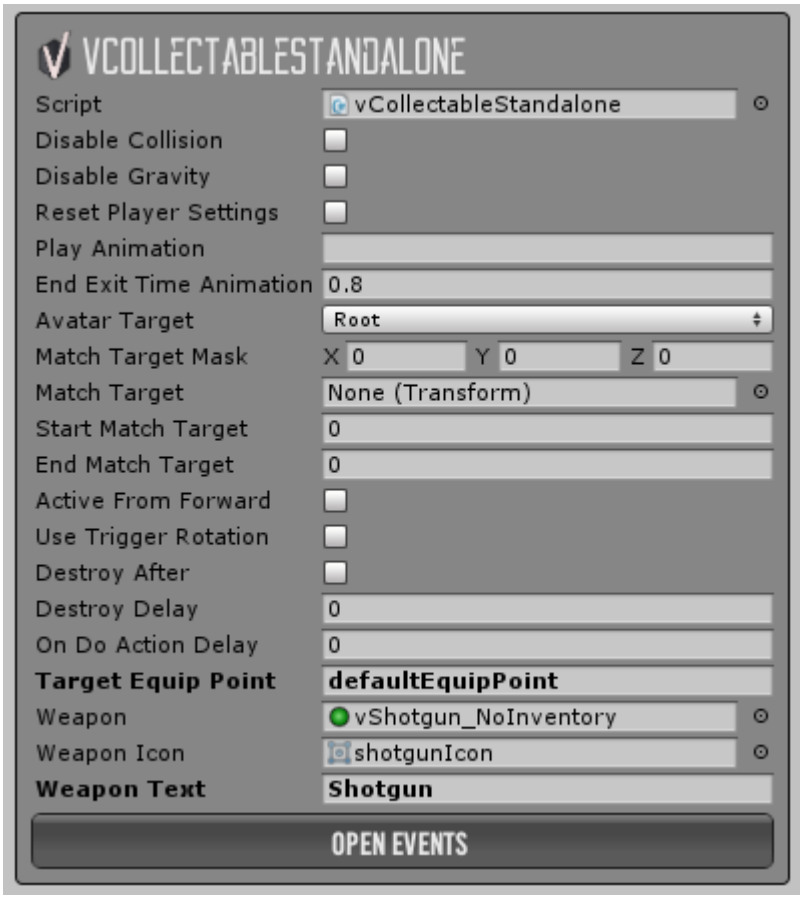
You need to create the defaultEquipPoint to equip weapons and assign inputs to drop them.



We also have a pretty simple example of a Display HUD to show what weapons you're equipped with, it's call "vShooterDisplayWeapon", search in the project folder and drag and drop the prefab into the scene.



For the ItemManager we need a prefab for the actual weapon that goes into the Player and another to be the Collectable, but in this case the CollectableStandalone is both. Take a look into one of the several example of collectables we have for both melee and shooter weapons.



It's important to assign the correct gameobjects into the Events, we turn off the collision and gravity of the weapons when equipped and turn on when you drop them.

CREATING AN ENEMY AI

Invector > Melee Combat > Create NPC and change the **Character Type** to **Enemy AI**

After hit the Create button, our scripts will handle all the most time consuming stuff and make the AI almost done to hit Play, you just need to BAKE a NavMesh on the Scene.

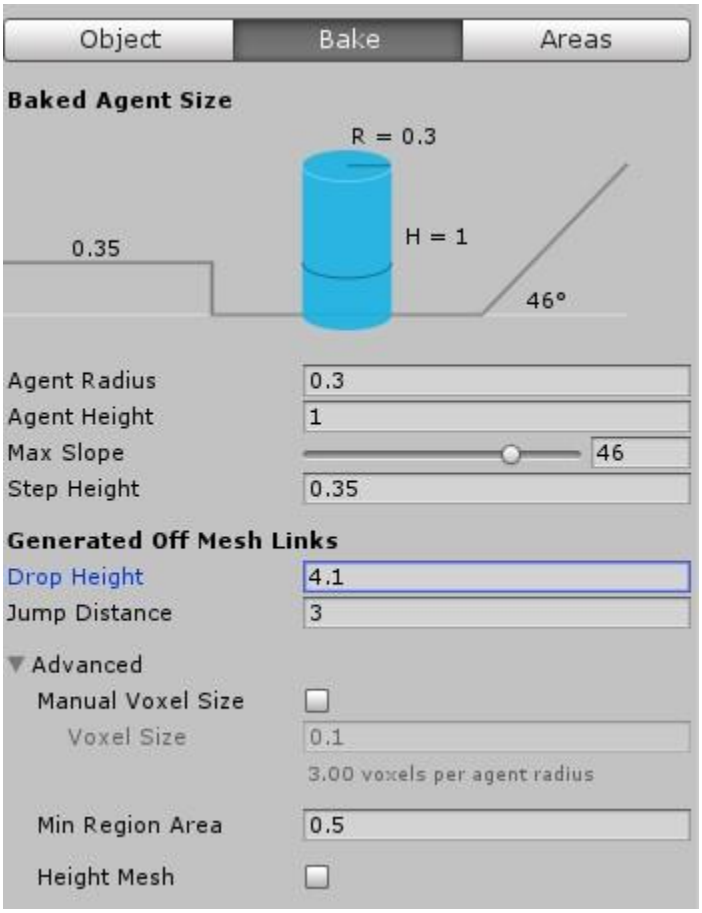
Locomotion - It works pretty much the same as the Character Controller, you still need to set up the

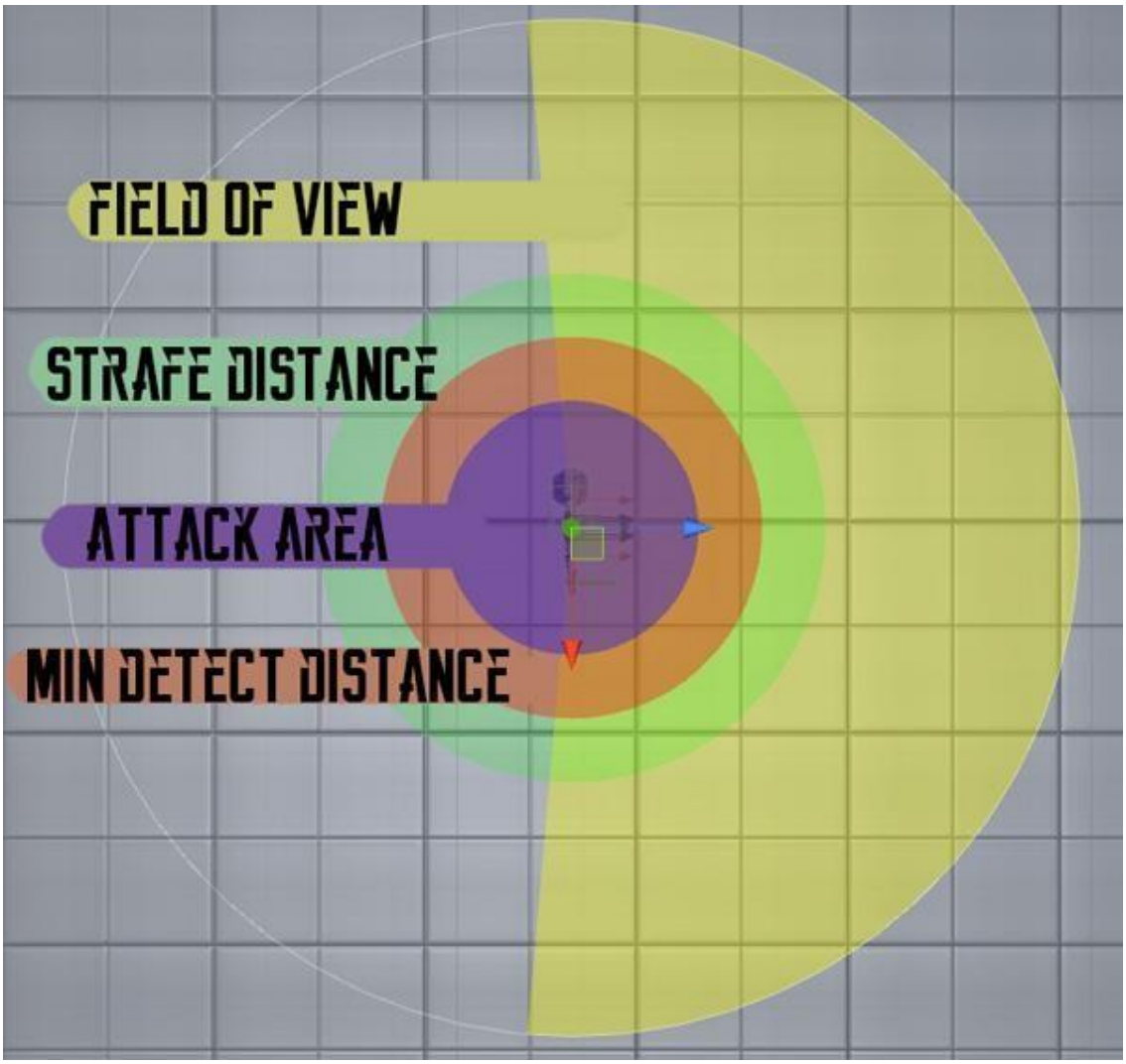
Layers – just like the Player, you need to set up a Layer for the AI (Enemy) and a layer for the Ground (Default). If you equip a Weapon on the character, don't forget to assign the Layer Triggers for this weapon.

Combat - Here you will have a lot of options to make very different combat behavior, you can add a chance to block (if equipped with a defense weapon), chance to roll, chance to defend an attack, change the attack frequency, strafe around the target, etc…

Waypoints - You can add waypoints for the AI to follow in sequence of activate the option to Random.

We manage to get better results with the NavMesh using this set up, but of course this will depend on your scene, terrain, meshs, etc…





Field of View > total range to detect the Player

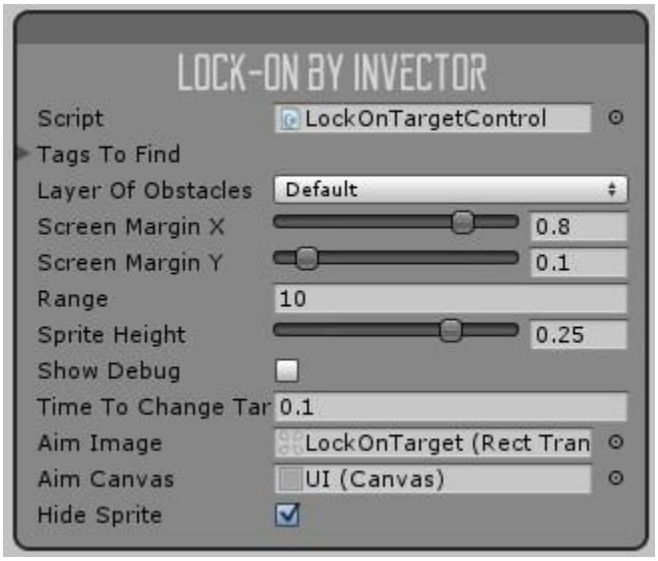
Strafe Distance > once in combat, the character will move Strafing

Attack Area > total range to attack

Min Detect Distance > Min distance to detect the player, even if the player is outside the Field of View range.

LOCK-ON TARGET

You can add a Lock-on component into the Camera by opening the 3rd Person Controller menu > Components > Lock-On. The component will be ready to use, you can set up the input that activate the Lock-on in the **ThirdPersonController** script, at the method **LockOnInput.**



You can also display a **Sprite Image** into the Target by assigning an Image and Canvas.

Hide Sprite will hide the sprite if the target if lock-on is false. Set off-set Y by changing the value of the **Sprite Height**.

This Lock-On currently works exclusively with our AI, it will not work out of the box with Non-Invector Characters because it need's the **vCharacter** interface to know if the target is alive. You can assign a **vCharacterStandalone** script into your gameobject, it contains health and a **TakeDamage** method to receive damage.

WAYPOINT SYSTEM

You can create a Waypoint Area by opening the 3rd Person Controller > Component > New Waypoint Area. To create a new **Waypoint**, just hold *Shift + Left Click* on any surface with a collider, to reposition the same waypoint hold *Shift + Right Click*. The same goes to create Patrol Points, but you will hold Ctrl instead of Shift. You can assign this Waypoint Area to many AI as you want, and limit the area / limit of AI that will access.

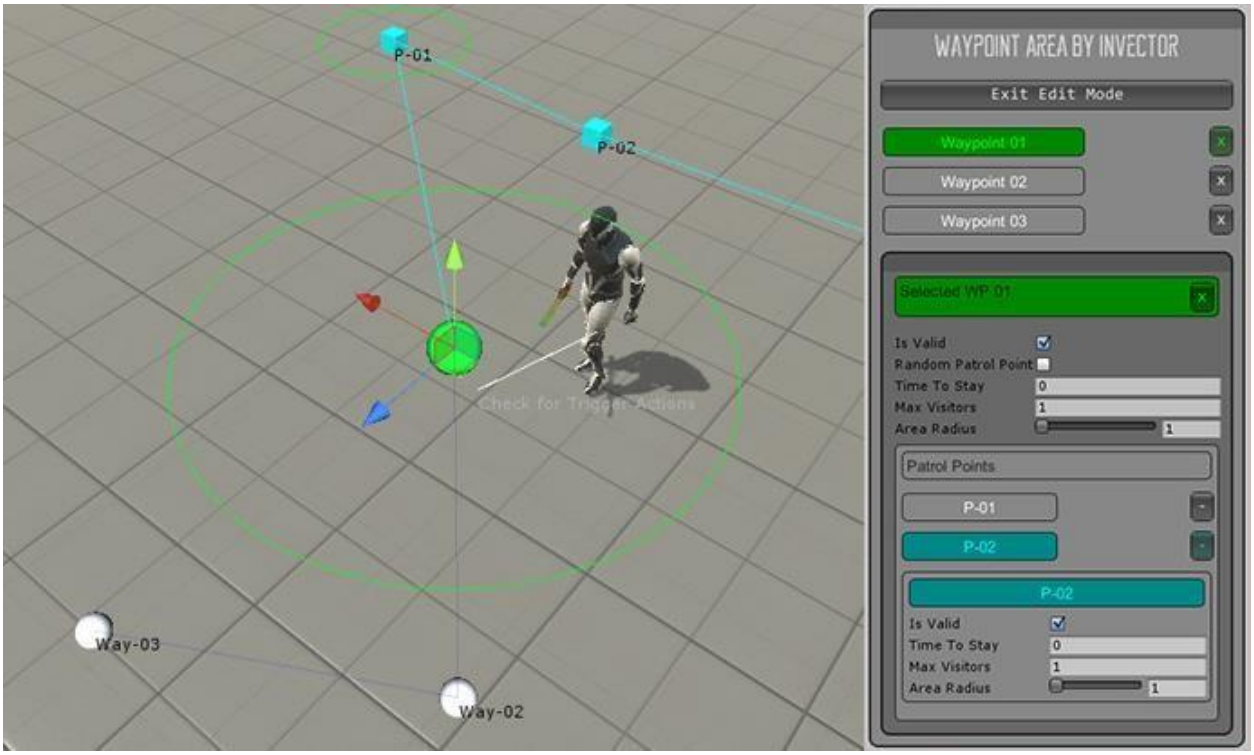
Patrol Points are points of interest that one waypoint has, for example if you have a corridor with 3 rooms, you can create 1 waypoint in the middle of the corridor and 3 patrol points with **Max Visitors of 1**, this means that if an AI is already on a room, the other AI will not come to the same room, he will go to the next one.

Time to Stay is how much time the AI will stand there.

isValid is a bool that you can turn on/off to disable a waypoints/patrol point in real time.

You can make the AI walks randomly at waypoints by selecting the option **Random Waypoints** on the AI Inspector. To make random patrol points, select the option **Random Patrol Point** on the Waypoint Inspector.

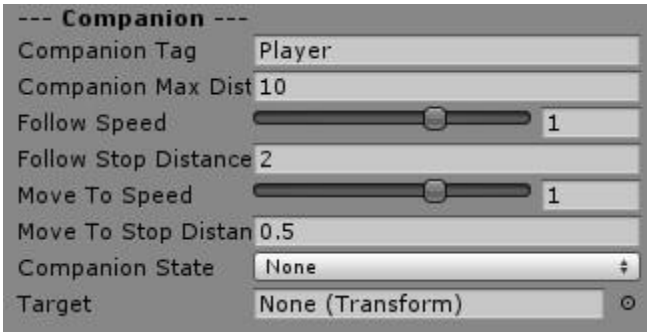
Waypoints are represented by Spheres and Patrol Points are represented by Cubes.



CREATING A COMPANION AI

Same process as creating a Player or EnemyAI, just select CompanionAI on the Character Type.

The Companion has the same variables as the EnemyAI, plus:



If you open the v_AICompanion script, you will see that we have a method call CompanionInputs and you can customize for your needs, this method contains the basic commands like Follow, Stay, Aggressive/Passive and MoveTo (you can send the AI to a specific spot by an Vector3)

Default Inputs:

- Stay
- Follow
- Aggressive/Passive
- Move to (moveToTarget)

*Notice that the transform target height can be no higher than 0.5f from the navmesh, otherwise he can't find a path to go.