



Third Person Controller - Shooter Template

(v1.1a 06/05/2017)

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 **Shooter Features**, there is another for the **Basic Locomotion** and **Melee Combat** in their respective folders.*

Summary

FIRST RUN.....	3
CREATING A NEW SHOOTER CONTROLLER.....	4
ITEM MANAGER (INVENTORY)	6
HOW TO ALIGN A SHOOTER WEAPON AND LEFTHAND IK	9
COLLECTABLE STANDALONE (NO INVENTORY)	11
MOBILE CONTROLS	13
HEAD TRACK.....	14
LOCK-ON TARGET	16

FIRST RUN

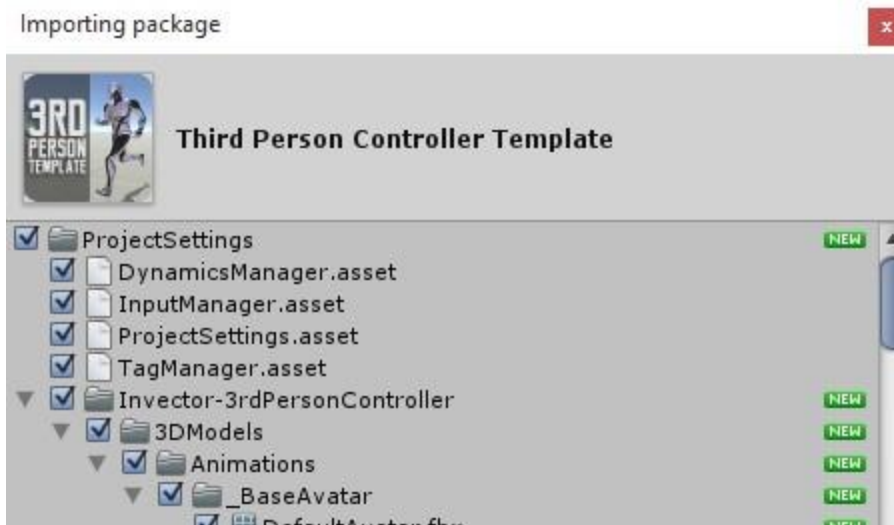
IMPORTANT

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*

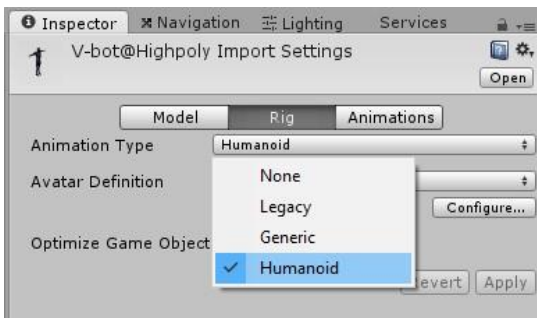
If you want to import into another project, you can UNCHECK some project settings to avoid conflicts or replace your project settings like the TagManager (which includes all the Layers), and add later the tags and layers that we use. We recommend to import the InputManager because it's kind of painful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



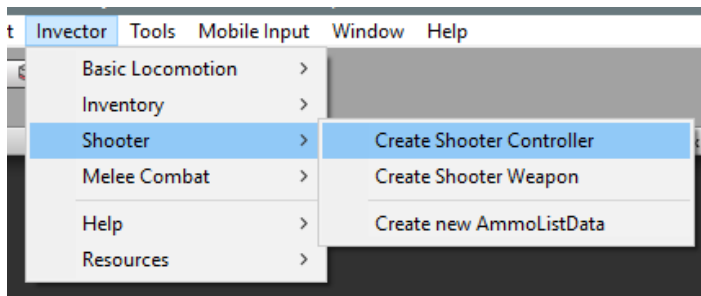
***Updates also need to be imported into a Clean Project, so MAKE SURE TO BACKUP your previous project and transfer the necessary files to your new project. ***

CREATING A NEW SHOOTER CONTROLLER

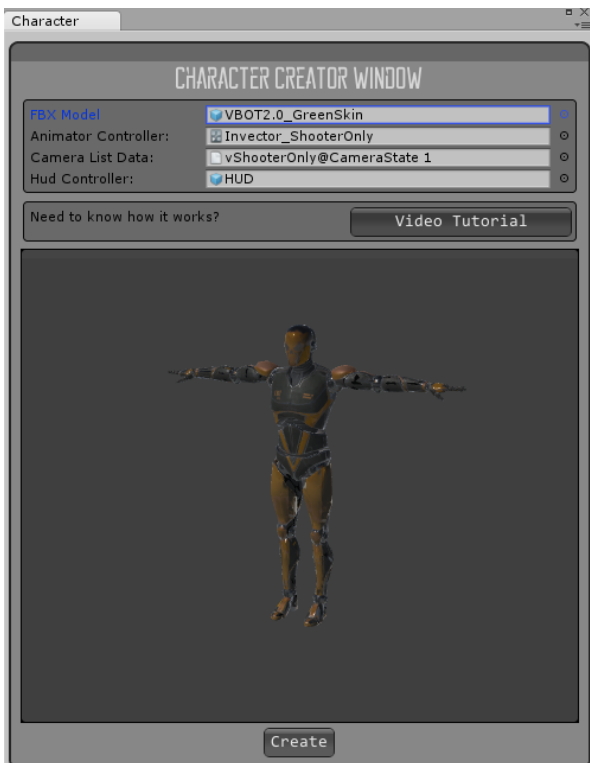
Make sure that your fbx character is set up as **Humanoid**



To setup a new character, go to the tab *Investor > Shooter > Create Shooter Controller*

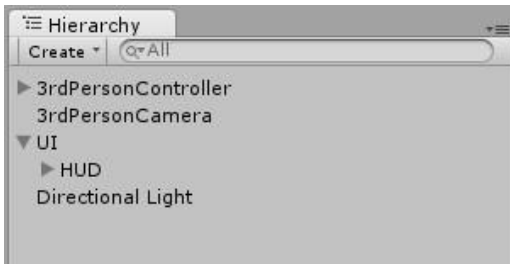


Make sure your Character is **Fully Rigged** and set up the FBX as a **Humanoid**, then assign the FBX to the field "FBX Model" and click on the button "Create".



Ps Make sure to select the **Invector_ShooterOnly** or **Invector_ShooterMelee** if you want to use both shooter and melee as your Animator Controller, you can find the file at the folder: Shooter > Animator, or just click on the little circle icon.*

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.

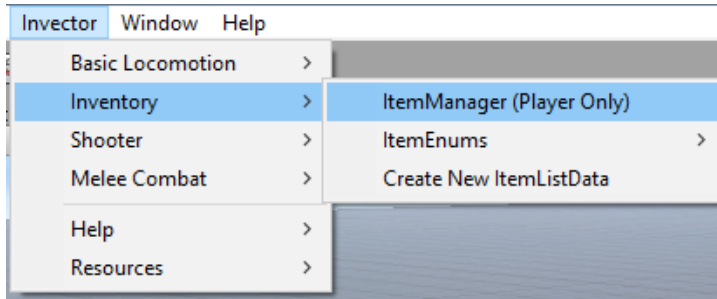


Your Capsule Collider settings will be based on your model proportions, if the capsule gets the wrong size, make sure that you rig is correct, and that your **model is using the correct Scale Factor** the same goes if the ragdoll **gets** weird.

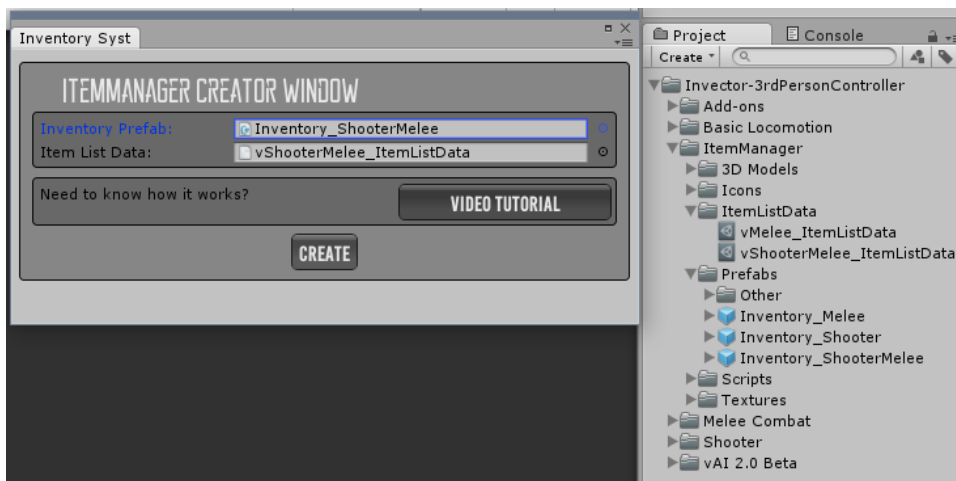
Hit Play and enjoy 😊

ITEM MANAGER (INVENTORY)

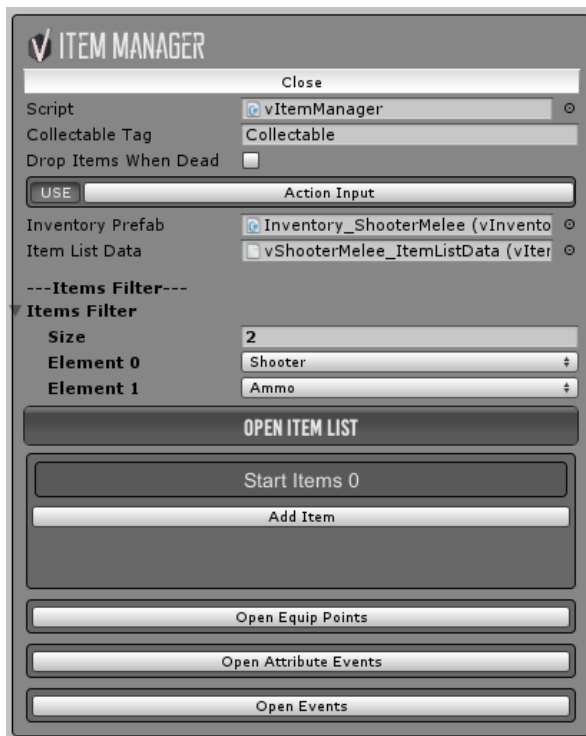
- Add the ItemManager into your Player from the menu **Invectro > Inventory > ItemManager**



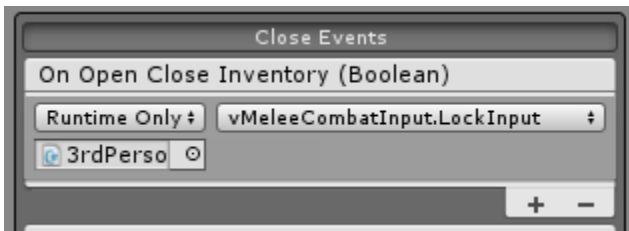
- Select the Inventory Prefab from the **Project > ItemManager > Prefabs**
- and a **ItemListData > vShooterMelee_ItemListData**



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



We also added **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 LockInput from the vMeleeCombatInput.



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.

Click in the **Open Item List** button, to manage or create new items



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 matches your item.

The image shows a detailed configuration panel for a weapon, specifically a Shotgun. At the top, it displays 'ID 12' and the name 'Shotgun' with a small icon of the weapon. Below this, there's a section for basic properties: 'Shotgun' (with an 'EditName' button), 'Description' (set to 'Tactical Shotgun'), 'Item Type' (set to 'Shooter'), and 'Stackable' (unchecked). The 'Icon' field shows 'shotgunIcon' with a preview of the weapon. The 'Original Object' is set to 'vShotgun' and the 'Drop Object' is set to 'vCollectableShotgun'. Below these are 'Attributes' with 'Add Attribute' button, 'Damage' set to 50, and 'AmmoCount' set to 8. The 'Custom Settings' section includes 'Script' set to 'vItem', 'Two Hand Weapon' checked, and 'Equipable Settings' with 'Equip ID' set to 3, 'Custom Equip Point' set to 'defaultPoint', and 'Equip Delay Time' set to 0.5.

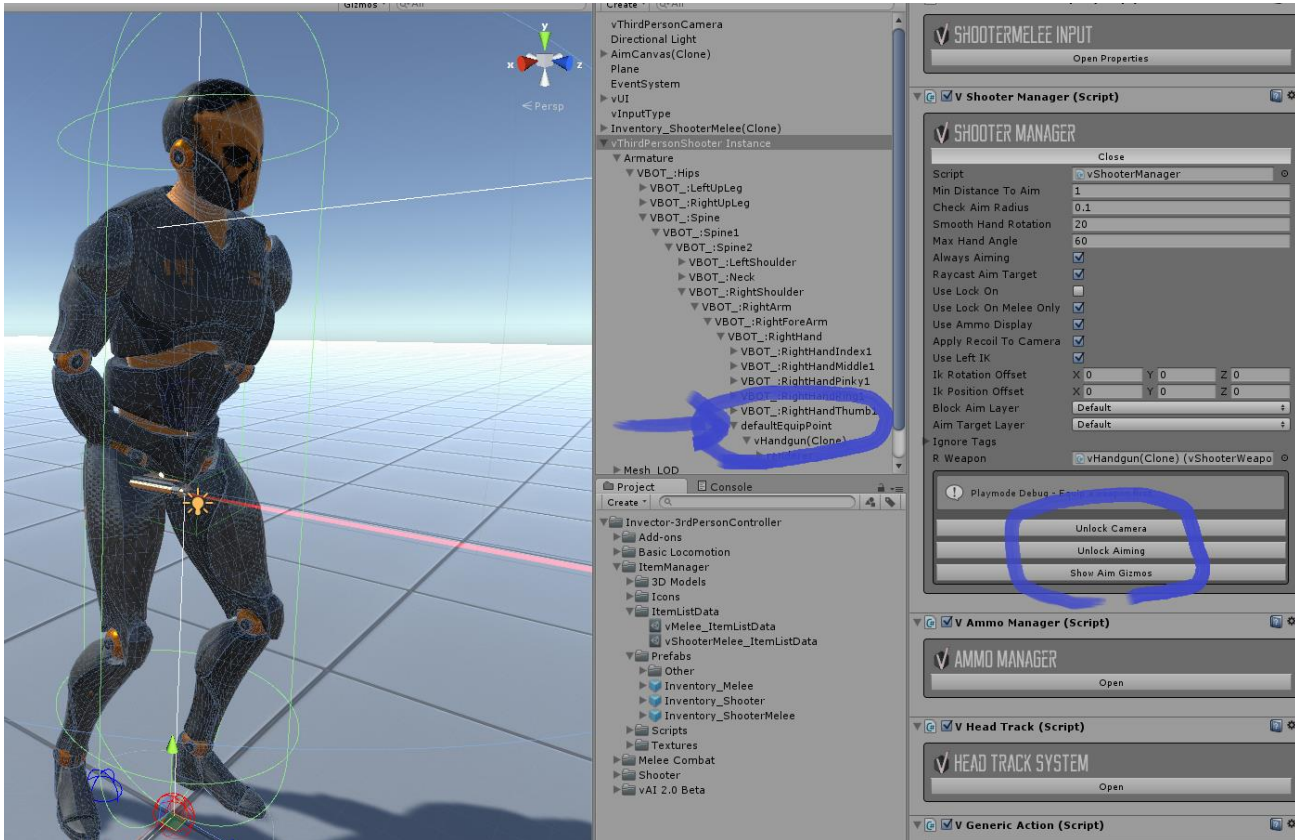
ID 12	
- Shotgun	
- Shooter	
Shotgun	EditName
Description	
Tactical Shotgun	
Item Type	Shooter
Stackable	<input type="checkbox"/>
Icon	shotgunIcon
Original Object	vShotgun
Drop Object	vCollectableShotgun
Attributes	
Add Attribute	
Damage	50
AmmoCount	8
Custom Settings	
Script	vItem
Two Hand Weapon	<input checked="" type="checkbox"/>
Equipable Settings	
Equip ID	3
Custom Equip Point	defaultPoint
Equip Delay Time	0.5

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

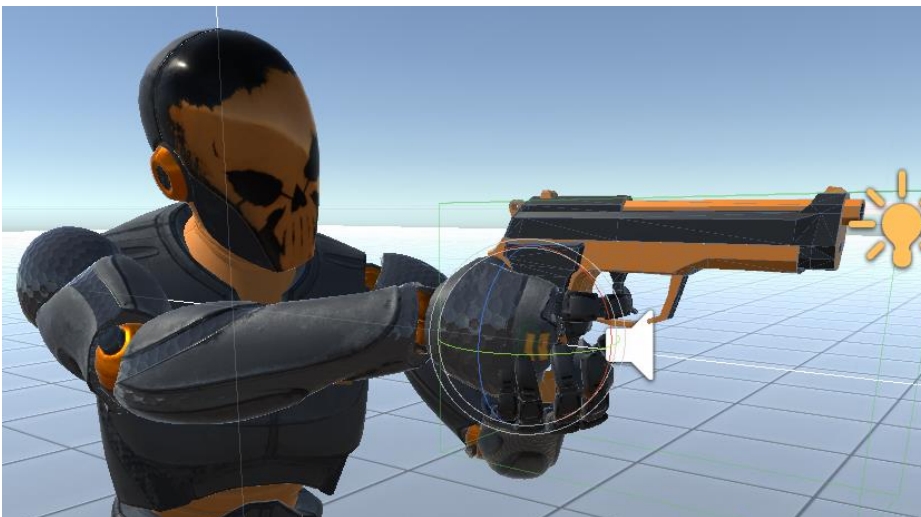
This Inventory Example goes further and further into options to customize, like consumable itens, if is stackable or not, and much more that is better explained on video tutorials that you can watch on our [Youtube Channel](#).

HOW TO ALIGN A SHOOTER WEAPON AND LEFTHAND IK

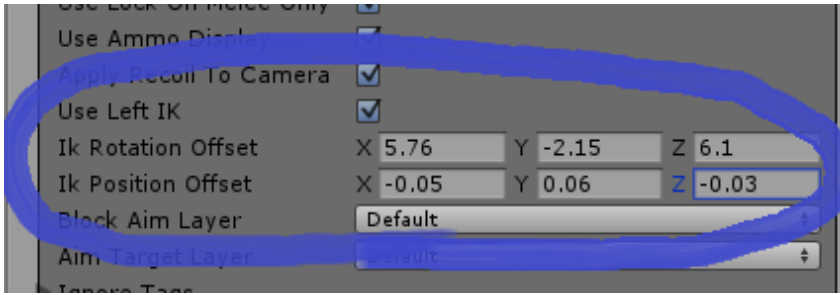
- try add a weapon into the inventory, hit play, equip the weapon and aim (you will see that the character will aim down)
- go to the inspector and find the **ShooterManager**, hit the button **LockCamera** and **LockAiming**



- to fix the position of the weapon and arms, change the transform of the **defaultEquipPoint** and not the weapon itself or the renderer inside, start by rotating the defaultEquipPoint so the IK can work to aim forward, then position the weapon into the hand.
- copy the transform component of the defaultEquipPoint and paste the values after exiting PlayMode, now all the weapons should be align correctly.



- Now to adjust the **LeftHandIK Rotation and Position OffSet** go to the **ShooterManager** and insert values until you get the results you need (each character rig will use different values). Don't forget to copy and paste this values after exiting PlayMode.



- to create new **ShooterWeapons**, it's easier to just modify the prefab of one already setup by dragging and drop into the scene and replace the model/values.

There is a crazy amount of options to customize your weapon, what projectile will shot, sounds, effects, offsets for IK, type of damage based on distance, events and much more...

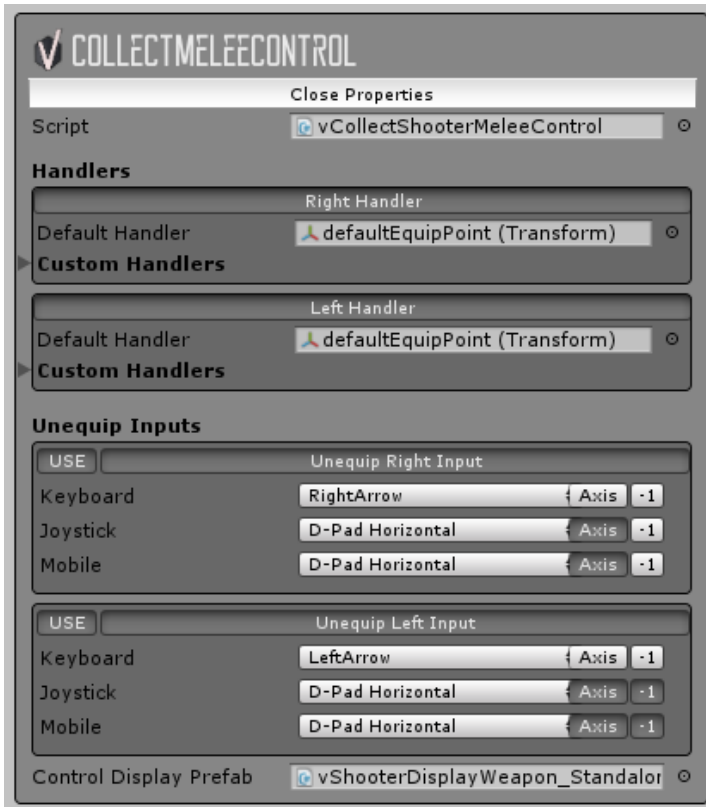
You can also make your Shooter Weapon be a MeleeWeapon by attaching the vMeleeWeapon Component with a hitbox just like a MeleeWeapon, we have several examples.

COLLECTABLE STANDALONE (NO INVENTORY)

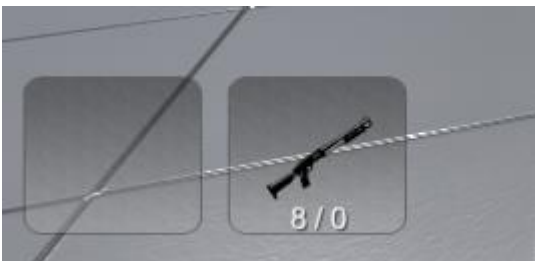
If you don't want to use the ItemManager to manage your items, we have another solution for 'on demand' collectables, notice that you can only equip 1 item, once you try to equip another the current item will drop.

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.

VCOLLECTABLESTANDALONE

Script

vCollectableStandalone

Disable Collision

☐

Disable Gravity

☐

Reset Player Settings

☐

Play Animation

End Exit Time Animation

0.8

Avatar Target

Root

Match Target Mask

X 0Y 0Z 0

Match Target

None (Transform)

Start Match Target

0

End Match Target

0

Active From Forward

☐

Use Trigger Rotation

☐

Destroy After

☐

Destroy Delay

0

On Do Action Delay

0

Target Equip Point

defaultEquipPoint

Weapon

vShotgun_NoInventory

Weapon Icon

shotgunIcon

Weapon Text

Shotgun

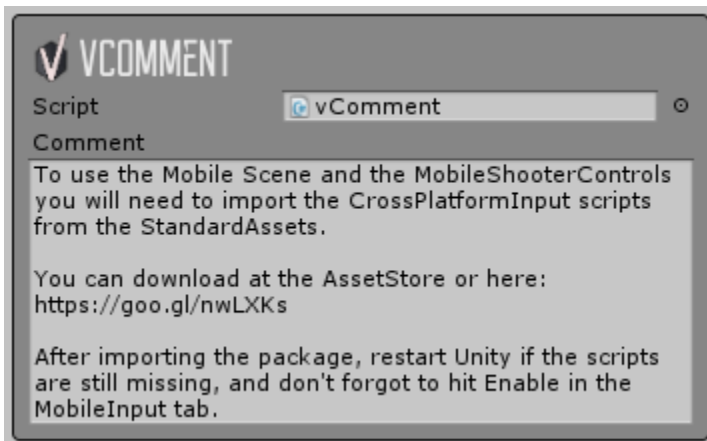
OPEN EVENTS

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.

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.

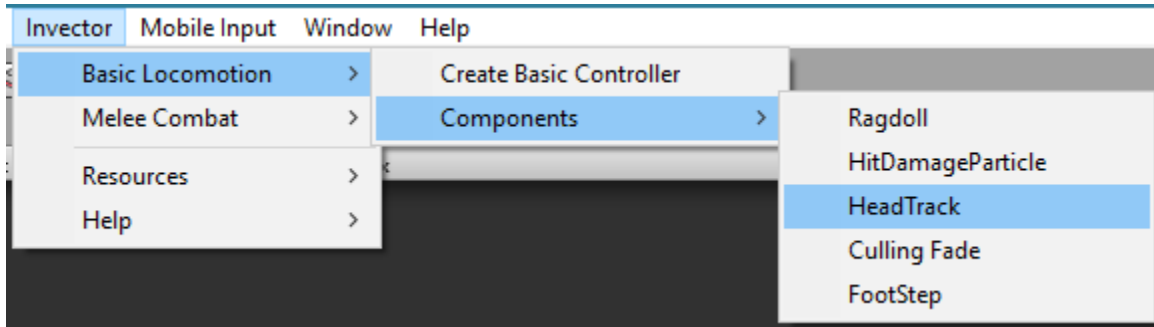


In order to have a **stable performance** on mobile devices, we recommend **compress all your textures**, set the **Quality Settings** to **Good** or **Simple**, and remove any **Camera Effects**.

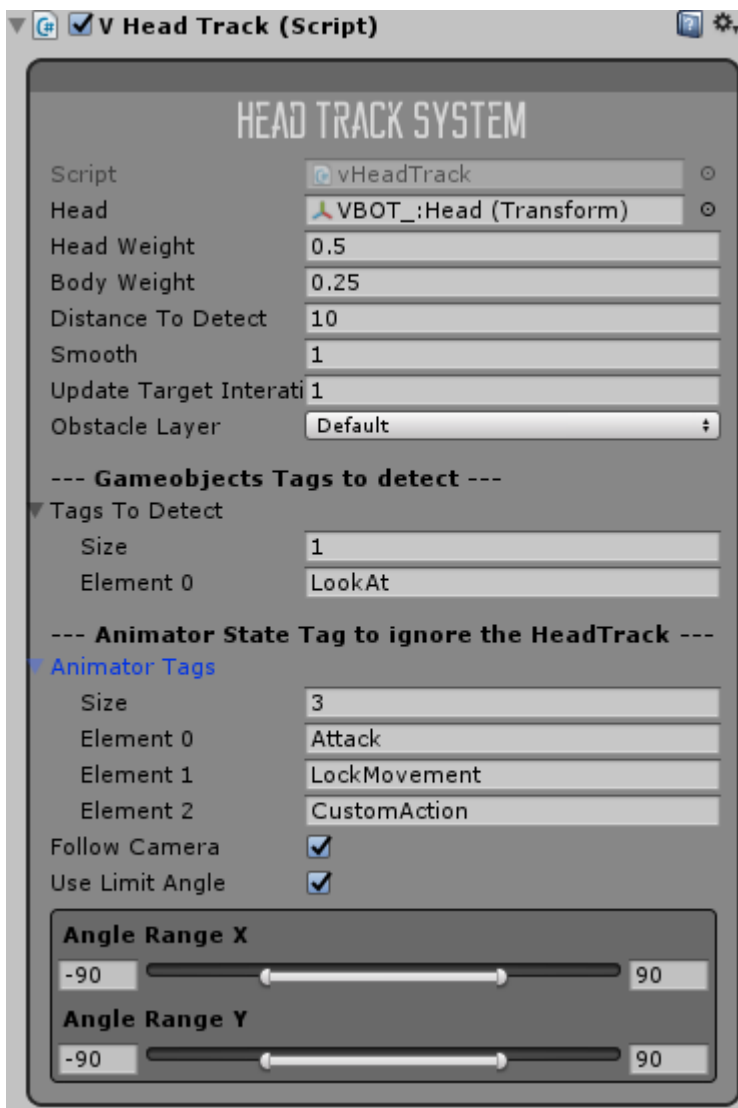
HEAD TRACK

ADD V2.0 - Now the Headtrack is a separated component and you need to add manually:

***Shooter** - automatically add's the headtrack in order to aim up/down

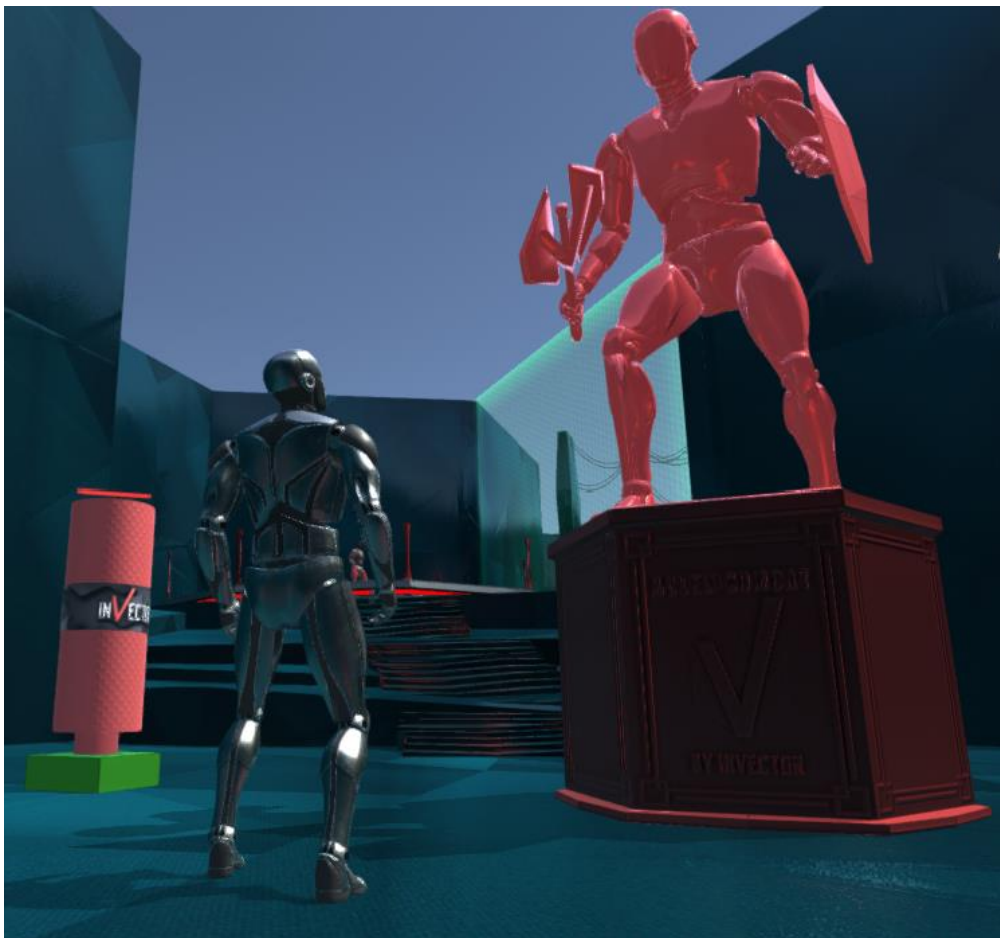
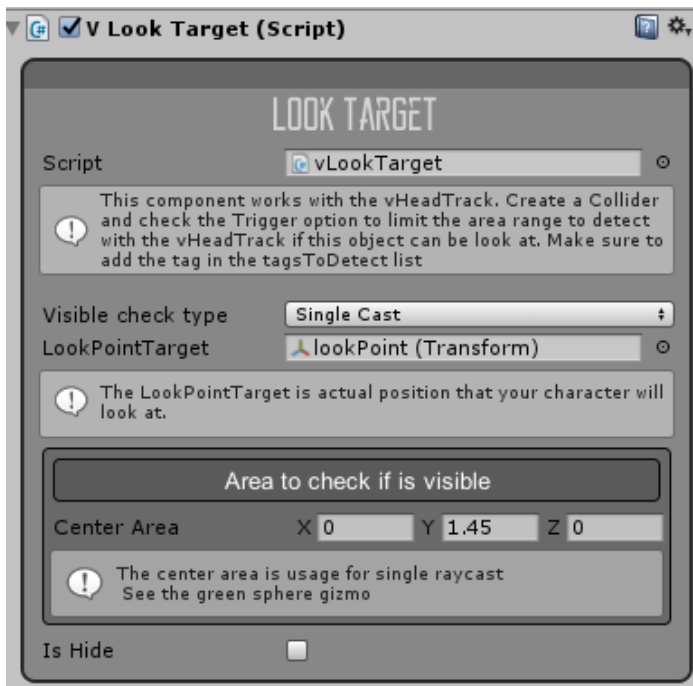


Now we have a lot of more options and we can use the LookAt feature as well.



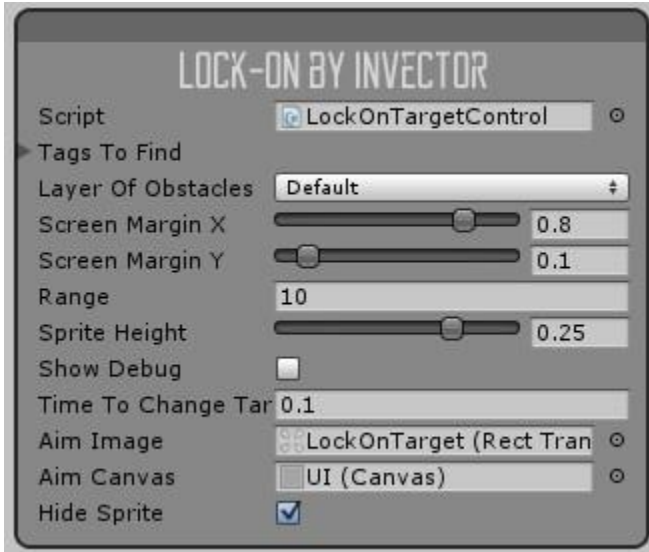
If you don't want the HeadTrack in a specific animation, you can add the Tag CustomAction into the animationState and the headtrack will turn off while this animation is playing.

To make the character look at an object, you need to add the component vLookTarget into the object, you can take a look at several examples in the DemoScenes.



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.

Shooter - You can use the Lock-On by checking the "Use Lock-On" option on the ShooterManager.