3rd Person Controller – Melee Combat v2.0

Thank you for support this asset, we develop this template because a lot of developers have good ideas for a 3rd 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 ---

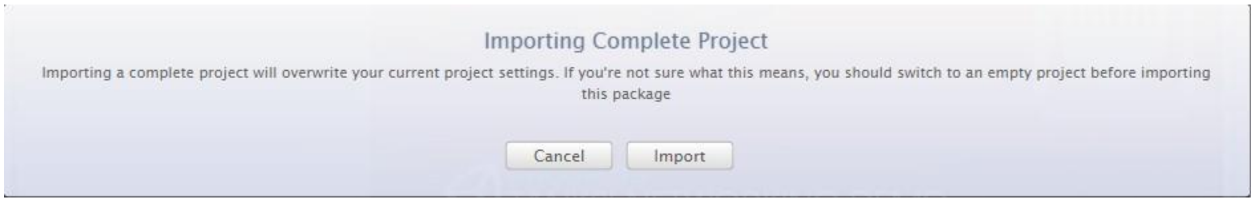
Summary

FIRST RUN....3
CREATING A NEW MELEE CHARACTER FIRST RUN....4
TOPDOWN INPUT....6
HOW IT WORKS?....7
CREATING A NEW CAMERA STATE....8
XBOX CONTROLLER SUPPORT....11
INPUTMANAGER....11
TAGS & LAYERS....12
RECOMMENDED MOBILE SETTINGS....13
HEAD TRACK....14
FOOTSTEP AUDIO SYSTEM....16
CREATING A RAGDOLL....19
HOW TO ADD NEW ANIMATIONS?....21
RAYCAST CHECKERS (V1.1)....22
CAMERA CULLING FADE (V1.1A)....23
WORKING WITH STANDARD ASSETS....25
MELEE MANAGER....26
CREATING A MELEE WEAPON....29
ITEM MANAGER....31
CREATING AN ENEMY AI....33
LOCK-ON TARGET....35
WAYPOINT SYSTEM....36
CREATING A COMPANION AI....37

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**.

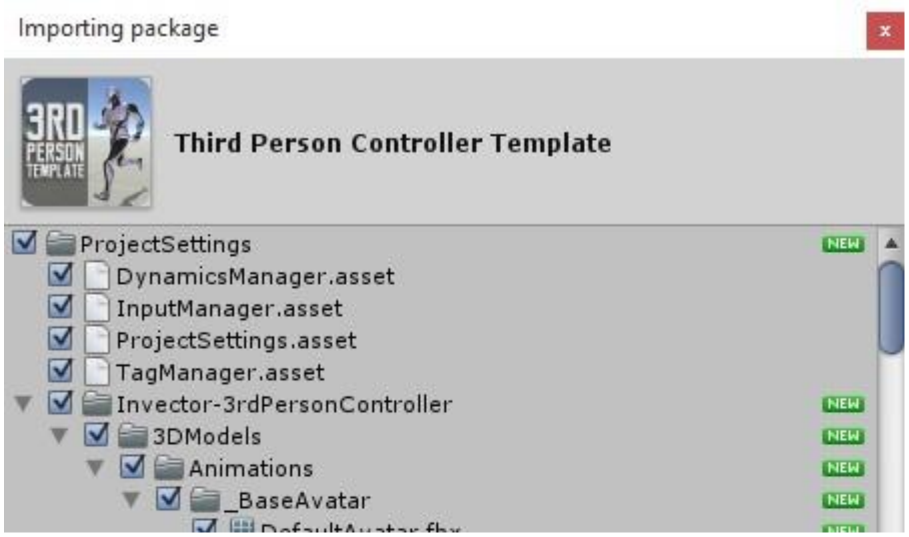


Importing Complete Project

Importing a complete project will overwrite your current project settings. If you're not sure what this means, you should switch to an empty project before importing this package

- *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).



Importing package

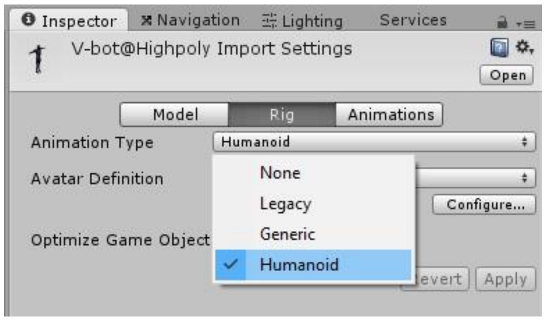
Third Person Controller Template

```
☑ ProjectSettings                              NEW
  ☑ DynamicsManager.asset
  ☑ InputManager.asset
  ☑ ProjectSettings.asset
  ☑ TagManager.asset
▼ ☑ Invector-3rdPersonController               NEW
  ▼ ☑ 3DModels                                 NEW
    ▼ ☑ Animations                             NEW
      ▼ ☑ _BaseAvatar                          NEW
```

***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. ***

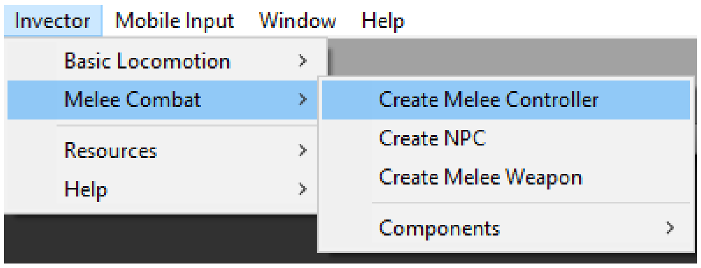
CREATING A NEW MELEE CHARACTER FIRST RUN

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

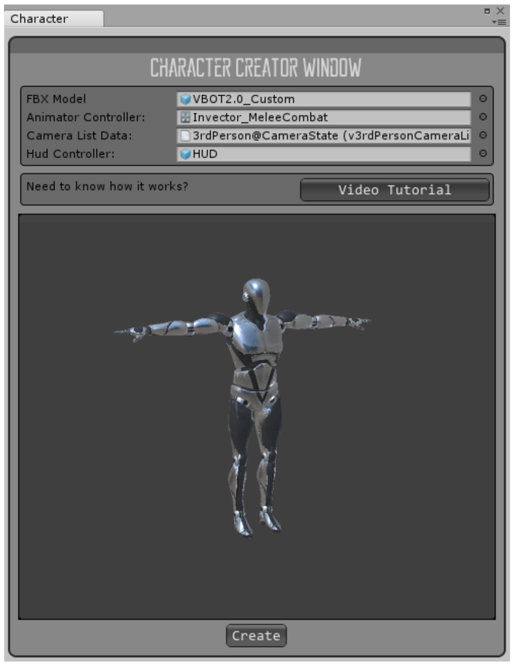


Animation Type: Humanoid
Avatar Definition
Optimize Game Object

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 3rd Person Controller".

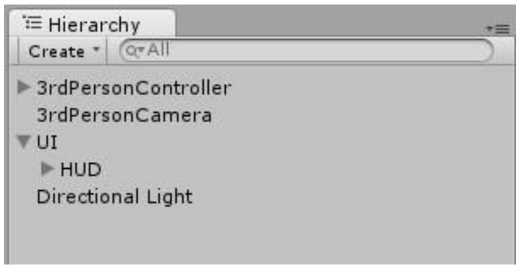


FBX Model: VBOT2.0_Custom
Animator Controller: Invector_MeleeCombat
Camera List Data: 3rdPerson@CameraState (v3rdPersonCameraLi
Hud Controller: HUD
Need to know how it works?

Ps Make sure to select the **Invector_MeleeCombat** as your Animator Controller, you can find the file at the folder: Melee Combat > Animator, or just click on the little circle icon.*

Done.

You don't have to do anything like dragging scripts, assign empty slots, etc… the **Character Creator** will take care of all the hard work automatically and set up everything for you. It will create the **3rdPersonController**, **3rdPersonCamera** and a UI Canvas with a **HUD** to display health, stamina and other information's.



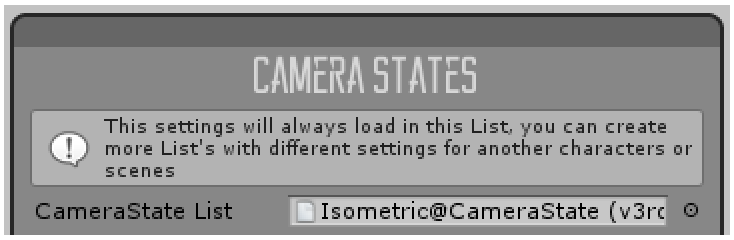
```
▶ 3rdPersonController
  3rdPersonCamera
▼ UI
    ▶ HUD
  Directional Light
```

The script will also adjust your Capsule Collider settings 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 SCALE 1** the same goes if the ragdoll **gets** weird.

Hit Play and enjoy ☺

TOPDOWN INPUT

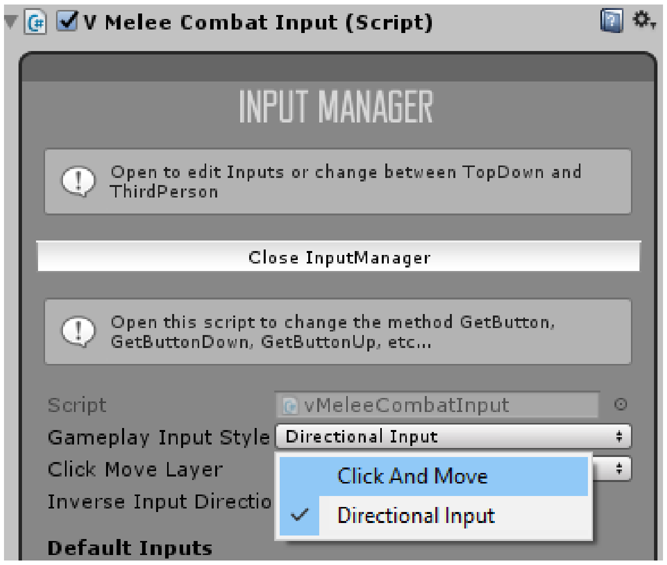
To turn your 3rd Person Controller into a **TopDown** or **Isometric** controller just go into your **3rdPersonCamera** and change the **CameraState** to **Isometric@CameraState**


CAMERA STATES

This settings will always load in this List, you can create more List's with different settings for another characters or scenes

CameraState List: Isometric@CameraState (v3rc

Select the **Player** and change your **Gameplay Input Style** to **Click and Move.**


V Melee Combat Input (Script)

INPUT MANAGER

Open to edit Inputs or change between TopDown and ThirdPerson

Close InputManager

Open this script to change the method GetButton, GetButtonDown, GetButtonUp, etc...

Script: vMeleeCombatInput

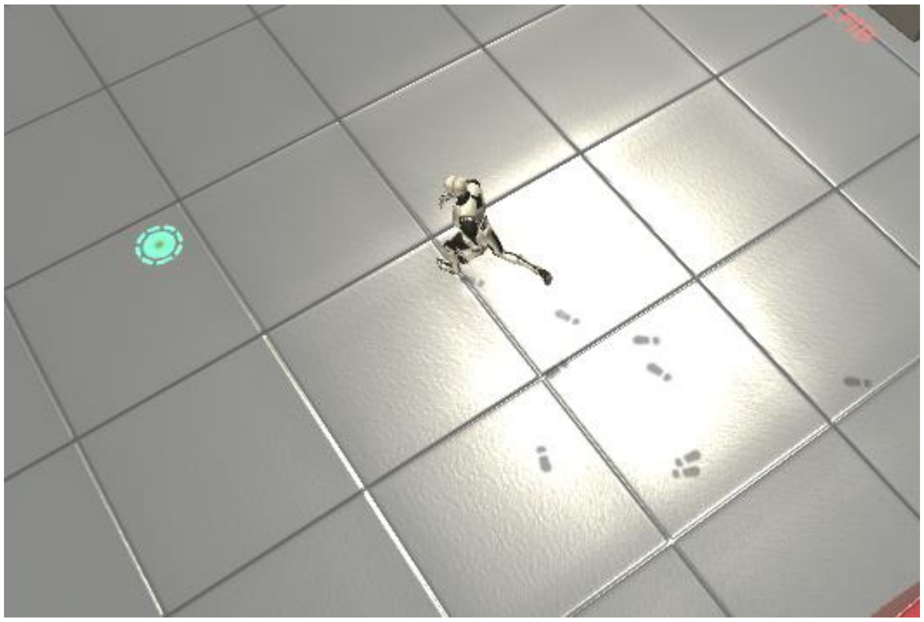
Gameplay Input Style: Directional Input

Click Move Layer

Inverse Input Directio

Default Inputs

We also include a **TopDownCursor** prefab that you just need to drag and drop into the scene, and assign the player at the TpInput field.



HOW IT WORKS?

The Controller works with **five main scripts**:

1- **vCharacter** takes care of Health/Stamina and has the method TakeDamage to apply damage.

2- **Third Person Motor** handles all the information of rigibody, colliders, verifications of ground distance, stepoffset, slope limit, etc…

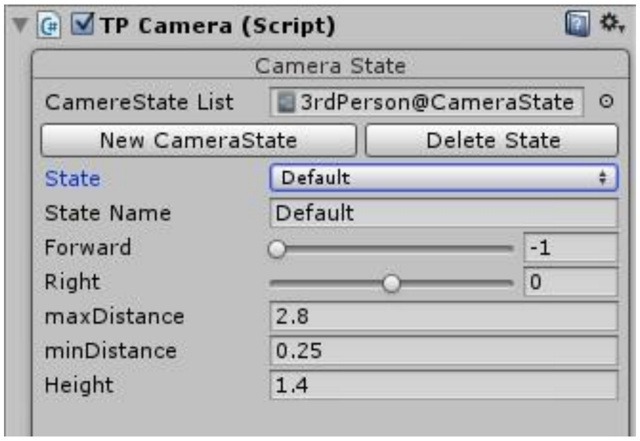
3- **Third Person Animator** is responsible to control the behavior of your animations, you can set bools, float, int and control the state of your animation.

4- **Third Person Controller** manage methods like sprint, crouch, roll, jump, etc…

5- **Third Person Input** receives all the input and call every method of the other scripts on Updates.

CREATING A NEW CAMERA STATE

On your 3rd Person Camera you can create new CameraStates to manage different values, states like “Default”, “Aiming”, “Crouch”, to set up new camera position, distance, height, etc.


TP Camera (Script)

Camera State

CamereState List	3rdPerson@CameraState
State	Default
State Name	Default
Forward	-1
Right	0
maxDistance	2.8
minDistance	0.25
Height	1.4

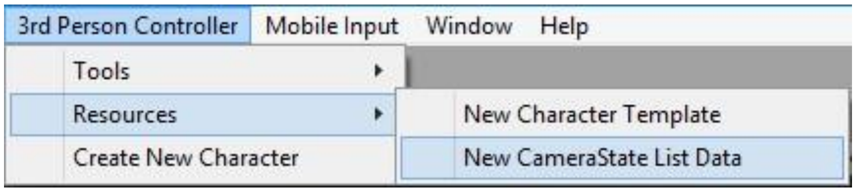
Then just change the CameraState on the method ControlCameraState() on the script TP_Motor.

Example:

if(aiming) tpCamera.ChangeState ("Aim", true);

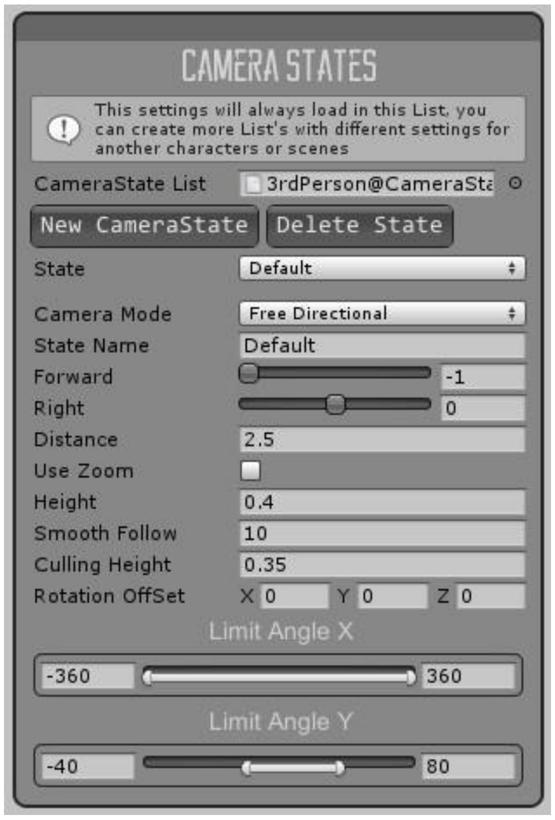
The first string value is the State Name that you created on the Camera Inspector, the second value is a bool, leave it true if you want a smooth transition to this state or false if not.

If you have more than one character and want to use different States, you can create a new **CameraState List Data** here (pic below) and assign on the CameraState List field on TP Camera Inspector.



CameraMode – Free Directional

This CameraMode offer a free directional – orbital around the character, with a lot of options to customize and make over the shoulders, or above the character, zoom (mouse only) etc…



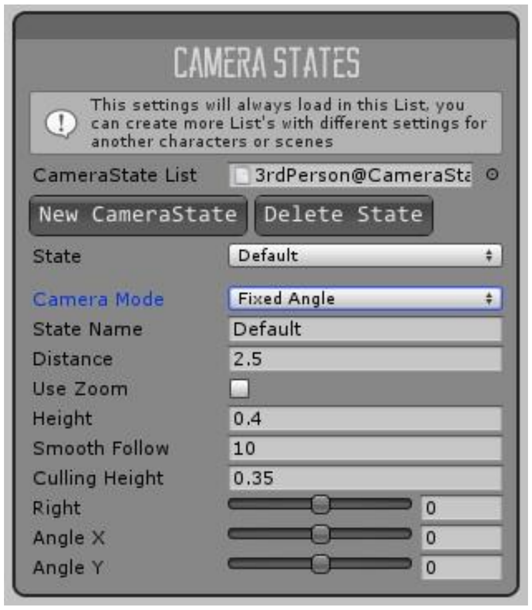
This settings will always load in this List, you can create more List's with different settings for another characters or scenes

CameraState List: 3rdPerson@CameraSta

State: Default
Camera Mode: Free Directional
State Name: Default
Forward: -1
Right: 0
Distance: 2.5
Use Zoom: ☐
Height: 0.4
Smooth Follow: 10
Culling Height: 0.35
Rotation OffSet: X 0 Y 0 Z 0
Limit Angle X: -360 360
Limit Angle Y: -40 80

CameraMode – Fixed Angle

This is a feature to use for Isometric or Topdown games, you can set up a fixed rotation for the camera and make games like Diablo or MGS 1.



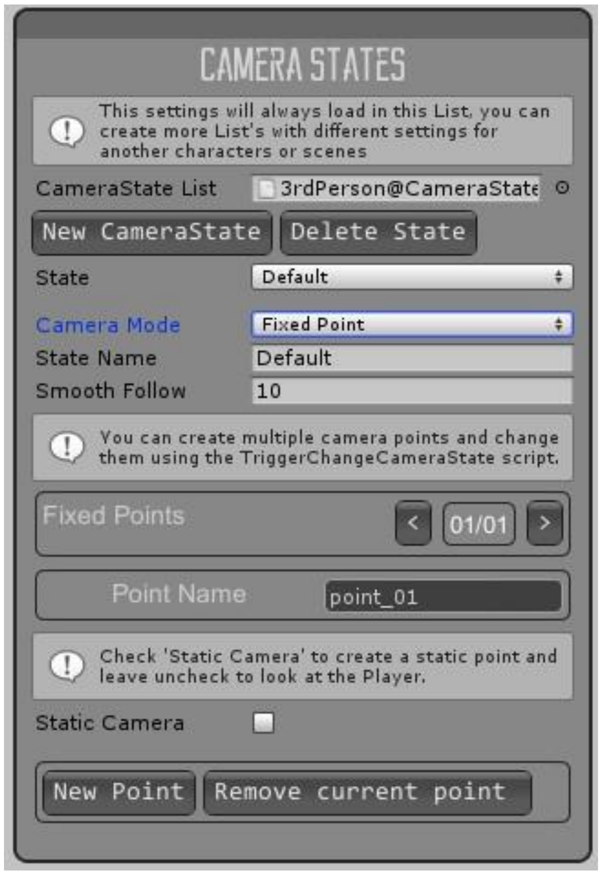
This settings will always load in this List, you can create more List's with different settings for another characters or scenes

CameraState List: 3rdPerson@CameraSta

State: Default
Camera Mode: Fixed Angle
State Name: Default
Distance: 2.5
Use Zoom: ☐
Height: 0.4
Smooth Follow: 10
Culling Height: 0.35
Right: 0
Angle X: 0
Angle Y: 0

CameraMode - Fixed Point

Fixed Points are states that you can create to use the Camera as a CCTV mode (Oldschool Resident Evil series), this state will follow the character by default or you can check Static Camera to make it fixed.


CAMERA STATES

This settings will always load in this List, you can create more List's with different settings for another characters or scenes

CameraState List: 3rdPerson@CameraState

State: Default

Camera Mode: Fixed Point

State Name: Default

Smooth Follow: 10

You can create multiple camera points and change them using the TriggerChangeCameraState script.

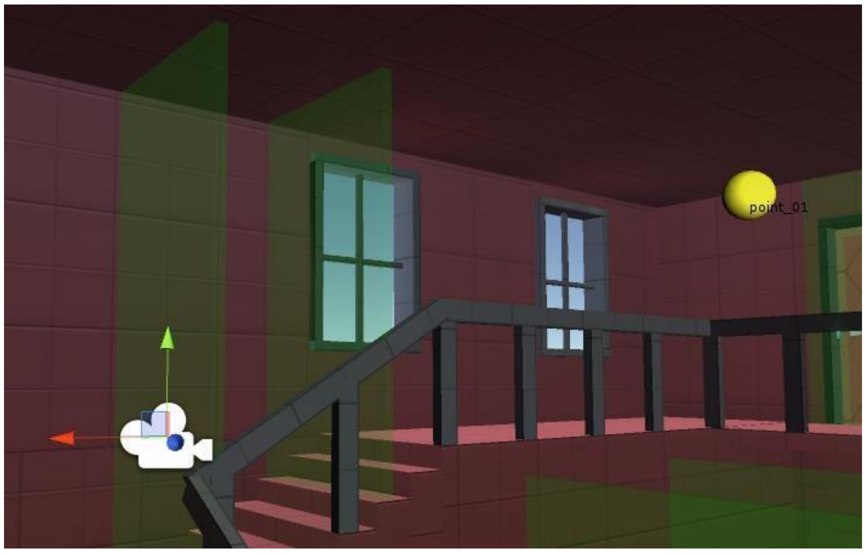
Fixed Points: 01/01

Point Name: point_01

Check 'Static Camera' to create a static point and leave uncheck to look at the Player.

Static Camera: ☐

You can also create multiple points and change with the **TriggerChangeCameraState** that has an option for smooth transition between points or not. *always leave a safe-space between triggers



XBOX CONTROLLER SUPPORT

This package works great with the **360 controller** and supports **vibration** (Windows only), make sure you compile your build according to your system. If you are using Windows 32bits make sure the build settings are set to x86 or if you are using Windows 64bits make sure the build settings are set to x86_x64.

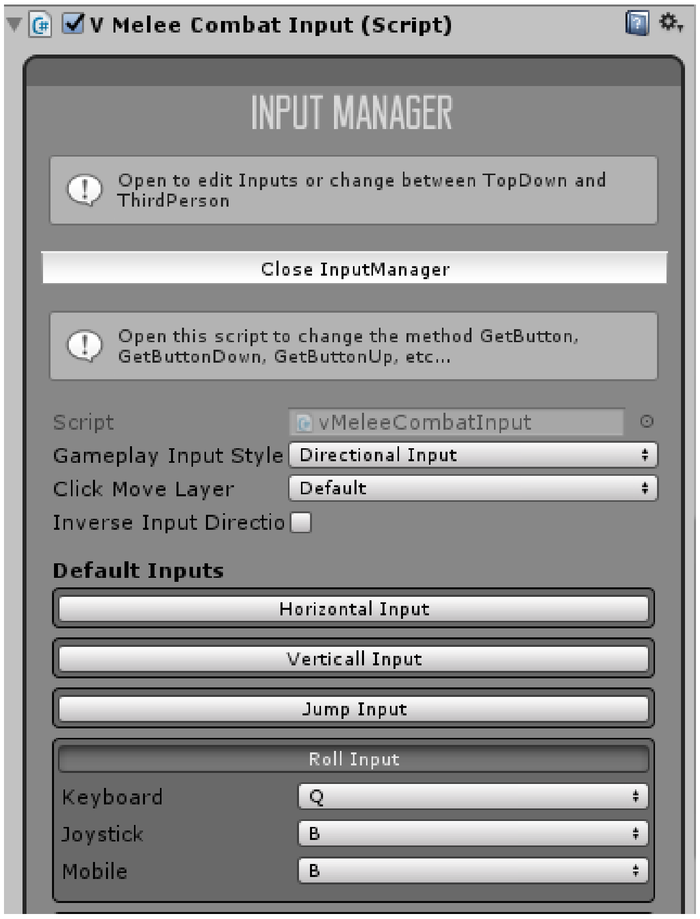
To apply the vibration, you can call the method by SendMessage to the player, for example:

target.SendMessage("GamepadVibration",0.25f,SendMessageOptions.DontRequireReceiver); The float value is the duration that you want for the vibration to last.

V1.1 add support for MFi iOS gamepad.

INPUTMANAGER

We create a InputManager so you can change the input of the character actions and movement.
If you created a Basic Locomotion it will use the vThirdPersonInput, if it's a Melee Combat character it will use the vMeleeCombatInput.



V Melee Combat Input (Script)

INPUT MANAGER

Open to edit Inputs or change between TopDown and ThirdPerson

Close InputManager

Open this script to change the method GetButton, GetButtonDown, GetButtonUp, etc...

Script	vMeleeCombatInput
Gameplay Input Style	Directional Input
Click Move Layer	Default
Inverse Input Directio	☐

Default Inputs

Horizontal Input

Verticall Input

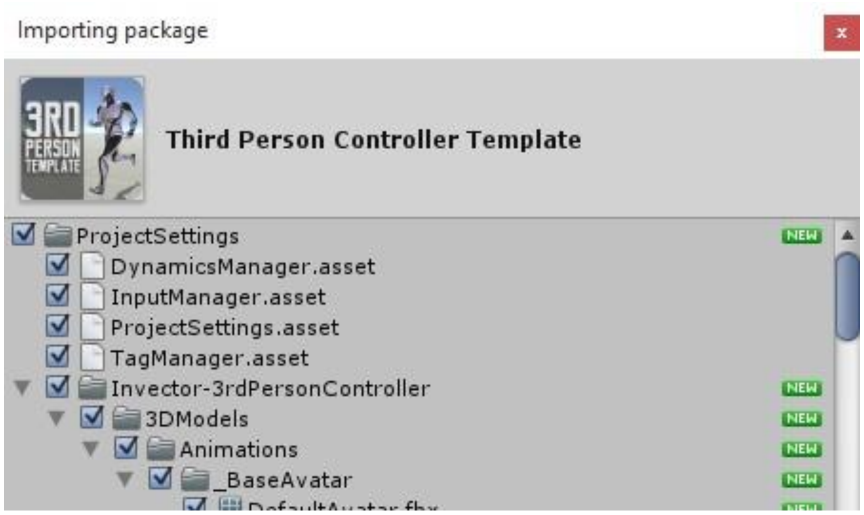
Jump Input

Roll Input

Keyboard	Q
Joystick	B
Mobile	B

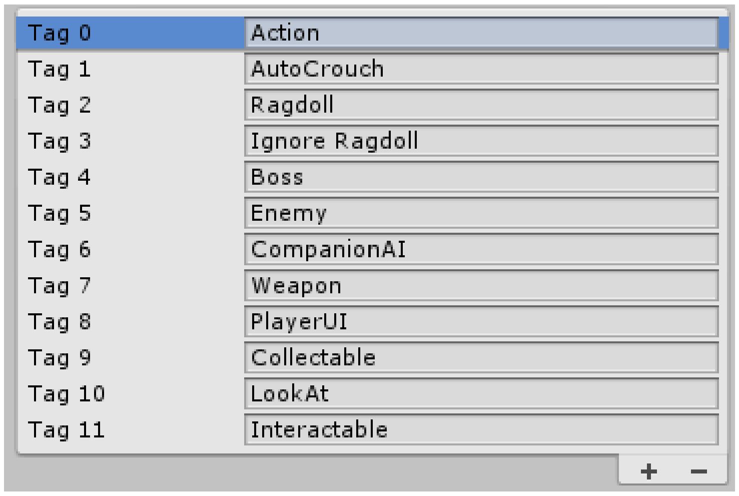
TAGS & LAYERS

V1.3 – If you will import the package into an existing project with your own tags and layers, you can uncheck the TagManager.asset and the system will automatically add our tags and layers without replace yours.



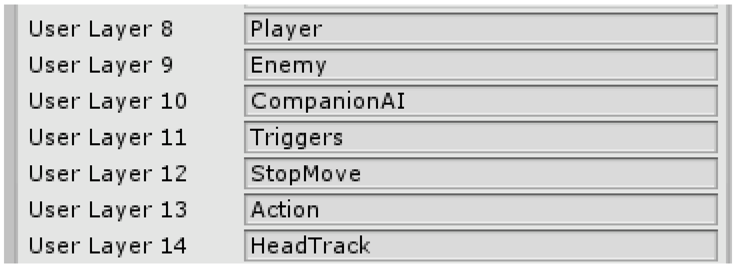
ProjectSettings
- DynamicsManager.asset
- InputManager.asset
- ProjectSettings.asset
- TagManager.asset
- Invector-3rdPersonController
  - 3DModels
    - Animations
      - _BaseAvatar

This is all the Tags we need to for the template work properly:



Tag 0	Action
Tag 1	AutoCrouch
Tag 2	Ragdoll
Tag 3	Ignore Ragdoll
Tag 4	Boss
Tag 5	Enemy
Tag 6	CompanionAI
Tag 7	Weapon
Tag 8	PlayerUI
Tag 9	Collectable
Tag 10	LookAt
Tag 11	Interactable

Layers:



User Layer 8	Player
User Layer 9	Enemy
User Layer 10	CompanionAI
User Layer 11	Triggers
User Layer 12	StopMove
User Layer 13	Action
User Layer 14	HeadTrack

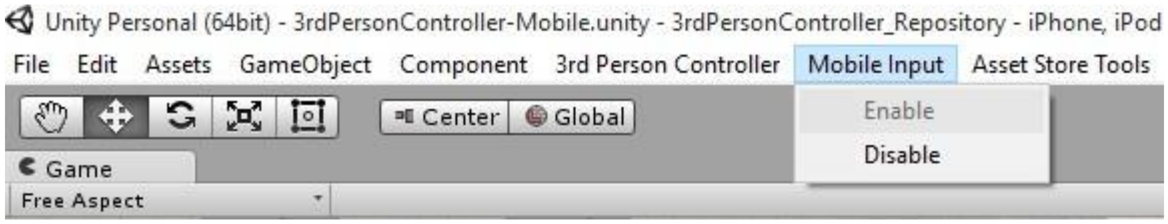
RECOMMENDED MOBILE SETTINGS

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.**

Change your platform to **Android** or **iOS** on the **Build Settings** and make sure you have the **SDK** installed.

Export the build with **ETC1** selected on Texture Compression and change your Shaders materials to **Mobile Diffuse** or **Legacy Diffuse** (this will improve performance a Lot in lower devices)

Don't forget to **Enable** the Mobile Input after change the platform, it should work right on the Editor.

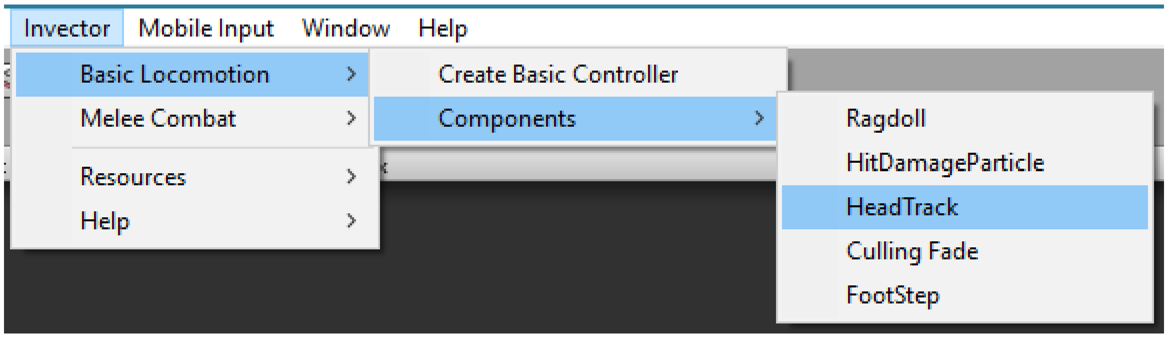


With these settings, we manage to get **stable 60fps** on several Android smartphones

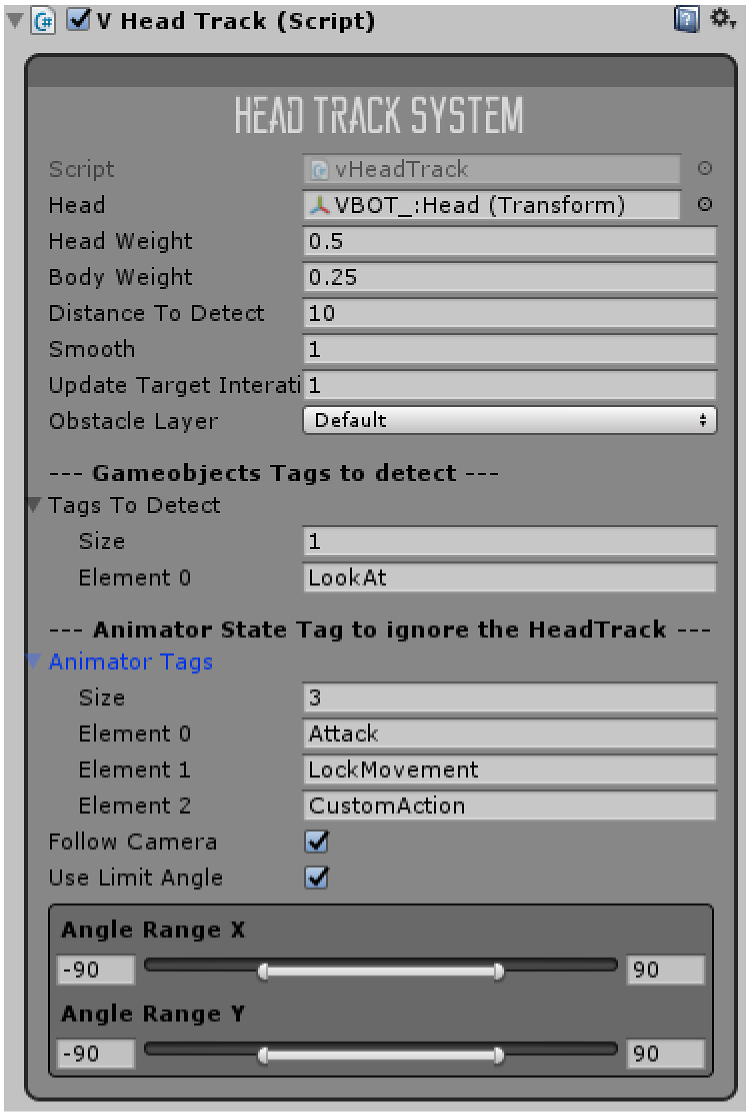
**Unity does no longer supports Tegra devices*

HEAD TRACK

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



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



Script vHeadTrack
Head VBOT_:Head (Transform)
Head Weight 0.5
Body Weight 0.25
Distance To Detect 10
Smooth 1
Update Target Interati 1
Obstacle Layer Default

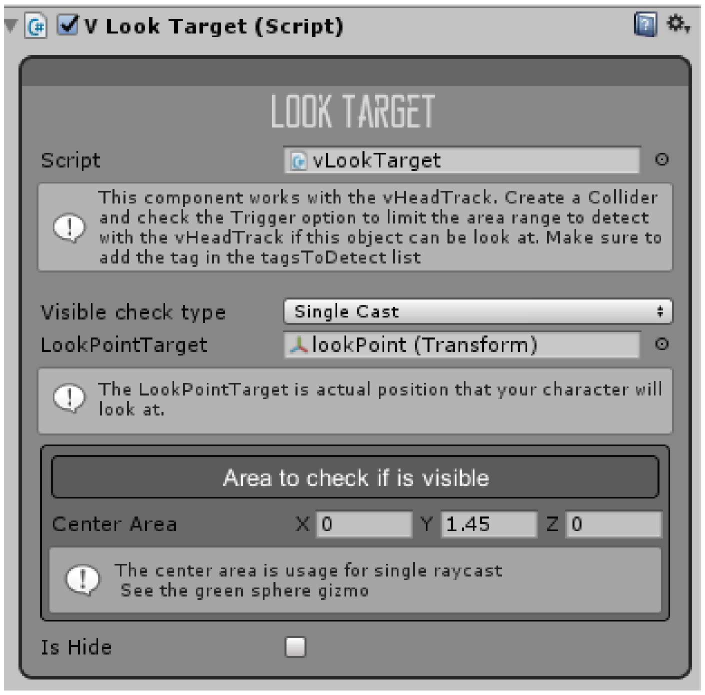
--- Gameobjects Tags to detect ---
Tags To Detect
Size 1
Element 0 LookAt

--- Animator State Tag to ignore the HeadTrack ---
Animator Tags
Size 3
Element 0 Attack
Element 1 LockMovement
Element 2 CustomAction
Follow Camera ✓
Use Limit Angle ✓

Angle Range X
-90 90
Angle Range Y
-90 90

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.


V Look Target (Script)

LOOK TARGET

Script: vLookTarget

This component works with the vHeadTrack. Create a Collider and check the Trigger option to limit the area range to detect with the vHeadTrack if this object can be look at. Make sure to add the tag in the tagsToDetect list

Visible check type: Single Cast

LookPointTarget: lookPoint (Transform)

The LookPointTarget is actual position that your character will look at.

Area to check if is visible

Center Area X 0 Y 1.45 Z 0

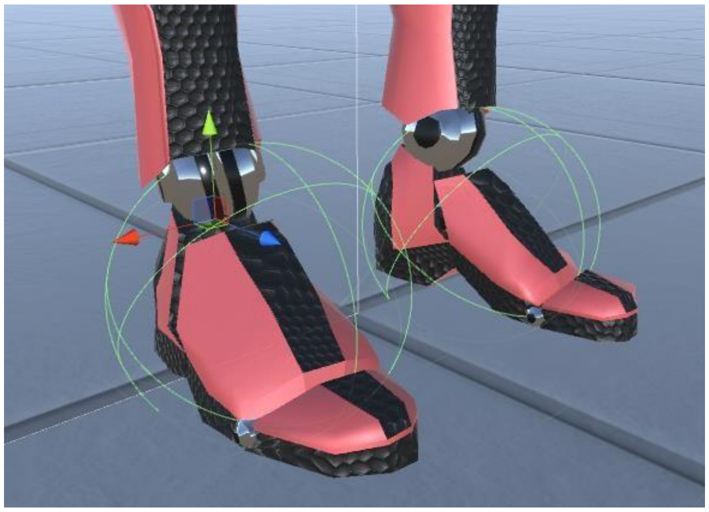
The center area is usage for single raycast
See the green sphere gizmo

Is Hide: ☐

FOOTSTEP AUDIO SYSTEM

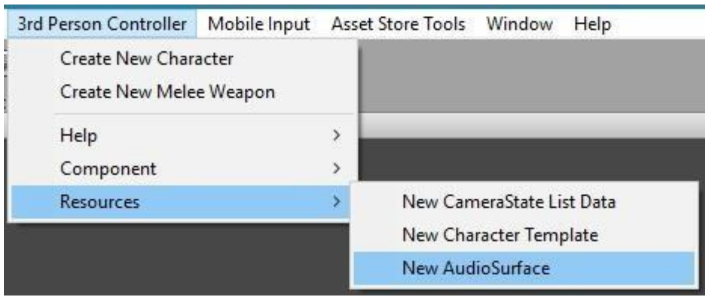
V1.2.1 New FootStep System.

When you create a new Character the FootStep component will be already attached, if you want to add a component into another Character go to the *3rdPersonController Menu* > *Component* > *FootStep*. The component will automatically create a **sphere collider** on the foot of your character, but you need to make sure that the Radius and Position of the sphere is **touching** the ground.

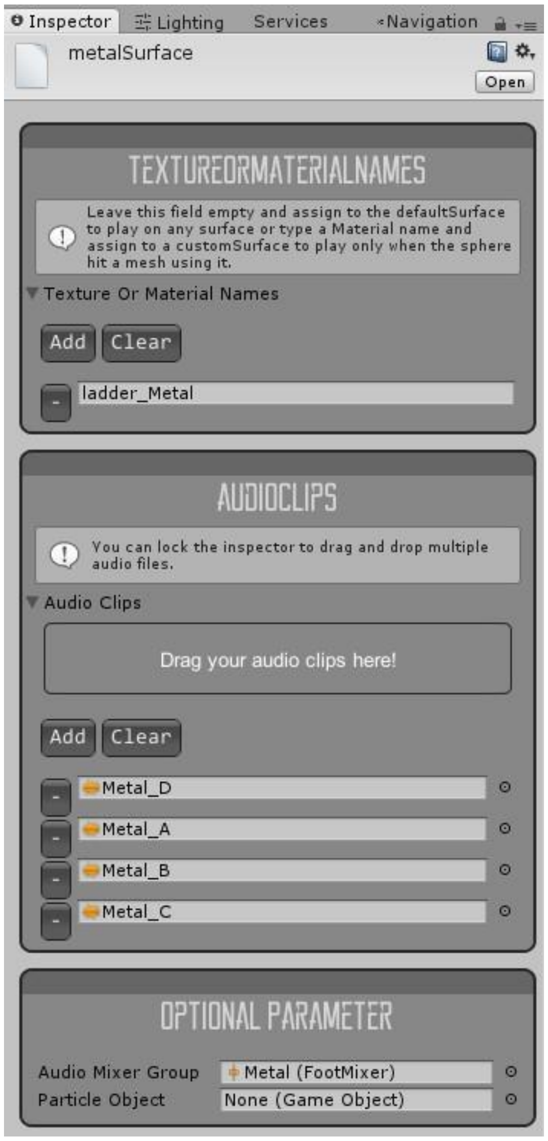


You can select the **LeftFoot** and **RightFoot** Sphere and manipulate the **Center XYZ** to position as you like, and change the **Collider Radius** too, the size of this sphere will depend on your Rig bone size. Assign the "***defaultSurface***" that comes with the package to have an example of how it works.

To create a new AudioSurface go to the 3rd Person Controller menu > Resources > New AudioSurface.



Now you can create **Custom Surfaces**, to play other audioclips based on the **material** that the sphere collider will hit. Assign the new CustomSurface to a new CustomSurface on the FootStep Inspector.



metalSurface

TEXTUREORMATERIALNAMES

Leave this field empty and assign to the defaultSurface to play on any surface or type a Material name and assign to a customSurface to play only when the sphere hit a mesh using it.

Texture Or Material Names

Add Clear

ladder_Metal

AUDIOCLIPS

You can lock the inspector to drag and drop multiple audio files.

Audio Clips

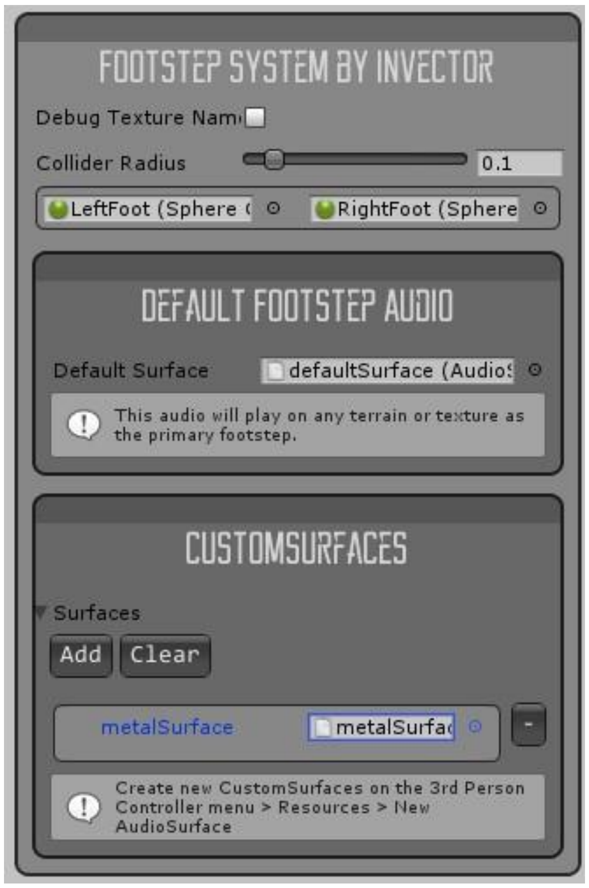
Drag your audio clips here!

Add Clear

Metal_D
Metal_A
Metal_B
Metal_C

OPTIONAL PARAMETER

Audio Mixer Group: Metal (FootMixer)
Particle Object: None (Game Object)



FOOTSTEP SYSTEM BY INVECTOR

Debug Texture Nam
Collider Radius: 0.1
LeftFoot (Sphere (RightFoot (Sphere

DEFAULT FOOTSTEP AUDIO

Default Surface: defaultSurface (Audio:

This audio will play on any terrain or texture as the primary footstep.

CUSTOMSURFACES

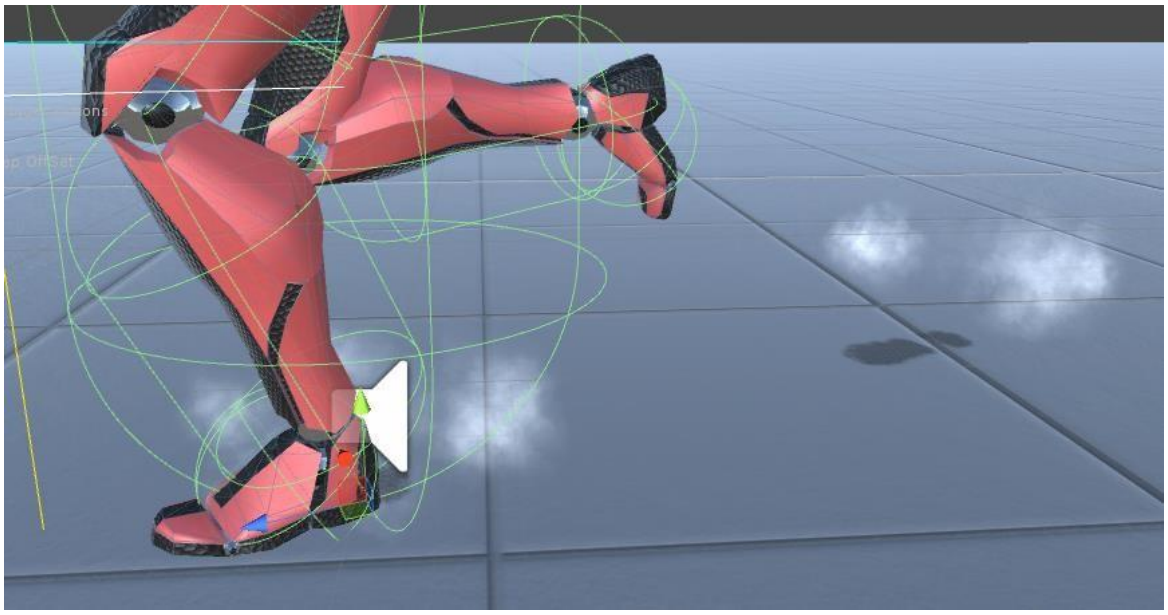
Surfaces

Add Clear

metalSurface metalSurfa

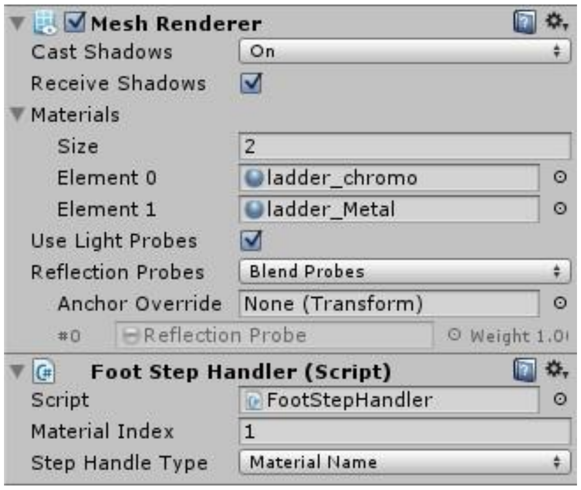
Create new CustomSurfaces on the 3rd Person Controller menu > Resources > New AudioSurface

You can assign a **AudioMixer** for better control the surfaces, and you can instantiate a **Particle** as well, see the example on the DefaultSurface call ‘smoke’ that also uses a **StepMark** sprite call SimpleStepMark.



V1.1 Using the FootStep system in objects with multiple Materials

If your gameobject has multiple materials and you need to play a specific material, you can use the FootStepHandler script and set the correct Material Index of your object. (*See example on the Ladder prefab)



Mesh Renderer

Cast Shadows	On
Receive Shadows	✓
Materials	
Size	2
Element 0	ladder_chromo
Element 1	ladder_Metal
Use Light Probes	✓
Reflection Probes	Blend Probes
Anchor Override	None (Transform)
#0	Reflection Probe Weight 1.0

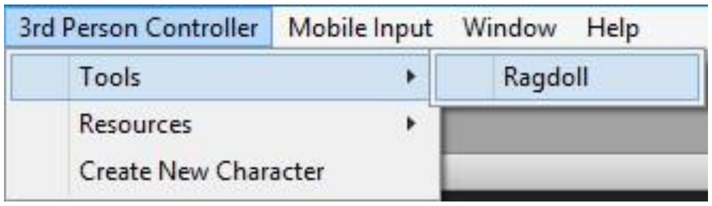
Foot Step Handler (Script)

Script	FootStepHandler
Material Index	1
Step Handle Type	Material Name

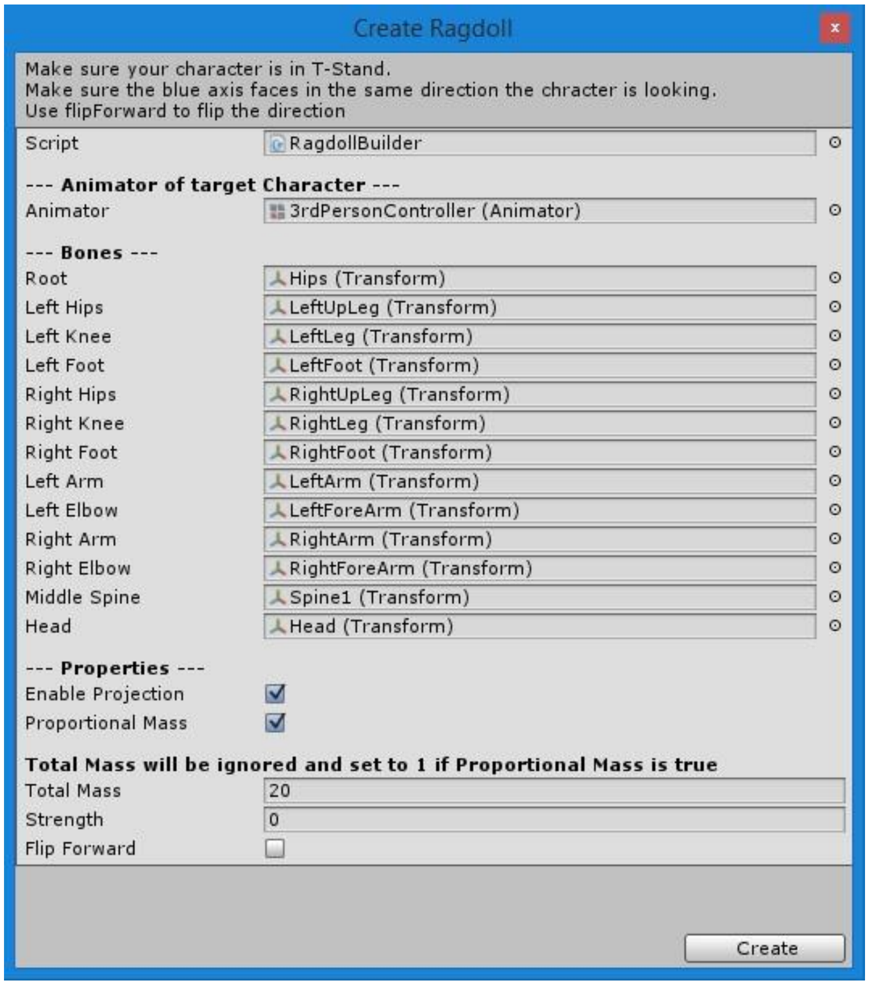
CREATING A RAGDOLL

Ps* Make sure to add the Ragdoll First and then equip the character with the MeleeManager and Weapons!

Creating a Ragdoll is just easy as creating your Character, just go to the tab "3rd Person Controller" > "Tools" > "Ragdoll".



If you have your character selected on the Hierarchy, all the fields will **autofill**, if not, just click on your character and it will autofill for you, this template was design to **save time**, so you don't have to waste your time dragging and drop every bone, instead just hit the "Create" button and it's ready to go.

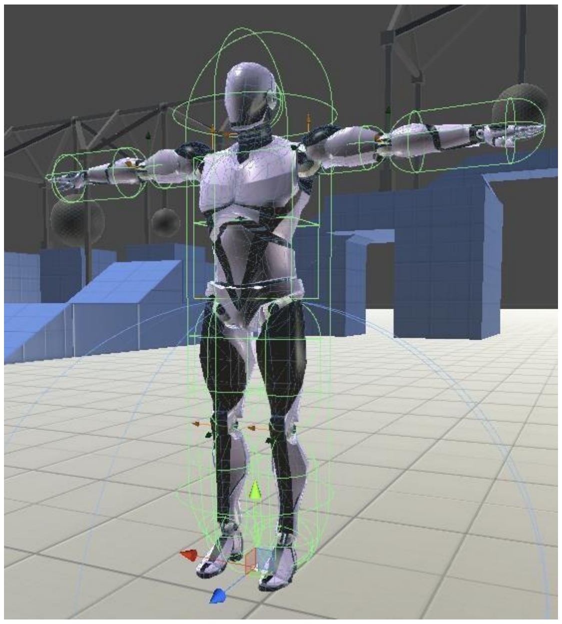


Create Ragdoll

Make sure your character is in T-Stand.
Make sure the blue axis faces in the same direction the chracter is looking.
Use flipForward to flip the direction

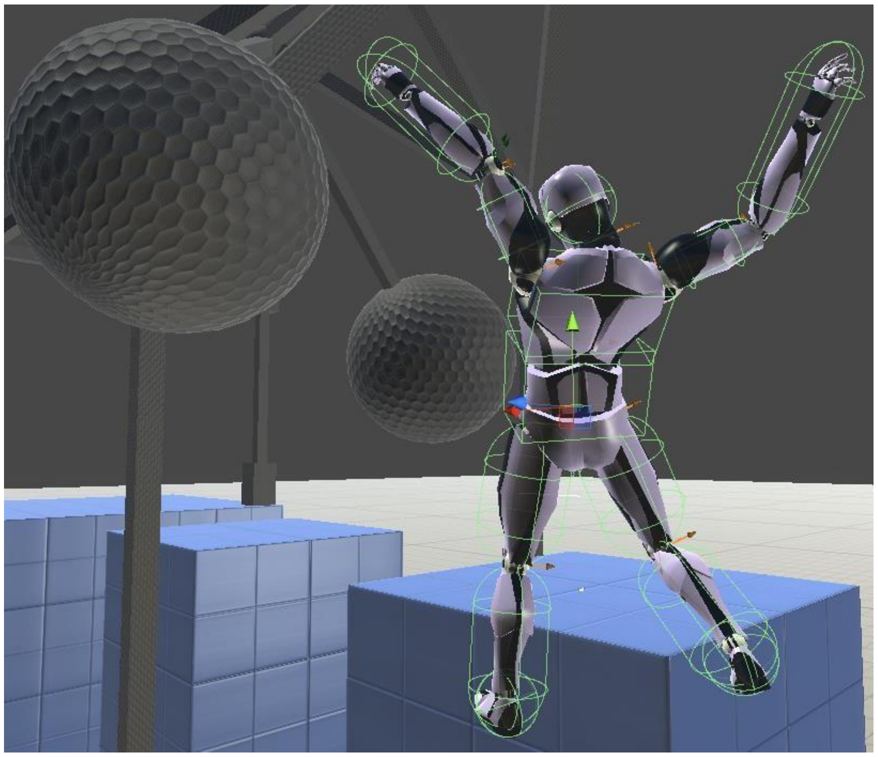
Script	RagdollBuilder
--- Animator of target Character ---	
Animator	3rdPersonController (Animator)
--- Bones ---	
Root	Hips (Transform)
Left Hips	LeftUpLeg (Transform)
Left Knee	LeftLeg (Transform)
Left Foot	LeftFoot (Transform)
Right Hips	RightUpLeg (Transform)
Right Knee	RightLeg (Transform)
Right Foot	RightFoot (Transform)
Left Arm	LeftArm (Transform)
Left Elbow	LeftForeArm (Transform)
Right Arm	RightArm (Transform)
Right Elbow	RightForeArm (Transform)
Middle Spine	Spine1 (Transform)
Head	Head (Transform)
--- Properties ---	
Enable Projection	✓
Proportional Mass	✓
Total Mass will be ignored and set to 1 if Proportional Mass is true	
Total Mass	20
Strength	0
Flip Forward	☐

We strongly recommend keep the **Enable Projection** and the **Proportional Mass** enabled, and do not forget to use **Scale Factor 1** on your **fbx** Model. This you provide better behavior of your ragdoll.

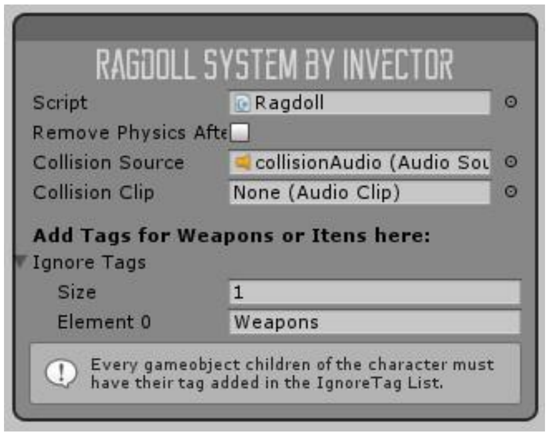


To enable the ragdoll, you can use the Script **ObjectDamage** or just call this line on the **OnCollisionEnter** method.

hit.transform.root.SendMessage ("ActivateRagdoll", SendMessageOptions.DontRequireReceiver);



v1.1b – Add "***Ignored Tags***" you can add a list of tags for objects that are children of the Player to keep the rotation correctly, otherwise it will mess up the rotation when the Ragdoll are on.



Script: Ragdoll
Remove Physics Afte: ☐
Collision Source: collisionAudio (Audio Sou
Collision Clip: None (Audio Clip)

Add Tags for Weapons or Itens here:
Ignore Tags
Size: 1
Element 0: Weapons

Every gameobject children of the character must have their tag added in the IgnoreTag List.

HOW TO ADD NEW ANIMATIONS?

The process is:

Set your animation clip as Humanoid and retarget to your T-Pose character

If it is an action like open a door, put the animation on the Action State of the Animator.

At the Motor script, create a variable like a bool to control the animation

At the AnimationControl script, tell what variable controls what animation

At the Controller script, run the method to trigger the animation

Here is a Video Tutorial showing the process to apply a Jump Animation:

[How to add new animations]

[Replacing Animations]

This is just an example, but of course, you may have to prepare the script to what your new actions are going to do, just as we prepare for the jump animation example.

RAYCAST CHECKERS (V1.1)

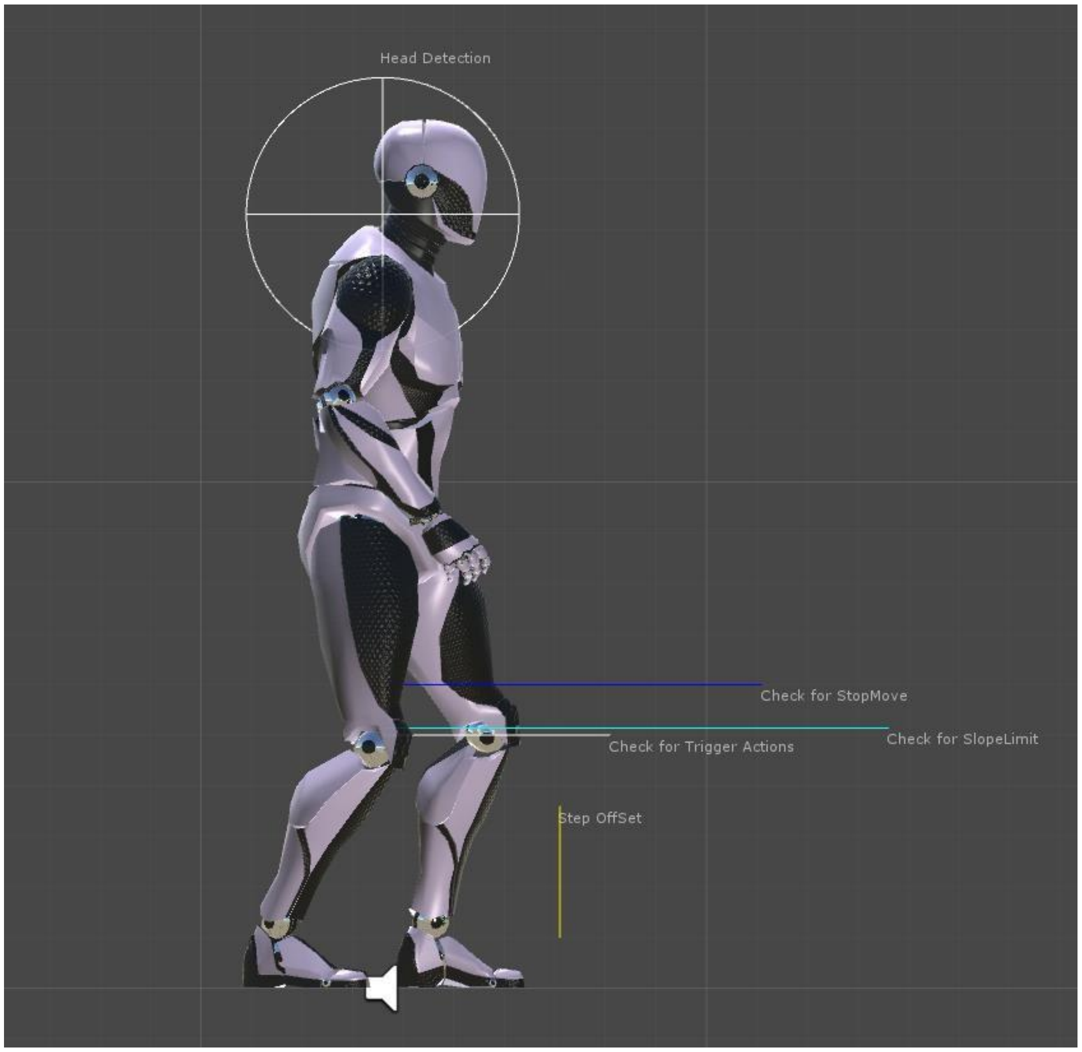
Head Detection is a SphereCast that will detect if has an object above, and keep the character crouched, use the same layer as the Ground Layer (Default). Just adjust to sync with the height of your capsule collider.

StopMove is a Raycast that detect any object with the layer (Default, StopMove) to prevent the character to walk in place, you can use a StopMove in an invisible wall for example, and the camera will not clip, because the culling layer is set to "Default".

SlopeLimit will prevent the character of walking in absurd angle heights, float customizable on the Player Inspector.

Trigger Actions is the raycast that check for objects with the component TriggerAction, you can trigger a specific action based on the tag, display information and pass a specific transform position and rotation using matchTarget.

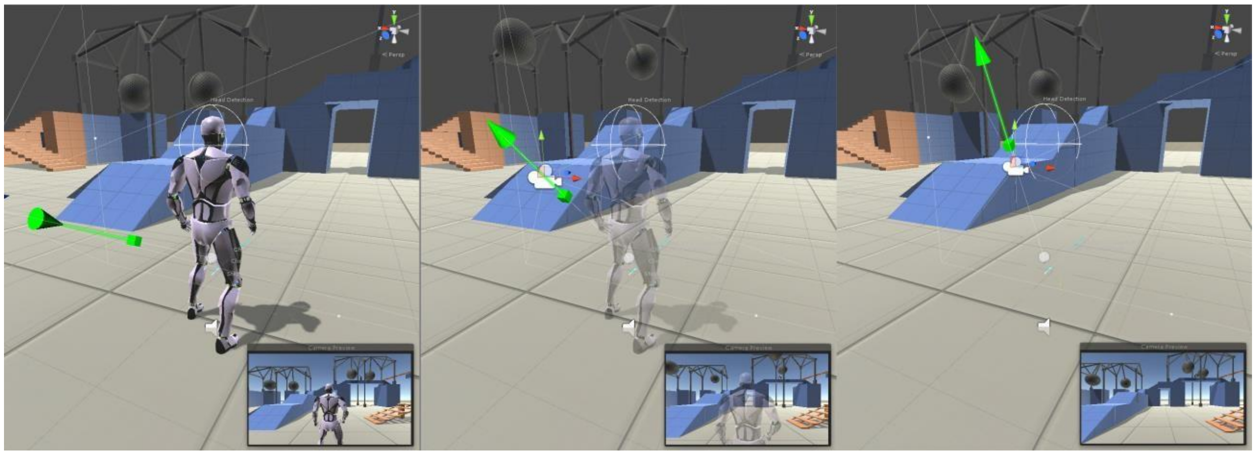
StepOffset is to help the character walk in custom height steps, adjust the values on the Player Inspector.



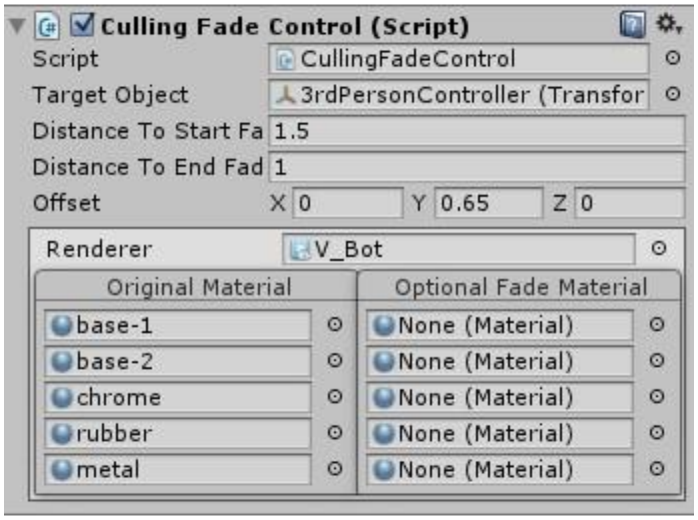
CAMERA CULLING FADE (V1.1A)

We add a Culling Fade script for the camera to avoid see through the character’s mesh, you can set up the distance to start fading and an offset.

Example:



Our Culling Fade will set up automatically for the default Stardard Shader of Unity’s, but you also can use custom shaders, just make an additional copy with the fade material and assign in the “Optional Fade Material” field.

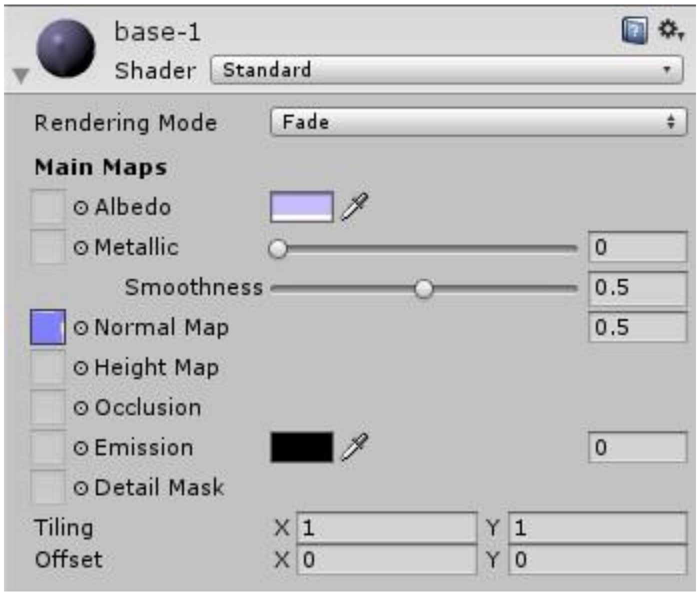


Culling Fade Control (Script)

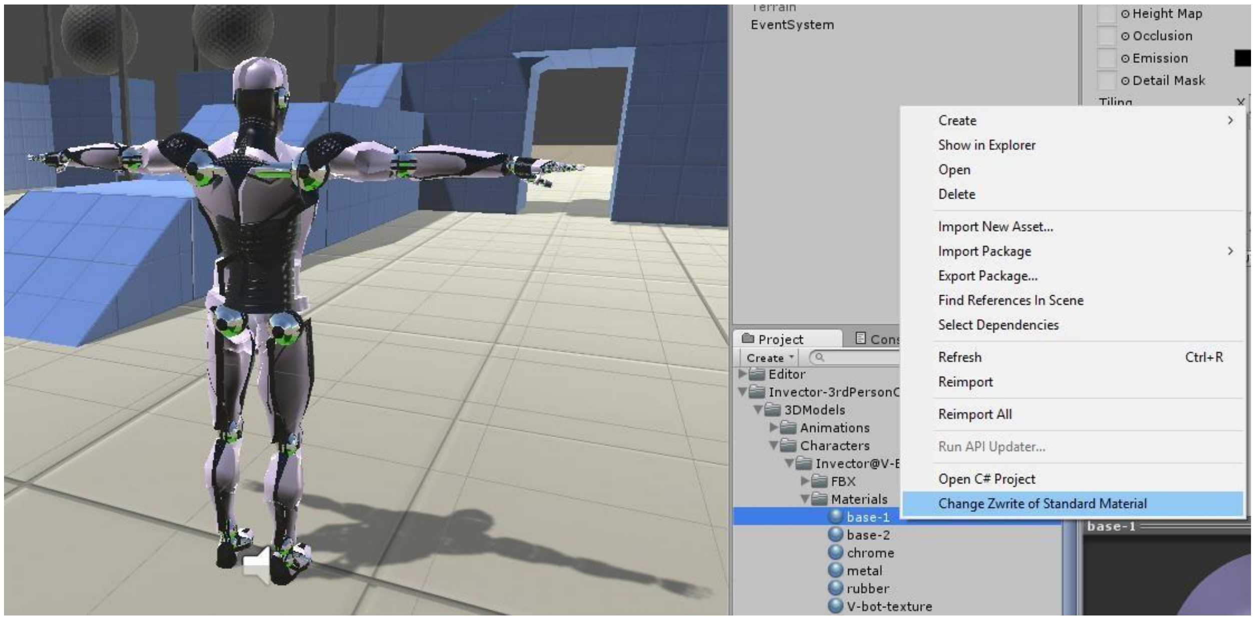
Script	CullingFadeControl
Target Object	3rdPersonController (Transfor
Distance To Start Fa	1.5
Distance To End Fad	1
Offset	X 0 Y 0.65 Z 0
Renderer	V_Bot

Original Material	Optional Fade Material
base-1	None (Material)
base-2	None (Material)
chrome	None (Material)
rubber	None (Material)
metal	None (Material)

If you are using the Standard Shader, just select the Rendering Mode “Fade” on the Material.


```
base-1
Shader      Standard
Rendering Mode    Fade
Main Maps
  Albedo
  Metallic                    0
      Smoothness              0.5
  Normal Map                  0.5
  Height Map
  Occlusion
  Emission                    0
  Detail Mask
Tiling     X 1      Y 1
Offset     X 0      Y 0
```

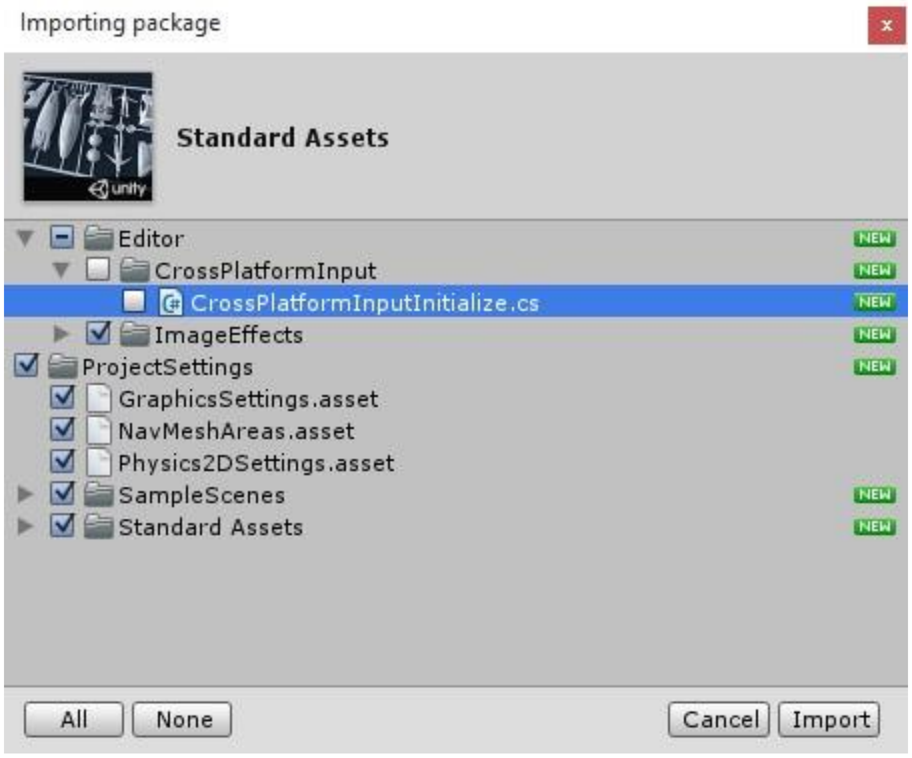
The character will look like this (picture below) but you can fix by right clicking at the material and “Change Zwrite of Standard Material”.



UPDATE V1.1B – now the script will be attached into the Controller just like the Ragdoll and the Footstep, It’s a modular feature.

WORKING WITH STANDARD ASSETS

Our template already comes with the **CrossPlataformInput** and the **ImageEffect AntiAliasing** imported, so if you want to import the **Standard Assets** package into the project, just make sure to ***UNCHECK*** the following item:



Importing package

Standard Assets

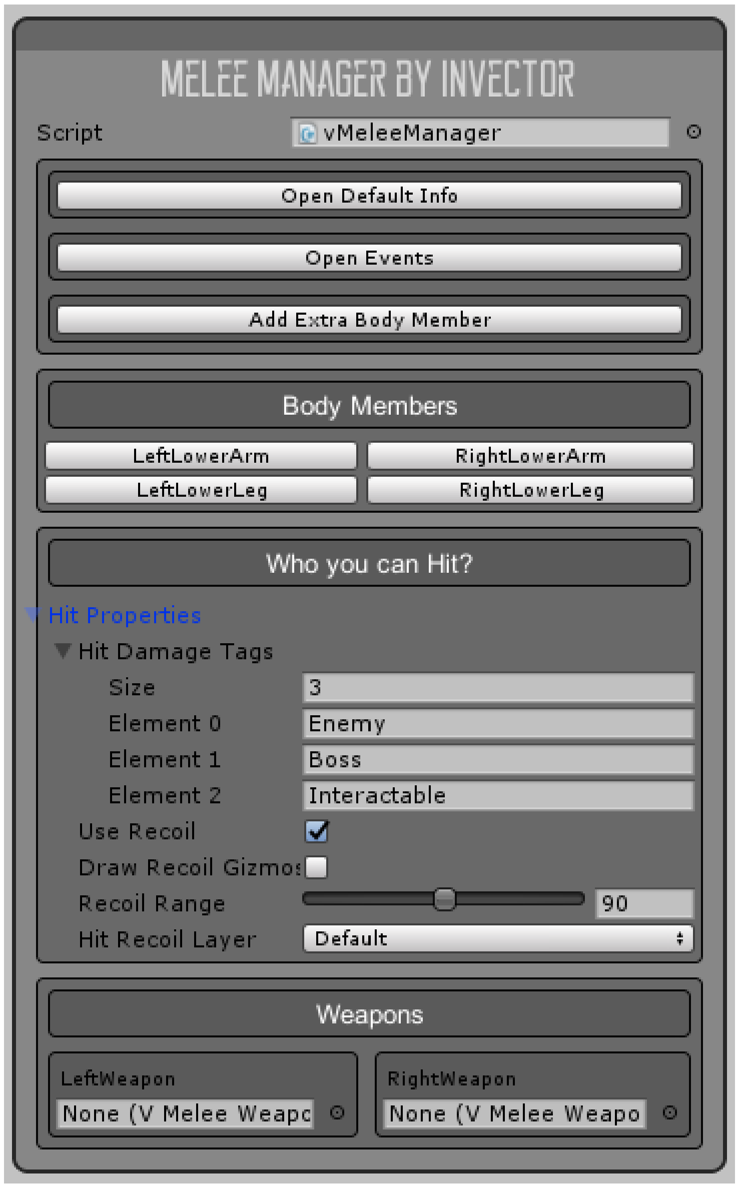
- [–] Editor NEW
  - [ ] CrossPlatformInput NEW
    - [ ] CrossPlatformInputInitialize.cs NEW
  - [✓] ImageEffects NEW
- [✓] ProjectSettings NEW
  - [✓] GraphicsSettings.asset
  - [✓] NavMeshAreas.asset
  - [✓] Physics2DSettings.asset
  - [✓] SampleScenes NEW
  - [✓] Standard Assets NEW

If you imported by mistake and are getting some errors, try deleting the folder **Editor** and the folder **Standard Assets** and ***reimport*** the Standard Assets package again, the errors will be gone. [Video Tutorial]

MELEE MANAGER

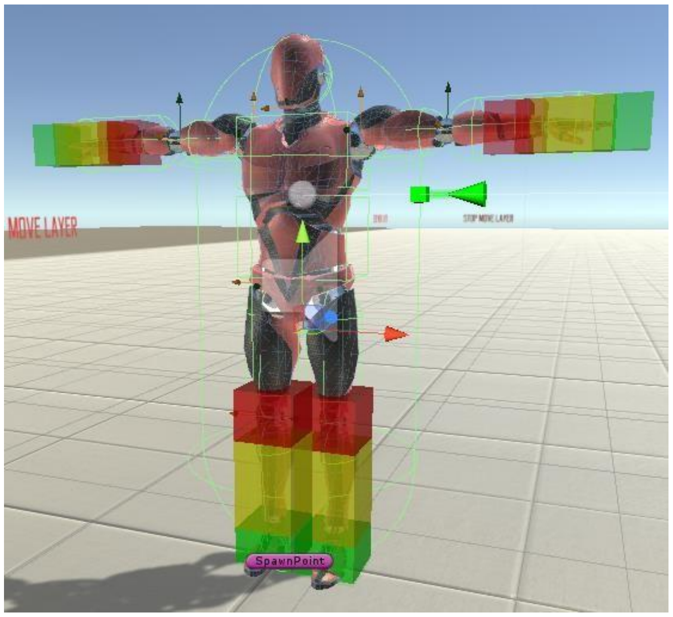
V2.0 - You can add a **Melee Manager** Component by opening the Invector tab > Melee Combat > Component

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)



Script: vMeleeManager
Size: 3
Element 0: Enemy
Element 1: Boss
Element 2: Interactable
Use Recoil: ✓
Draw Recoil Gizmos: ☐
Recoil Range: 90
Hit Recoil Layer: Default
LeftWeapon: None (V Melee Weapc
RightWeapon: None (V Melee Weapo

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.

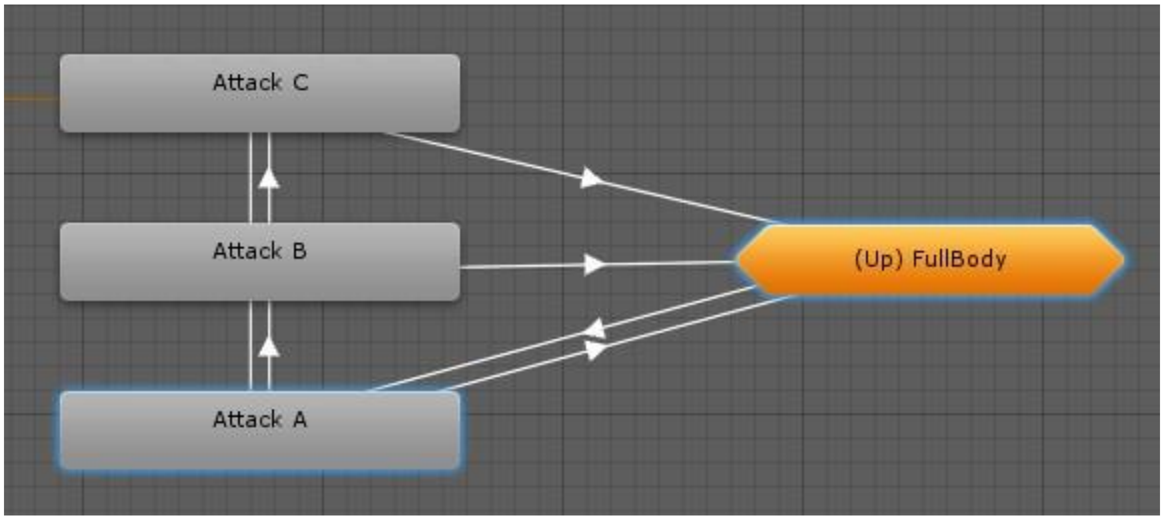


The animations for the hand to hand combat can be set up in the **Unarmed** state machine, trigger by the **ATK_ID 0** and the defense **DEF_ID 1** on the UpperBody Layer**, Default Defense.**

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.

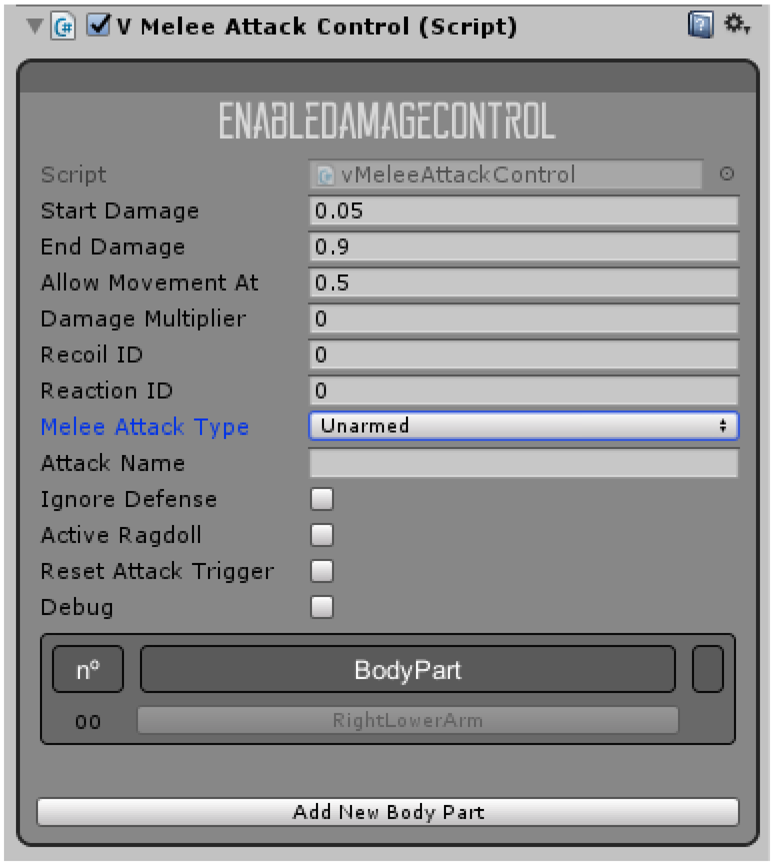
You can use **UpperBody** to attack as well, this way you can move the character and attack at the same time.

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


V Melee Attack Control (Script)

ENABLEDAMAGECONTROL

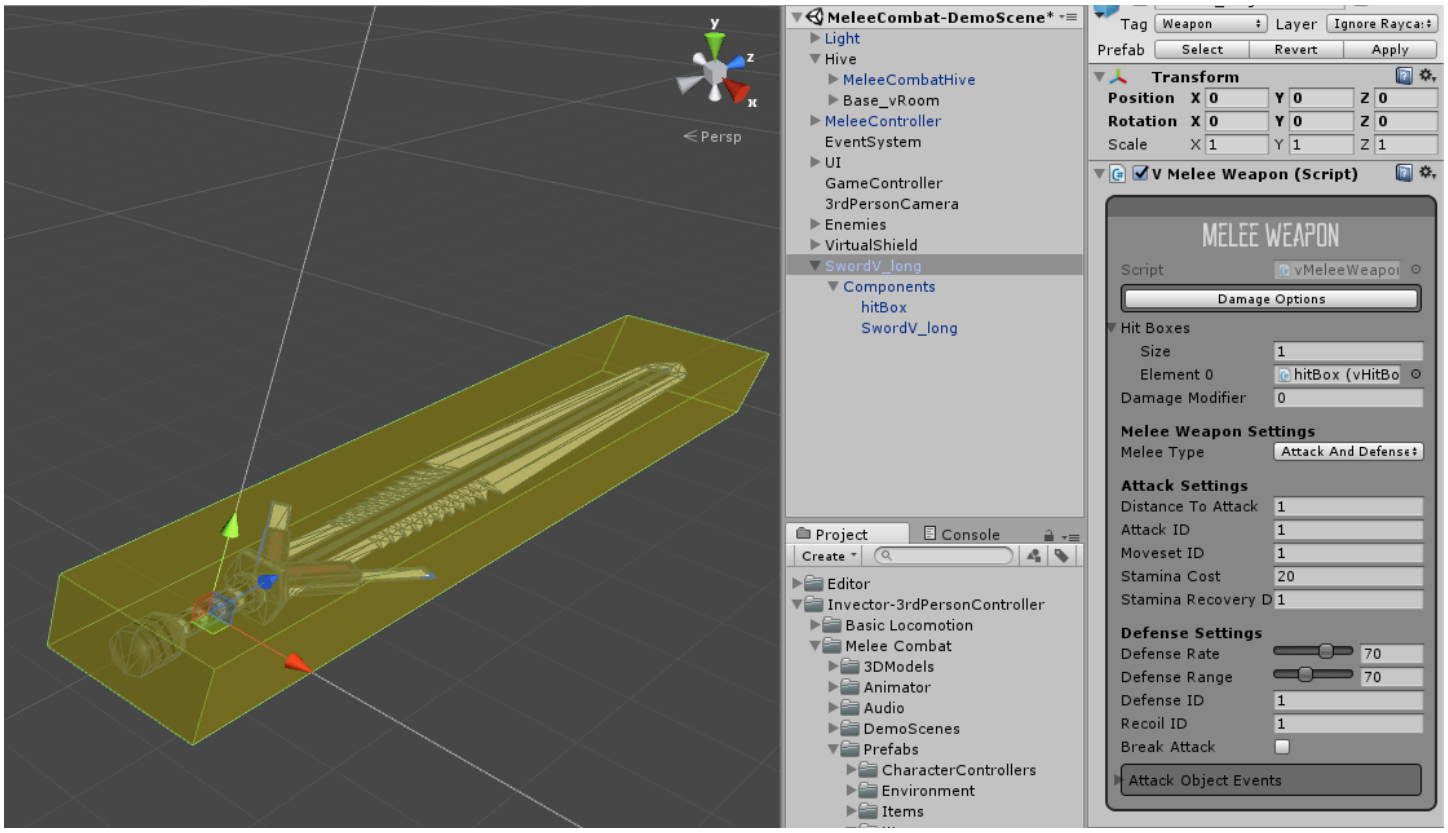
Script	vMeleeAttackControl
Start Damage	0.05
End Damage	0.9
Allow Movement At	0.5
Damage Multiplier	0
Recoil ID	0
Reaction ID	0
Melee Attack Type	Unarmed
Attack Name	
Ignore Defense	☐
Active Ragdoll	☐
Reset Attack Trigger	☐
Debug	☐

nº	BodyPart
00	RightLowerArm

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.



Tag: Weapon — Layer: Ignore Raycast

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

V Melee Weapon (Script)

MELEE WEAPON
Script: vMeleeWeapon

Hit Boxes
Size: 1
Element 0: hitBox (vHitBox)
Damage Modifier: 0

Melee Weapon Settings
Melee Type: Attack And Defense

Attack Settings
Distance To Attack: 1
Attack ID: 1
Moveset ID: 1
Stamina Cost: 20
Stamina Recovery D: 1

Defense Settings
Defense Rate: 70
Defense Range: 70
Defense ID: 1
Recoil ID: 1
Break Attack: ☐

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.

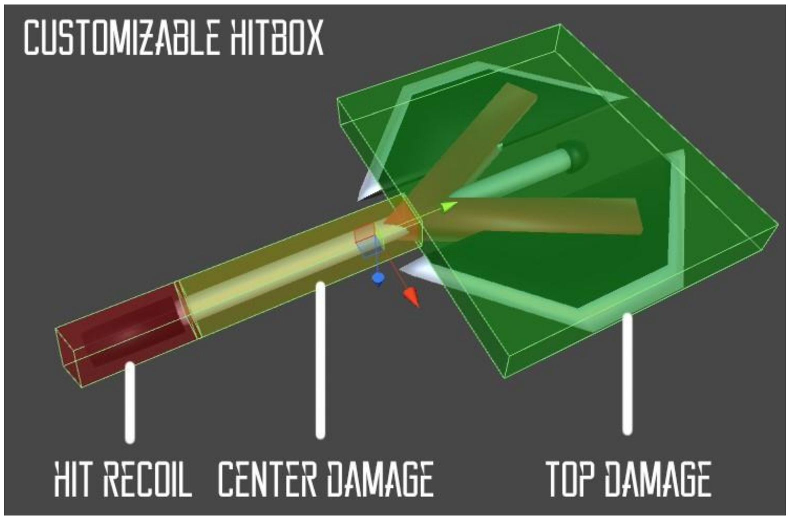
IMPORTANT – don't forget to set your **Weapon Layer to Ignore Raycast and the Tag to Weapon**, if you put a weapon into an Enemy or Player and change the Layer and children's, you need to set the weapon layer to Triggers again

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



**V1.2 image for custom hitbox reference*

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:

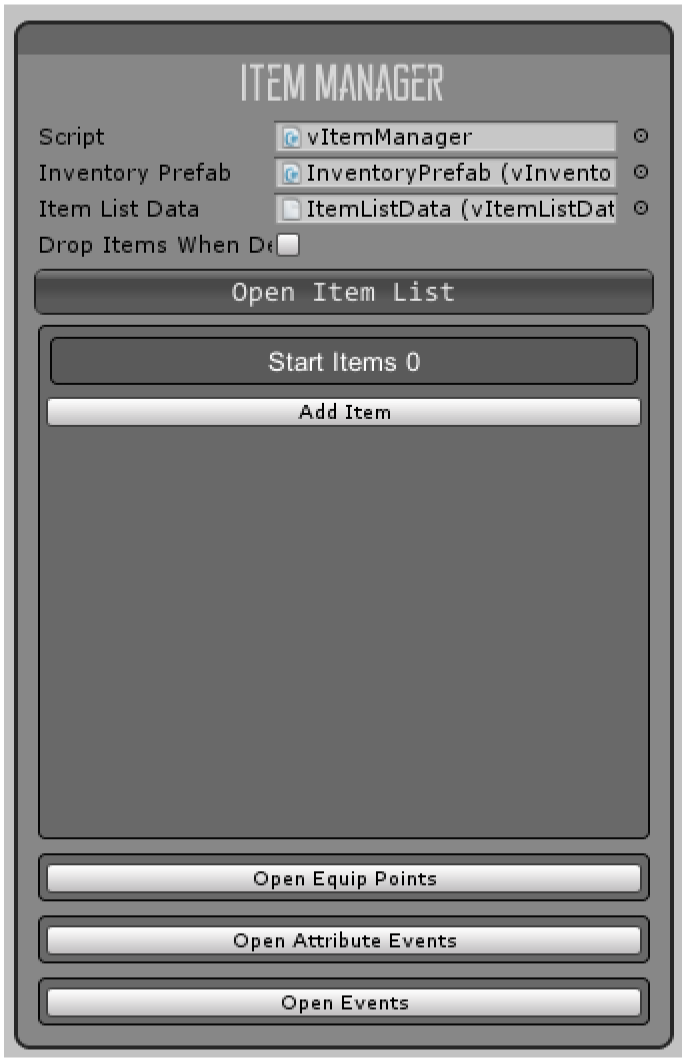
DEF ID > ID of your defense animation
Recoil_ID > Trigger a recoil animation
Defense Rate > how much damage you can defend from an attack
Defense Range > When you select the shield, a Gizmos will appear to help you see how much of Defense Range you need.
Break Attack > Trigger a Recoil Animation on the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

ITEM MANAGER

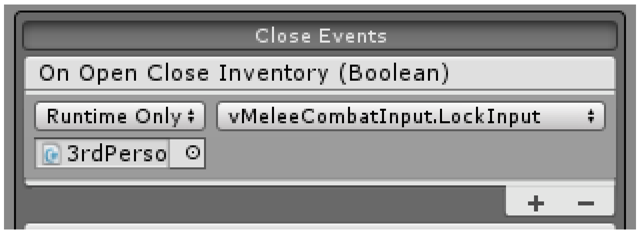
To add an Item Manager, select your Player and go to Invector > Melee Combat > Item Manager.

After that the Item Manager Component will be attach into your Character and the **Inventory Prefab** will be auto assign with our example that you can fully customize and apply your own skin.

The **ItemListData** is a resource file, and you can create a new one by going into Inventor > Resources > NewI temListData, just like the CameraState ListData or the Audio Surface.


ITEM MANAGER
Script vItemManager
Inventory Prefab InventoryPrefab (vInvento
Item List Data ItemListData (vItemListDat
Drop Items When De

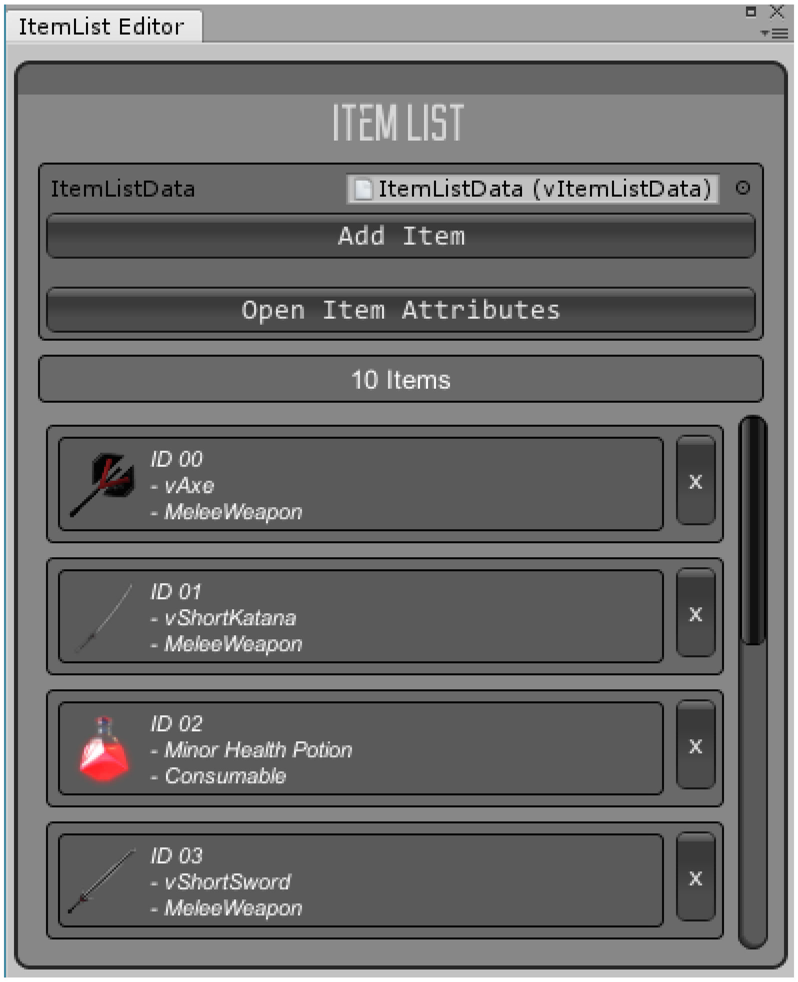
We have added a window for **Events** that are very useful, for example if you want to lock the input of the character while the Inventory is Open, just assign the Character and call the method.



On Open Close Inventory (Boolean)
Runtime Only vMeleeCombatInput.LockInput
3rdPerso

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.



When creating a Weapon Item, you need to assign the **Original Object** (that instantiate into the Player with a vMeleeWeapon) and a **DropObject** which we have a prefab called **"CollectableEquipment"** that you can use.

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.

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.

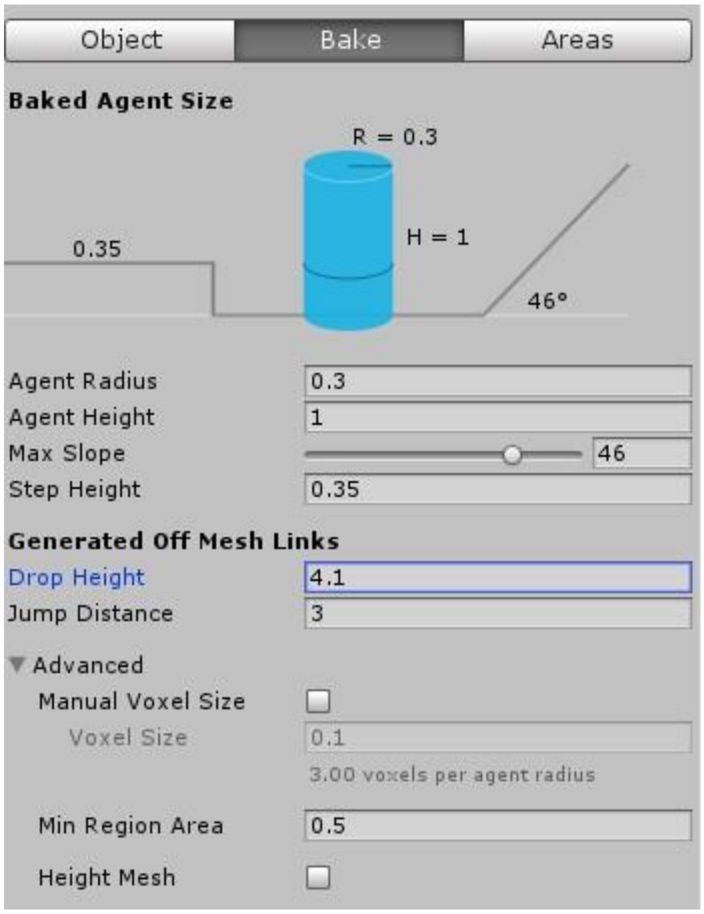
Locomotion - It works pretty much the same as the Character Controller, you still need to set up the

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…



Baked Agent Size

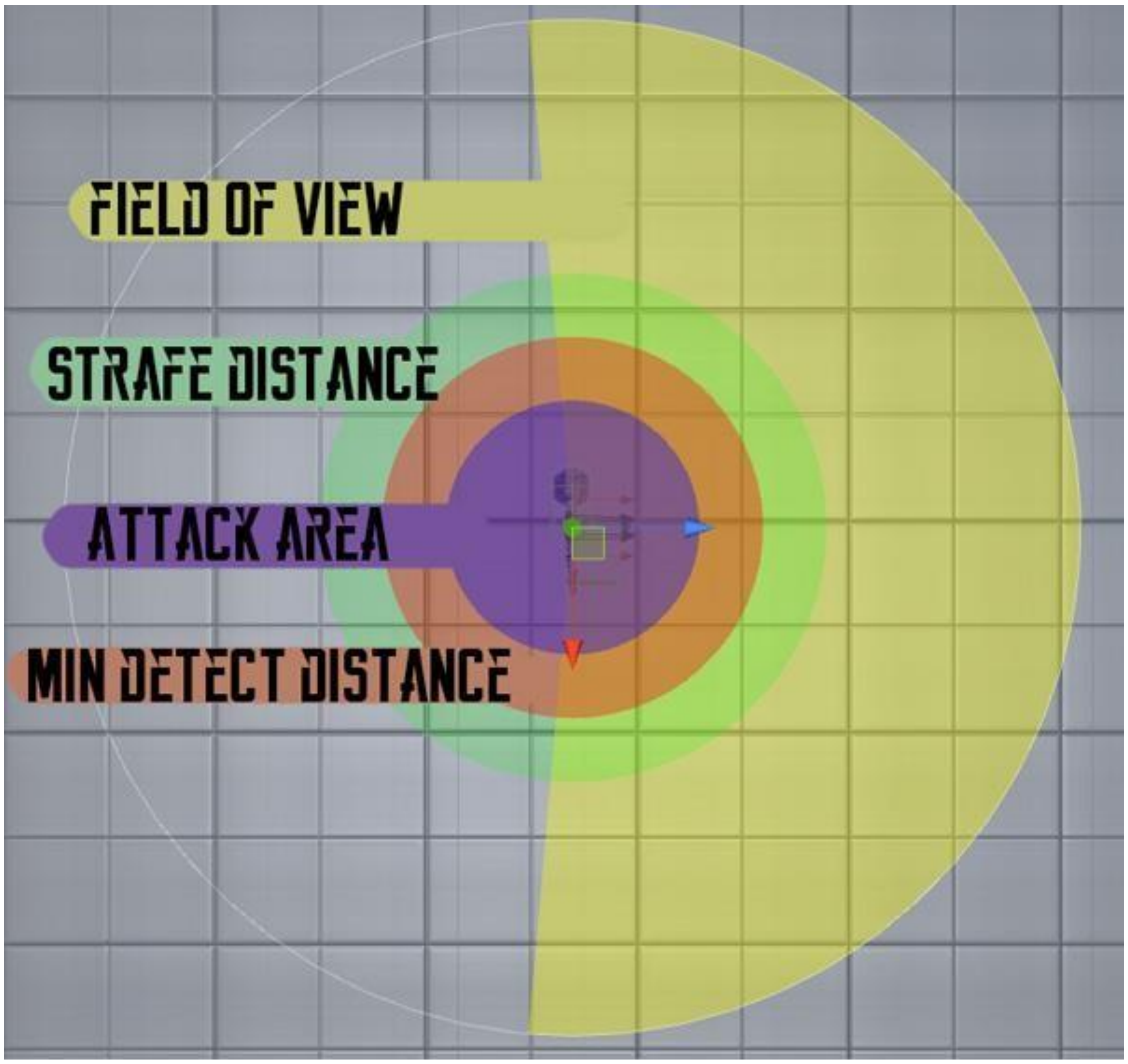
Agent Radius	0.3
Agent Height	1
Max Slope	46
Step Height	0.35

Generated Off Mesh Links

Drop Height	4.1
Jump Distance	3

▼ Advanced

Manual Voxel Size	☐
Voxel Size	0.1
	3.00 voxels per agent radius
Min Region Area	0.5
Height Mesh	☐



Field of View > total range to detect the Player

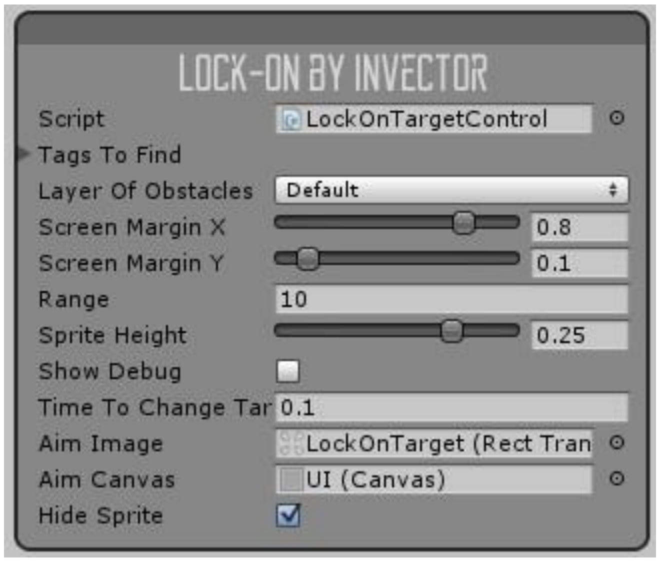
Strafe Distance > once in combat, the character will move Strafing

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.**


LOCK-ON BY INVECTOR

Script	LockOnTargetControl
Tags To Find	
Layer Of Obstacles	Default
Screen Margin X	0.8
Screen Margin Y	0.1
Range	10
Sprite Height	0.25
Show Debug	☐
Time To Change Tar	0.1
Aim Image	LockOnTarget (Rect Tran
Aim Canvas	UI (Canvas)
Hide Sprite	✓

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.

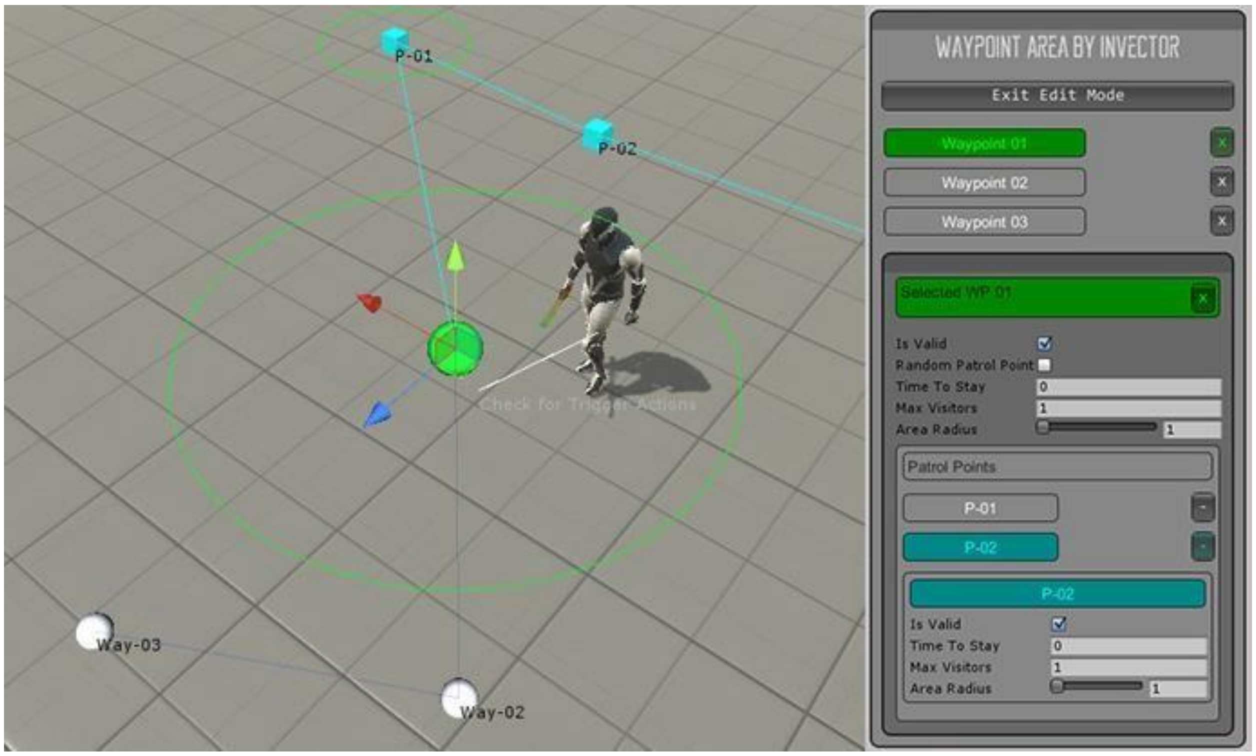
Patrol Points are points of interest that one waypoint has, for example if you have a corridor with 3 rooms, you can create 1 waypoint in the middle of the corridor and 3 patrol points with **Max Visitors of 1**, this means that if an AI is already on a room, the other AI will not come to the same room,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



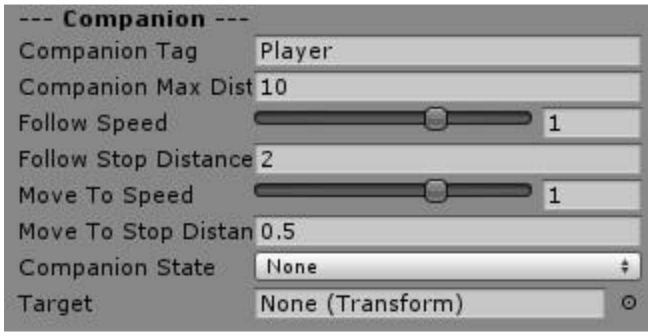
Is Valid ☑
Random Patrol Point ☐
Time To Stay 0
Max Visitors 1
Area Radius 1

P-02
Is Valid ☑
Time To Stay 0
Max Visitors 1
Area Radius 1

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:



--- Companion ---

Companion Tag	Player
Companion Max Dist	10
Follow Speed	1
Follow Stop Distance	2
Move To Speed	1
Move To Stop Distan	0.5
Companion State	None
Target	None (Transform)

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.