3rd Person Controller – Basic Locomotion

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.

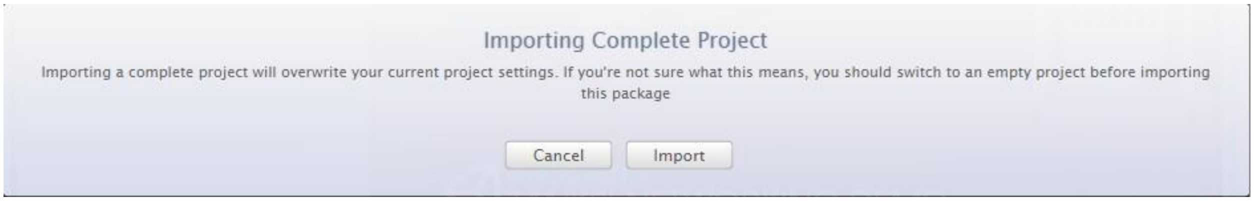
Invector Team

Summary

First Run

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**.

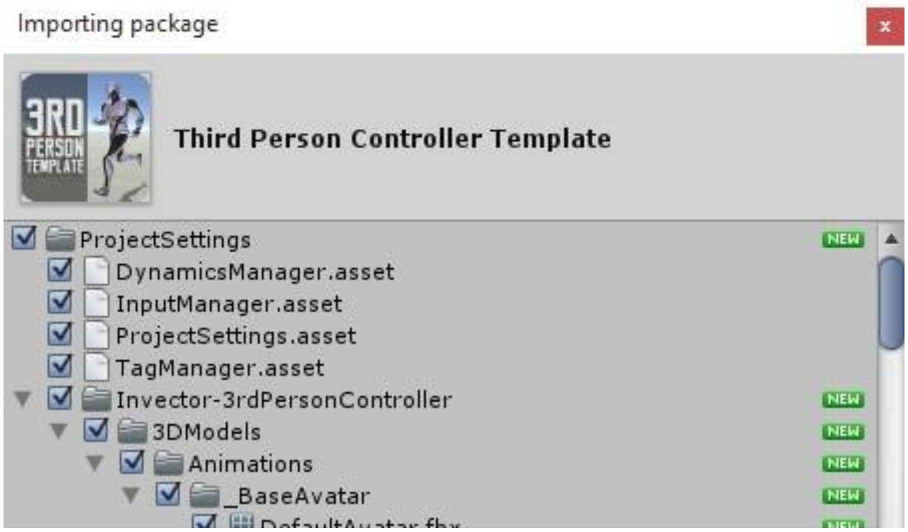


Importing Complete Project

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

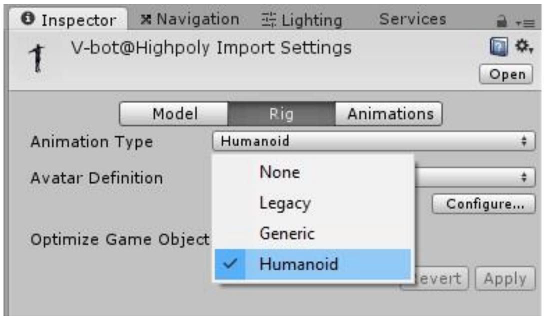
Third Person Controller Template

☑ ProjectSettings NEW
☑ DynamicsManager.asset
☑ InputManager.asset
☑ ProjectSettings.asset
☑ TagManager.asset
☑ Invector-3rdPersonController NEW
☑ 3DModels NEW
☑ Animations NEW
☑ _BaseAvatar NEW
☑ DefaultAvatar.fbx 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. ***

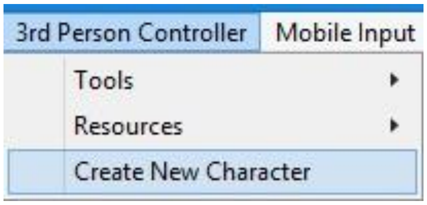
Creating a new Character

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

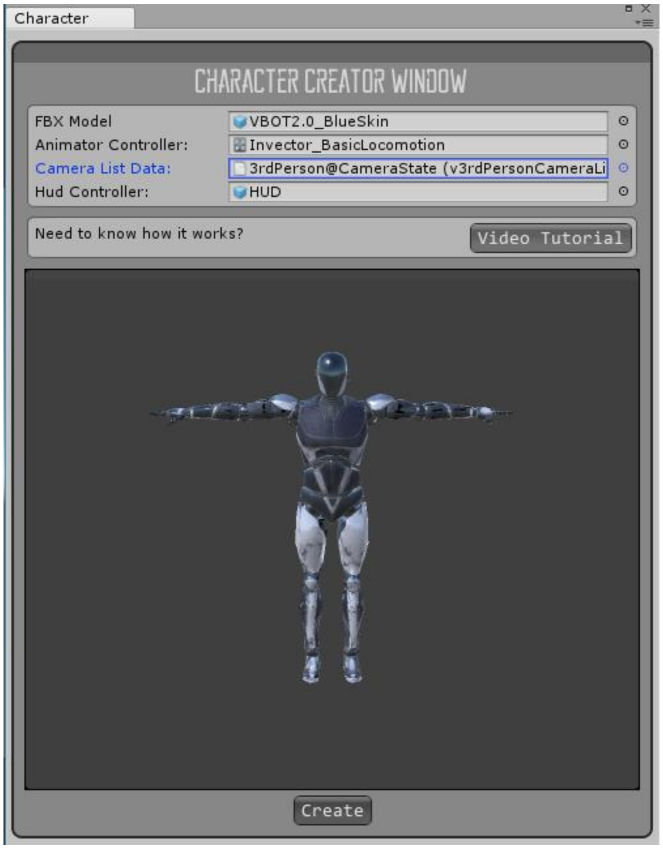


Animation Type: Humanoid

To setup a new character, go to the tab “3rd Person Controller” and click “Create New Character”



Make sure your Character is **Fully Rigged** and set up the FBX rig option to **Humanoid**, then assign the FBX to field Humanoid”, the **Animator Controller**, select a **CameraState** and click the button “Create”.



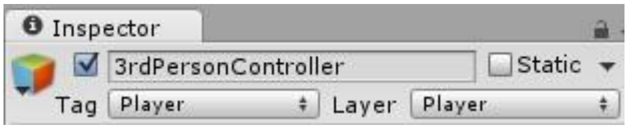
FBX Model: VBOT2.0_BlueSkin

Animator Controller: Invector_BasicLocomotion

Camera List Data: 3rdPerson@CameraState (v3rdPersonCameraLi

Hud Controller: HUD

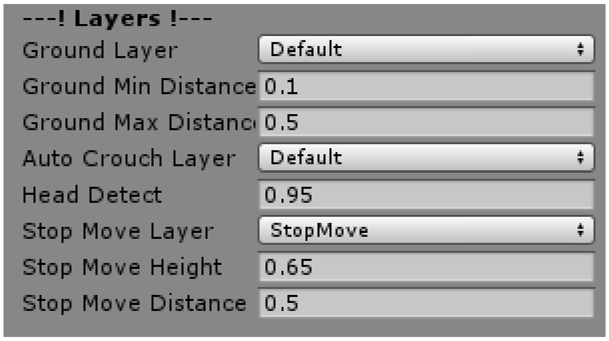
1- ***ADD V1.1* -** Make sure to set up a different **Layer** for the **Player** and the **Ground Layer**, if your project does not contain this Layers, create then on Edit > Project Settings > Tags and Layers.


Inspector

☑ 3rdPersonController □ Static

Tag Player Layer Player

Make sure that your layers are assign correctly on the Player Inspector:


---! Layers !---

Ground Layer Default

Ground Min Distance 0.1

Ground Max Distanc 0.5

Auto Crouch Layer Default

Head Detect 0.95

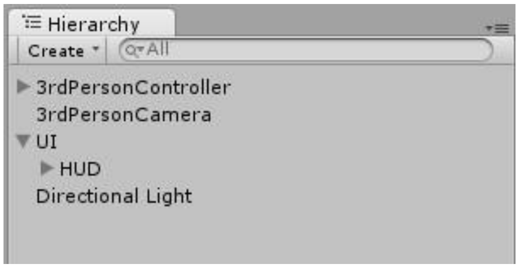
Stop Move Layer StopMove

Stop Move Height 0.65

Stop Move Distance 0.5

2- **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.

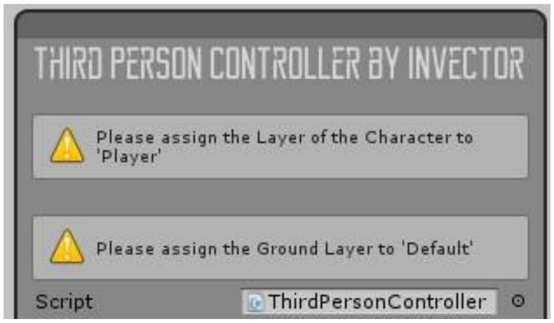

Hierarchy

Create All

▶ 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.

***ADD V1.2* -** The character inspector will show warnings accusing if the Layers are not correct


THIRD PERSON CONTROLLER BY INVECTOR

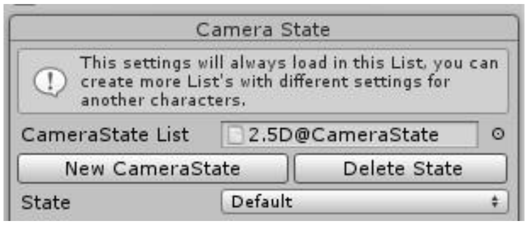
Please assign the Layer of the Character to 'Player'

Please assign the Ground Layer to 'Default'

Script ThirdPersonController

3- Hit Play and enjoy ☺

V1.1 Additional Info:
3rd Person and **2.5D** bot use the **3rd Person Controller Template**, just change the CameraStateList.



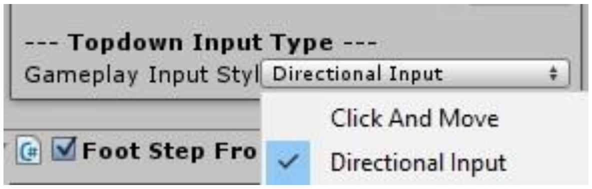
Camera State

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

CameraState List: 2.5D@CameraState

State: Default

Topdown and **Isometric** both use the **Topdown Controller Template**, you can change to input style to Mouse and Click or control with inputs on the Player Inspector.



--- Topdown Input Type ---

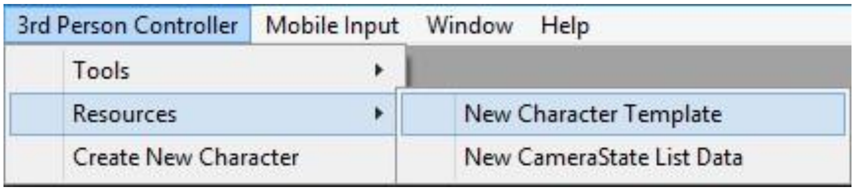
Gameplay Input Styl: Directional Input

☑ Foot Step Fro

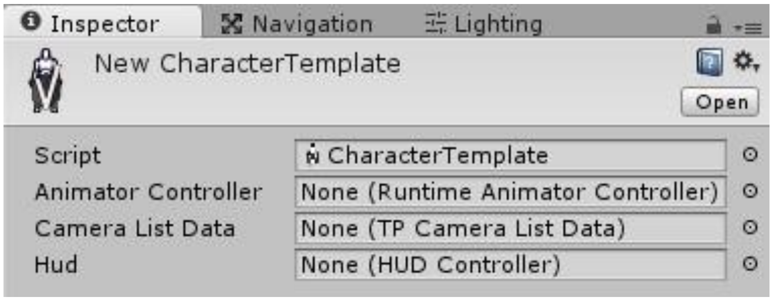
Also, remember to **Uncheck** the **HeadTrack** option at the Player Inspector if you are using **Topdown** or **Isometric** mode.

Creating a new Character Template

You can set up the Character Creator to create **custom controllers** that you have modified, to create a new template go to “3rd Person Controller” tab, “Resources” and click on “New Character Template”.



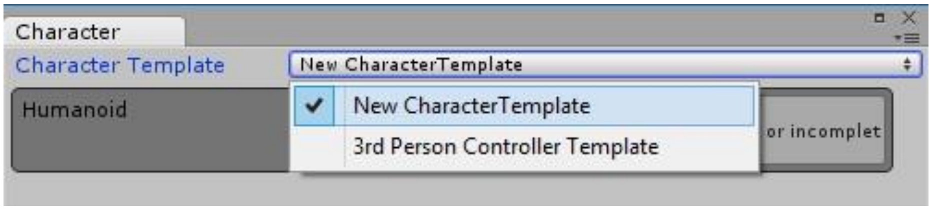
Assign your **modified prefab** of an **Animator Controller**, **Camera List Data** and the **HUD Controller**.



New CharacterTemplate

Script: CharacterTemplate
Animator Controller: None (Runtime Animator Controller)
Camera List Data: None (TP Camera List Data)
Hud: None (HUD Controller)

The next time you create a new controller, choose the new template.



Character Template: New CharacterTemplate

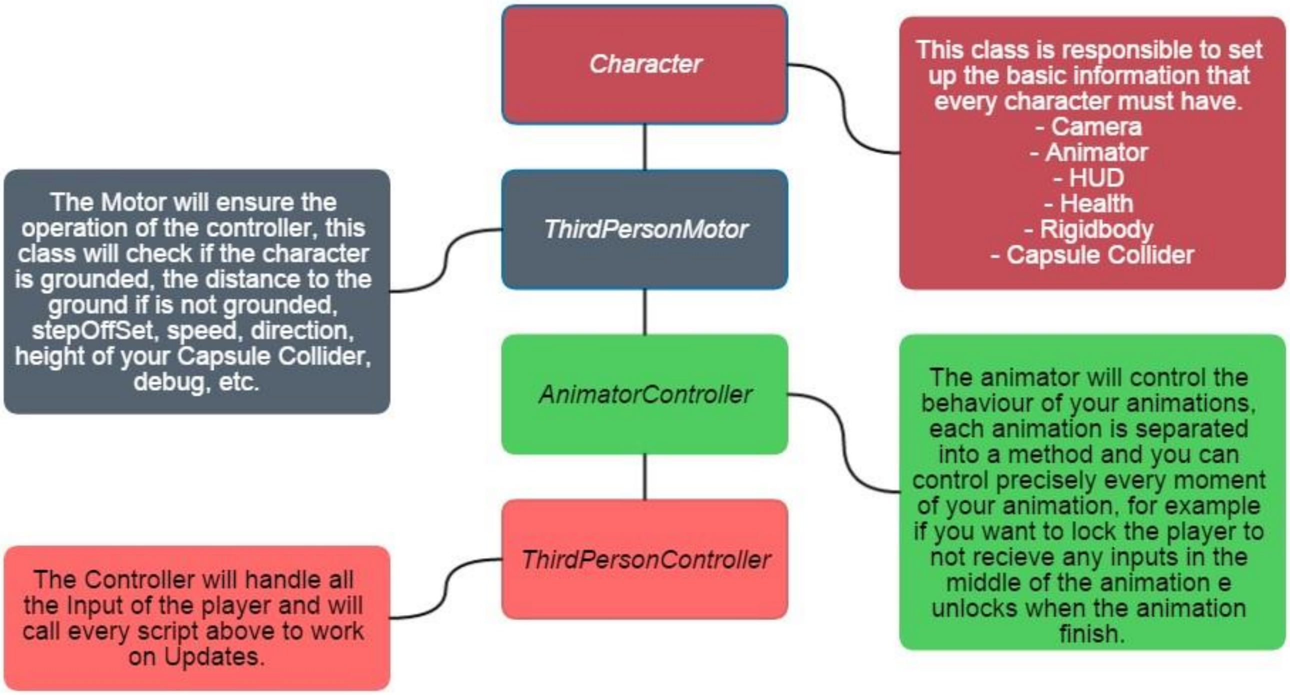
How it works?

The Controller works with **four main scripts**, Character, Motor, Animator and the Controller.

Character will take care of the basic components in order to make the controller works, things like a Camera, Animator, Rigidbody, Capsule Collider, etc..

Third Person Motor handles all the verifications of ground distance, stepoffset, slope limit, etc..

Animator Controller is responsible to control the behavior of your animations, you can set set bools, float, int and control the state of your animation. **Third Person Controller** receives all the input and call every method of the other scripts.



Character

This class is responsible to set up the basic information that every character must have.
- Camera
- Animator
- HUD
- Health
- Rigidbody
- Capsule Collider

ThirdPersonMotor

The Motor will ensure the operation of the controller, this class will check if the character is grounded, the distance to the ground if is not grounded, stepOffSet, speed, direction, height of your Capsule Collider, debug, etc.

AnimatorController

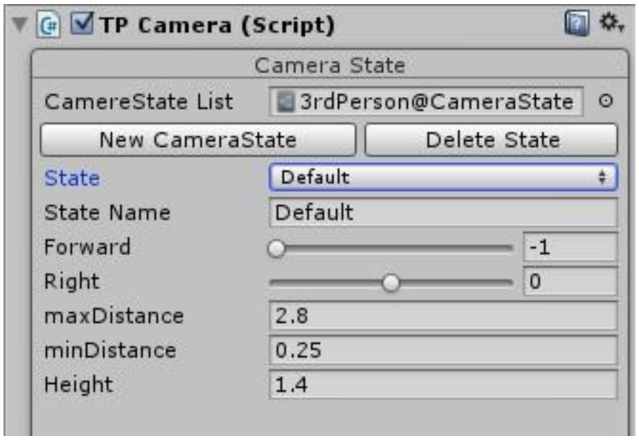
The animator will control the behaviour of your animations, each animation is separated into a method and you can control precisely every moment of your animation, for example if you want to lock the player to not recieve any inputs in the middle of the animation e unlocks when the animation finish.

ThirdPersonController

The Controller will handle all the Input of the player and will call every script above to work 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.



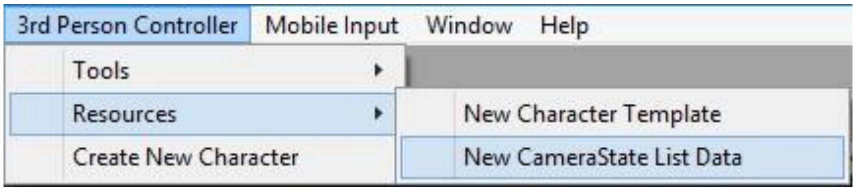
CamereState List: 3rdPerson@CameraState
State: Default
State Name: Default
Forward: -1
Right: 0
maxDistance: 2.8
minDistance: 0.25
Height: 1.4

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

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

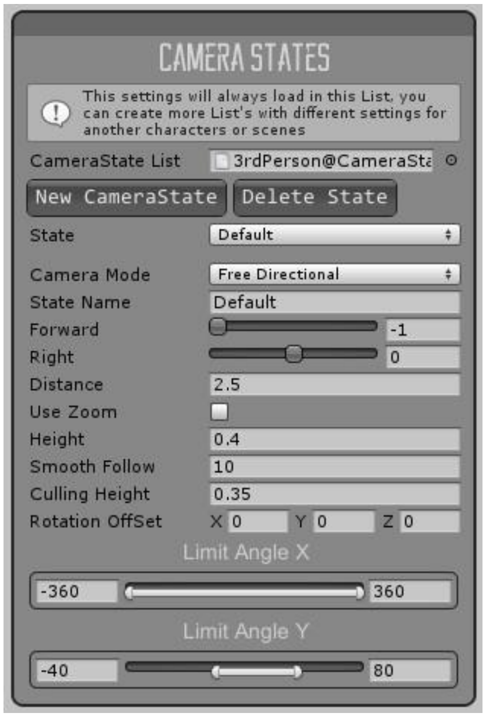
The first string value is the State Name that you created on the Camera Inspector, the second value is a bool, leave it true if you want a smooth transition to this state or false if not.

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…

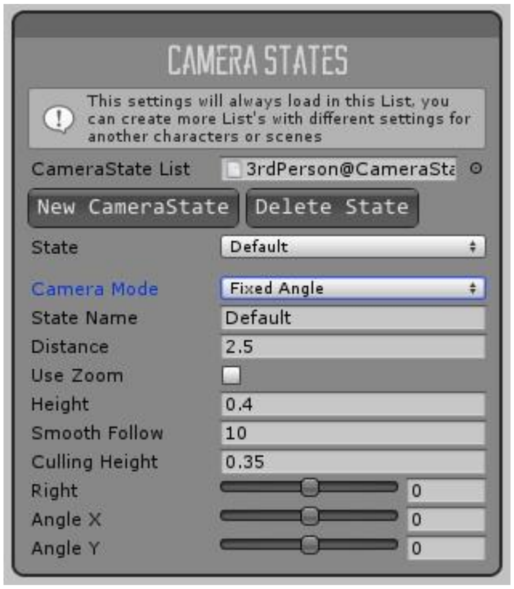


This settings will always load in this List, you can create more List's with different settings for another characters or scenes

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.

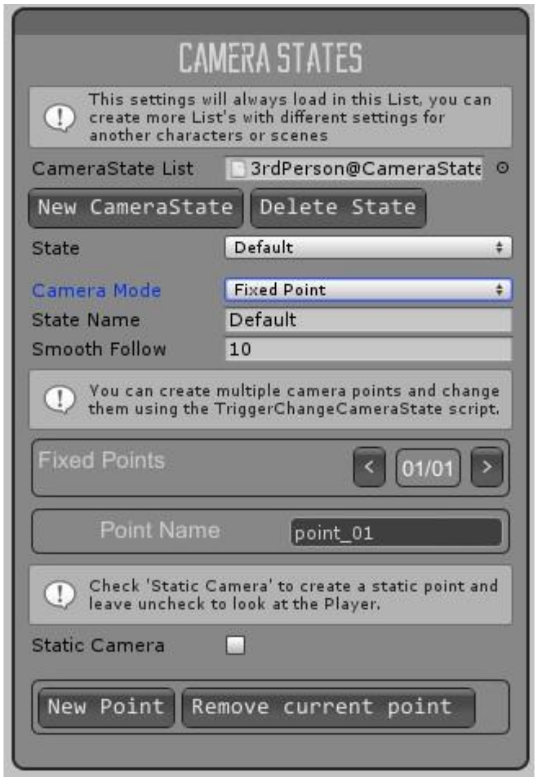


This settings will always load in this List, you can create more List's with different settings for another characters or scenes

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.



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.

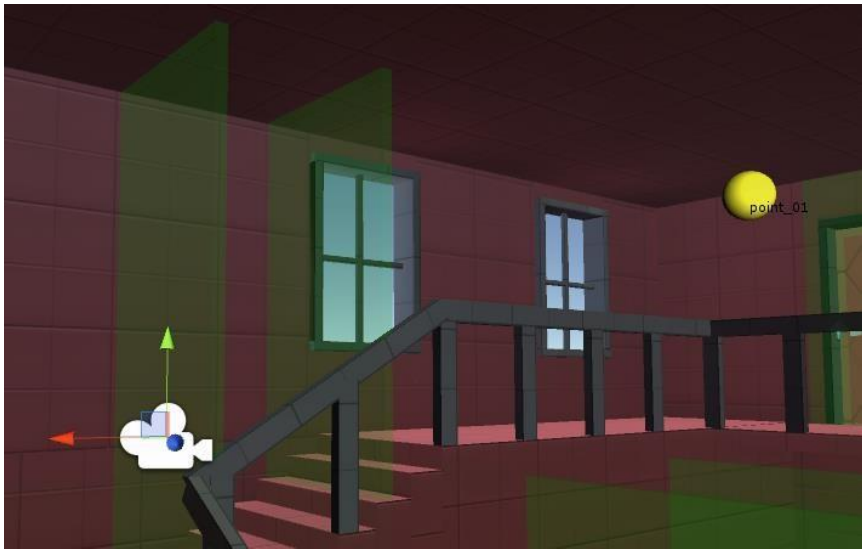
Fixed Points

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 360 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.

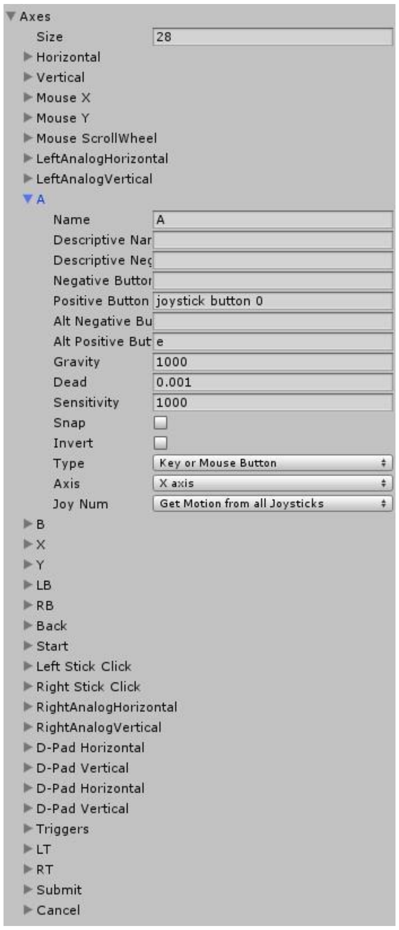
To apply the vibration, you can call the method by SendMessage to the player, for example:

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

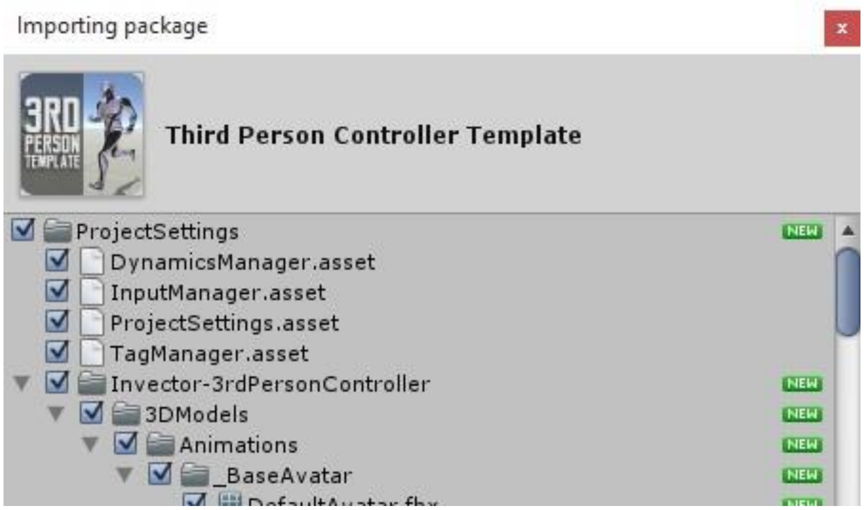
Our input manager is mapped with the 360 controller, you can modify the inputs of mobile, keyboard/mouse or joystick on the **vThirdPersonInput** script.



```
▼ Axes
    Size               28
  ▶ Horizontal
  ▶ Vertical
  ▶ Mouse X
  ▶ Mouse Y
  ▶ Mouse ScrollWheel
  ▶ LeftAnalogHorizontal
  ▶ LeftAnalogVertical
  ▼ A
      Name               A
      Descriptive Nar
      Descriptive Neg
      Negative Buttor
      Positive Button    joystick button 0
      Alt Negative Bu
      Alt Positive But   e
      Gravity            1000
      Dead               0.001
      Sensitivity        1000
      Snap               □
      Invert             □
      Type               Key or Mouse Button
      Axis               X axis
      Joy Num            Get Motion from all Joysticks
  ▶ B
  ▶ X
  ▶ Y
  ▶ LB
  ▶ RB
  ▶ Back
  ▶ Start
  ▶ Left Stick Click
  ▶ Right Stick Click
  ▶ RightAnalogHorizontal
  ▶ RightAnalogVertical
  ▶ D-Pad Horizontal
  ▶ D-Pad Vertical
  ▶ D-Pad Horizontal
  ▶ D-Pad Vertical
  ▶ Triggers
  ▶ LT
  ▶ RT
  ▶ Submit
  ▶ Cancel
```

Tags & Layers

V1.3 – If you 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.

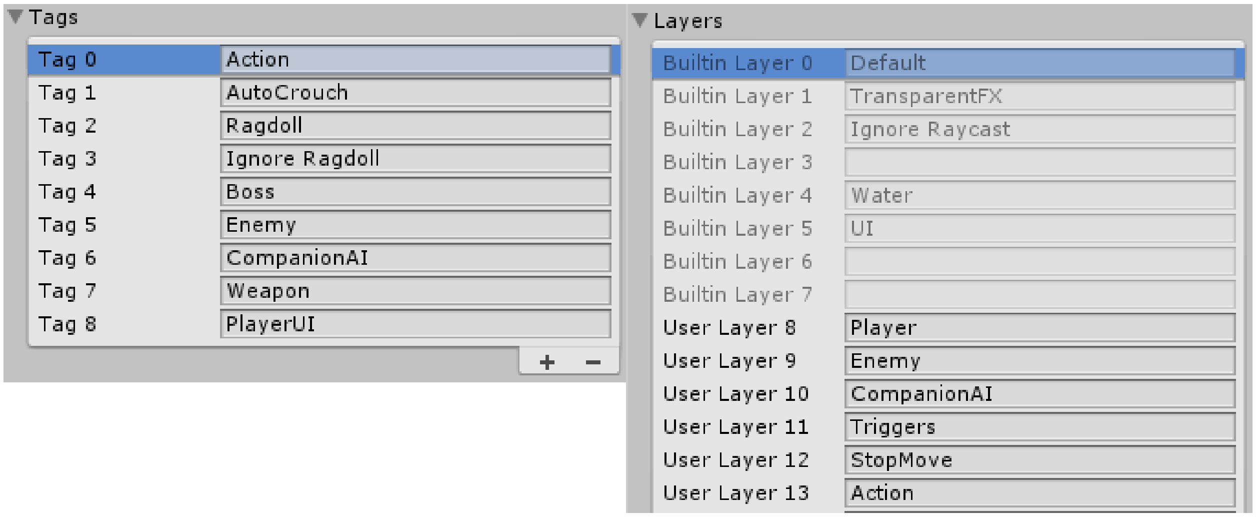


Importing package

Third Person Controller Template

- ☑ ProjectSettings NEW
  - ☑ DynamicsManager.asset
  - ☑ InputManager.asset
  - ☑ ProjectSettings.asset
  - ☑ TagManager.asset
- ☑ Invector-3rdPersonController NEW
  - ☑ 3DModels NEW
    - ☑ Animations NEW
      - ☑ _BaseAvatar NEW
        - ☑ DefaultAvatar.fbx NEW

This is all the layers and tags we need to for the template work properly:



Tags

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

Layers

Builtin Layer 0	Default
Builtin Layer 1	TransparentFX
Builtin Layer 2	Ignore Raycast
Builtin Layer 3	
Builtin Layer 4	Water
Builtin Layer 5	UI
Builtin Layer 6	
Builtin Layer 7	
User Layer 8	Player
User Layer 9	Enemy
User Layer 10	CompanionAI
User Layer 11	Triggers
User Layer 12	StopMove
User Layer 13	Action

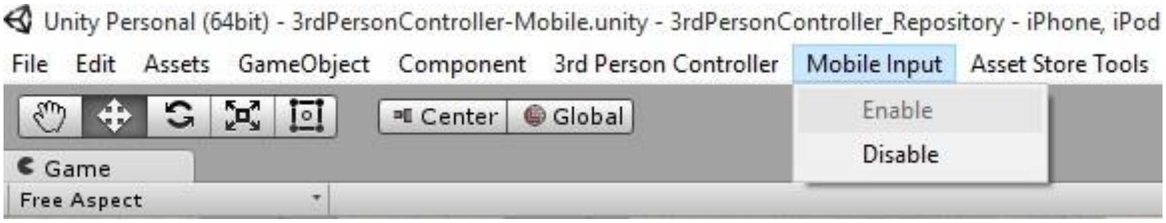
Recommended Mobile Settings

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)

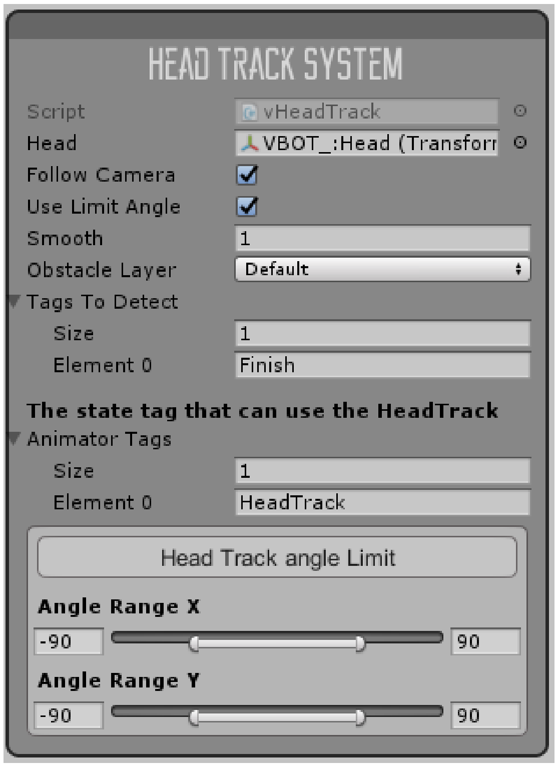
Don’t forget to **Enable** the Mobile Input after change the platform, it should work right on the Editor.



With these settings, we manage to get **stable 60fps** on several Android smartphones
**Unity does no longer supports Tegra devices*

Head Track

You can add the Head Track component by clicking on the 3rd Person Controller menu, Components > HeadTrack. You can make the character **follow the camera** rotation and set angle limits X and Y.



HEAD TRACK SYSTEM

Script: vHeadTrack
Head: VBOT_:Head (Transforr
Follow Camera: ☑
Use Limit Angle: ☑
Smooth: 1
Obstacle Layer: Default
Tags To Detect
Size: 1
Element 0: Finish

The state tag that can use the HeadTrack

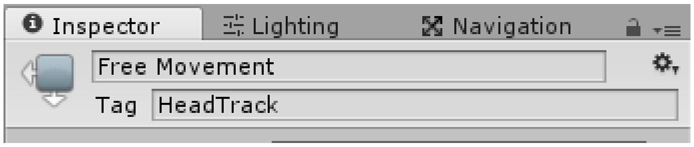
Animator Tags
Size: 1
Element 0: HeadTrack

Head Track angle Limit

Angle Range X
-90 — 90

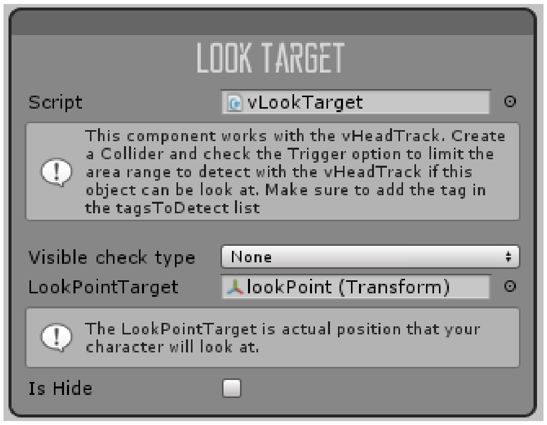
Angle Range Y
-90 — 90

To enable the head track, you need to apply the **Tag** “HeadTrack” into the **AnimationState** on the Animator Controller, usually the main locomotion states.



Free Movement
Tag: HeadTrack

There is also the option to LookAt objects, check more information on the **LookTarget** 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: None
LookPointTarget: lookPoint (Transform)

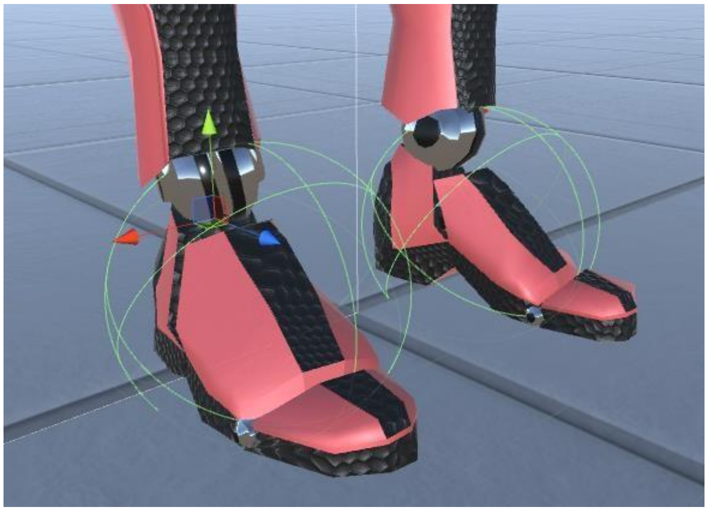
The LookPointTarget is actual position that your character will look at.

Is Hide: □

FootStep Audio System

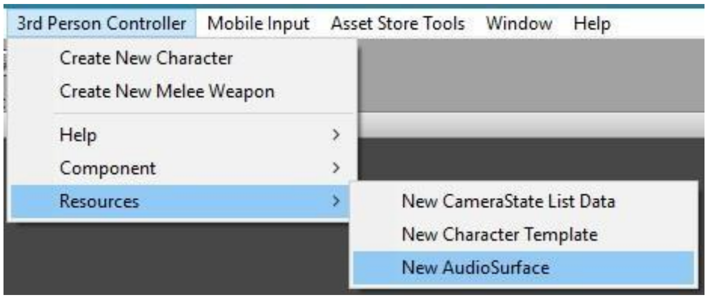
V1.2.1 New FootStep System.

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.

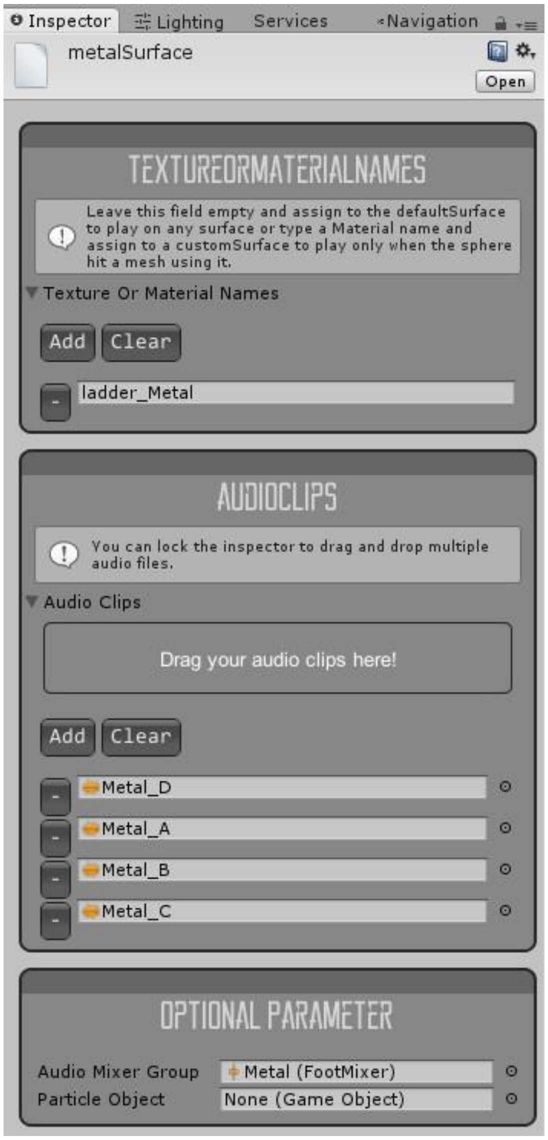


You can select the **LeftFoot** and **RightFoot** Sphere and manipulate the **Center XYZ** to position as you like, and change the **Collider Radius** too, the size of this sphere will depend on your Rig bone size. Assign the "***defaultSurface***" that comes with the package to have an example of how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

To create a new AudioSurface go to the 3^{rd} 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

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

ladder_Metal

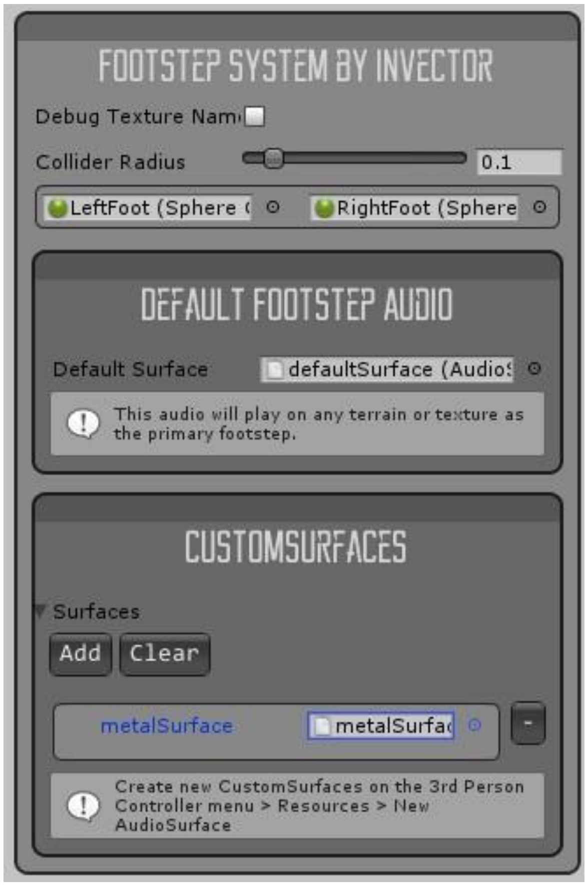
You can lock the inspector to drag and drop multiple audio files.

Audio Clips

Drag your audio clips here!

Metal_D
Metal_A
Metal_B
Metal_C

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



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

Default Surface: defaultSurface (Audio!

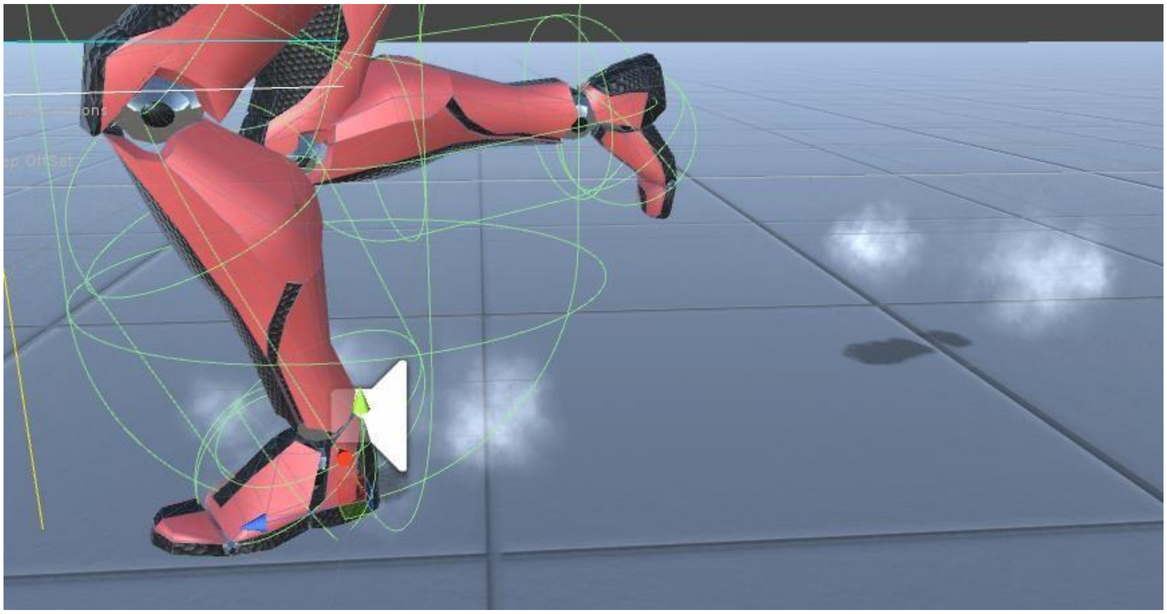
This audio will play on any terrain or texture as the primary footstep.

Surfaces

metalSurface metalSurfac

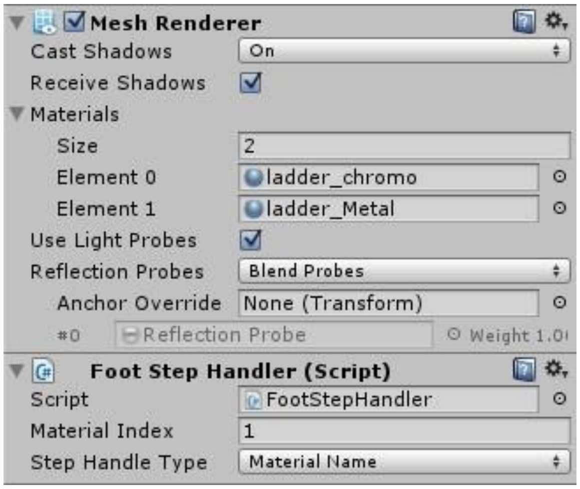
Create new CustomSurfaces on the 3rd Person Controller menu > Resources > New AudioSurface

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)

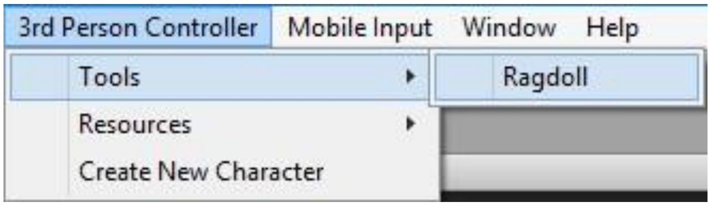
Script: FootStepHandler

Material Index: 1

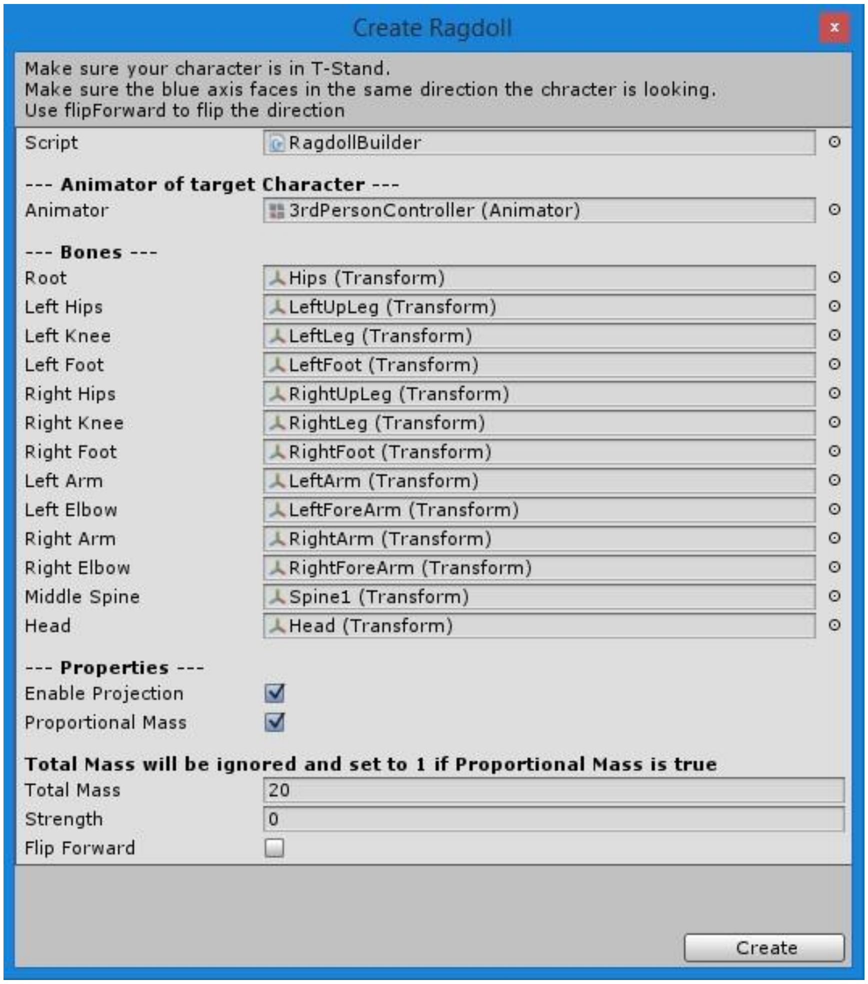
Step Handle Type: Material Name

Creating a Ragdoll

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



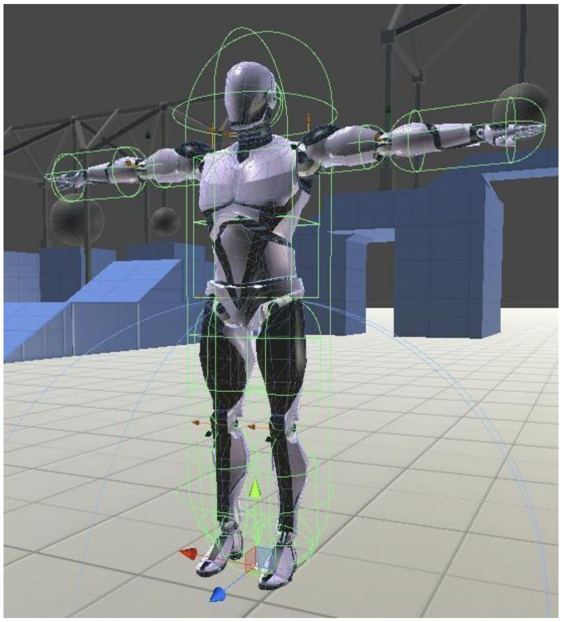
If you have your character selected on the Hierarchy, all the fields will **autofill**, if not, just click on your character and it will autofill for you, this template was design to **save time**, so you don’t have to waste your time dragging and drop every bone, instead just hit the “Create” button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



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

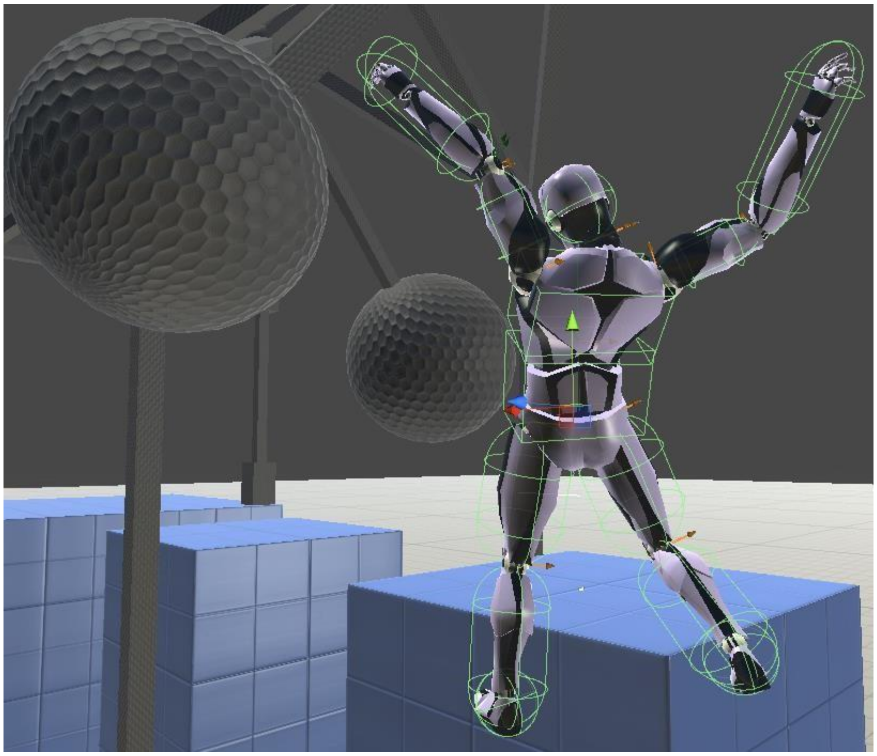
Script	RagdollBuilder
--- Animator of target Character ---	
Animator	3rdPersonController (Animator)
--- Bones ---	
Root	Hips (Transform)
Left Hips	LeftUpLeg (Transform)
Left Knee	LeftLeg (Transform)
Left Foot	LeftFoot (Transform)
Right Hips	RightUpLeg (Transform)
Right Knee	RightLeg (Transform)
Right 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.

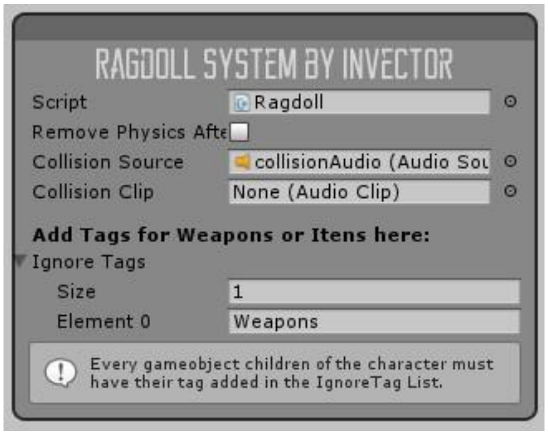


To enable the ragdoll, you can use the Script **ObjectDamage** or just call this line on the **OnCollisionEnter** method.

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.



RAGDOLL SYSTEM BY INVECTOR

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 ThirdPersonMotor script, create a variable like a bool to control the animation
- At the ThirdPersonAnimator script, tell what variable controls what animation
- At the ThirdPersonController 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.

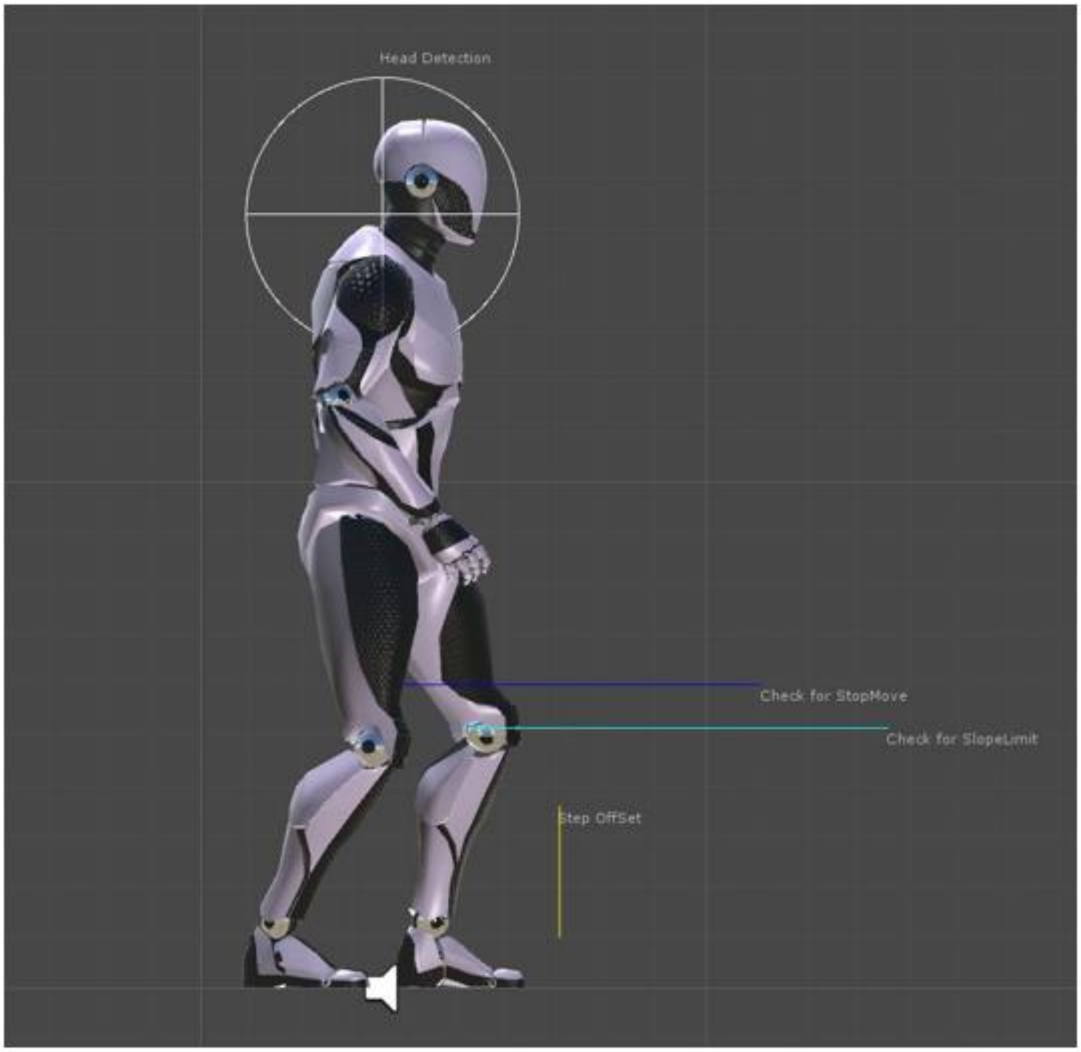
RayCast Checkers (v1.1)

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.

2- **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”.

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

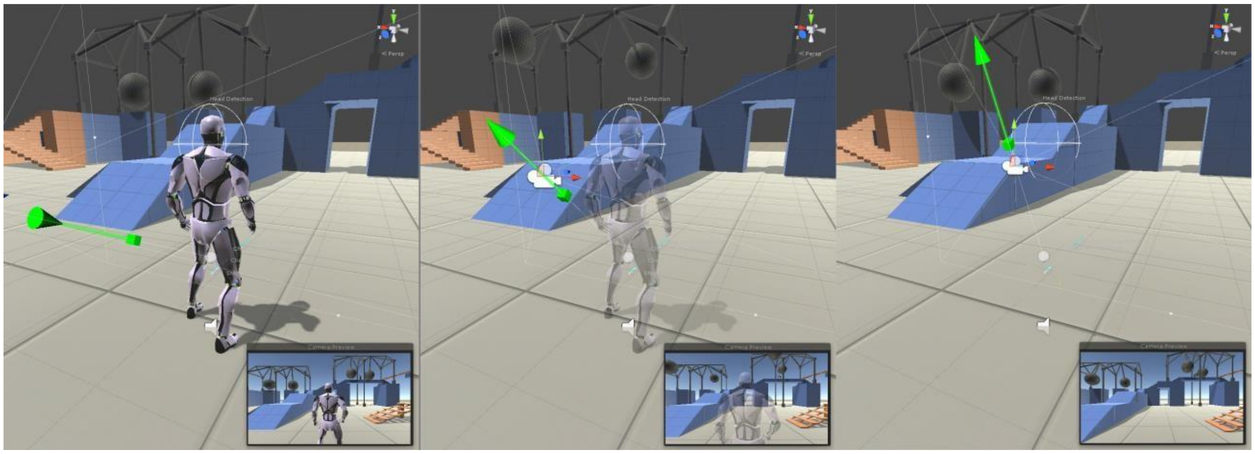
4- **StepOffset** is to help the character walk in custom height steps, adjust the values on the Player Inspector.



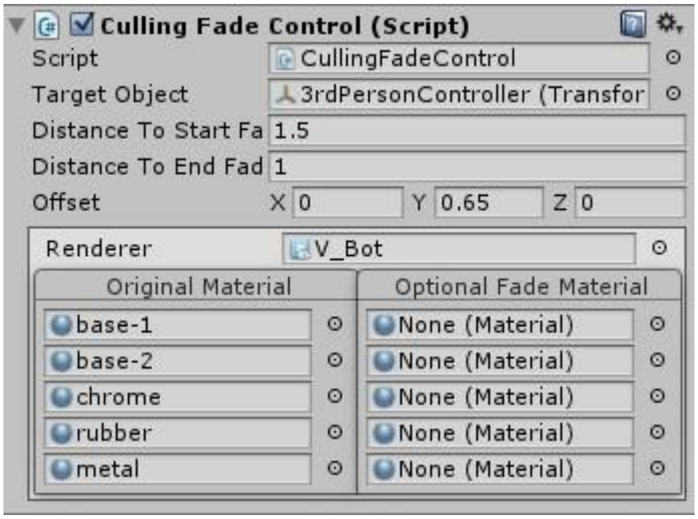
Camera Culling Fade (v1.1a)

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.



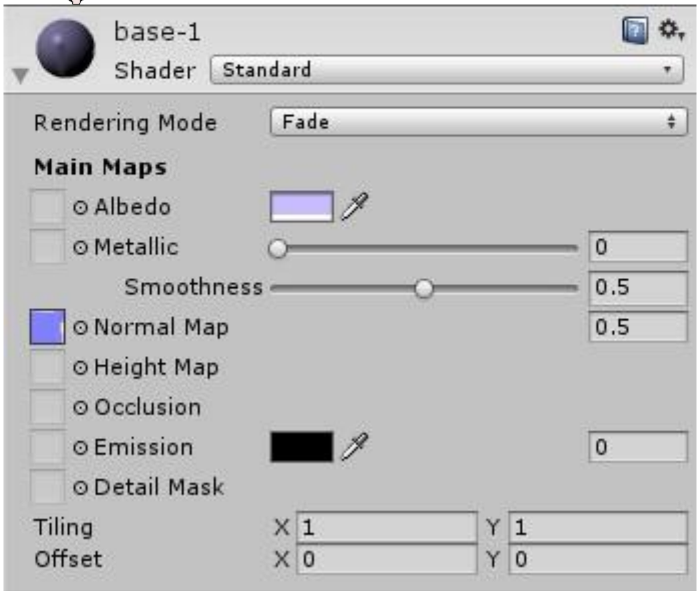
☑ **Culling Fade Control (Script)**

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

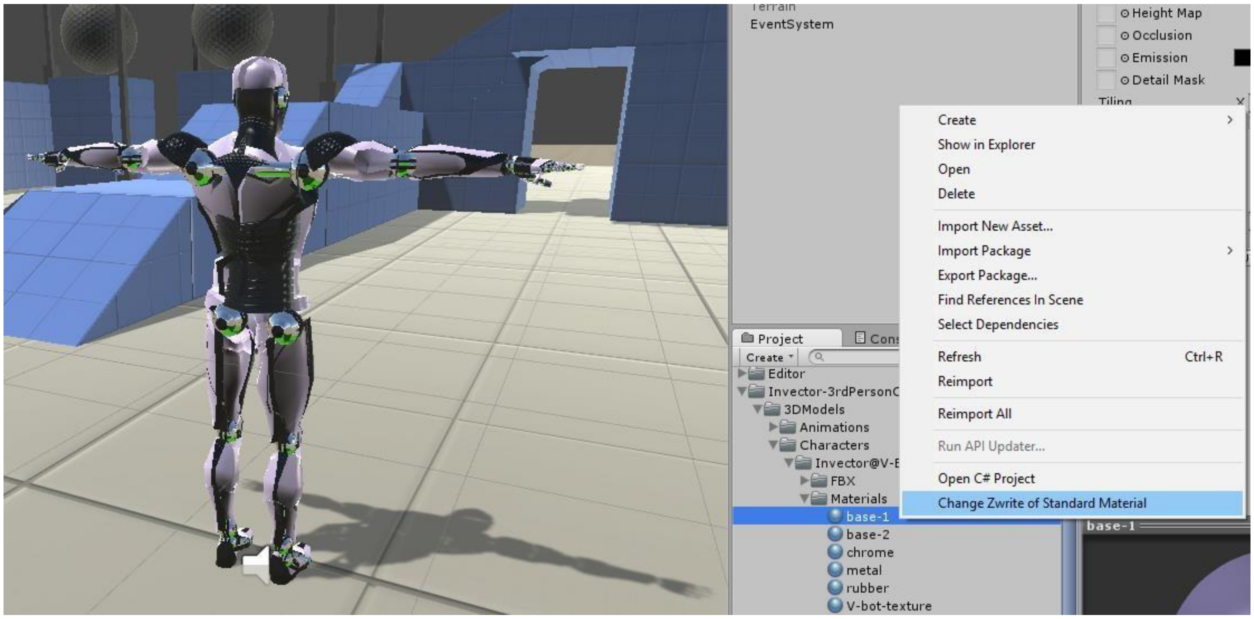
Emission: 0

Detail Mask

Tiling: X 1, Y 1

Offset: X 0, Y 0

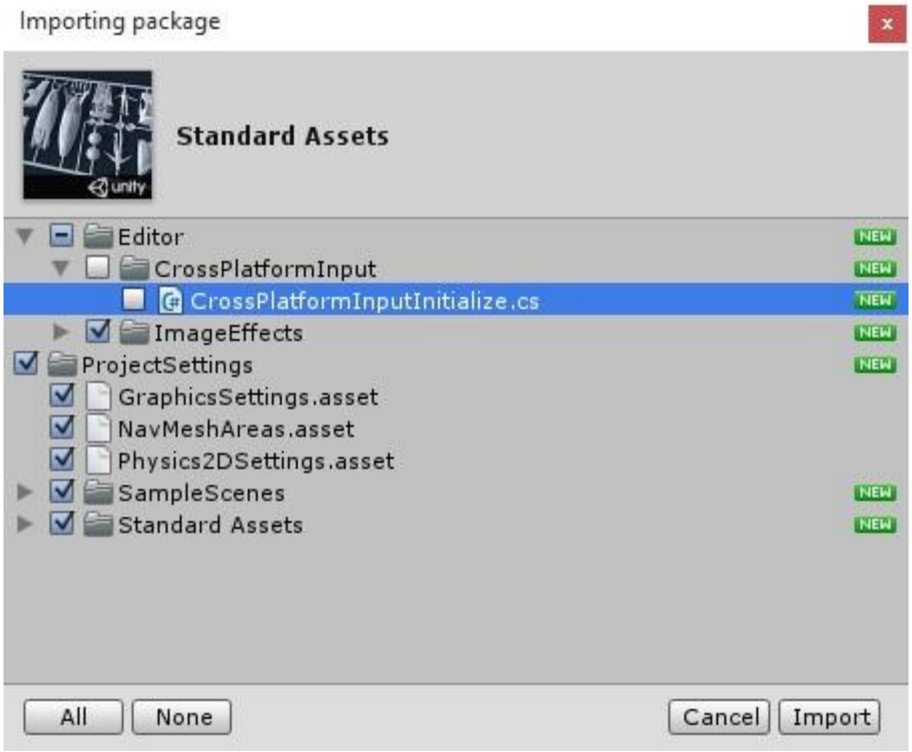
The character will look like this (picture below) but you can fix by right clicking at the material and “Change Zwrite of Standard Material”.



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:



- ⊟ 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]