

## **Readme**

Thank you for buying this model pack.

To use the models, simply drag the masterprefab, located in the “Prefabs” folder into your scene.

## **Customization**

This pack contains an animated character with modular weapons/equipment. Several texture sets for the character are included too.

The weapons/equipment can be attached to the character.

## **PSD Source File**

The PSD can be used to modify/change the textures. It also provides all necessary texture maps for making new ones with quixel mixer or substance painter.

## **Contact Information**

You can contact me here: [msgdi@yahoo.de](mailto:msgdi@yahoo.de). Please include your invoice number.

I always appreciate ratings/reviews on the asset store if you think this pack deserves it. Thank you.

Facebook: <https://www.facebook.com/msgdi>

Twitter: <https://twitter.com/MSGameDev>

Deviantart: <https://www.deviantart.com/msgamedevelopment/gallery/>

Sketchfab: <https://sketchfab.com/MSGDI>

Artstation: <https://www.artstation.com/msgdi>

3D Rigging/Modeling Support: Royal Skies LLC

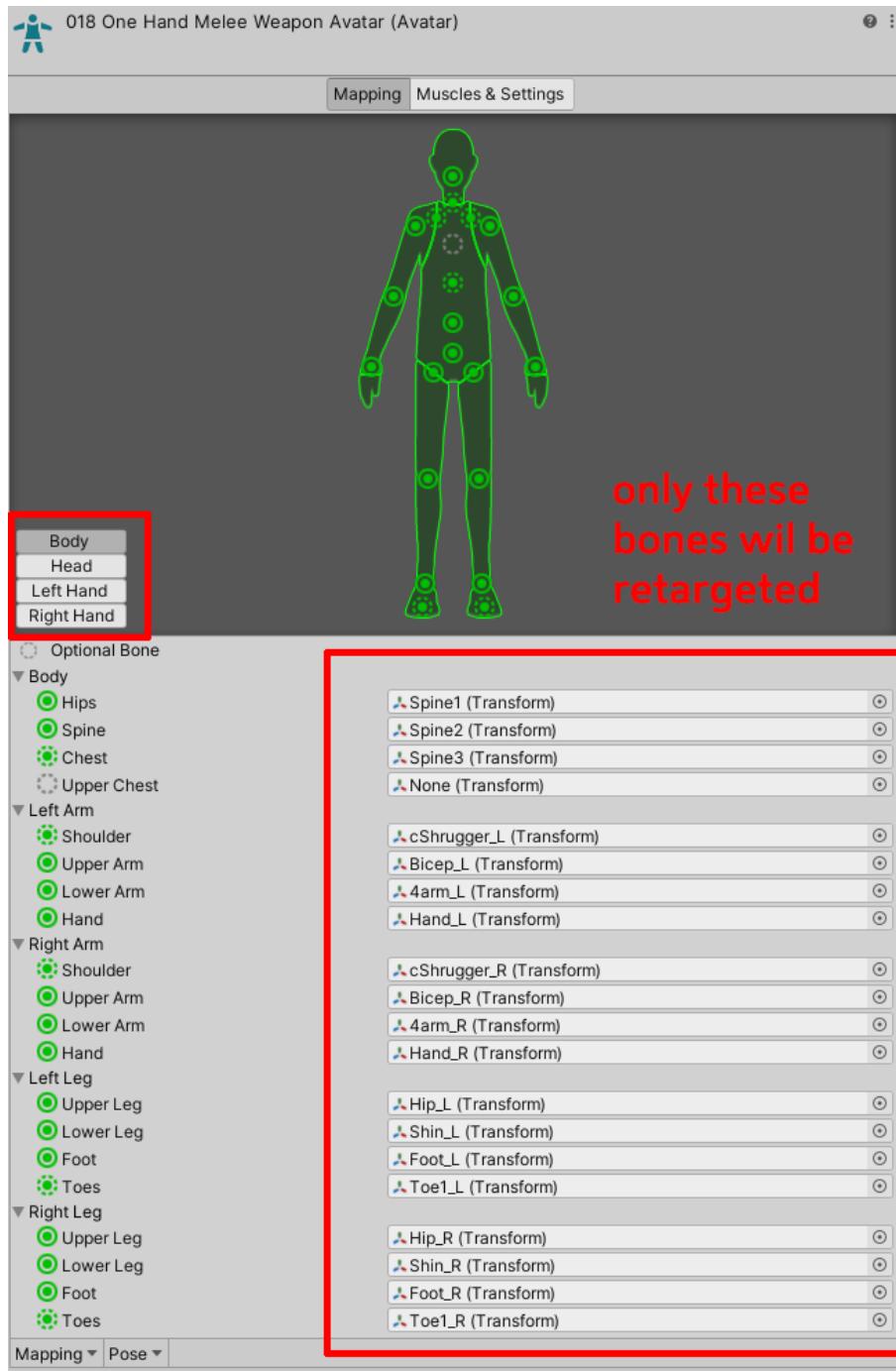
## Character Settings

The character uses a humanoid rig setup and is compatible with Unity's Mecanim system. That means animations from other sources that also use a humanoid rig can be retargeted on the character.

If you're unfamiliar with animation retargeting and how it works: <https://docs.unity3d.com/Manual/Retargeting.html>

However there are a few limitations you should be aware of.

By default only the bones specified in the avatar mapping setup will be retargeted - and nothing else.



The solution to that is using an “avatar mask”: <https://docs.unity3d.com/Manual/class-Avatar-Mask.html>

This mask defines what bones beside the humanoid rig bones should be retargeted. But keep in mind that this only works if both the animation file rig and the target character rig have the same bone structure + bone names.

This character uses the avatar mask for retargeting the boob and hair bones so that these parts will be animated on the target character.

If you e.g. don’t want animated boobs/hair, you can deactivate these bones in the avatar mask and they won’t be animated anymore.

The avatar mask for this character is located in FBX/Animations. It is called “FemaleRigAvatar-Mask”

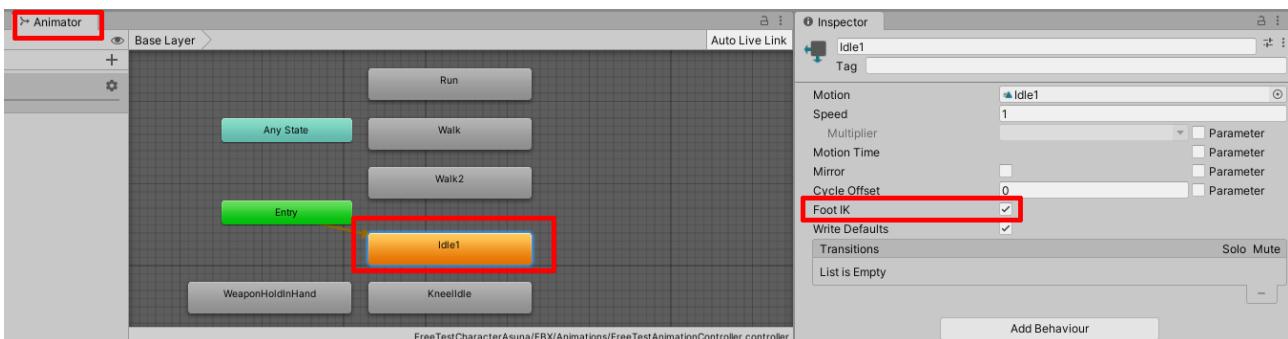
**Be aware: if you use animations from another source they will most likely not have boob/hair bones. This means these bones on the target character won’t get any animation data and will just stay in place.**

## Animation Type

All animations are “in place”. This means they don’t have root motion. The character must be moved with a character controller (which is not included - I’m only a 3d artist, not a programmer).

## Shuffling / Stuttering Feet

If you encounter that the feet move around especially during an idle animation you have to turn on “Foot IK” in the animator component. It is turned off by default.



## Eye colors

There are 17 different eye textures included. You can change them by replacing the albedo / illumination textures in the eye shader.

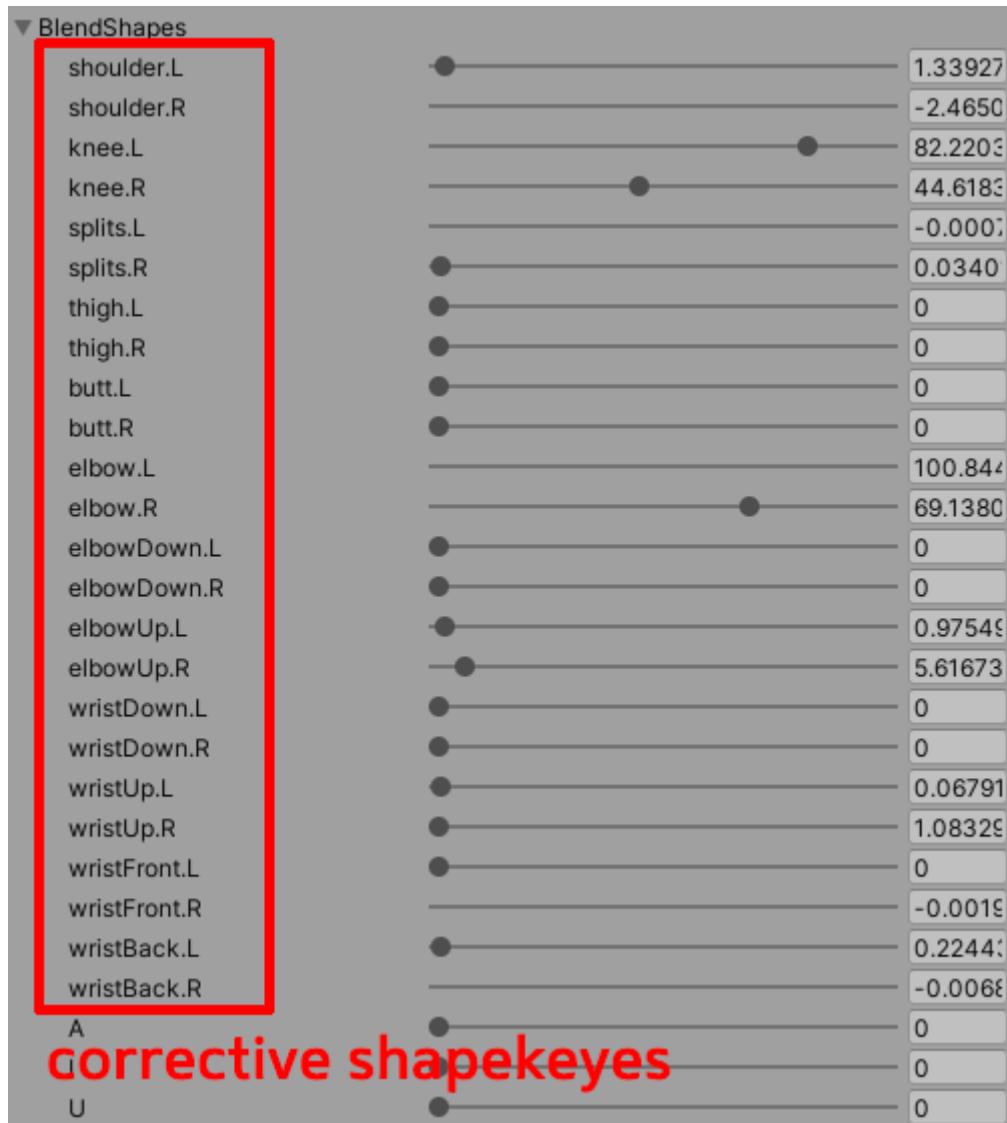
## Hair colors / Eyelashes / Brows

There are 4 different hair textures included. It is **highly recommended** to use a **double sided shader** for the hair / eyelashes and brows. The hair will look much better then. Sadly the standard shader is only single sided.

## Shapekeyes / Blendshapes

This character uses shapekeyes (Unity calls them blendshapes but they are the same) for the facial expressions and as corrective shapekeyes during animation.

Using blendshapes in Unity: <https://docs.unity3d.com/Manual/BlendShapes.html>



The corrective shapekeyes are there to compensate for bad mesh deformation that will always happen during a more extreme bone rotation/bending. In Blender they are setup as drivers that trigger the shapekey if a specific condition is true, e.g. if the forearm is rotated more than 70 degree on the x-axis the elbow shapekey is activated and starts deforming the mesh around the elbow to compensate for the bad deformation.

As the bones move and rotate during an animation various corrective shapekeyes are triggered automatically with a range from 0 to 100%. These shapekeyes are part of the skinned mesh named "BodyMesh".

Shapekeyes are only retargeted to other characters if the skinned mesh has the same name (BodyMesh) and the same shapekeyes. **This is true for my characters.**

If you retarget animations from **another source** however this won't be the case anymore. E.g. another animation file won't have a skinned mesh named "BodyMesh" and or not the same corrective shapekeyes.

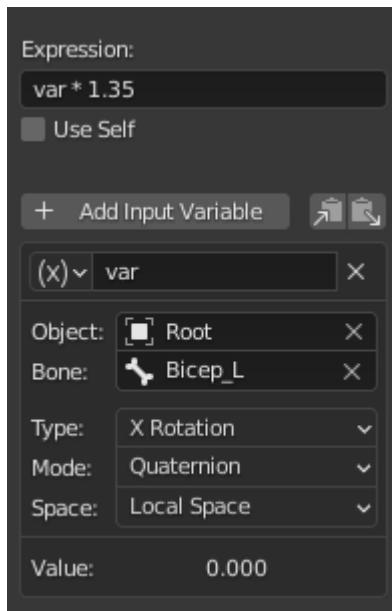
The result will be that the shapekeyes on the target character won't be triggered automatically anymore and thus won't compensate for bad mesh deformation/bending.

Luckily there is a solution for that: You have to access and trigger the corrective shapekeyes by script. (See the link above for accessing shapekeyes via script)

Basically you have to remake the driver conditions from Blender in your script.

For that purpose all the drivers from Blender and their conditions are listed below. Only the drivers for the left side are listed because they are the same for the right side only the bones must be replaced with the right side bones (Biceps\_R instead of Biceps\_L, etc.)

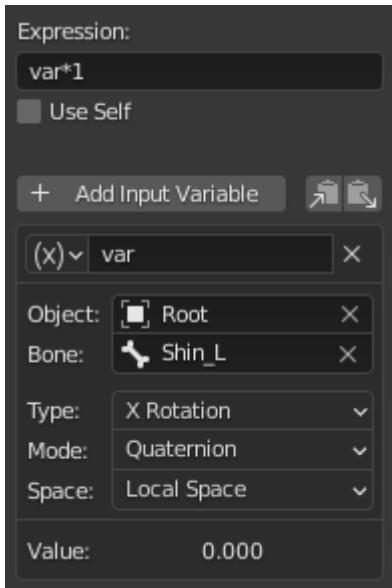
### shapekey name: shoulder.L



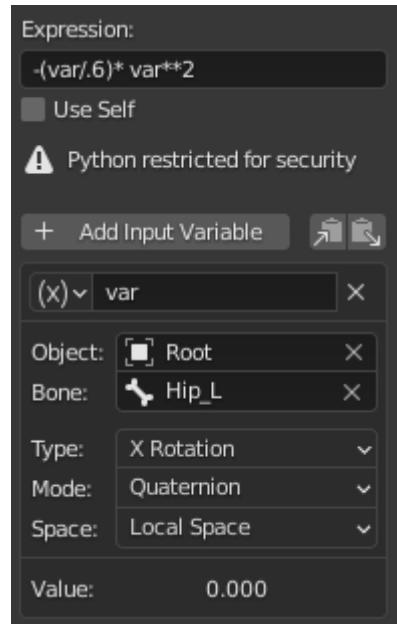
Example explanation: the shapekey shoulder.L is triggered if the bone Biceps\_L is rotated on its X-Axis in local space.

The shoulder.R shapekey is triggered by the bone Biceps\_R accordingly.

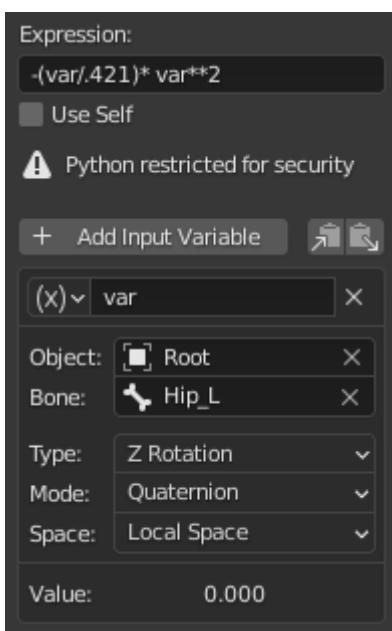
**shapekey name: knee.L**



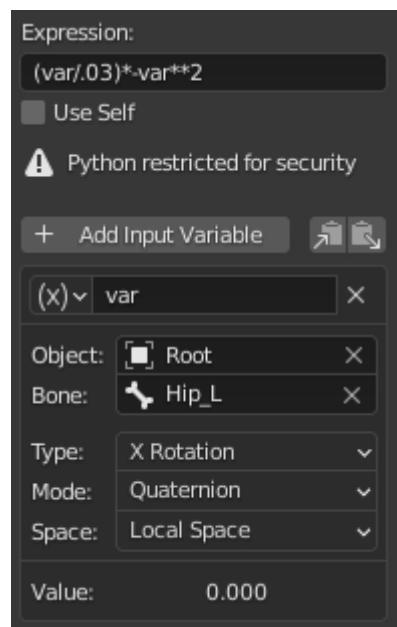
**shapekey name: thigh.L**



**shapekey name: splits.L**



**shapekey name: butt.L**



**shapekey name: elbow.L**

Expression:  
 $\min(1 + (\text{yar}/1.1 \dots (\text{var}/.26) * \text{var}^{**2})$

Use Self

**⚠ Python restricted for security**

+ Add Input Variable

(x) <input type="button" value="▼"/> var	x	
Object:	[ Root	x
Bone:	[ 4arm_L	x
Type:	X Rotation	
Mode:	Quaternion	
Space:	Local Space	
Value:	-0.000	

(x) <input type="button" value="▼"/> yar	x	
Object:	[ Root	x
Bone:	[ Bicep_L	x
Type:	Y Rotation	
Mode:	Quaternion	
Space:	Local Space	
Value:	-0.049	

(x) <input type="button" value="▼"/> zar	x	
Object:	[ Root	x
Bone:	[ Bicep_L	x
Type:	Z Rotation	
Mode:	Quaternion	
Space:	Local Space	
Value:	-0.001	

**shapekey name: elbowDown.L**

Expression:  
 $-(\text{var}/.32) * \text{var}^{**2}$

Use Self

**⚠ Python restricted for security**

+ Add Input Variable

(x) <input type="button" value="▼"/> var	x	
Object:	[ Root	x
Bone:	[ Bicep_L	x
Type:	Y Rotation	
Mode:	Quaternion	
Space:	Local Space	
Value:	-0.049	

**shapekey name: elbowUp.L**

Expression:  
 $(\text{var}/.32) * \text{var}^{**2}$

Use Self

**⚠ Python restricted for security**

+ Add Input Variable

(x) <input type="button" value="▼"/> var	x	
Object:	[ Root	x
Bone:	[ Bicep_L	x
Type:	Y Rotation	
Mode:	Quaternion	
Space:	Local Space	
Value:	-0.049	

**shapekey name: wristDown.L**

Expression:  
`-(var/.34)*var**2`

Use Self

**⚠ Python restricted for security**

+ Add Input Variable  

(x)  var 
Object:  Root 
Bone:  Hand_L 
Type: Z Rotation
Mode: Quaternion
Space: Local Space
Value: 0.000

**shapekey name: wristFront.L**

Expression:  
`-(var/.53)*var**2`

Use Self

**⚠ Python restricted for security**

+ Add Input Variable  

(x)  var 
Object:  Root 
Bone:  Hand_L 
Type: Y Rotation
Mode: Quaternion
Space: Local Space
Value: 0.049

**shapekey name: wristUp.L**

Expression:  
`(var/.34)*var**2`

Use Self

**⚠ Python restricted for security**

+ Add Input Variable  

(x)  var 
Object:  Root 
Bone:  Hand_L 
Type: Z Rotation
Mode: Quaternion
Space: Local Space
Value: 0.000

**shapekey name: wristBack.L**

Expression:  
`(var/.35)*var**2`

Use Self

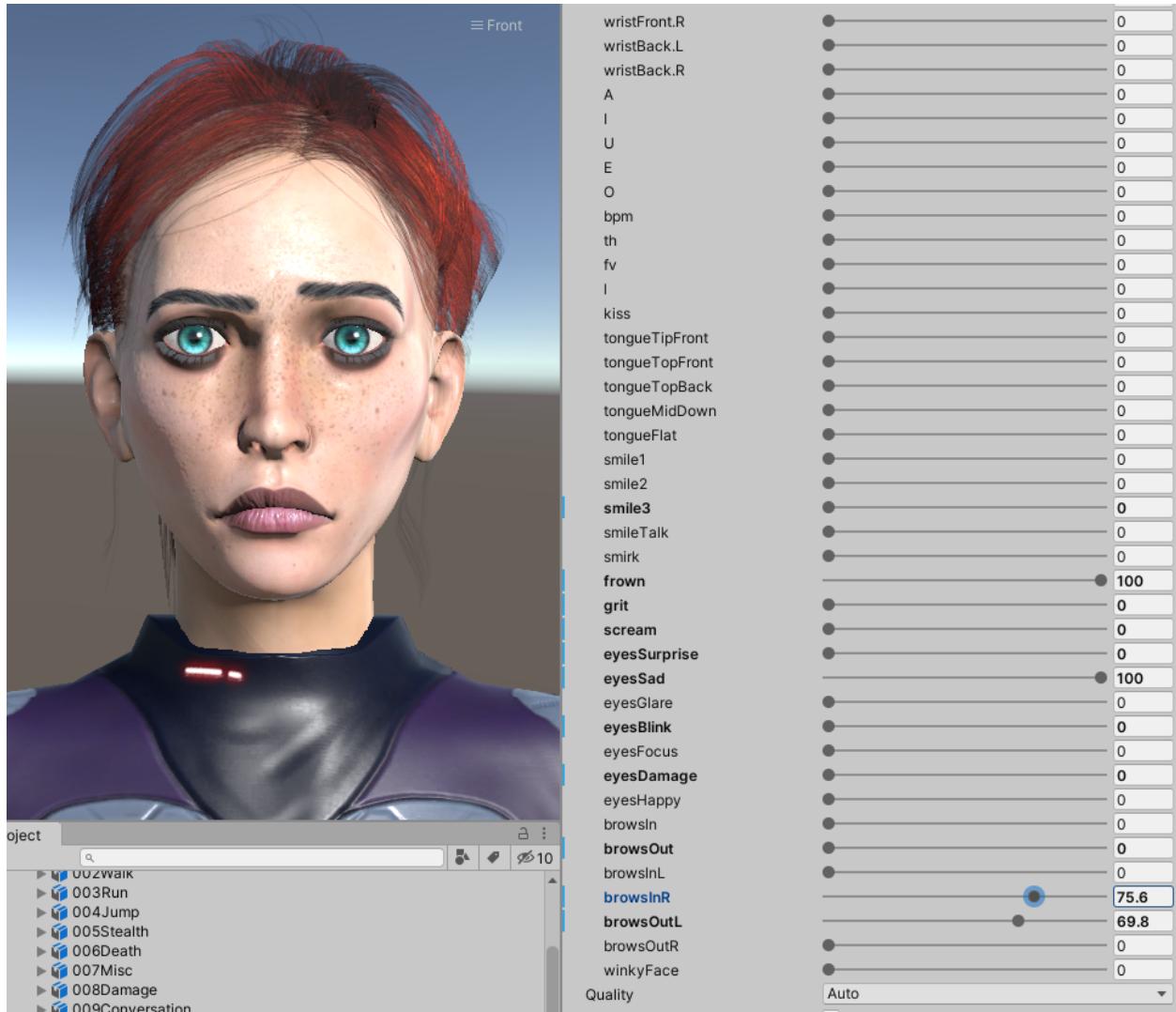
**⚠ Python restricted for security**

+ Add Input Variable  

(x)  var 
Object:  Root 
Bone:  Hand_L 
Type: Y Rotation
Mode: Quaternion
Space: Local Space
Value: 0.049

## Facial expressions

Apart from the corrective shapekeys there are the facial expressions. These must be accessed by script and triggered for e.g. a conversation or something similar. As with all shapekeys several of them can be mixed together to create different looking faces.



## **Compatibility with Mixamo and other 3rd party animation sources**

First I have to repeat again what won't work (see above) with Mixamo/3<sup>rd</sup> party animations:

- only humanoid bones will be retargeted. Any additional bones (boobs/hair) won't get animation data and thus won't be animated
- Shapekeys won't be retargeted.

Apart from that the character can use Mixamo animations or animations from other sources as long as they are humanoid and use Unity's mecanim system.

**However they won't work out of the box and will most likely need some adjustments.**

The following example demonstrates the problem with Mixamo animations and how to get them to work correctly.

- Download an animation from Mixamo (in this example it's a walk animation), import it to Unity
  - in the rig tab: set the rig to humanoid, avatar definition: "from this model"
  - in the animation tab: Anim. Compression: "off", (Loop Time/Loop Pose if needed), Motion/Root Motion Node: <Root Transform>
- drag the animation into the animator right click on it and "set as layer default state"
- Hit play - your character should now play the Mixamo animation

However you'll notice that there is difference between the original and the retargeted animation. In our example its the feet that are messed up.



## **What is causing this and how to correct it?**

The problem is that the Mixamo animation is in a specific pose on the first frame of the animation.

Whereas my character is in a T-pose at the first frame of the animation FBX file.

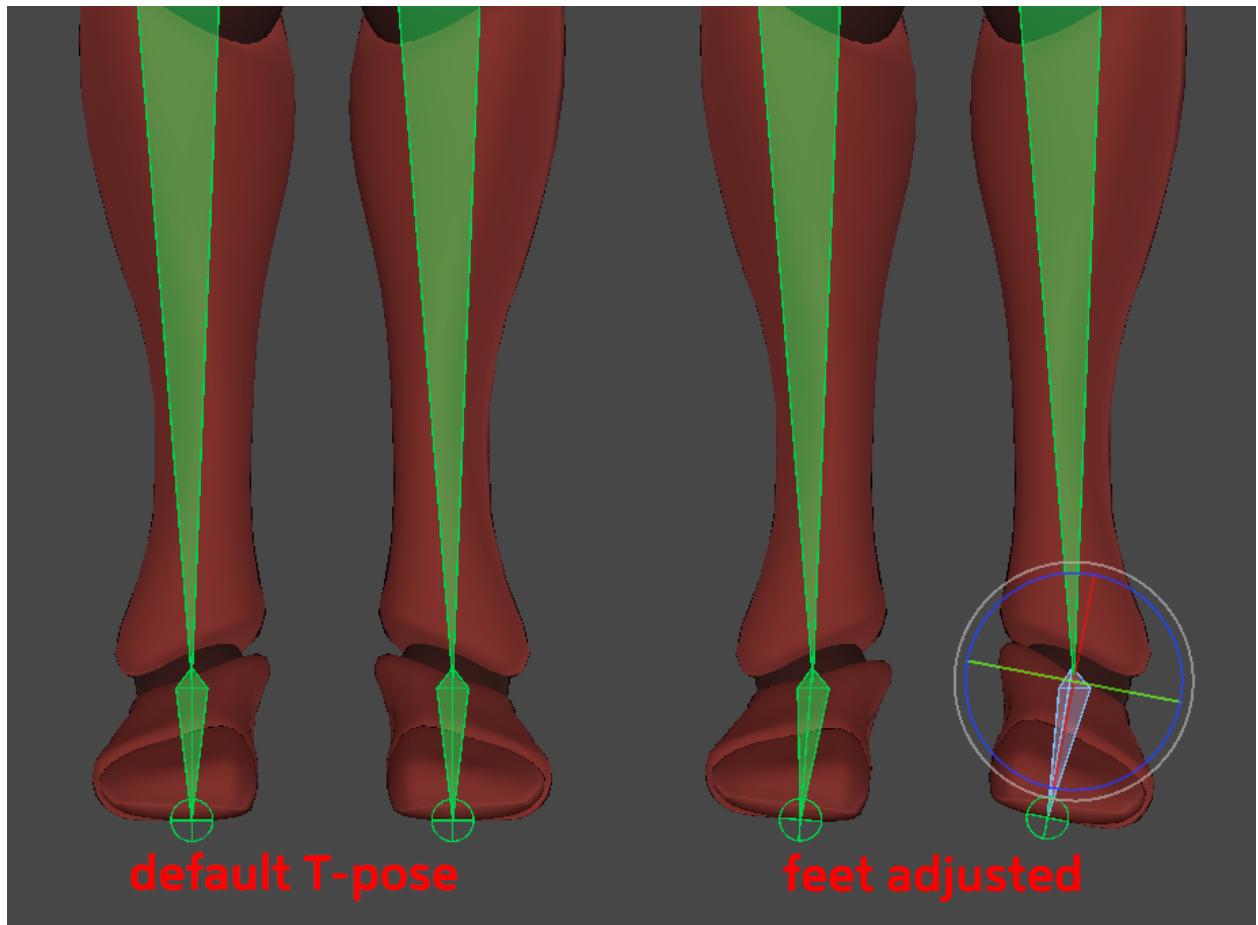
The Mixamo rig is in a T-pose in the avatar configuration (rig tab), too. So theoretically this should compensate for the walking pose on frame 1 but for some reason it doesn't. When the animation is retargeted the target character receives a wrong input and this causes the feet to be tilted left.



## How to fix this?

You have to manually adjust the bones of the Mixamo avatar.

- In the rig tab click “configure” to open the avatar.
- from the front view rotate the feet bones slightly to the left
- Hit apply and done
- press play and look how the retargeted animation looks now
- If it still doesn't look good try again. This is guesswork and may need a lot of trial and error before it is adjusted correctly.





Keep in mind that this isn't limited to the feet. This problem can arise with other body parts as well but the solution is the same as described above.

## Using weapons / equipment on the character

The character animations are designed to work with the weapons / equipment I released so far (and will release in the future). You can ofc use weapons from other sources, however there is no guarantee they will work seamlessly with my animations.

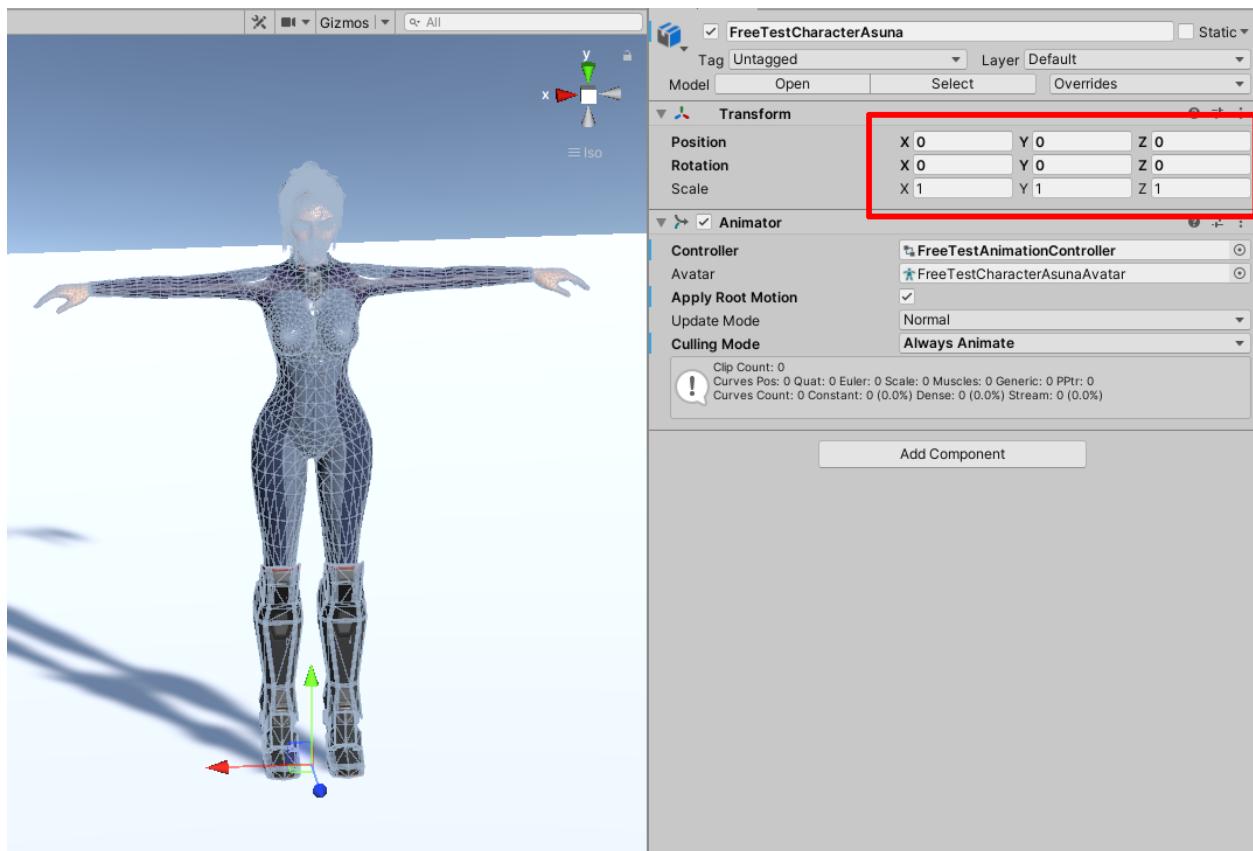
If you inspect the character you'll see that there are many sockets attached to the bones. These sockets are weapon attachment points. They have the correct position and rotation for use with the animations. These sockets can be turned on/off during animations, e.g. holster a weapon: weapon socket on the right hand deactivated, holster socket is activated.



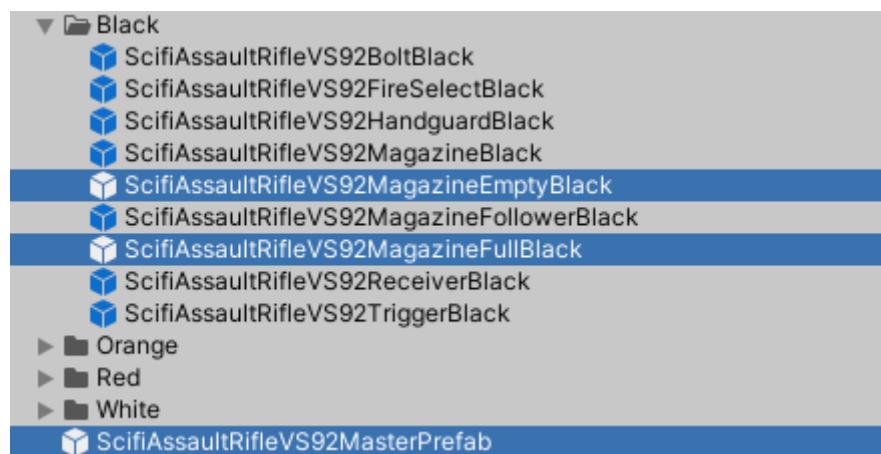
## Setting up the weapons

This example demonstrates the weapon setup with the Scifi Assault Rifle VS92 (not included) but the overall process is the same with all other weapons

1. Import the character and the weapon into your project.
2. drag the character masterprefab into the scene and make sure its position and rotation is 0.

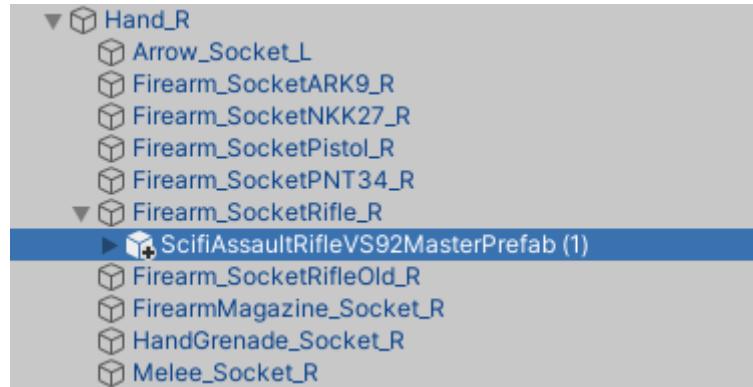


3. drag the “ScifiAssaultRifleVS92MasterPrefab” and the full and empty magazines into the scene and set their position and rotation to 0 too.

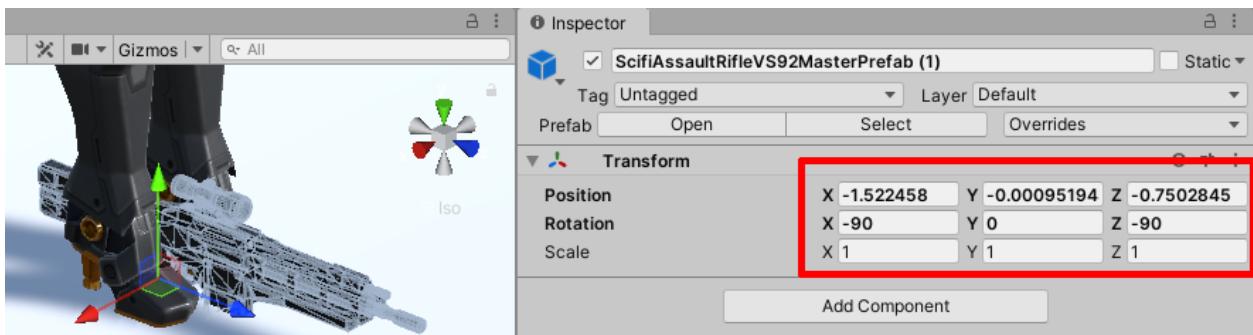


4. make a copy of the rifle masterprefab and the empty magazine (select them and press CTRL+D). You should have now 2 rifles, 2 empty and 1 full magazine in the scene.

5. In the hierarchy expand the character and navigate to the right hand bone. There you'll find the "Firearm\_Socket\_Rifle\_R". This is the attachment point for the rifle on the right hand. Now select one of the rifles in the hierarchy and drag it onto the Firearm\_Socket\_Rifle\_R



The rifle is now parented onto the socket, however its position is wrong. Remember the rifle was set to position/rotation 0. It hasn't changed its actual position/rotation but the values are different now. That is because the rifle is now a child of the socket and its position/rotation is relative to the socket.

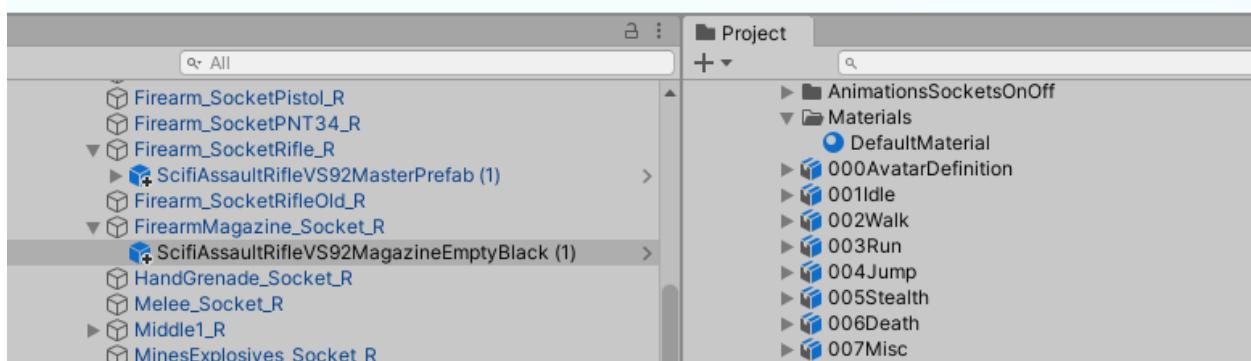


6. With the rifle still selected enter 0 for all position/rotation values. The rifle will change its position and rotation to match the position/rotation of the weapon socket in the right hand. The rifle is now positioned correctly.



7. Now select one of the empty magazines and parent it to the "FirearmMagazine\_Socket\_R" which is also on the right hand bone. Repeat the steps above zeroing out its position/location.

It should look like this now:



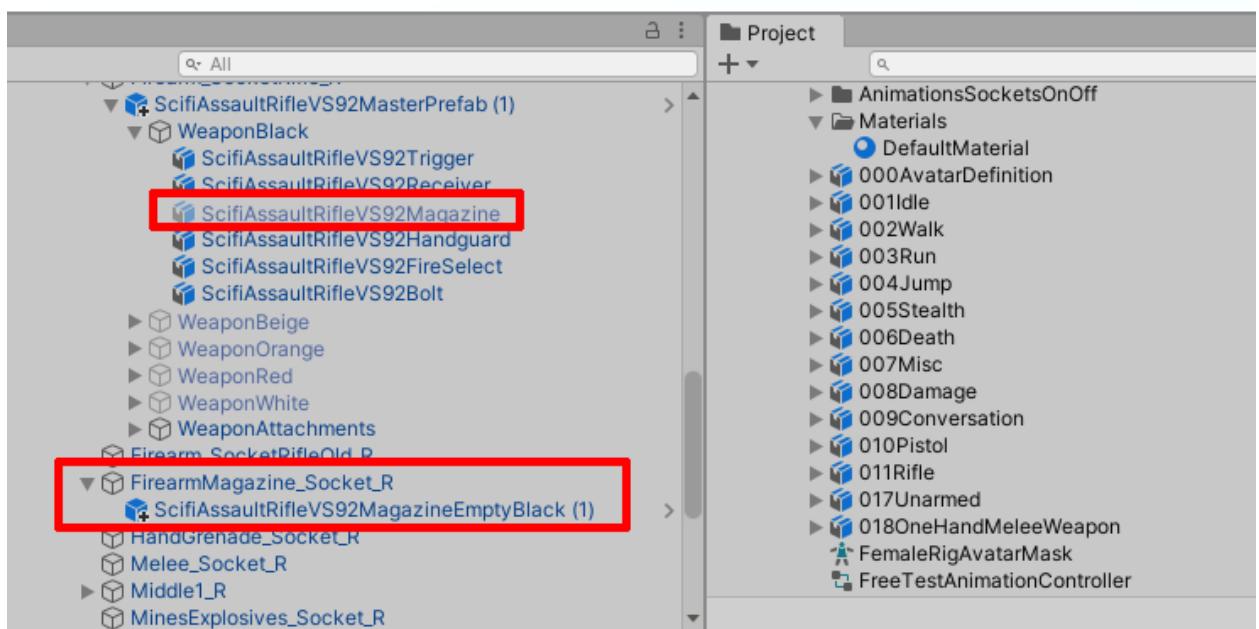
Note that the FirearmMagazine\_Socket\_R is a general socket for all magazines of all firearms attached to the right hand.

8. The empty magazine must be placed correctly now. This means it must be placed where the magazine of the rifle masterprefab is. You can use Unity's automatic vertex snapping tool for that. With the empty magazine selected press and hold "V". Move the cursor over the empty magazine and you'll see a rectangle appear. With "V" still pressed, click and hold the LMB and move the empty magazine over to the magazine of the rifle masterprefab. The magazine will automatically snap to the vertices of the rifle magazine. Move it until the selected vertex of the empty magazine snaps to the same vertex of the rifle magazine. The magazines are now aligned perfectly.

Now you have to **deactivate the magazine on the rifle prefab** (making it invisible): select the rifle magazine and press Shift+Alt+A



**magazines aligned perfectly**



9. All that's left to do is setting up the reload magazines on the left hand and the holster weapon on the back. But that's much easier (repeat steps 5 and 6 from above for each object):

- parent the 2. rifle to the RifleHolster\_Socket\_Back on the spine3 bone
- parent the 2. empty magazine to the ReloadMagazineEmptyRifle\_Socket\_L on the left hand bone
- parent the full magazine to the ReloadMagazineRifle\_Socket\_L on the left hand bone

The rifle will move to the back of the character, the magazines will go into the left hand.

Now all parts are parented to the character correctly and have the correct position for the animations.

## Multiple weapons / magazines on one socket

Depending on your game you can have dozens of weapons parented to the sockets. This will lead to a situation where several weapons / magazines are parented to the same socket, e.g. all rifles will be parented to the "Firearm\_SocketRifle\_R" and all magazines (including the magazines from all other weapon types) to the "FirearmMagazine\_Socket\_R" on the right hand. The same is true for the holster sockets and the reload magazine sockets on the left hand.

If a socket is activated/deactivated during an animation it will always activate/deactivate all parented objects along with it.

This means you have to control what weapons / magazines are turned off and which ones are not. This will most likely depend on what weapon your character has equipped.

You have to do this with a script.



## Reloading/Draw/Holster Weapons

During the reload, draw and holster animations the sockets on the model have to be turned on/off at a specific time to have a seamless transition. The activation/deactivation of sockets has to be controlled with a script.

### Example Draw Assault Rifle VS92:

The VS92 rifle uses the “RifleDraw” animation:

- At the start of the animation the “RifleHolster\_Socket\_Back” must be activated - this means the rifle on the back is visible.
- The “Firearm\_SocketRifle\_R” and “Firearm\_Magazine\_Socket\_R” on the right hand must be deactivated - this means the rifle + magazine in the right hand are invisible.
- When the “RifleDraw” animation is played the weapon switch happens on frame 210 in the FBX animation file “011Rifle”. This frame is called a **contact frame** or **CF**. If this frame is reached during an animation the following happens (and must be scripted):
  - the “RifleHolster\_Socket\_Back” is deactivated - the rifle on the back becomes invisible
  - the “Firearm\_SocketRifle\_R” and “Firearm\_Magazine\_Socket\_R” on the right hand are activated - the rifle + magazine in the right hand becomes visible
- The weapon switch is now complete. The character pulled his weapon from the back and holds it in his hands.

### Example Reload Assault Rifle VS92:

The same procedure applies for the reload animation:

- At the start of the animation the “Firearm\_SocketRifle\_R” and “Firearm\_Magazine\_Socket\_R” on the right hand must be activated.
- The “ReloadMagazineRifle\_Socket\_L” and “ReloadMagazineEmptyRifle\_Socket\_L” on the left hand must be deactivated.
- When the “RifleReload1” animation is played there are 3 magazine switches during the animation:
  - on frame 275 the “Firearm\_Magazine\_Socket\_R” is deactivated and the “ReloadMagazineEmptyRifle\_Socket\_L” is activated. The empty magazine is now pulled out of the weapon (MagOut).
  - on frame 290 the “ReloadMagazineEmptyRifle\_Socket\_L” is deactivated and the “ReloadMagazineRifle\_Socket\_L” is activated. The empty magazine is replaced with a full one (MagReplace).
  - on frame 305 the “ReloadMagazineRifle\_Socket\_L” is deactivated and the “Firearm\_Magazine\_Socket\_R” on the right hand is activated again. The full magazine is inserted into the weapon (MagIn).

You can find all needed bone/socket data below in the **“Weapon Animation Compatibility”** list

## Weapon Animation Compatibility

The following list shows which animations are compatible with the weapons I have published so far. Some animations can be used for multiple weapons. It also shows the **contact frames (CF)** for the weapon/magazine switch during the draw/holster/reload animations.

You can use these contact frames to turn on/off magazines/weapon sockets by script.

There are two contact frame values listed. The first one refers to the frame position in the whole FBX file. The second one (red value) is the frame position after the FBX file is split into separate clips.

The list shows all needed sockets that are used during the animations. Weapons and magazines must be parented to these sockets as described above under “Setting up the weapons”

### Scifi Pistol MNL21 (included in this pack), Scifi Pistol STN33

<https://assetstore.unity.com/packages/3d/props/guns/scifi-pistol-stn-33-159662?aid=1011IGjR>

bone	socket
Hip_R	Pistol_Holster_Socket_R
Hand_R	Firearm_SocketPistol_R
Hand_R	Firearm_Magazine_Socket_R
Hand_L	ReloadMagazineEmptyPistol_Socket_L
Hand_L	ReloadMagazinePistol_Socket_L

PistolHoldInHand (blend only right hand over other animations, e.g. idle, walk, run, etc.)

PistolAim

PistolFire

PistolDraw CF: 205 / 5

PistolHolster CF: 245 / 15

PistolReload1 CF MagOut: 275 / 5 CF MagReplace: 290 / 20 CF MagIn: 305 / 35

PistolReload2 CF MagOut: 325 / 5 CF MagReplace: 340 / 20 CF MagIn: 355 / 35

### Scifi Rifle WLT78 (included in this pack), Assault Rifle VS92

<https://assetstore.unity.com/packages/3d/props/guns/scifi-assault-rifle-vs-92-159656?aid=1011IGjR>

bone	socket
Spine3	RifleHolster_Socket_Back
Hand_R	Firearm_SocketRifle_R
Hand_R	Firearm_Magazine_Socket_R
Hand_L	ReloadMagazineEmptyRifle_Socket_L
Hand_L	ReloadMagazineRifle_Socket_L

RifleHoldInHand (blend both arms + hands over other animations, e.g. idle, walk, run, etc.)

RifleAim

RifleFire

RifleDraw CF: 210 / 10

RifleHolster CF: 240 / 10

RifleReload1 CF MagOut: 275 / 5 CF MagReplace: 290 / 20 CF MagIn: 305 / 35

RifleReload2 CF MagOut: 325 / 5 CF MagReplace: 340 / 20 CF MagIn: 355 / 35

## One Hand Melee Weapon

These animations are unspecific and can be used for all kind of one handed melee weapons, e.g. swords, hammers, axes,...

<b>bone</b>	<b>socket</b>
Spine3	1HMeleeHolsterBack_Socket_L
Spine3	1HMeleeHolsterBack_Socket_R
Spine1	1HMeleeHolsterLeg_Socket_L
Hand_R	Melee_Socket_R
Hand_L	Melee_Socket_L

WeaponHoldInHand (blend left or right or both hands over other animations, e.g. idle, walk, run, etc.)

CombatStance1H  
IdleToCombatStance1H  
CombatStance1HToldle  
DrawWeapon1HFromLLeg CF: 310 / 10  
HolsterWeapon1HToLLeg CF: 340 / 10  
DrawWeaponRHandFromBack CF: 370 / 10  
HolsterWeaponRHandToBack CF: 400 / 10  
DrawWeaponLHandFromBack CF: 430 / 10  
HolsterWeaponLHandToBack CF: 460 / 10  
DrawWeaponBothHandsFromBack CF: 490 / 10  
HolsterWeaponBothHandsToBack CF: 520 / 10  
CombatStance1HBothHands  
IdleToCombatStance1HBothHands  
CombatStance1HBothHandsToldle  
AttackRightHand  
AttackLeftHand  
AttackBothHands

## **Animation Keyframe Range**

File **001Idle** contains all idle animations

```
take 1 Idle: 20 - 80
take 2 Idle2: 100 - 160
take 3 IdleVariation1: 180 - 300 (turn head left)
take 4 IdleVariation2: 320 - 440 (turn head)
take 5 IdleVariation3: 490 - 530 (scratch butt)
take 6 IdleVariation4: 550 - 600 (wipe forehead)
take 7 StandToKneel: 620 - 650
take 8 KneelIdle: 660 - 720
take 9 KneelToStand: 730 - 760
take 10 TurnRight: 780 - 820
take 11 TurnLeft: 840 - 880
```

File **002Walk** contains all walking animations

```
take 1 Walk: 100 - 132
take 2 WalkStrafeRight: 180 - 210
take 3 WalkStrafeLeft: 220 - 250
take 4 WalkStrafeRightFront: 260 - 290
take 5 WalkStrafeLeftFront: 300 - 330
take 6 WalkStrafeRightBack: 340 - 370
take 7 WalkStrafeLeftBack: 380 - 410
take 8 WalkBack: 420 - 450
take 9 Walk2: 460 - 492
```

File **003Run** contains all running animations

```
take 1 Run: 10 - 34
take 2 RunStrafeRight: 50 - 74
take 3 RunStrafeLeft: 90 - 114
take 4 RunStrafeRightFront: 130 - 154
take 5 RunStrafeLeftFront: 170 - 194
take 6 RunStrafeRightBack: 210 - 230
take 6 RunStrafeLeftBack: 250 - 270
take 7 RunBack: 290 - 310
```

File **004Jump** contains all jumping animations

```
take 1 JumpStart: 20 - 45
take 2 Freefall: 50 - 100
take 3 JumpLand: 120 - 150
```

File **005Stealth** contains all stealth/sneak animations

```
take 1 StealthIdle: 40 - 100
take 2 StealthWalk: 110 - 170
take 3 StealthWalkStrafeRight: 180 - 240
take 4 StealthWalkStrafeLeft: 250 - 310
take 5 StealthWalkStrafeRightFront: 320 - 380
take 6 StealthWalkStrafeLeftFront: 390 - 450
take 7 StealthWalkStrafeRightBack: 460 - 520
take 8 StealthWalkStrafeLeftBack: 530 - 590
take 9 StealthWalkBack: 600 - 660
```

File **006Death** contains all death animations

```
take 1 Death1: 20 - 60
take 2 DeadOnGround1: 70 - 75
```

File **007Misc** contains all miscellaneous animations

```
take 1 PickUpItemGround: 20 - 80
take 2 PickUpItemTable: 100 - 145
take 3 PickUpItemShelf: 165 - 210
```

File **008Damage** contains all damage animations

```
take 1 HitFront: 20 - 28
take 2 HitFrontRightSide: 40 - 48
take 3 HitFrontLeftSide: 60 - 68
take 4 HitBack: 80 - 88
take 5 HitBackRightSide: 100 - 108
take 6 HitBackLeftSide: 120 - 128
```

File **009Conversation** contains all talking/conversation animations need to be combined with face shape keyes (e.g. mouth movement)

```
take 1 Conversation1: 20 - 105
take 2 Conversation2: 120 - 180
take 3 Conversation3: 200 - 290
take 4 ConversationNo: 310 - 350
take 5 ConversationYes: 370 - 410
take 1 ConversationMaybe/Don'tKnow: 430 - 460
```

File **010Pistol** contains all pistol animations

```
take 1 PistolHoldInHand: 100 - 110
take 2 PistolAim: 140 - 170
take 3 PistolFire: 180 - 185
take 4 PistolDraw: 200 - 220
take 5 PistolHolster: 230 - 250
take 6 PistolReload1: 270 - 310
take 7 PistolReload2: 320 - 370
```

File **011Rifle** contains all rifle animations

```
take 1 RifleHoldInHands: 100 - 110
take 2 RifleAim: 140 - 170
take 3 RifleFire: 180 - 185
take 4 RifleDraw: 200 - 220
take 5 RifleHolster: 230 - 250
take 6 RifleReload1: 270 - 310
take 7 RifleReload2: 320 - 370
```

File **017Unarmed** contains all unarmed combat animations

```
take 1 UnarmedPose: 30 - 70
take 2 IdleToUnarmed: 90 - 110
take 3 UnarmedToldle: 120 - 140
take 4 PunchRight: 160 - 170
take 5 PunchLeft: 180 - 190
take 6 Kick: 200 - 225
take 7 Smash: 240 - 265
```

File **018OneHandMeleeWeapon** contains all 1 handed melee combat animations

```
take 1 WeaponHoldInHand: 100 - 110
take 2 CombatStance1H: 180 - 220
take 3 IdleToCombatStance1H: 240 - 260
take 4 CombatStance1HToldle: 270 - 290
take 5 DrawWeapon1HFromLLeg: 300 - 320
take 6 HolsterWeapon1HToLLeg: 330 - 350
take 7 DrawWeaponRHandFromBack: 360 - 380
take 8 HolsterWeaponRHandToBack: 390 - 410
take 9 DrawWeaponLHandFromBack: 420 - 440
take 10 HolsterWeaponLHandToBack: 450 - 470
take 11 DrawWeaponBothHandsFromBack: 480 - 500
take 12 HolsterWeaponBothHandsToBack: 510 - 530
take 13 CombatStance1HBothHands: 550 - 590
take 14 IdleToCombatStance1HBothHands: 610 - 630
take 15 CombatStance1HBothHandsToldle: 640 - 660
take 16 AttackRightHand: 680 - 700
take 17 AttackLeftHand: 710 - 730
take 18 AttackBothHands: 740 - 775
```