FIMPOSSIBLE CREATIONS

TAIL ANIMATOR V2

USER MANUAL

About Tail Animator

- Tail Animator is animating chains of transforms (it can be 3D/2D model bones) simulating physical behaviour and can be highly customized

- Package is providing mesh skinning API which can be used to create skinned mesh renderers inside editor or in playmode (runtime)

- Tail Animator component is providing highly customized inspector window (GUI) to help user design desired motion responsively

- Package is providing many example scenes presenting different features which can be unpacked to project with “Tail Animator - New Demo Scenes” unitypackage file

- In package’s demo scenes are used models with dense meshes (not optimized for mobile for example) so package is providing few optimized tail models for custom usage or to be used as rigging reference and it can be unpacked to project with “Tail Animator - Tail Models for custom usage” unitypackage file

- Package is providing tools to convert old Tail Animator V1 scripts to new one, it can be unpacked to project with “Tail Animator - V1 to V2 Converter + Old Examples”

- Tail Animator motion can be combined with skeleton keyframe animations to make it more elastic and smooth

Contact and other links you will find in Readme.txt file

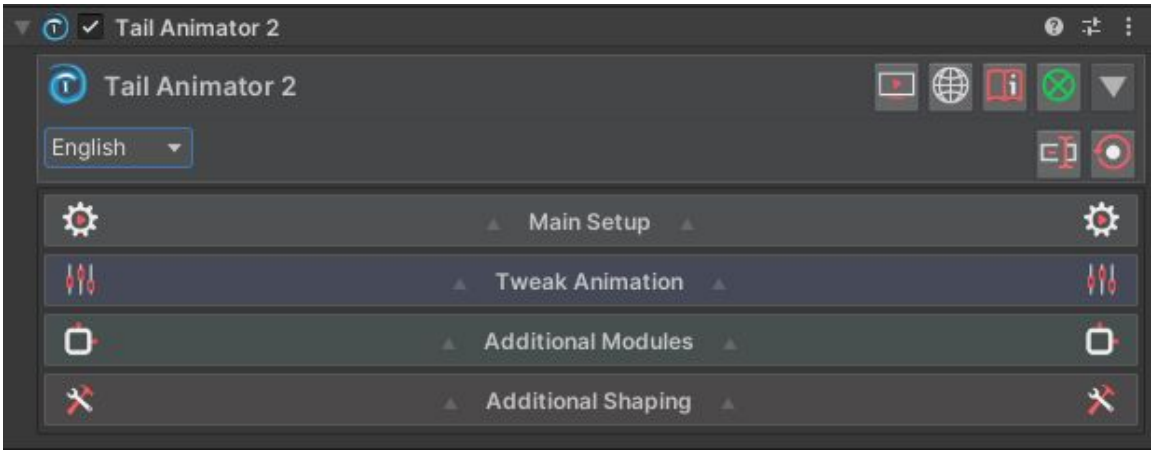
Index

1: Getting Started

Tail Chain:

Tail Chain is list of Transforms (skeleton bones are also transforms) which will be animated by tail animator in playmode.
After adding Tail Animator 2 to your game object algorithm will try to find a skinned mesh renderer and assign the first bone from it as the start of the tail chain.
Algorithm will automatically go through child transforms of the start bone to last one and assign it as end bone of tail chain.
You can freely define other Start Bone and other End Bone than algorithm by using inspector window GUI.

Inspector Window:

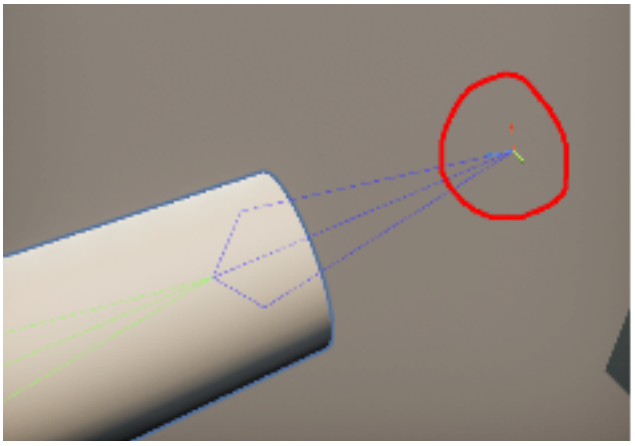


Tail Animator inspector window is divided in 5 foldable tabs: Header, **Main Setup**, **Tweak Animation**, **Additional Modules** and **Additional Shaping**.

Header tab contains buttons with links to my **Youtube Channel**, Asset Store Page, This Manual File, button to **switch drawing gizmos** in scene view, **foldout button** -> **Language Selector**, **Rename Header Title Button** and **Default Inspector Switch**.

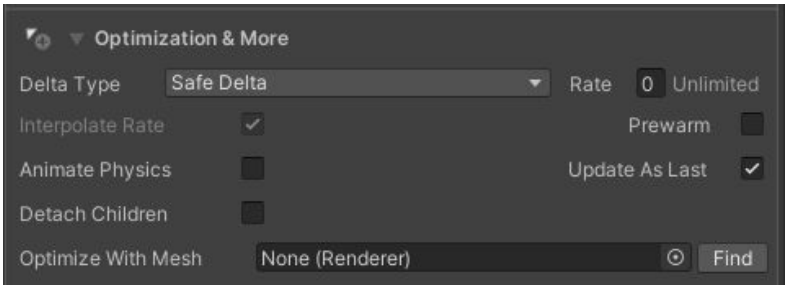
Customizing Tail Chain:

By assigning “Start Bone” Tail Animator will go through child transforms to create a preview tail chain, if it goes through wrong child bones you can assign “End Bone” manually.



When you want to use Tail Animator on single bone then using “**Offset**” toggle next to “End Bone” can be useful, with it you can adjust bone offset for desired bone animation reactions
(visible in scene view)

Optimization Settings and Others:



Tail Animator motion can look different when your game framerate is below 60 FPS or much higher like 1000 FPS.
To make it look the same in different FPS you can set **fixed update rate** with parameter next to “Delta Type”.
When you set this value to 60 then the tail animator will simulate tail motion every 1/60 of second (~0.0166 sec) even if the game FPS is higher like 1000, it can help optimize component a bit. If after setting fixed FPS rate motion is a bit jittery when object is moving you can try enabling “**Interpolate Rate**” (with it optimization will not be noticable)

If Tail Animator is doing some jiggling when game starts and you don’t want it then you can enable “**Prewarm**” which will simulate some update frames on Tail Animator at Start()

When you use Unity Animator with “Update Mode” set to “**Animate Physics**” enable the same named toggle in Tail Animator to synchronize it with keyframed animation.
“**Update as Last**” is queueing the Tail Animator component to be updated in LateUpdate() after other components which are using LateUpdate().
It can be helpful when you use Tail Animator on root bone and then other tail animators inside the chain as branches in this situation root should be updated first and then child Tail Animators, so child Tail Animators should have enabled “Update as Last” and root bone should have it disabled.

“**Detach Children**” will remove child/parent relation from the tail chain and **it can give a boost in performance up to 5 times faster!** But it can’t be used on keyframe animated models.

“**Optimize With Mesh**” will disable Tail Animator algorithm when the selected mesh will not be visible in game / scene view camera.

2: Animation Tweaking

Base Parameters:

Tail Animator V2 is highly dependent on three base parameters:

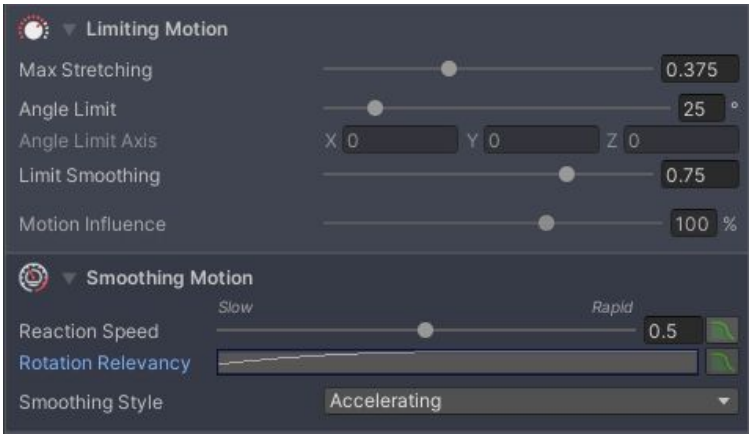
Best practice is to start a new scene, put one of tail model prefabs in scene and play with parameters to get acquainted with them.
By pressing the **green curve icon** you will enable spreading parameter value over each segment of tail chain with curve (for advanced users)

Tail Animator Amount is controlling how much of tail motion should be applied to the model but in most cases you will probably leave it at 100%

Slithery is the most powerful parameter here, it blends stiff - branch like behaviour with bendy/smooth like tail/tentacle motion.
It is very important to know that more bones = more bendy/slithery motion.
When you model have more bones along length then bending will be more sensitive.

When you want tail motion to be more bouncy you can adjust "**Springiness**" parameter but with it you should boost "**Curling**" parameter too to make spring effect less rapid.

Limiting and Smoothing:



Max Stretching is limiting distance between each tail segment, when you set this value high you allow the tail to stretch more and it can provide more smooth effect for tail motion, but sometimes full limiting (to value = 0) can provide much different and interesting motion effect.

With an **angle limit** you can restrict how each segment of tail can bend.
You can also restrict it for selective axis ranges.
Sometimes it can be tricky to set, depending on tail model source hierarchy setup.

Motion Influence can limit or extrapolate motion effect for tail movement in world space, when your character with tail is moving on scene at very high speeds try lowering down this parameter.

Reaction Speed works like Tail Animator V1 "Position Speeds" (if you set "Smoothing Style" to Quick) it is slowing down tail segments position changes in procedural animation and similar effect in "**Rotation Relevancy**" parameter which is doing same process but on tail segments rotations.

Additional Parameters (Experimental):

Sustain will make tail wave for some more time after moving but effect will be erased when using too much of smoothing or too less of “Springiness” parameter also boosting “Springiness” too much can make this parameter react too much so adjust it with care.

When bones chain is dense in bones (have many of them along the model) or there are only a few bones and your model is kinda short then tail motion can look very different in these two extreme situations.
You can adjust it using more “Curling” or lowering “Reaction Speed” but sometimes it can be not enough, then you can open the “Additional Parameters” tab and change the value of “**Unify Bendiness**” parameter.
Unify bendiness will automatically make your tail bend more, but it will be more sensitive for short models with few bones. When you boost up “Unify Bendiness” then you can lower “Curling” and boost “Springiness” to easier shape desired motion.

Limit Axis for 2D will prevent tail segments to move in 3D space so segments will not rotate to depth direction. When you use the “Auto Waving” tail can rotate to depth direction anyway so please adjust “Auto Waving” axis if you need 2D space motion.

3: Additional Modules

Auto Waving:

Auto waving is just automatically and additively rotating tail chain root bone to simulate waving effect over whole tail chain.
“Advanced” **Waving Type** is using perlin noise to generate tail rotation.
“Simple” is using just sinus, rotation axis is dependent on bones source orientations and can rotate in different rotations on different models so please check different axis values when using “Simple” waving.

Collisions:

Enabling collision detection for each tail segment in the chain.
You can try using **World Space Collision** detection which is working on unity's rigidbodies but **I suggest using "Selective"** collision which is more optimal and precise. Selective world collision needs a list of defined colliders to collide with but Tail Animator V2 provides experimental "**Dynamic World Colliders Inclusion**" feature which automatically includes and removes near colliders using trigger collider.
After enabling collision always remember to adjust colliders by "**Colliders Setup**" tab and scene view gizmos.

Collision Slippery is an experimental parameter which can make tail sliding on colliders a bit less.

Collision Reflection will make tail reflect from collider instead of align to it which can be helpful when adjusting collisions with "Slithery" animation tweaking parameter setted high.

Collide with Disabled Colliders can be really helpful when setting up for example clothes with tail animator, you can add unity colliders to character bones and disable it, then include them in colliders list, collision with them will be detected by tail anyway but not by other game colliders. (no need of creating additional layer)

Partial Blend:

Can be used to spread tail animator motion effect over tail segments individually.

Inverse Kinematics:

Enabling Inverse Kinematics (CCD IK) on the tail chain which can be used to have more control over tail animation.

You can assign a **target** towards which tail will rotate and try to reach it with a tip.
You can **blend** IK effect and smoothly disable if needed with this parameter.

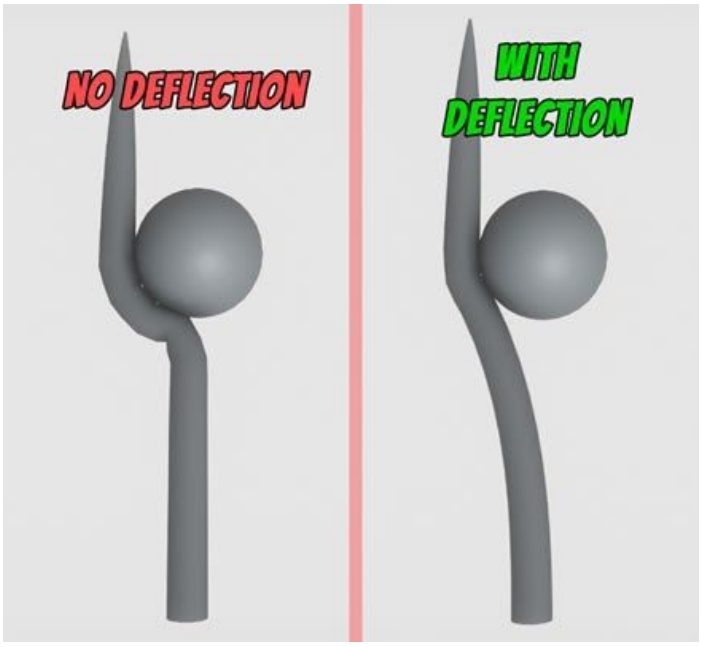
IK Animator Blend will apply some of keyframe animation to IK animation if there is keyframe animation applied to tail chain bones.

Reaction Quality will trigger more iterations of CCDIK every frame to make this reaction faster.
Smoothing will restrict IK to not exceed some limits, sometimes it can help some jitter issues or sometimes it can cause it so please use it with care.

Continuous Solve will make IK follow target in sequence instead of refreshing it's reference pose every frame (like with **Continuous Solve** turned off)
It will provide two much different looking behaviours.

When tail segments are bending too much with IK try adjusting **weights** and **angle limits** to prevent it.

Deflection (In Developement):



Deflection is a feature in development which allows tail segments to have influence over parent segments when child segments are rotated at some angle values.

You can disable **Deflect Only Collisions** so the deflection algorithm will try to make the tail more stiff when **Start Angle** will be exceeded on any tail segment.

Disable when Far:

You can use this feature for optimizations to smoothly disable the tail animator component when it is far from the main camera or custom transform.

Physical Effects:

Use gravity value to simulate world space gravity weight effect.

Wind Effect is an experimental feature to add some outside force to tail animator bones. Use **Tail Animator Wind component** to customize wind effect amount and behaviour.

Additional Shaping:

You can give tail customized additional shape by offsetting root bone rotation, curving each tail segment (can react in a bit different way when setting “Slithery” tweak animation parameter very low or very high) or changing length of whole tail.

4: Mesh Skinning API and User Methods

Editor Tail Skinner:

You can use this component to skin static meshes inside Unity Editor.
It can be used only on simple meshes like Tails, Capes etc. no branching like Spine->Legs->Arms etc.

You can watch this tutorial video to see how you can adjust skinner to your model. After hitting “Skin It” new .asset file will be generated in source mesh directory, thanks to it you can make prefab out of generated object without losing skinned mesh reference (without saving .asset file skinned mesh would be remembered only by scene file, not by project)

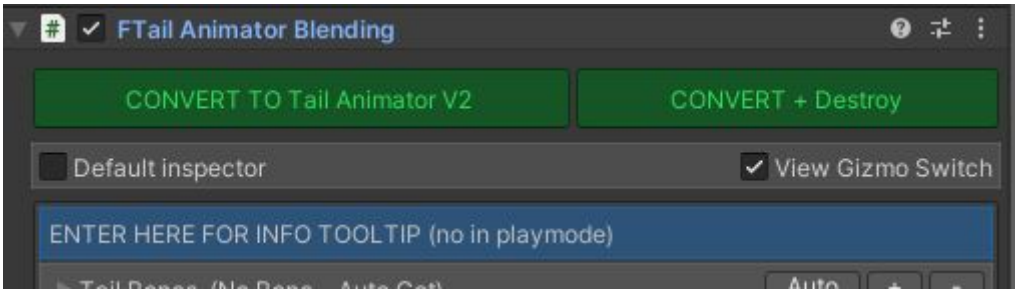
Inside the demo package there are examples how to skin models at runtime with Tail Skinning API, please take a look on “TailDemo_SkinnedMeshGenerator.cs” file if you’re interested in it.

Tail Animator V2 provides methods with “User_” prefix which for example allows to change count of tail segments dynamically.

Take a look on “TailDemo_SegmentedTailGenerator.cs” file and “S_Demo_TailAnimator2D Dynamic Generating” scene if you’re interested in it.

5: Tail Animator V1 Converter to V2

If you was using Tail Animator before V2 release, you can import package **“Tail Animator V1 to V2 Converter.unitypackage”**
Convert buttons will appear above Tail Animator V1 components.



I suggest setting tail animator motion parameters from the scratch after conversion. Tail Animator V2 is prioritizing different parameters but if you want algorithm to work with logics almost the same like Tail Animator V1 you should set “Slithery” value to 1 or 1.2, Curling very low, Springiness Very Low, then under “Smoothing” tab set “Animation Style” to quick, now Reaction Speed and Rotation Relevancy will react in very similar way like “Position Speeds” and “Rotation Speeds” parameters of Tail Animator V1.

If you like this package please visit my asset store page for more or write a review for this asset ;)