Magic !

Smart actions for posing

Instructions

In this product you will find three scripts :

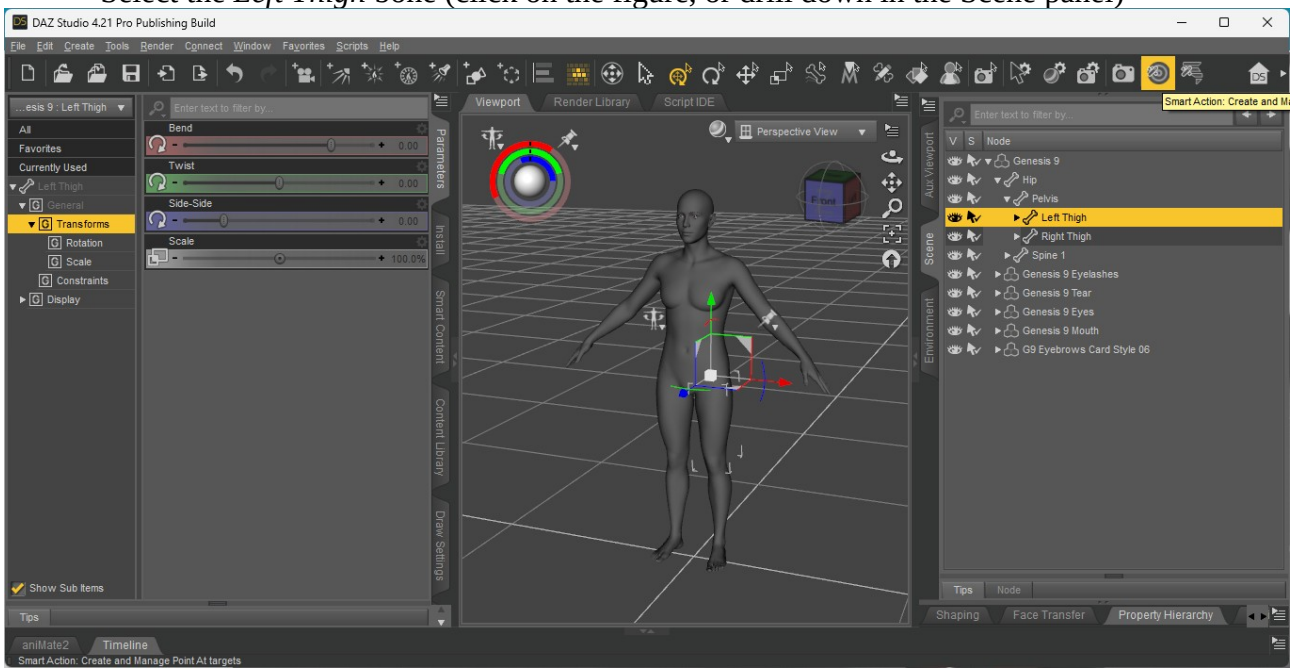
- Install Actions.dse : Creates the Custom Actions for Magic Pointer and Magic Twister, and add them to the Menu and Toolbar
- Magic Pointer.dse : Smart action to create and manage Point At targets
- Magic Twister.dse : Smart action to create and manage Twist/Bend controllers

Part 1 : What is Magic Pointer and what does it do ?

Magic Pointer is a « smart action », a script between a tool and a utility. It is designed to be very versatile, and yet laser-focused in its functionality. It is best used in tandem with the **Universal Tool**.

Here is a basic use cases :

- Load a Genesis 9 figure
- Select the *Left Thigh* bone (click on the figure, or drill down in the Scene panel)

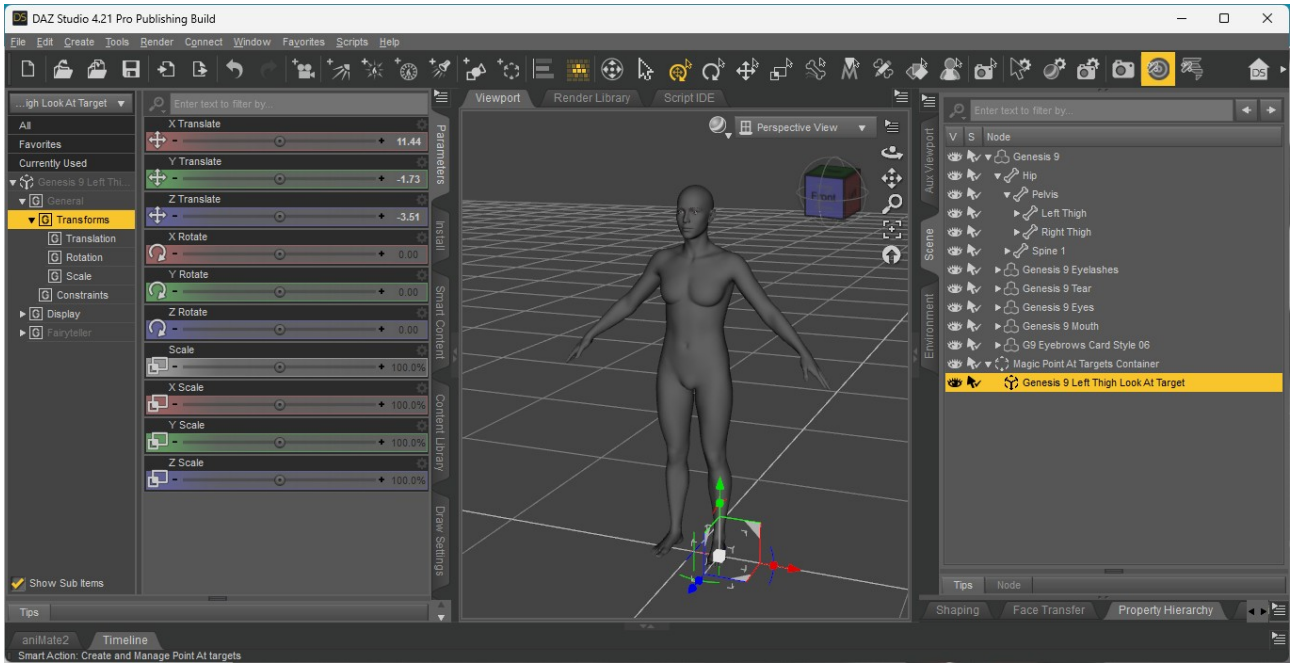


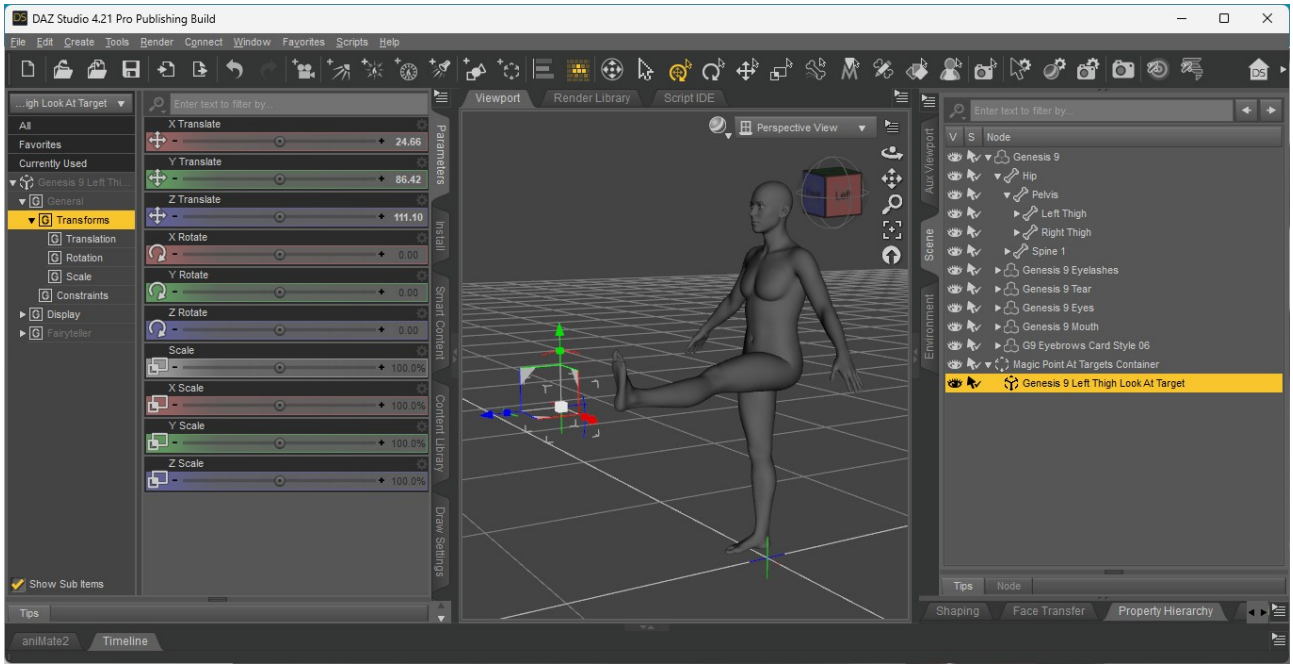
- Press Magic Pointer

You now have two new nodes : a **Group** named « *Magic Point At Targets Container* » and a **Target** Null named « *Genesis 9 Left Thigh Look At Target* »

The **Group** will be the default container for all created null targets. You can safely delete it whenever you want/need, it will be recreated automatically as needed.

The **Target** null was positioned exactly where the *Left Thigh* was pointing before, a short distance away from the bone. The *Left Thigh* has its « Point At » parameter set to point at the Target. Additionally, the *Genesis 9 Left Thigh Look At Target* is now selected, ready to move around with the Universal Tool.

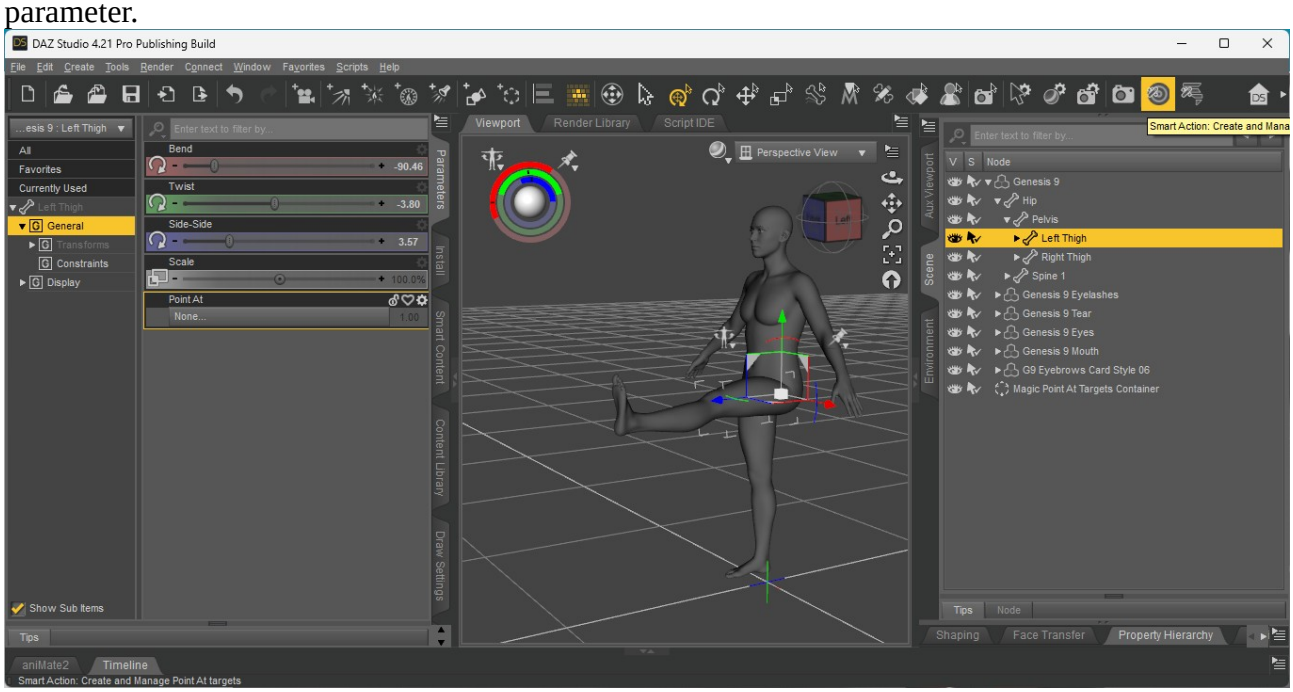




Move the target somewhere, notice that the *Left Thigh* keeps pointing at it.

Baking the Target's position back to the Bone rotations

Press **Magic Pointer** again, this time with the « *Genesis 9 Left Thigh Look At Target* » node selected.
The Target was « **baked** » back into the *Left Thigh*, saving its Rotation and clearing the « Point At » parameter.



The Magic Pointer Action « guesses » what you're trying to do depending on what you have selected. If a bone is selected, it will create a target and place it where it should be so that the pose isn't changed. If a target is selected, it will « bake it back » into the rotation of the bone.

But what if you have a bone with a target, and you select the bone again and press Magic Pointer ? Will it create a duplicate target ?

No ! In this case, Magic Pointer will simply select the target. It's useful when you have a lot of Nulls on your screen and don't know which one is the target you need. You can click on the bone in the viewport, press Magic Pointer and it will switch to the existing target.

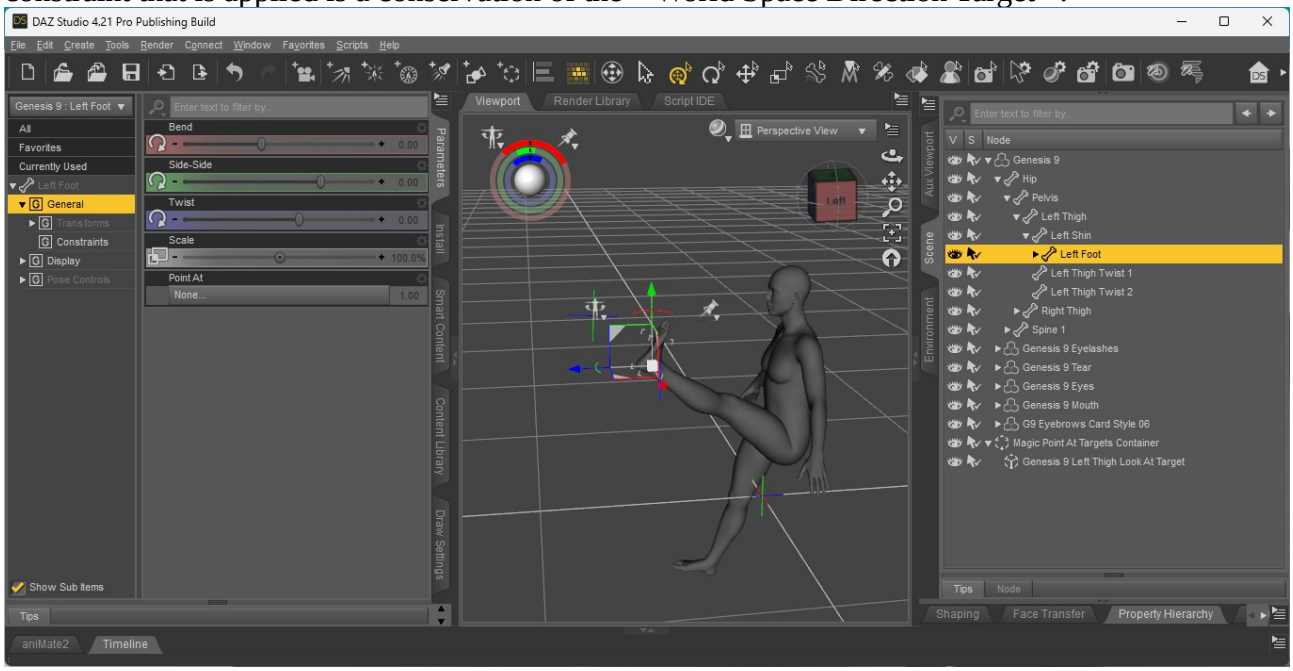
What if I don't like the new bone position ? Can I « reset » or cancel my action ?

You can either **Undo** (Ctrl+Z, Cmd+Z on Mac) the target bake/creation, or simply **Delete** the target node. Both behave slightly differently. In general I use Undo to undo a small change, and Delete the target to restore the bone to its « natural » pose.

It's worth noting that thanks to this Smart Action, **Targets are Cheap**. Create them, bake them, delete them, it's all just one click or one keypress away, and it's always the same button or key. You can create a target just to keep a bone in place while you move another part of the body.
In fact let's do exactly that :

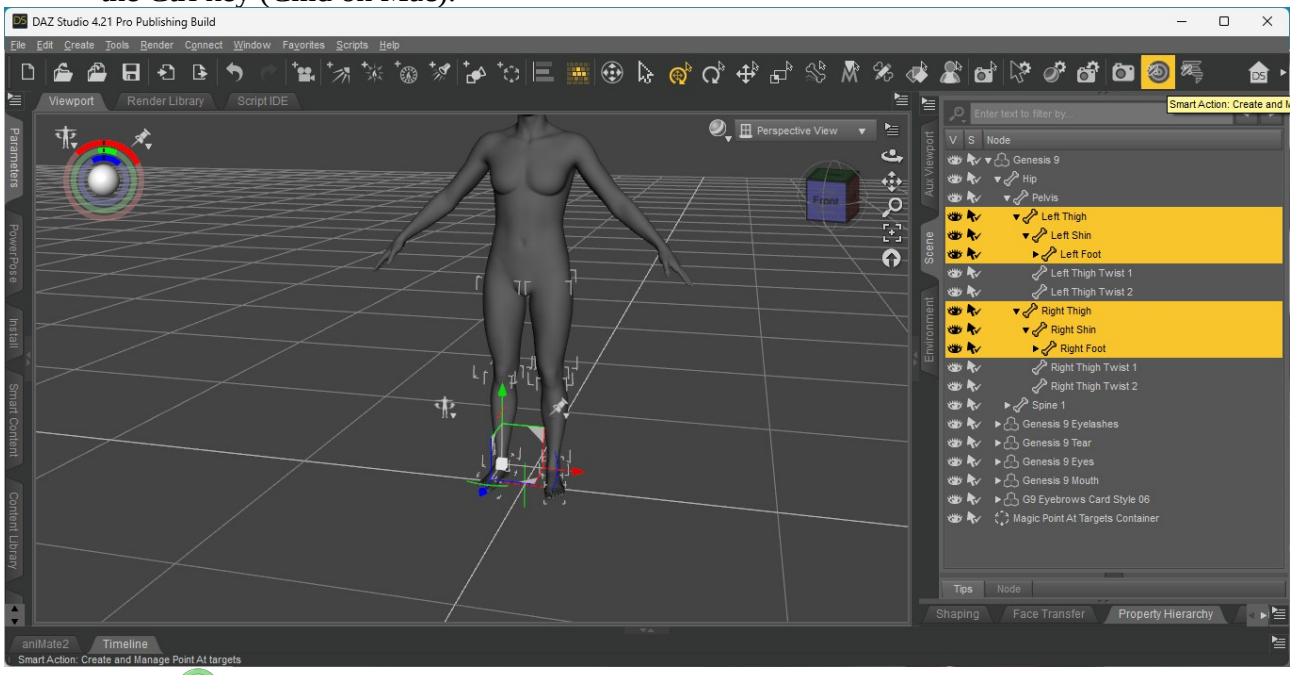
- Make sure the *Left Thigh* has a Target
- Try to Pose the *Left Foot* using the translate option of the Universal Tool

It takes a while to get used to, but as you can probably notice, the movement of the foot behaves differently from the other Posing tools, such as Pins, IK, Puppeteer, PowerPose… here the constraint that is applied is a conservation of the « World Space Direction Target ».



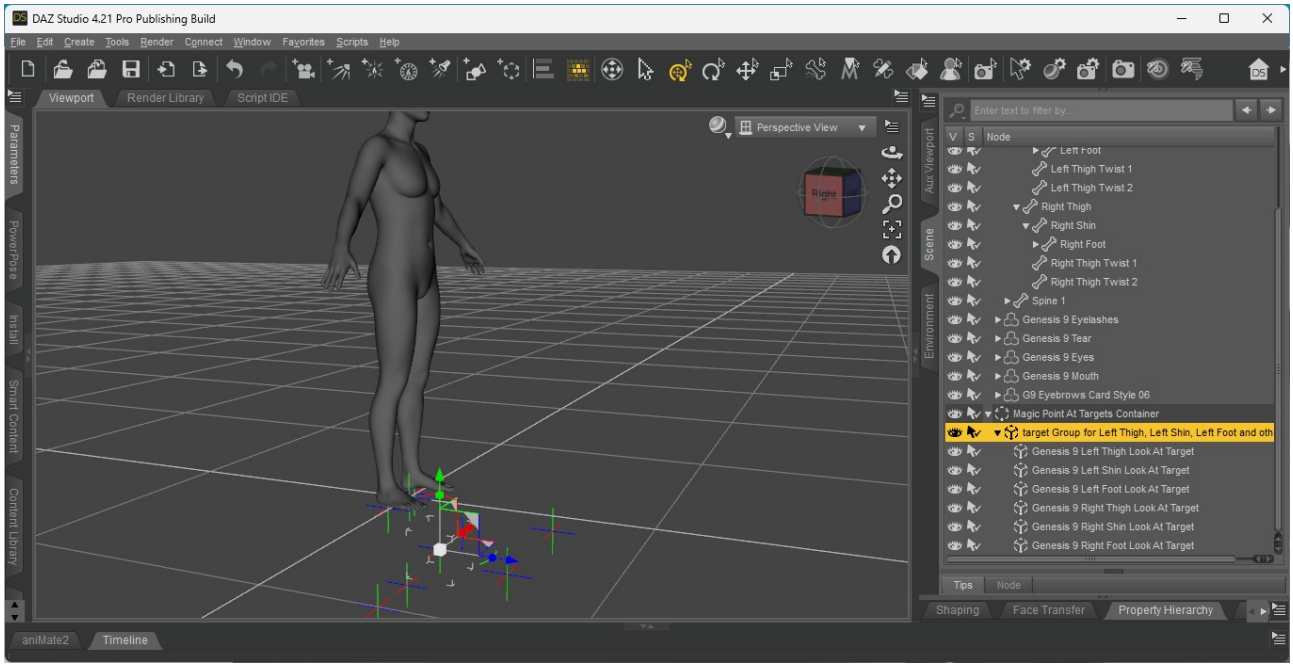
Now let’s talk about my favourite part : Multiselection !

- Select the Left Thigh, Left Shin, Left Foot, Right Thigh, Right Shin, Right Foot, by holding the Ctrl key (Cmd on Mac).

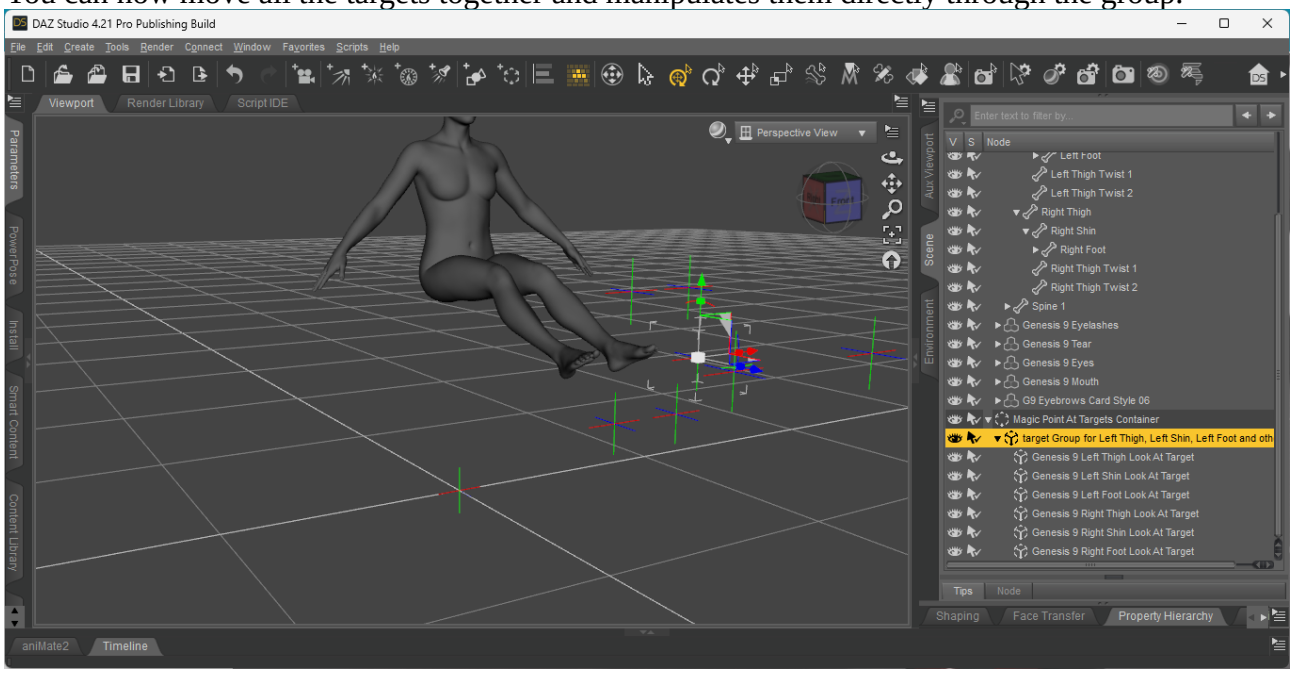


- Press Magic Pointer

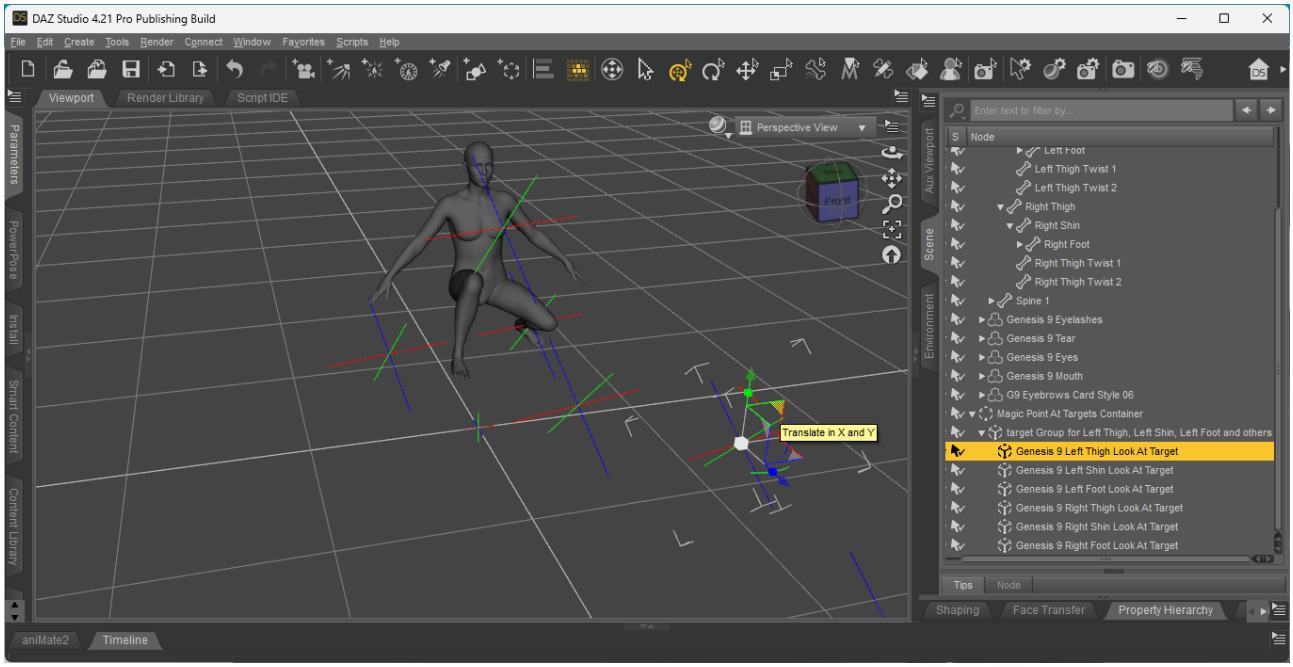
This action now creates a target for each selected bone, and additionally creates a Target Group named « Target Group for Left Thigh, Left Shin, Left Foot and others », positioned at the Center of all the created targets, sets the Target Group as the parent, and Selects the Target Group.



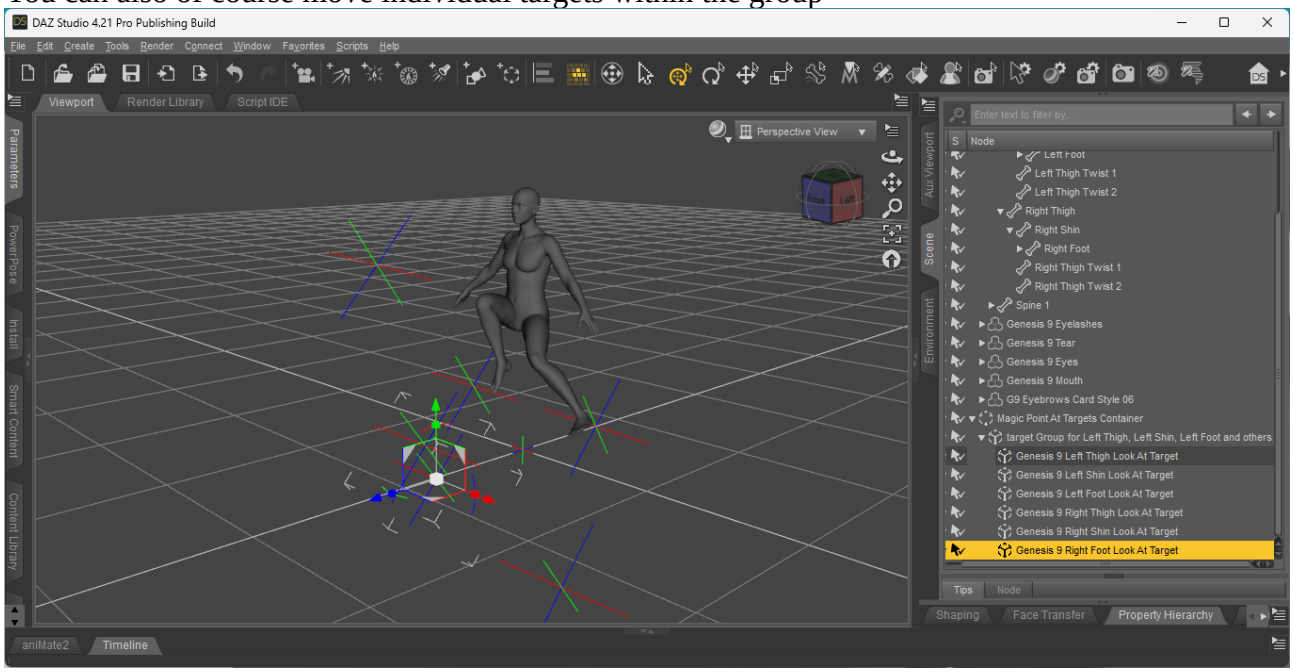
You can now move all the targets together and manipulates them directly through the group.



You can now also alter the Target Group's Scale and Rotation to move the individual bone targets away from each other, closer to each other or spread in a variety of directions.

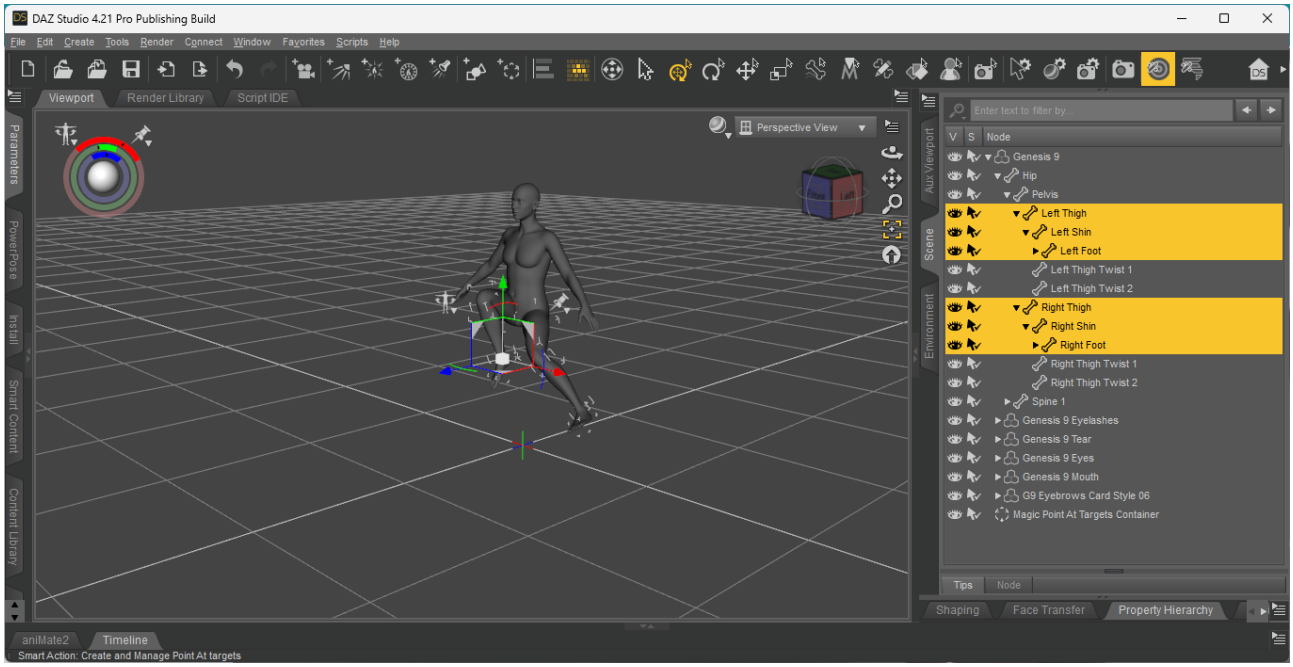


You can also of course move individual targets within the group



Keep playing with it until you see a pose you like, then select the group and press Magic Pointer again.

Every bone’s position and rotation is baked back in ! Now you can export your pose and apply it to another character. Or keep posing and playing with it.

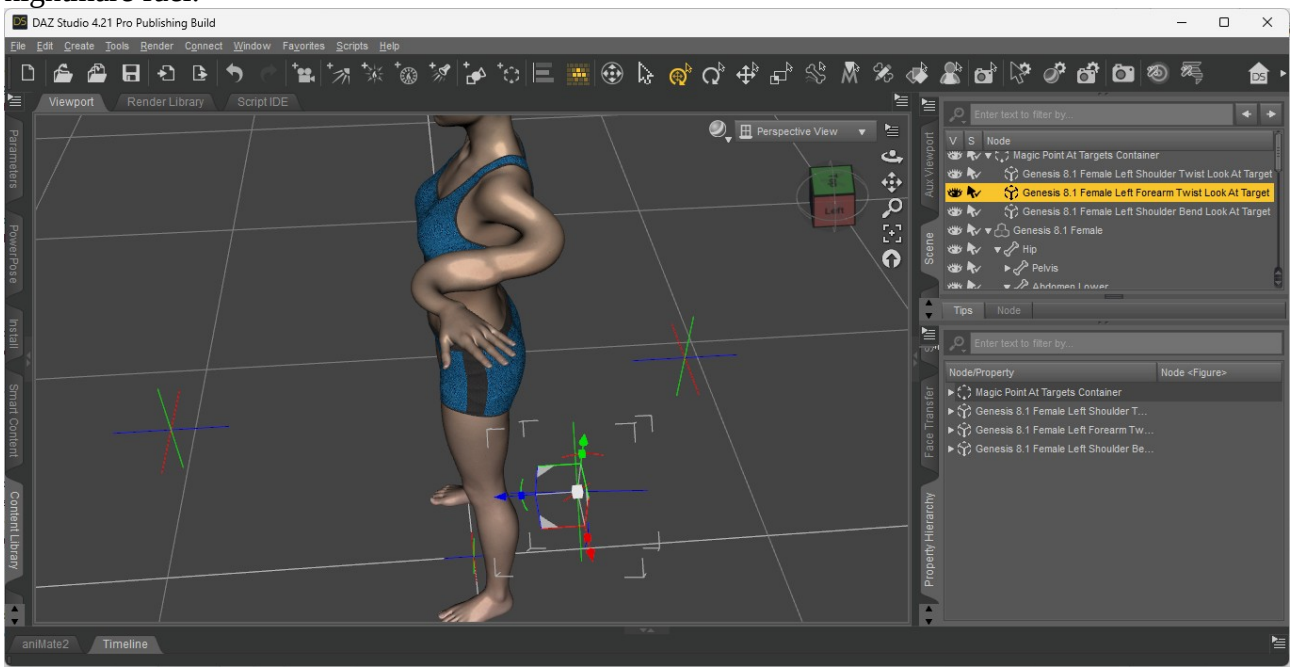


The way the Magic Pointer « guesses » what to do with multiple selections is pretty basic. If the selection has no target, it will create and/or switch to Targets of the selected nodes. If the selection is made only with targets or target groups, it will bake the targets back to the nodes that are looking at them. If it’s a mix of targets and nodes, Magic Pointer will simply do nothing.

Experiment away !

One caveat : Twist Rotations

First, one notice for Genesis 8/8.1 figures : Don’t use Magic Pointer on Twist Bones. It’s nightmare fuel.

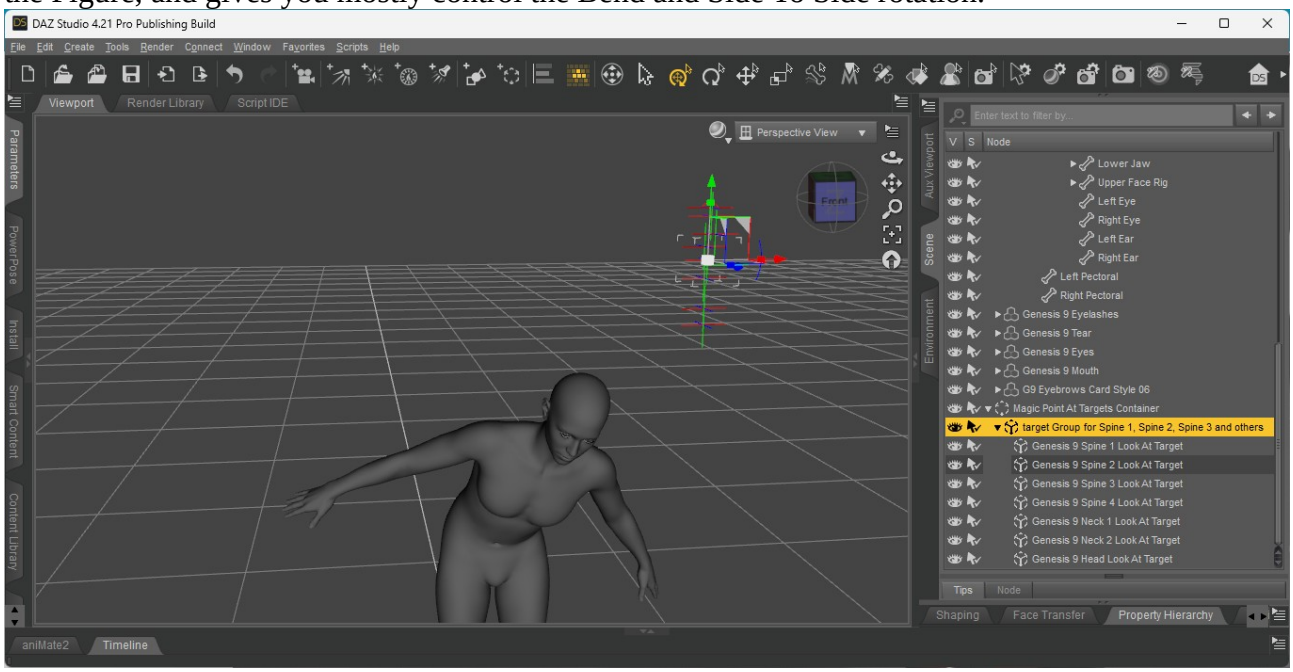


For everything else, these Point At Targets will greatly enhance your workflow, but they are not perfect. One thing they can’t do on their own, is control the Twist rotation of a bone. When moving

bones around, you'll notice that Twist value is difficult to predict and control. Sometimes you feel like you can do it, othertimes it does not do what you'd like.

The way it works is… complicated. For limbs or eyes, it works well, since these bones are naturally aligned with how our brain understands the concept of a target.
Some bones in the Genesis 9 figure however are not in the right direction : they are Vertical.

These bones are the Pelvis, the various parts of the Spine/Abdomen, the parts of the Neck, and of course the Head. If you use Magic Pointer on them, you'll see the natural targets are placed Above the Figure, and gives you mostly control the Bend and Side To Side rotation.



Moreover, while these bones have a target, you cannot control their Twist Rotation parameter at all. This is a hard limitation on Magic Pointer.

Fortunately, there is a process to solve that problem.
Unfortunately, the process is long, involved and complex with a lot of advanced manipulations, and it is difficult to clean up and export a pose.
Fortunately, there is **Magic Twister**.

Part 2 : What is Magic Twister and what does it do ?

To put it in simple terms, some bones are « Horizontal » which means you want to mostly use the Bending and Side to Side rotations to point at a target (example : eyes, arm, leg, hand, finger, foot, toes), but other bones are « Vertical », which means you want to mostly use the Twisting and Bending rotations to point at a target (example : head, neck, spine, pelvis).

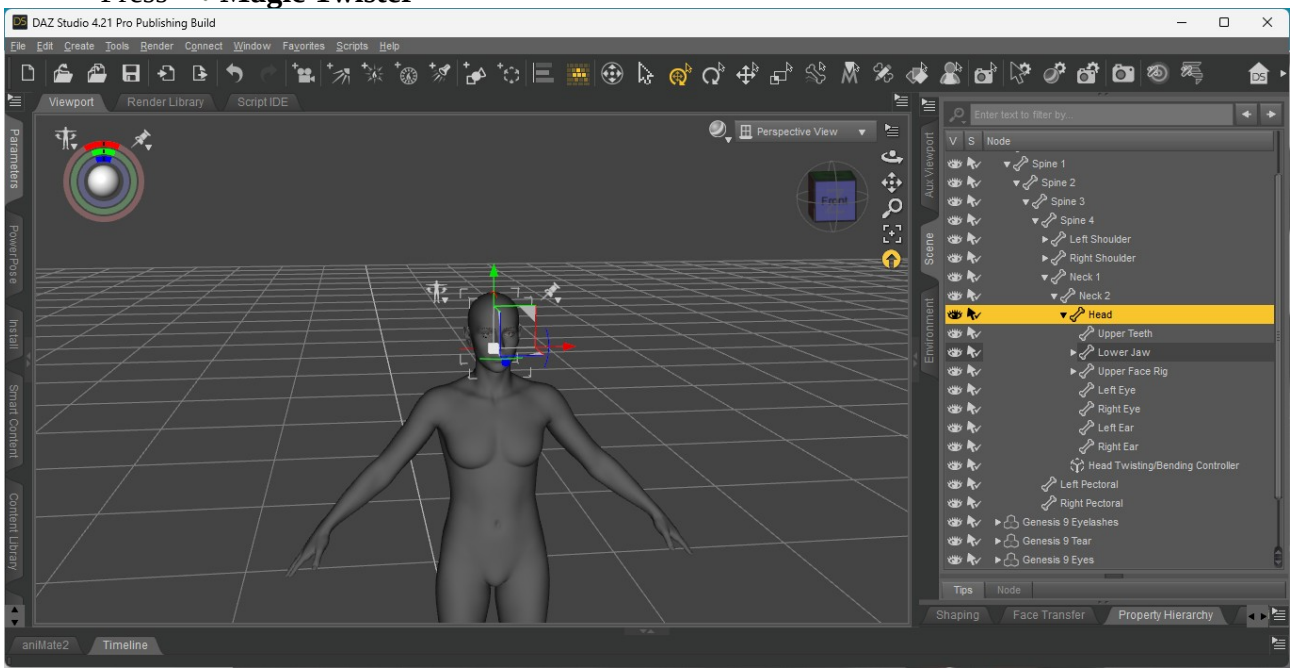
The Point At property, under the hood, mostly controls the Bending and Side to Side rotations, by design. So you can't always use it to properly control the Head.

The official approach to this difficult problem involves a complex process : you create an invisible « controller node », positioned properly, with proper Horizontal bone alignment, and you bind its Side to Side Rotation to the Twisting rotation of the bone you want to control.

If you want to learn more about it, check out this video from the official Daz 3D youtube channel : https://www.youtube.com/watch?v=FOPzgexT54I

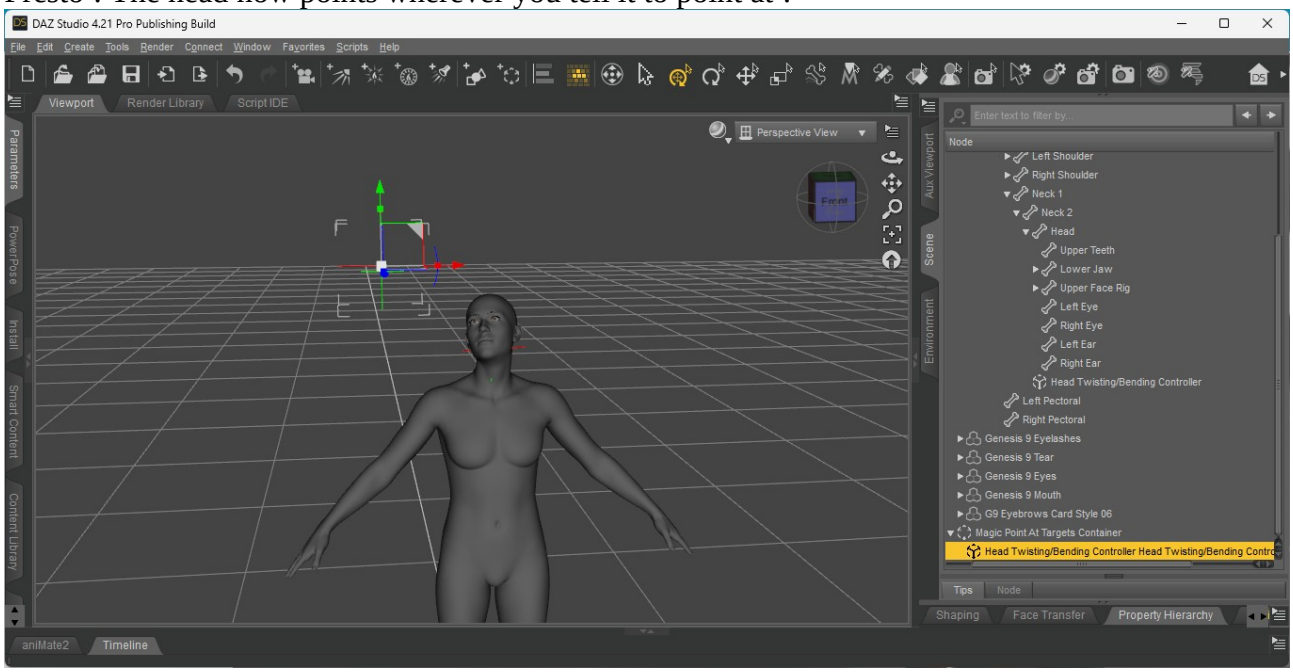
Magic Twister makes this 5-10 minutes process instant, reversible, bakeable, with near real-time updates, and – of course – handles multiselection. Let's get started !

- Load a Genesis 9 figure
- Select the *Head* bone
- Press **Magic Twister**



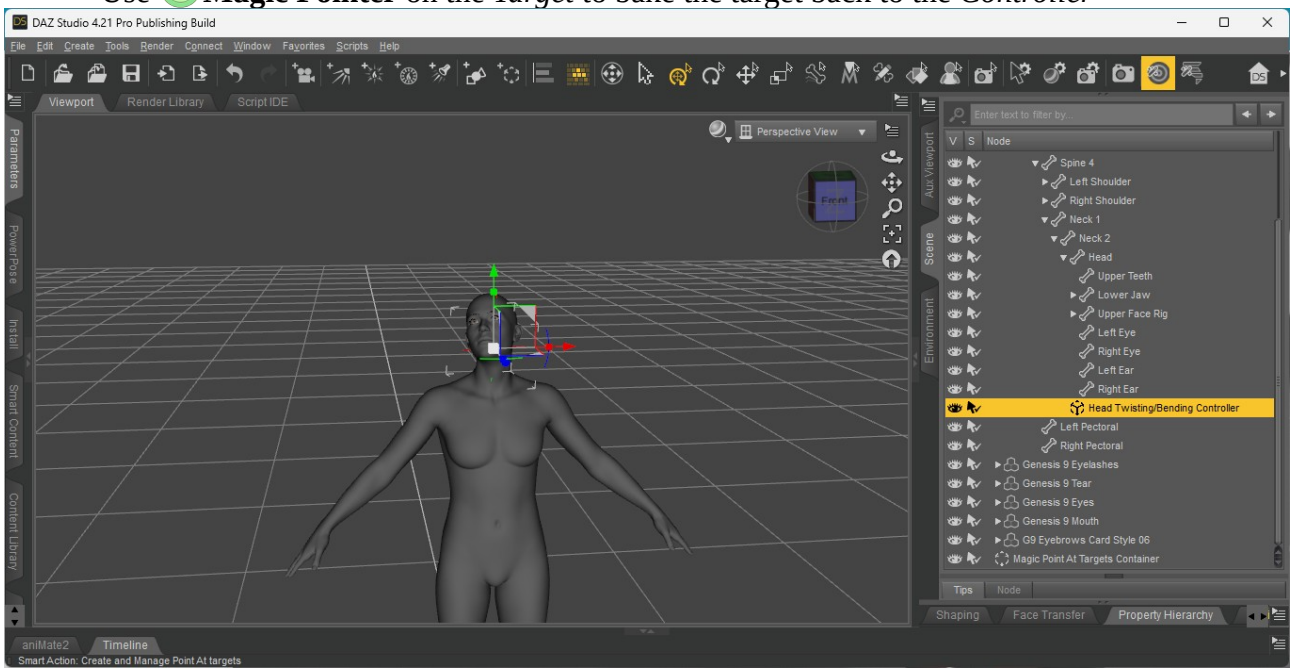
Look at the Scene node hierarchy : The *Neck 2* bone now has an additional Child bone : a Null node named « *Head Twisting/Bending Controller* ».

Select the **Controller**, and press **Magic Pointer** !
Presto ! The head now points wherever you tell it to point at !

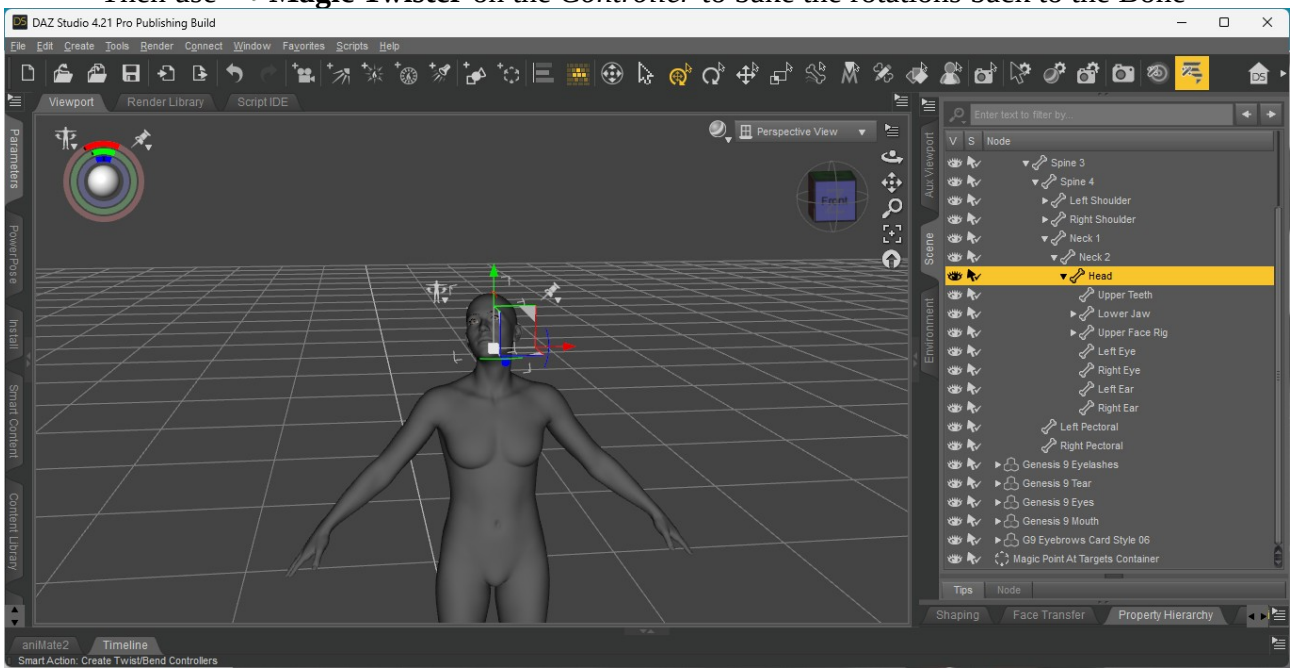


But what if I want to export the pose to another figure ? I don’t want to keep this controller.
This is a simple two steps process :

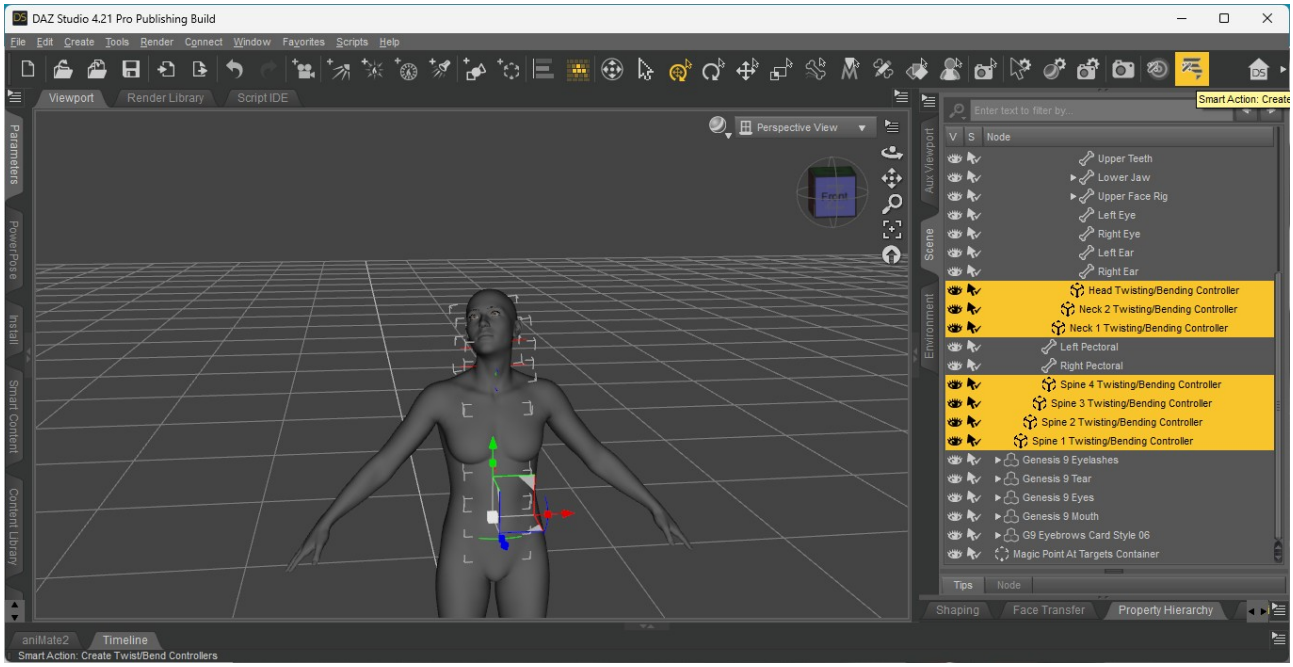
- Use **Magic Pointer** on the *Target* to bake the target back to the *Controller*

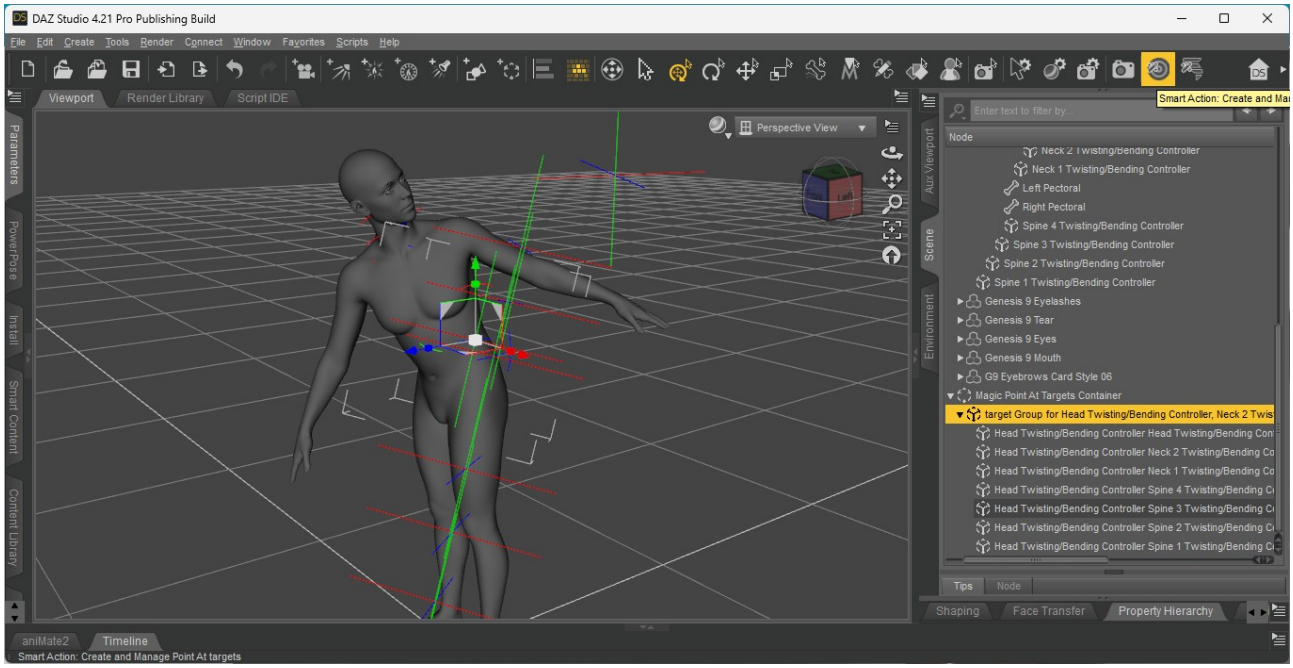


- Then use **Magic Twister** on the *Controller* to bake the rotations back to the Bone

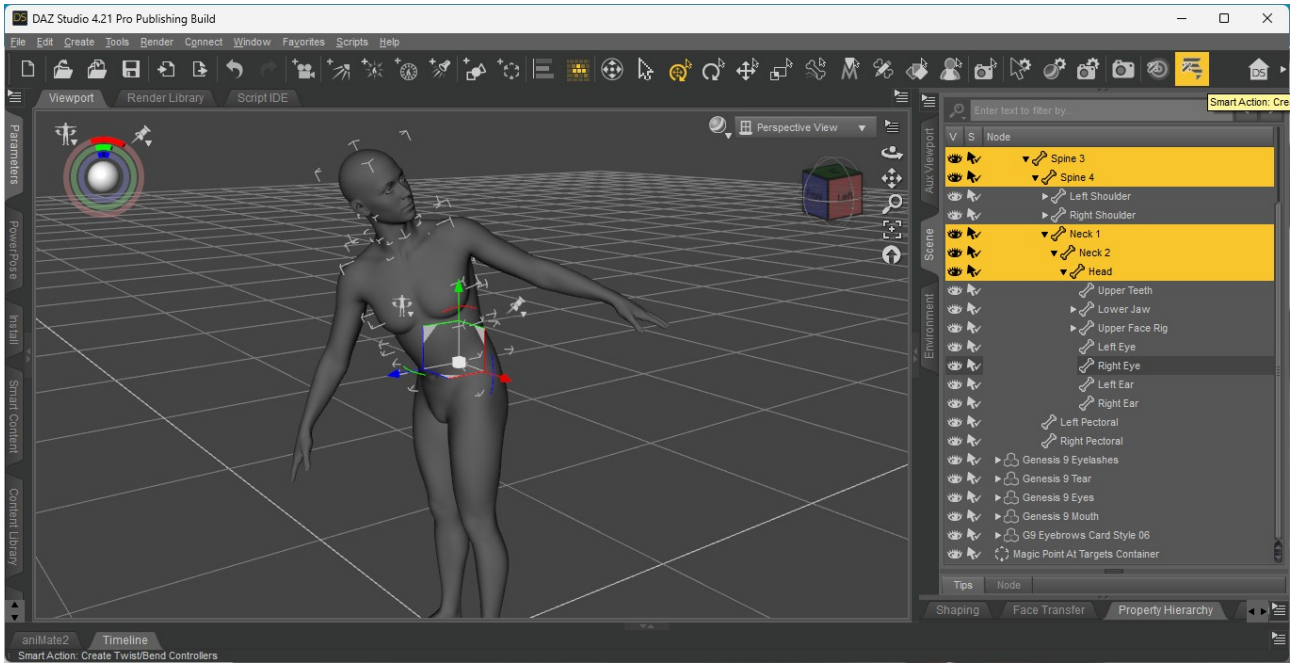


Of course it all works with multiple selections and all the goodness.





And you can of course multiselect bake the *controllers* back to the bones, too



Now you know pretty much everything about **Magic Pointer** and **Magic Twister**.

Get Posing !

Appendix : Technical details and algorithms used, for the nerds out there :-)

Magic Pointer

position of the target

The position of the target is determined using the following algorithm :

- First we normalize the vector between the Origin and End Point of the node
- We multiply this vector by 50 units and add it to the End Point. This gives the « baseline » position for the target.
- We store the initial Bend and Side to Side rotations of the Node
- We set the Point At property on the Node to the Target
- For performance reasons during reverse look-up, we set the « is pointed at by » property on the target to the Node
- This initial position works fine when the rotations are zero, but becomes slightly or greatly incorrect if the rest of the skeleton also has heavy rotations. This is what the final step is for : we correct up to 100 times the position of the target so that the Bend and Side to Side rotations of the Node both fall within 0.001 degrees of the initial rotations.

The Twisting rotation is never checked and assumed to be a constant. This can cause problems, so don’t hesitate to use other methods and tools to perfect your pose.

Target Groups

The target group is always placed at the barycenter of all the other targets. Every target has the same weight. It is worth experimenting and finding your own favourite groups. I like using one group for each individual leg, one for each individual arm, one for each « row » for fingers (eg : one group for the first bones of all fingers of the left hand).

Scaling and rotating target groups give great results.

Magic Pointer works well on props and other figures. Make sure you avoid twist bones though, they generally give pretty bad results.
For some props, it can be worth using the Joint Editor to position the origin and end point where you'd like them. For a sword prop for instance, you can place the Origin at the handle and End Point at the tip.

 **Magic Twister**
position and direction of the Controller
The position is the same as the Origin of the Node
We determine the Forward Vector using :

signX is +1 or -1 depending on the sign of (nodeEndPoint.x – nodeOrigin.x)
signY is +1 or -1 depending on the sign of (nodeEndPoint.y – nodeOrigin.y)
The forward Vector is then obtained with the formula in pseudocode :

```
fwVec = node.getOrientation().multVec([[signX, 0,0],[0,signY,0],[0,0,1]][node.getRotationOrder().secondAxis]) ;
```

The Rotation Order for the Controller is then set to ZYX

We then create two ERC Links on the Node, one on the Xrotate property with the Xrotate of the Controller set to « Add », and one on the Yrotate property with the Yrotate of the Controller also set to « Add »

Clean-up and Update Callbacks
Every Controller you add with Magic Twister registers four callbacks on the callback stack :

magic_kinemagics_onControllerAboutToBeRemoved
magic_kinemagics_onControllerAboutToDelete
magic_kinemagics_onControllerAboutToDestroy
magic_kinemagics_onControllerTransformChanged

- three of those callbacks are dedicated to cleaning up everything when the controller is removed/deleted, so those ERC Links don't needlessly pollute the parameters of the figure, and so the callbacks don't stay forever in the application callback stack.
- one callback is called whenever the Controller's position and orientation changes, as an Event which will be executed once the application has some spare time. It simply « refreshes » the rotations of the controlled node. This is useful during posing, to get an « almost real-time » feedback when you pose other parts of the scene that indirectly influence the controller's rotation.

These Callbacks are not saved with the Scene and need to be rebuilt each time. The Magic Twister script rebuilds everything automatically the first time you press the button – it's good practice to press it once after a Scene is loaded.

Even if loading the save and using Magic Twister to rebuild the controllers *should work*, when in doubt : bake everything, and save.