Actual VRChat SDK3 Documentation HERE:

https://docs.vrchat.com/docs/playable-layers

Adding Playable Layers

Click the Default layers to add your own instead
You'll probably only need to use the Gesture, Action, and FX Layers

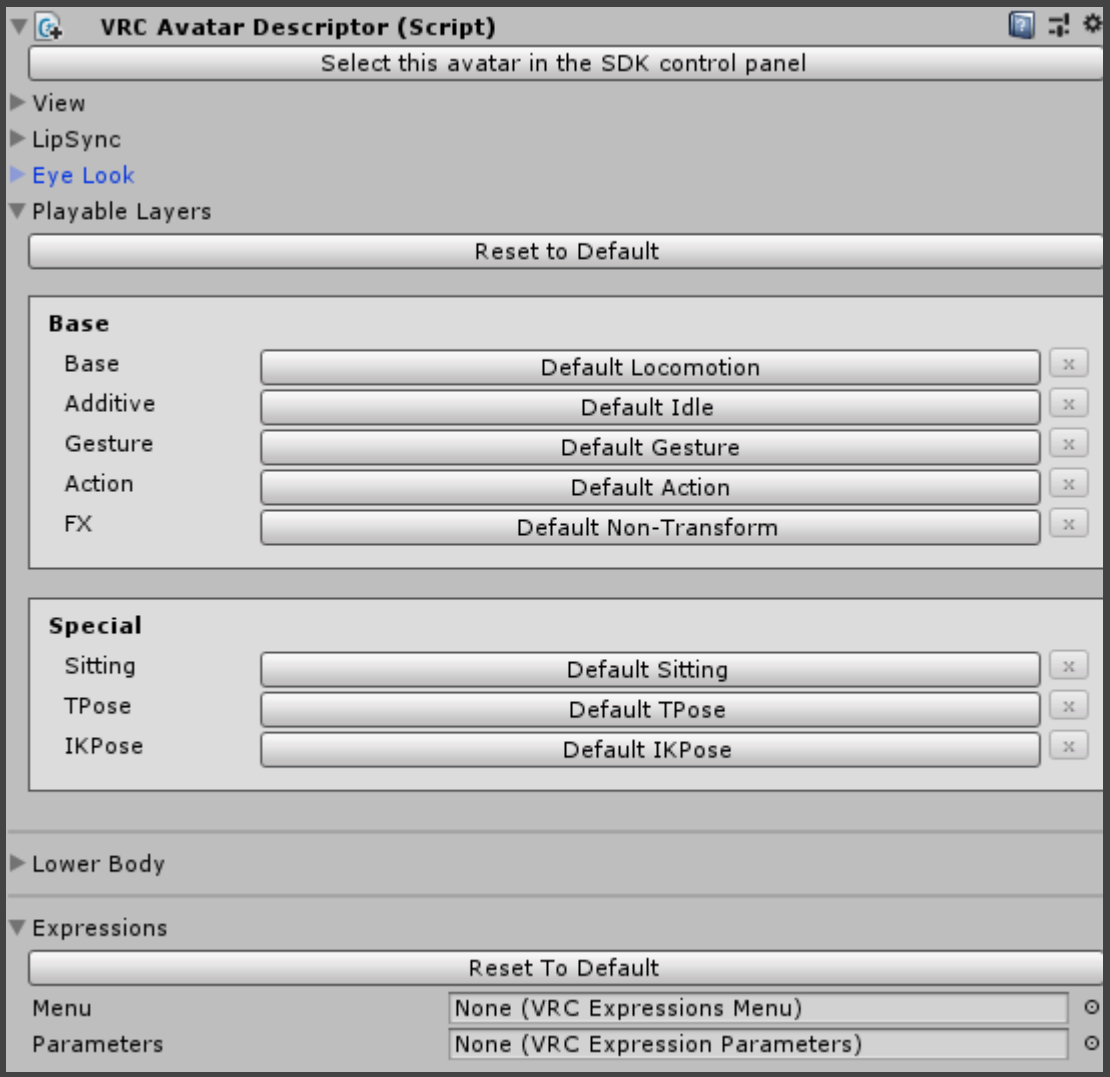
Locomotion is difficult, don't touch it unless you know what you're doing

The Idle layer is for doing additional animation on top of the base animation, and can be complicated, it's easiest to skip it for now

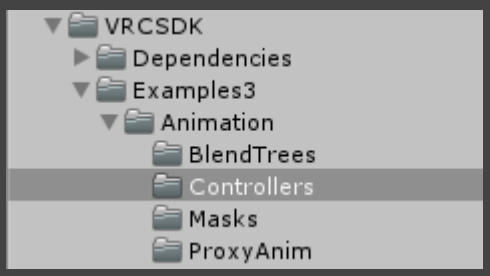
Gesture is like the Idle layer, but it overrides it instead. It's how the hands work, by default.

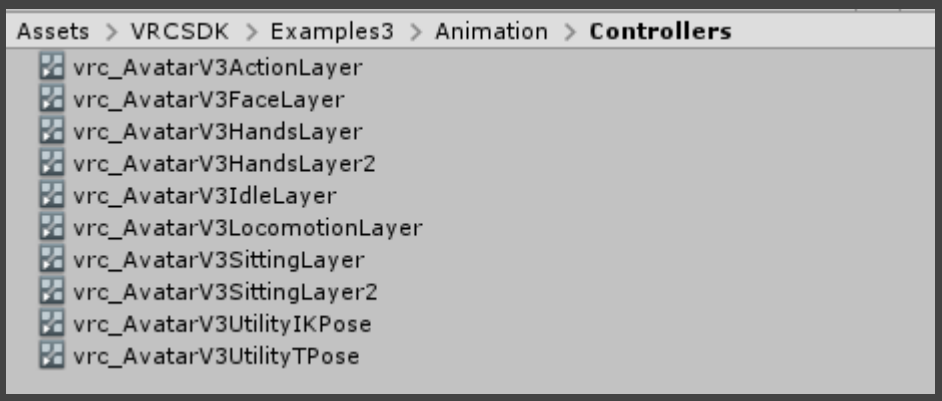
Action overrides the entire animation, but is off by default. This is where all your emotes go

FX is the special layer, because it's for everything that isn't actual bone animation. This one's the big one for all the blendshapes etc.



Head to The examples in the VRCSDK folder to see how the defaults are set up, and give you a basis to work from





For this example, we'll grab the action layer, since we're going to be doing our new emotes!

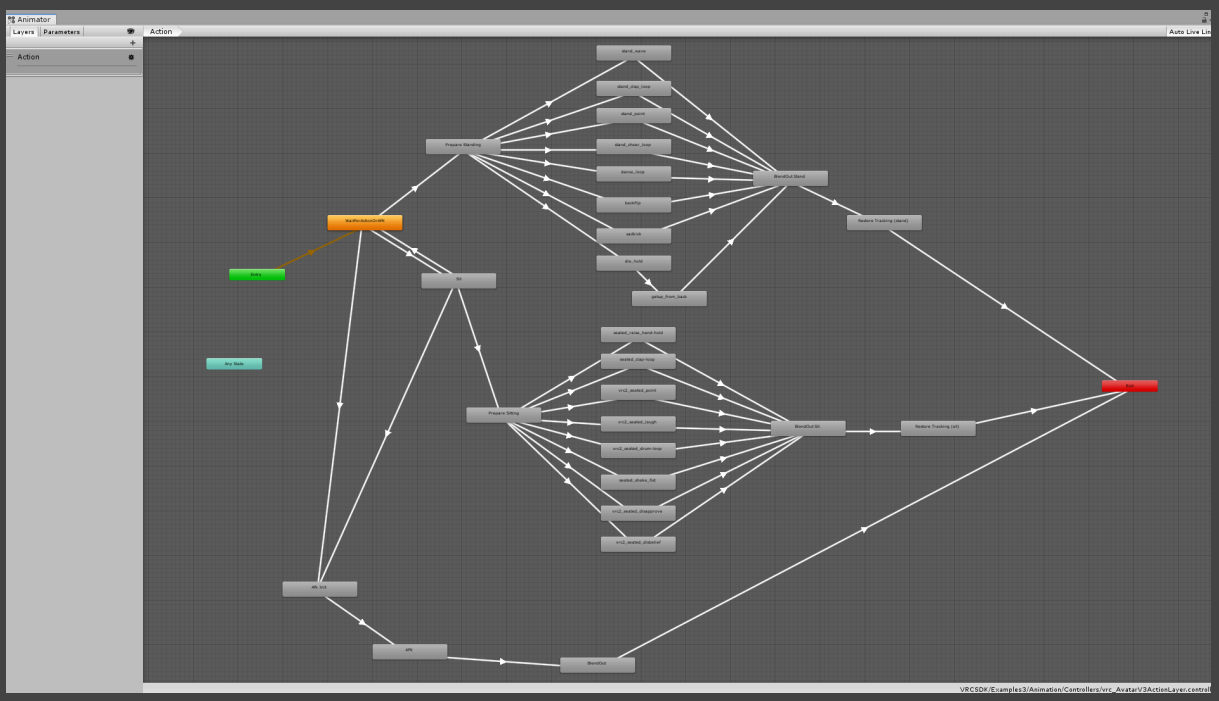
The Animator

Add tab-> Animator

You're going to be going over to it a lot

It's basically a flow chart, detailing how the animation is set up for a specific layer
You only get one animation per layer playing at a time, so if you'll be adding a lot of these if you're doing a lot of small things like options to open and close your mouth, or wiggle your left ear independently of your right etc

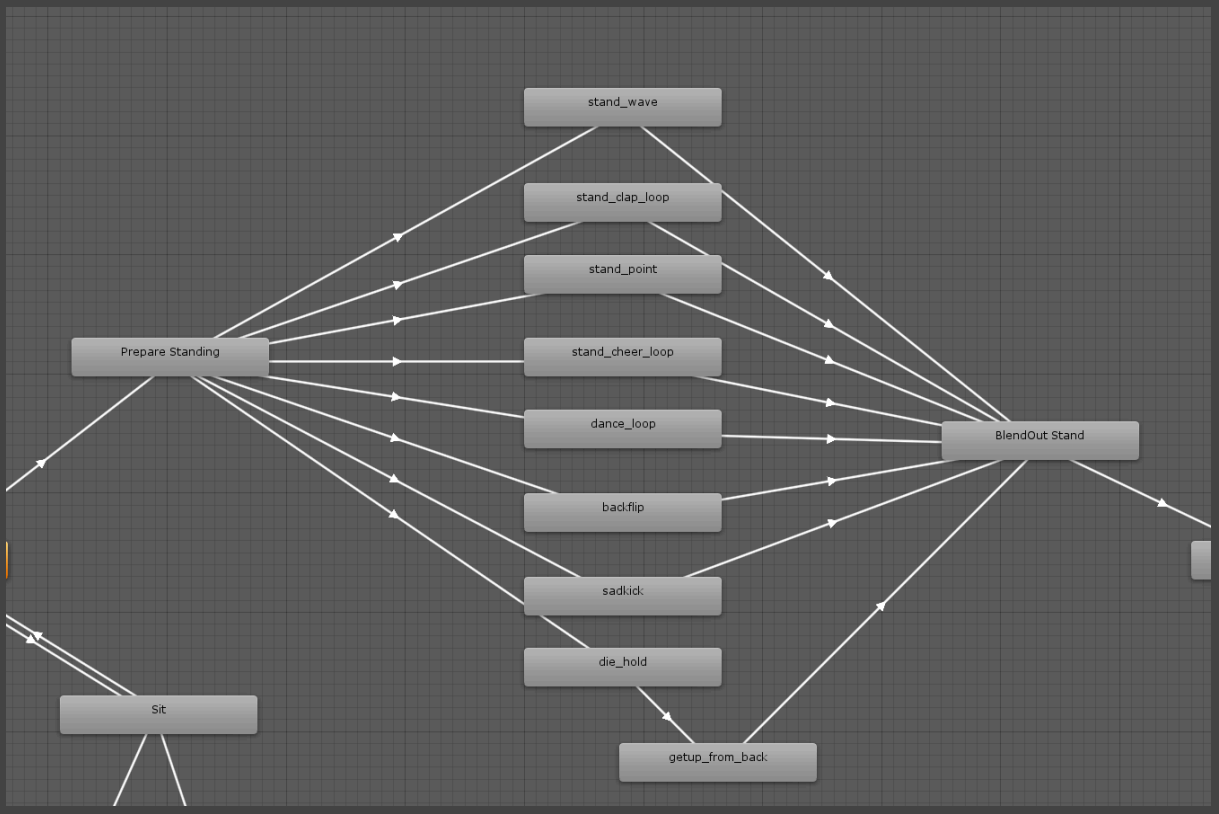
The action (Emotes) layer looks like this by default



Oh god

Thankfully that bottom layer is just for your afk, and middle diamond is for sitting emotes, which we can ignore for now.

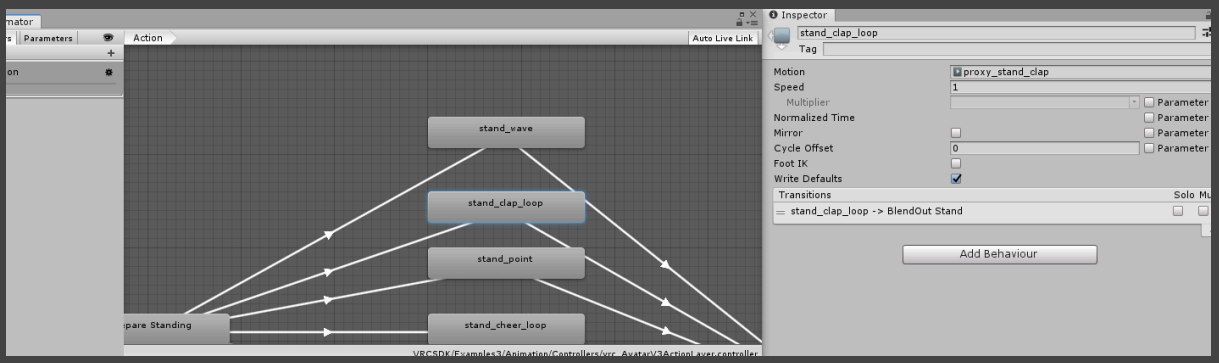
We just want this top diamond



As you can see, there are three stages to this part of the animator

It prepares the emote and overrides the default standing animation
It plays one of 8 emotes
And then it ends and sets you back to normal animation

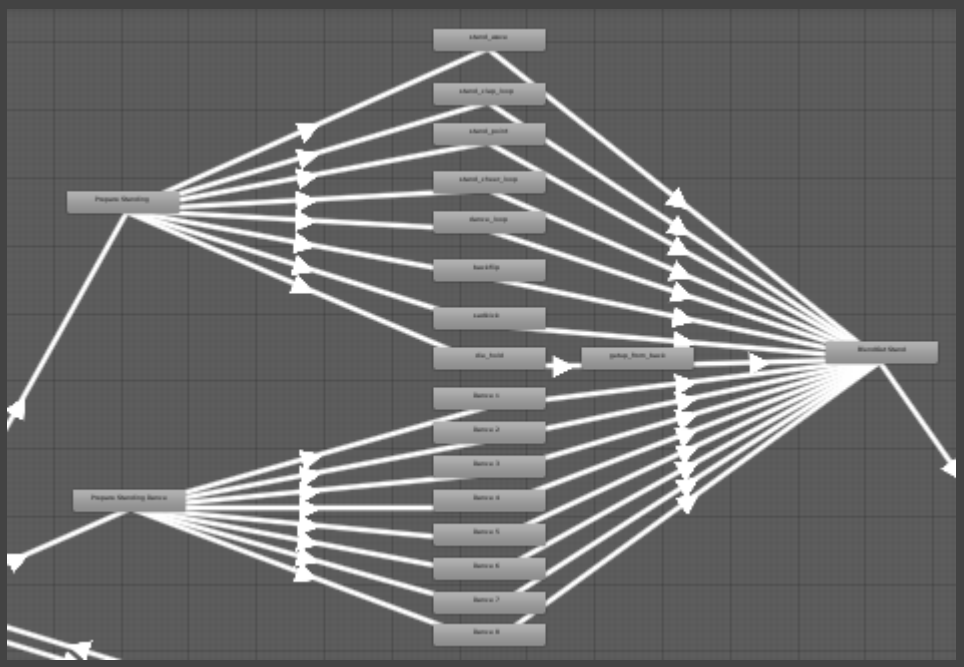
So to change from the default animations, all you've gotta do is click on one of these nodes, head over to the inspector, and replace the Motion with your custom animation!



Aaaaand yer done. Unless...

Doubling Your Emote Slots

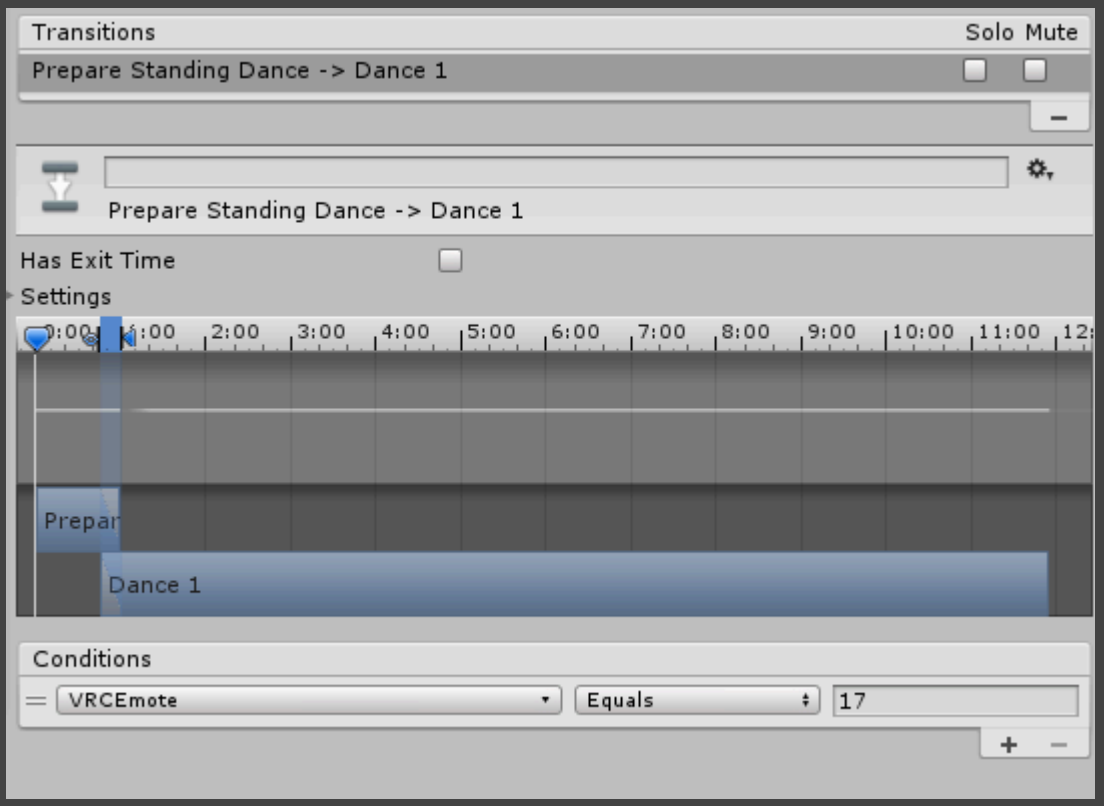
You can make it more complicated than that, though.



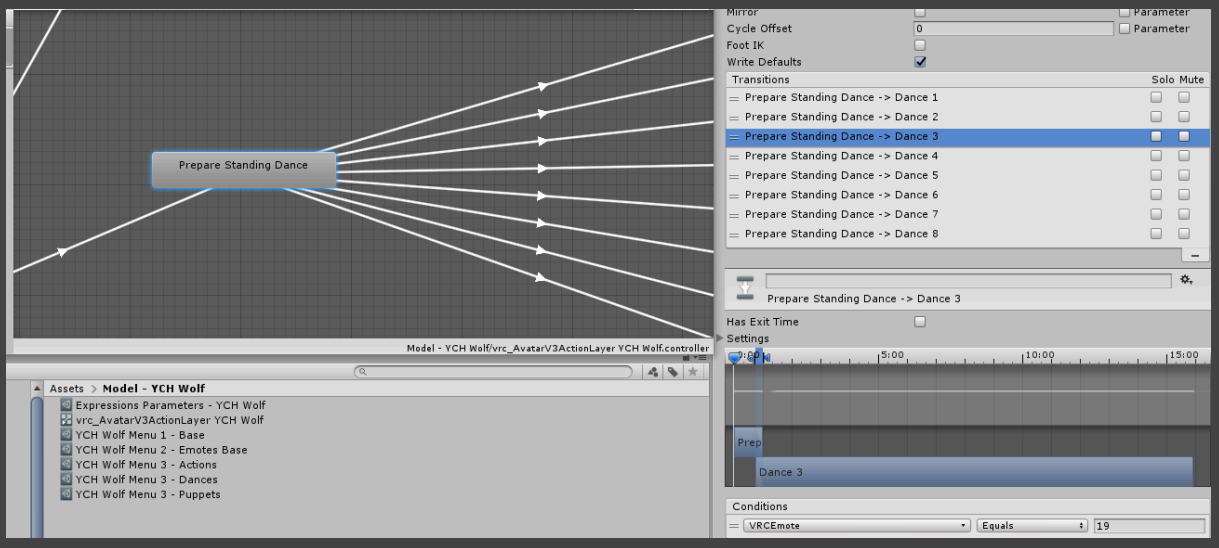
Copy and past the main 8 emote slots, along with the prepare standing node (It’s cleaner and easier to look at, even if it’s duplicating effort, as opposed to setting another transition trigger to the first)
And then get buckled do manually changing the transition efforts for each of these lines

If you click on the transition arrow and look at the inspector, it’ll tell you some interesting things, but for now all we’re worried about is the Conditions

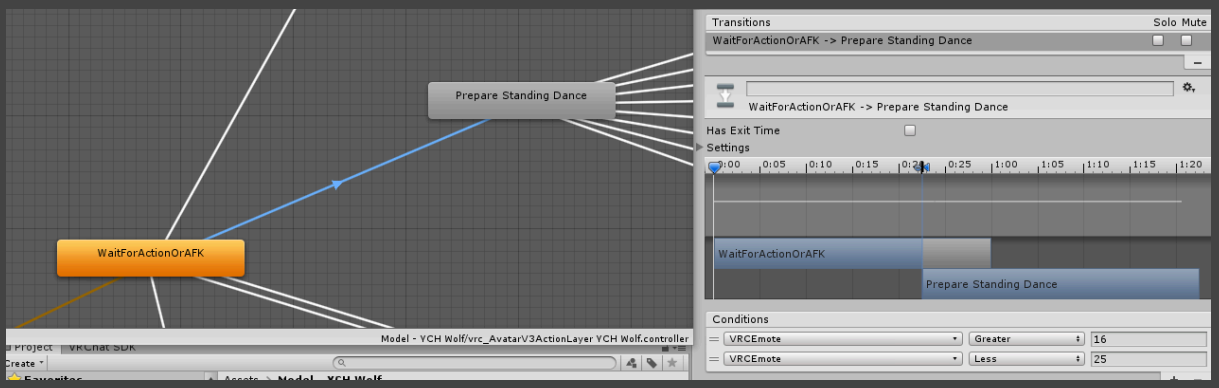
section.



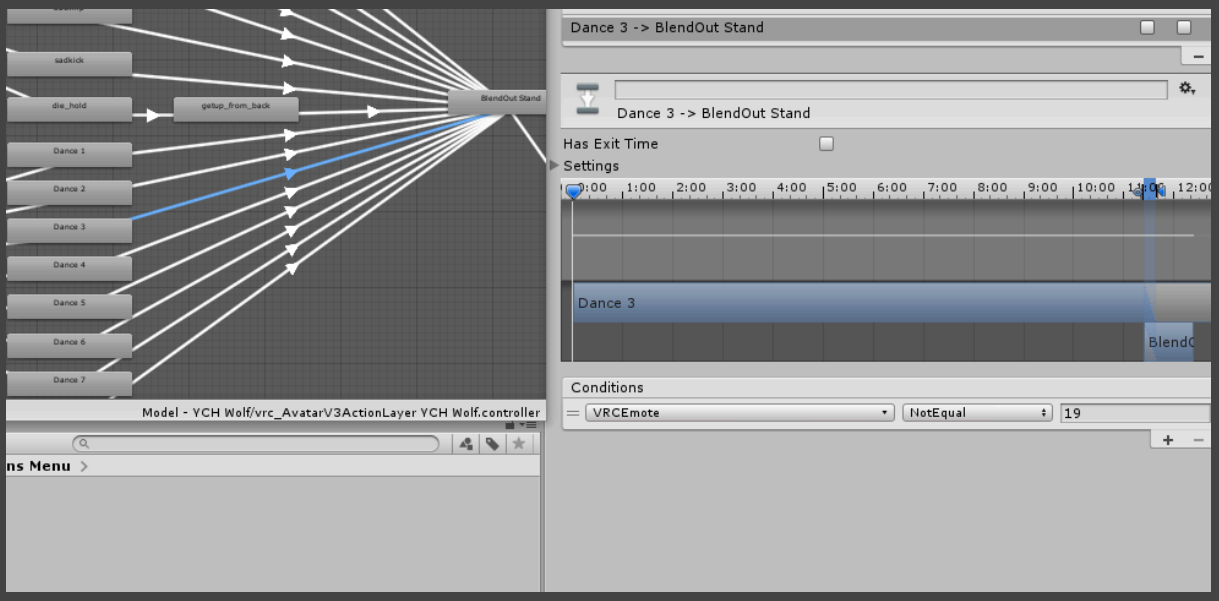
By default the VRCEmote parameter 1-8 is your default emote animations, and 9-16 are your sitting animations. So we're going to shove on a few more on the end of that, changing the Condition to increment from 17-25 as necessary. To speed this up, you can see every transition from a node by clicking on it, and changing them from there:



We’ll do the same for our duplicated Prepare Standing node, with two conditions this time (If there are multiple conditions, BOTH must be satisfied for it to trigger)



You’ll also need to edit all the transitions OUT, so that the emotes will stop if you change or close them



If it’s something like a dance or a looping animation, make sure to untick “Has Exit Time”, else it will continue playing until it reaches a defined point even after you untoggle.

Now you’ve got 8 more emote slots, you’ll need to build a way to access them.

Making Your Own Menu

Actual VRChat SDK Documentation:
https://docs.vrchat.com/docs/expression-menu-and-controls

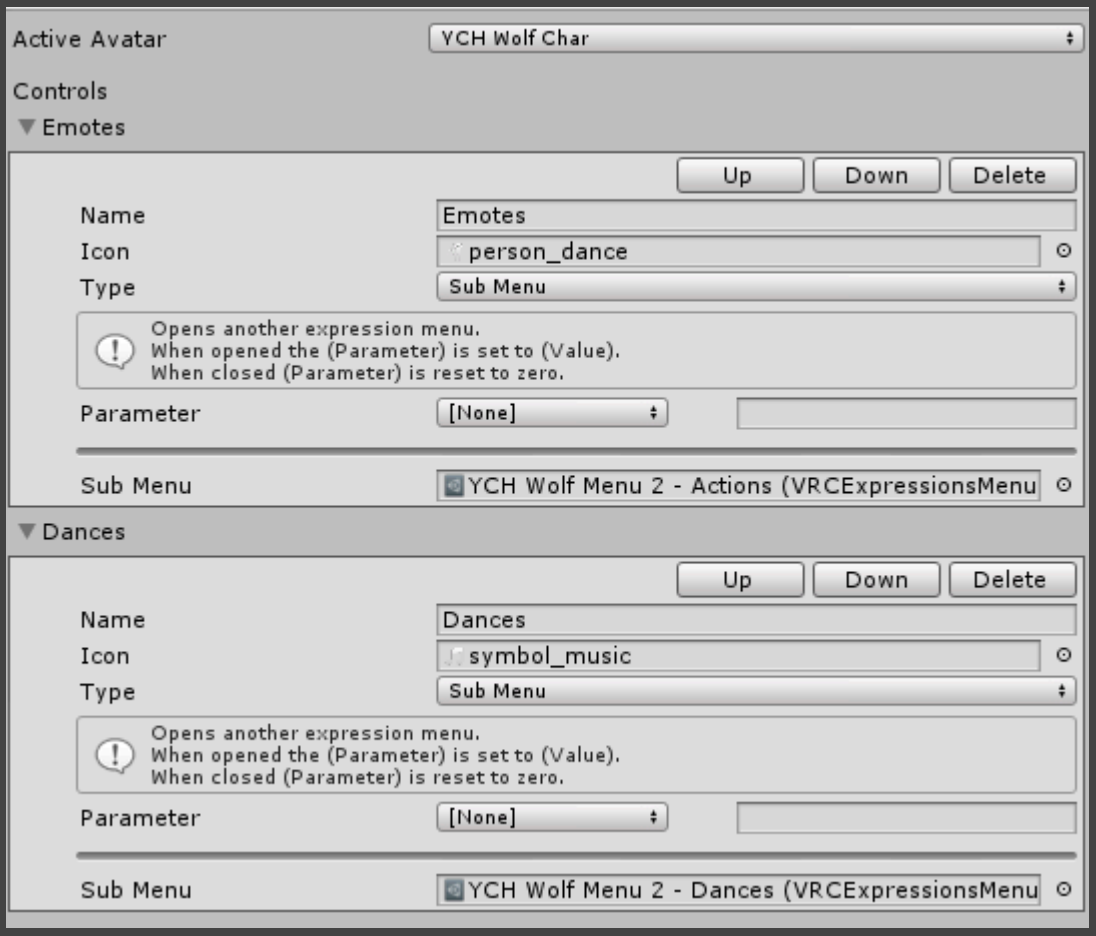
Yeah I know, the actual one kinda sucks, doesn't it?

Back in the VRC_AvatarDescriptor, add in the option to set your own Expressions Menu (And you might want Parameters whilst you're at it if you want even more complicated custom stuff, in the next section)

And grab the defaults from the examples folder.

Unfortunately, the default menu is also real blank. Copy that a few times, so you've got three. They'll be our Base Level Menu, our Default Emotes Submenu, and Our New Emotes Submenu

Add in a couple of controls, and set both of their to be submenus, these will be the default animations menu, and our new animations menu, respectively (I've named them Actions, and Dances)

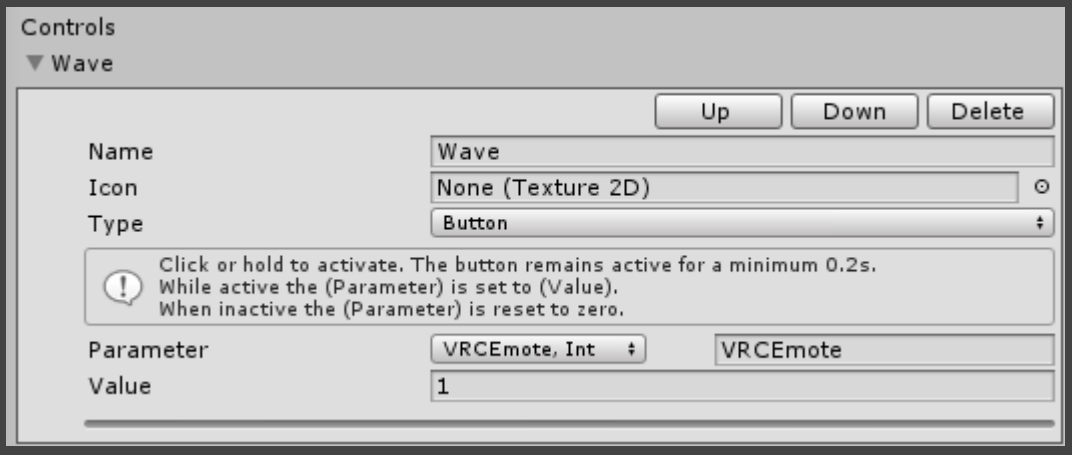


There's also example icons in the VRCSDK folder, to help organise and make things look pretty.

The SubMenus

For these submenus, just keep clicking add new control until you've got all 8 slots

For these, you'll want them to be buttons (or toggles, for looping animations like Clap and Dance)
And you'll want to set the parameter to VRCEmote
And the actual number from 1-8. This fills in clockwise, so feel free to reorder them, just remember which parameter is which animation.

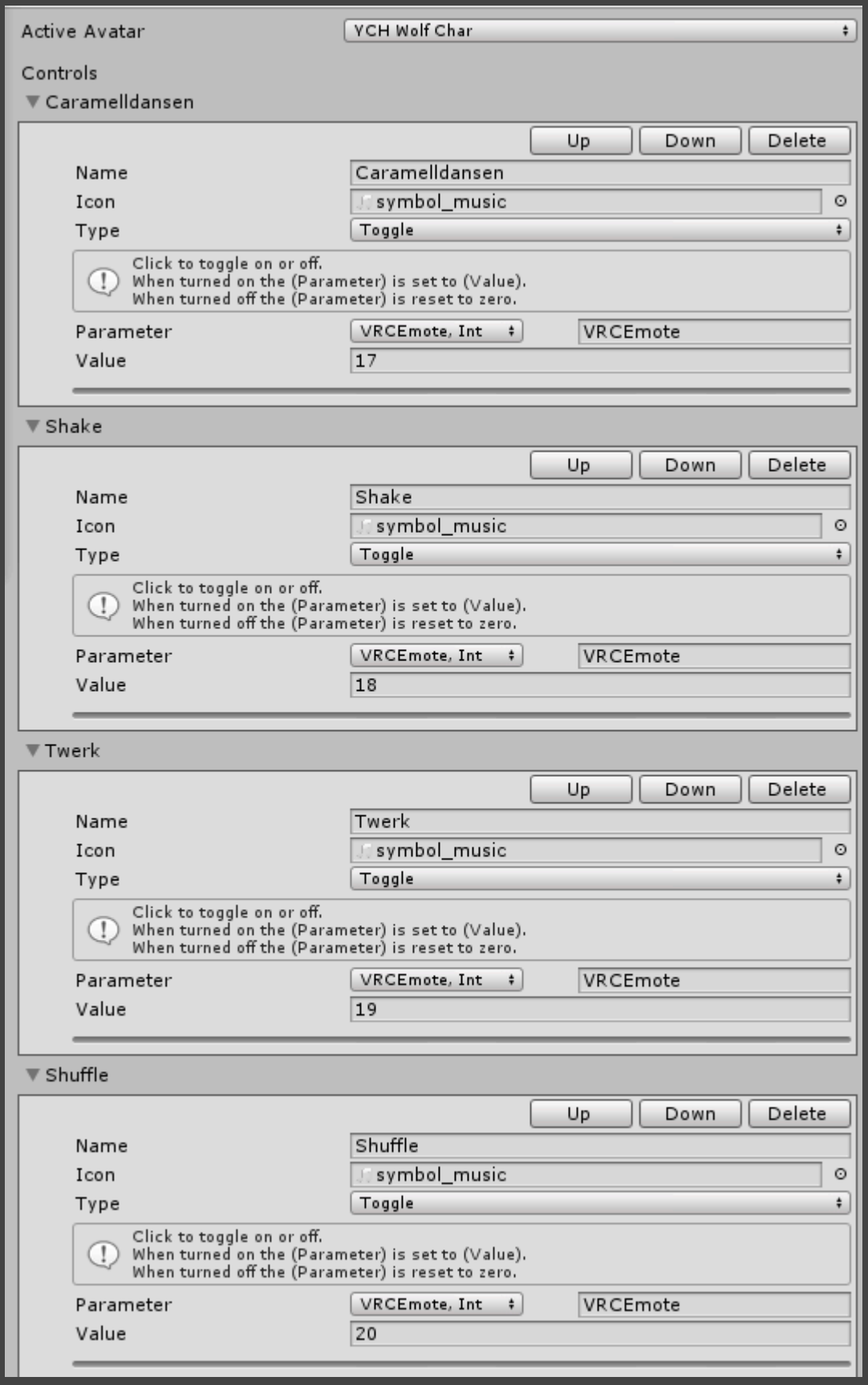


For your convenience, the default ones (and what type they should be) are:

1. Wave (Button)
2. Clap (Toggle)
3. Point (Button)
4. Cheer (Toggle)
5. Dance (Toggle)
6. Backflip (Button)
7. Sad (Button)
8. Die (Toggle)

If you change any of the default 8 animations, you'll rename them here (Also check whether the exit time checkbox is ticked or not depending if your overwriting animation is looping or not)

And then do the same for your new 8 emotes, which for me, are all dances and therefore toggles.



Now Open up VRChat and enjoy your 16 degrees of emotability.

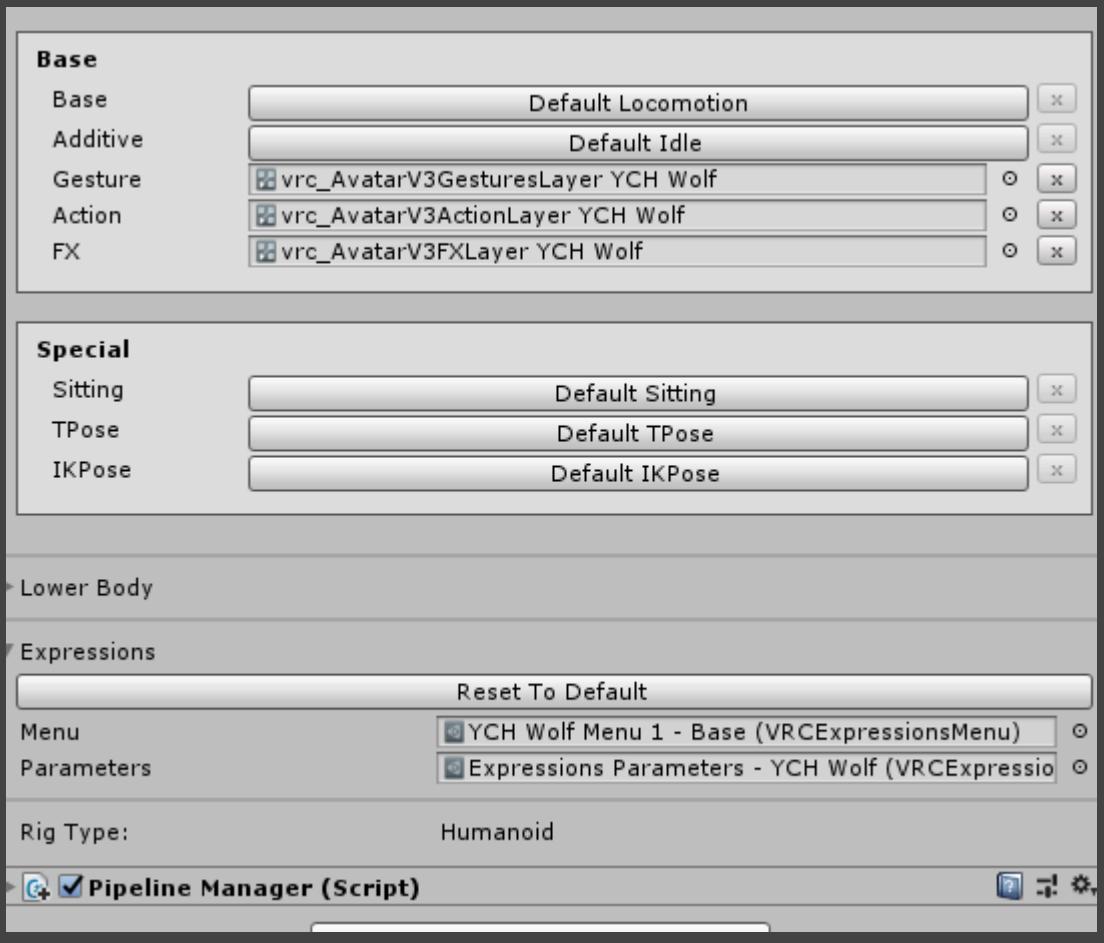
Theoretically, you can just keep going and adding more menus, if you've got enough animations. Integers go up to 255 after all.

Puppetry aka. A Basic Guide to The Complicated Stuff

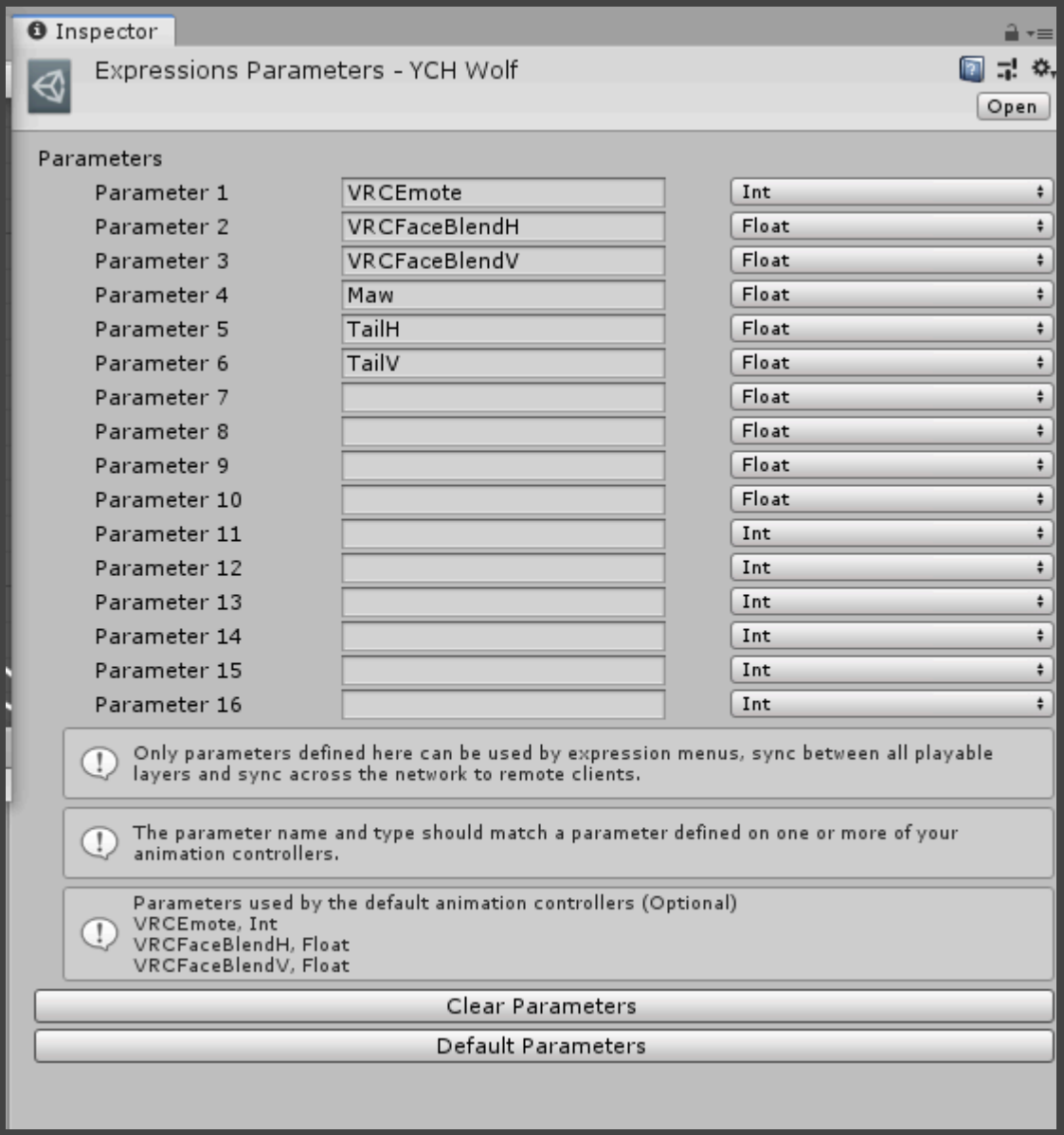
BUT amongst the capability to fill your wheel with submenus upon submenus for every possible animation you can find.

You can also do fun stuff with them.

Go grab the Gesture Layer and FX Layer Defaults and Get ready for ANOTHER layer of complications. And a lot more animator layers.

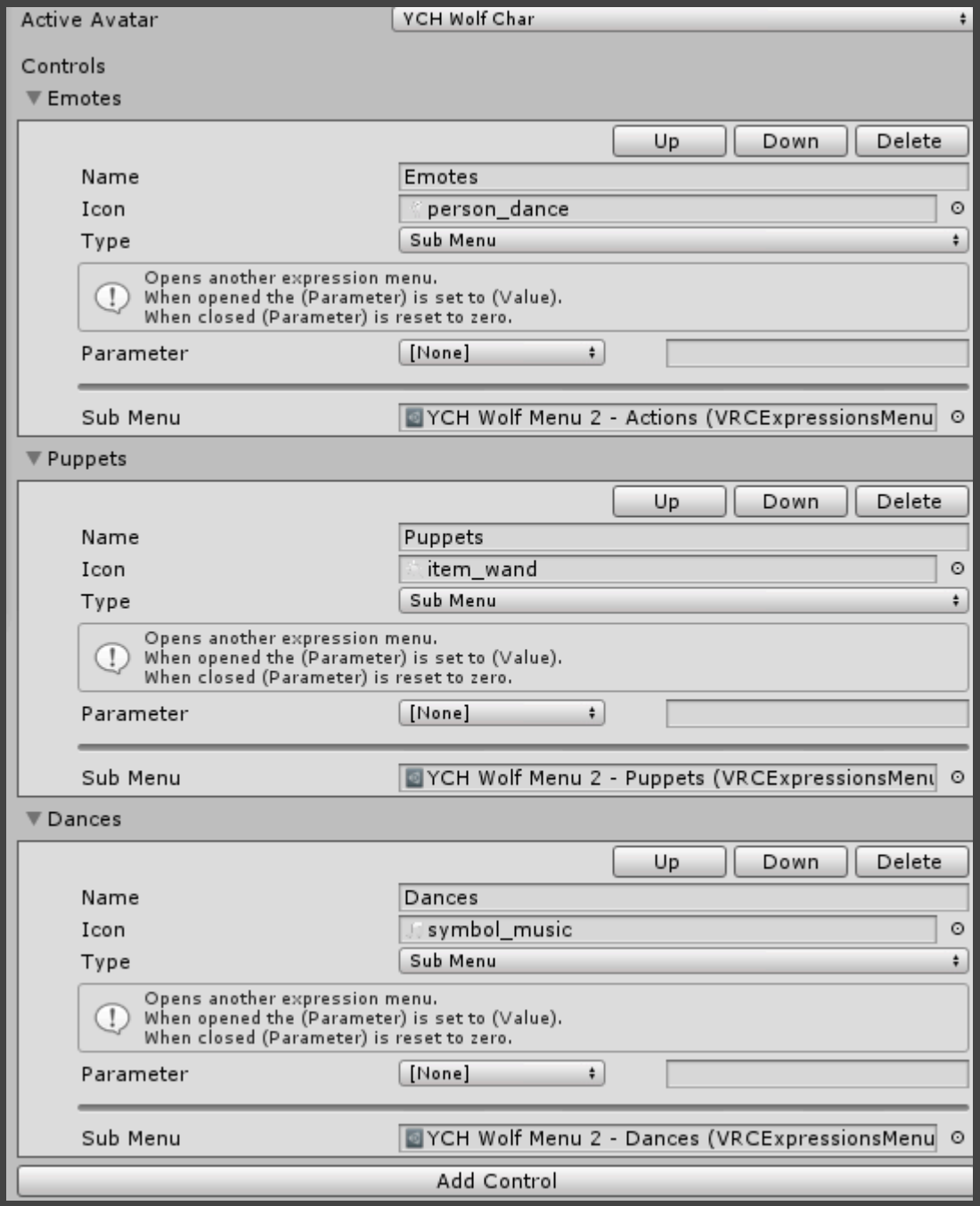


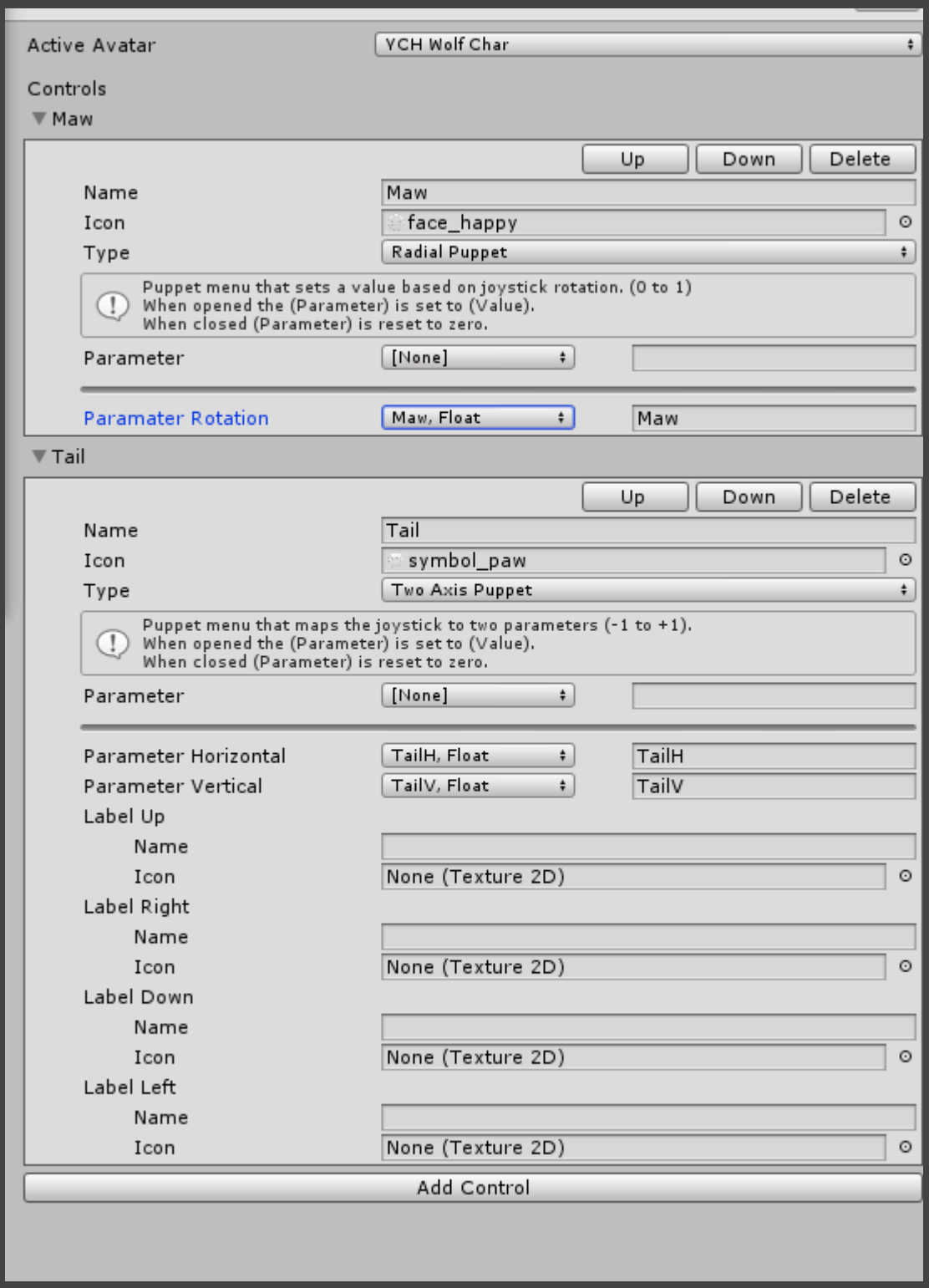
Head back to the Expressions Parameters and add in a couple of extra parameters. We'll set these to Floats instead of integers, since we'll be controlling them as decimals from -1 to 1



In this case, we're going to use two different puppets, one for the mouth to open and close it, and another to wag the tail manually. The important thing is to add a Horizontal and a Vertical Parameter for the tail, since we'll be controlling it in the X and Y axes.

Whilst we're at it, add in another submenu to our base menu, for Puppets, and a submenu for the puppets themselves



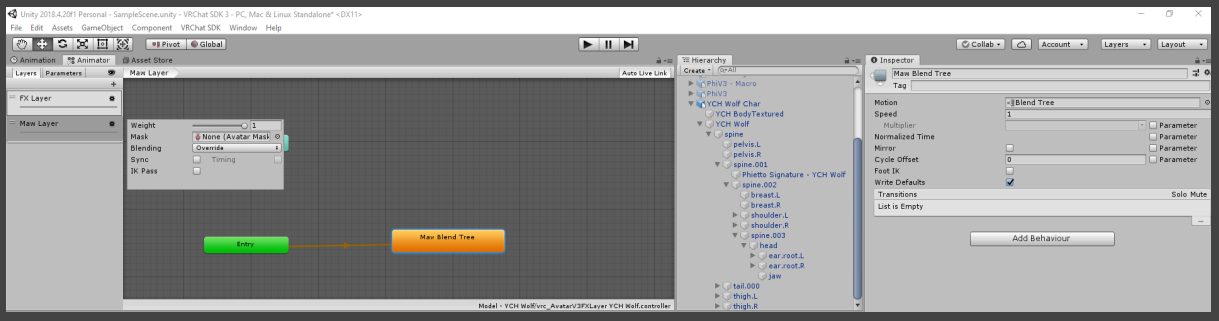


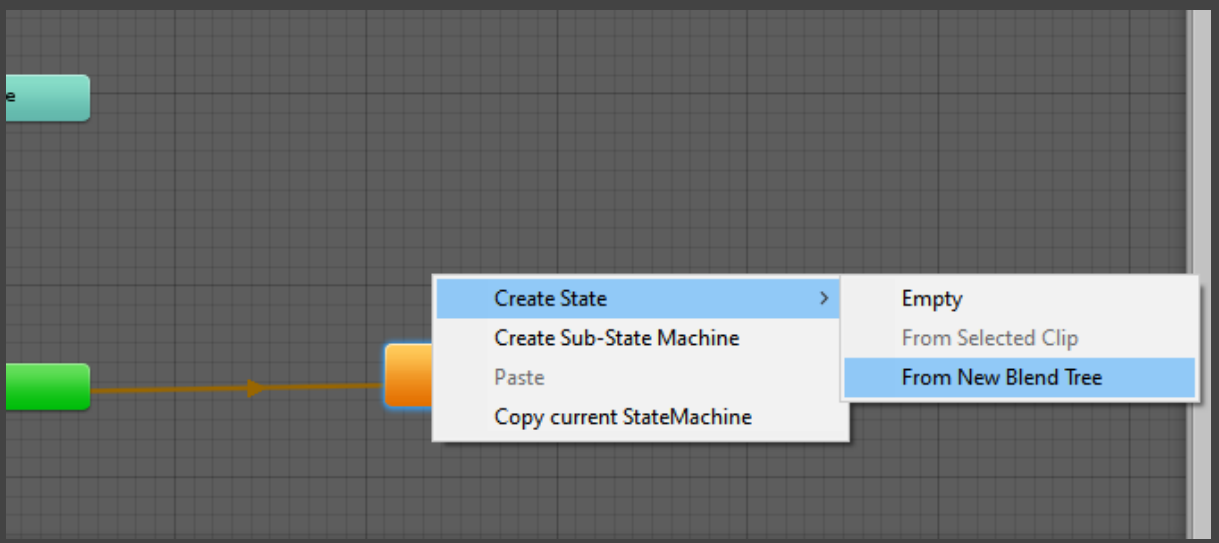
As the tooltips say, we'll be using two different types of puppet to control these (mostly for example purposes), so you should change these to suit your needs.

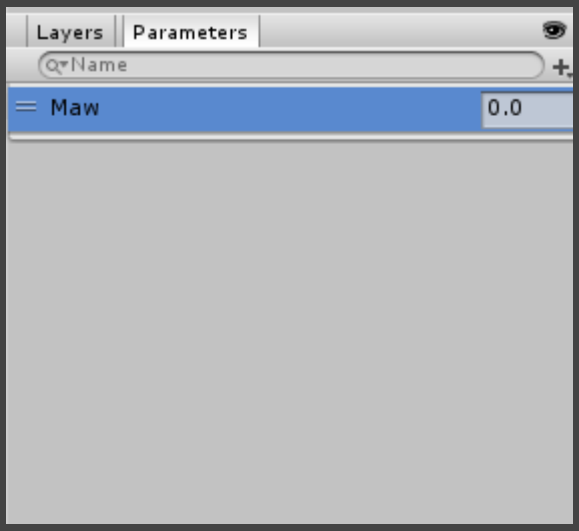
The FX Layer

We'll do the FX Layer first, since we'll be using a blendshape to switch how much our mouth is open. The FX Layer is for all the stuff like blendshapes, so this'll get pretty busy. Thankfully, it's not too complicated.

From the Animator, you'll want to create a new Layer (Make sure its weight is set to 1 so that it's always on! And in that layer we'll create a single node, as a blendtree. We'll also add a Float Parameter, in the other tab next to layers, to tell it to look for the Parameter we just added.



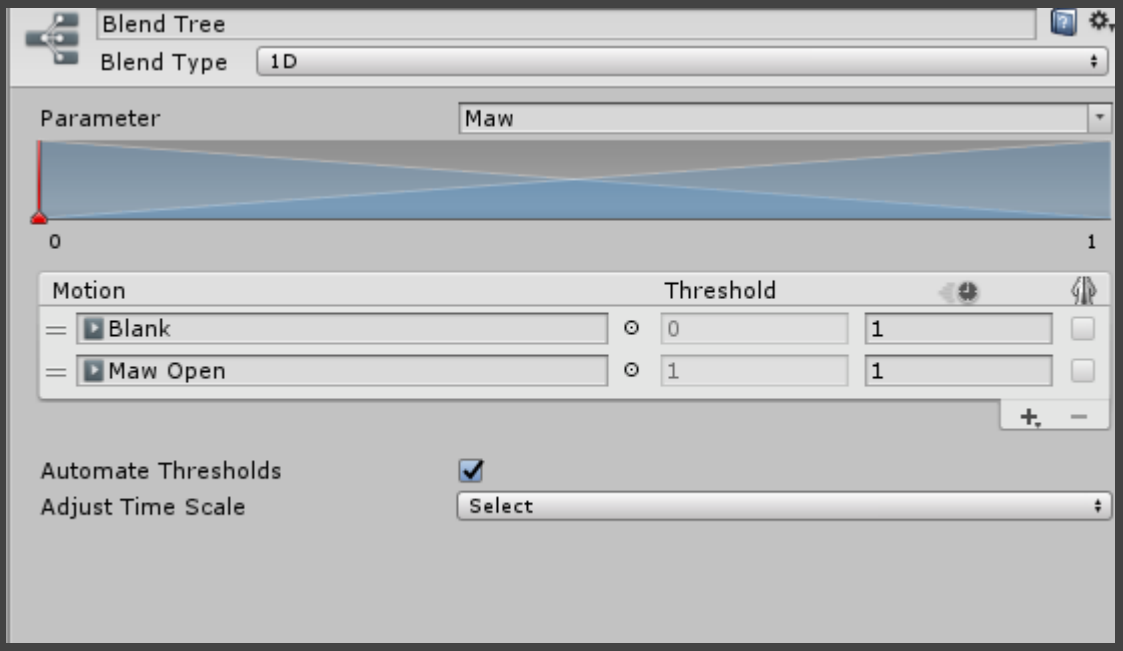


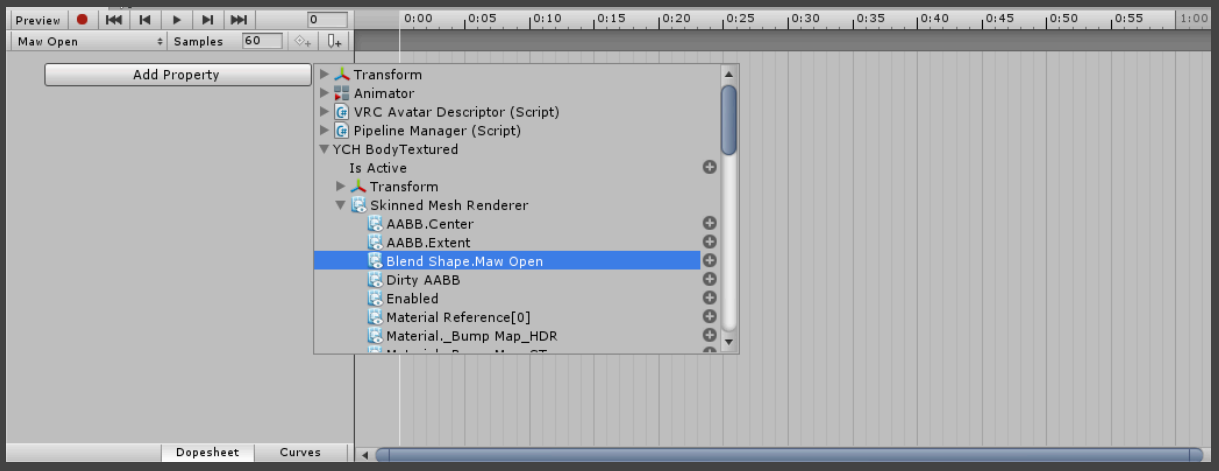


We could do this with two states, an open and a close state, but a blend tree will let us smoothly choose a point between our animation states.

Double click on the blend tree to edit it:

We'll set it to a 1D tree, and set its Parameter to the one we just created
Then we'll add two motion fields, and two animation clips to go in them: A blank, empty animation, and another one that just has a single keyframe of our blendshape at maximum.

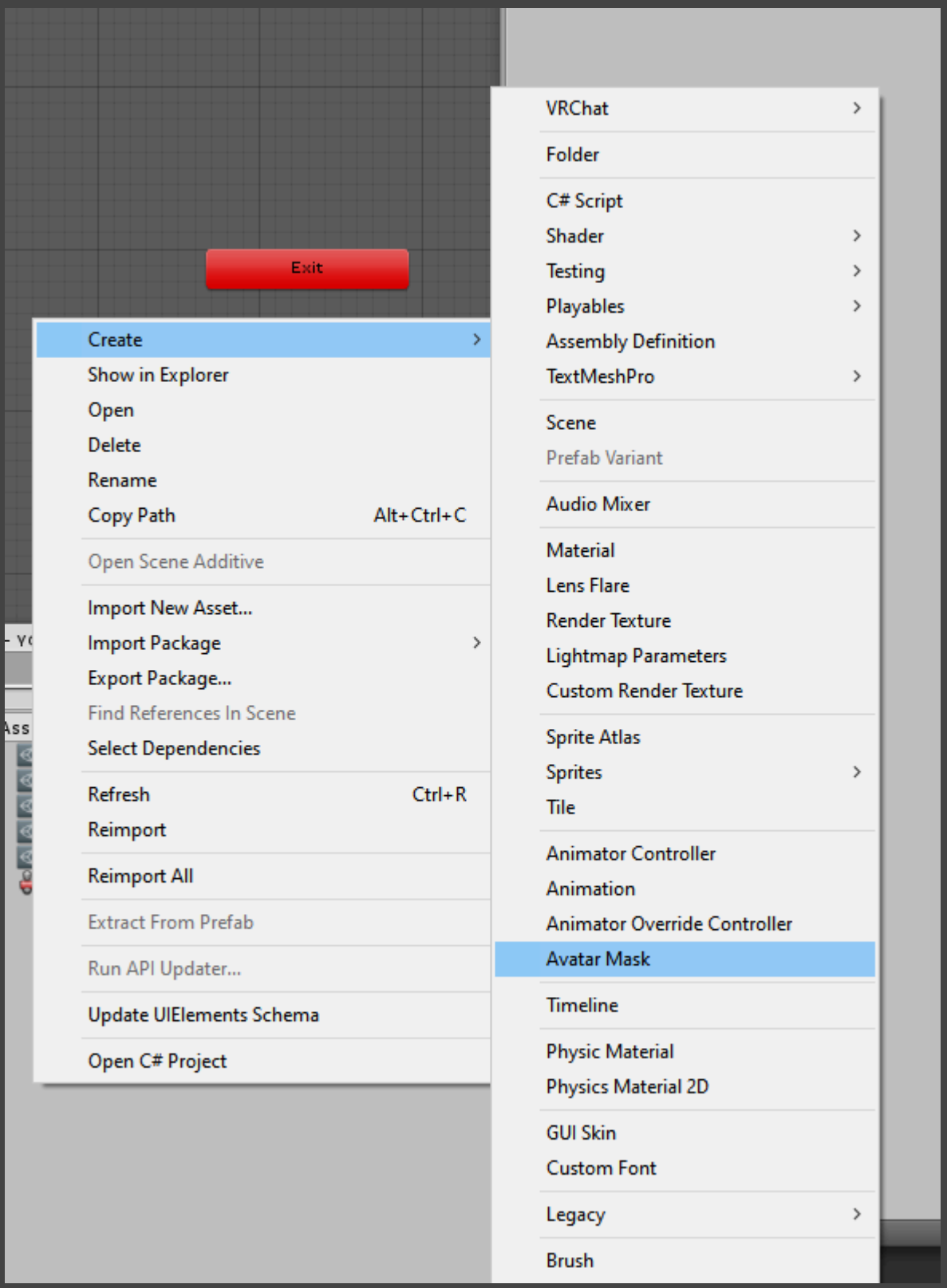


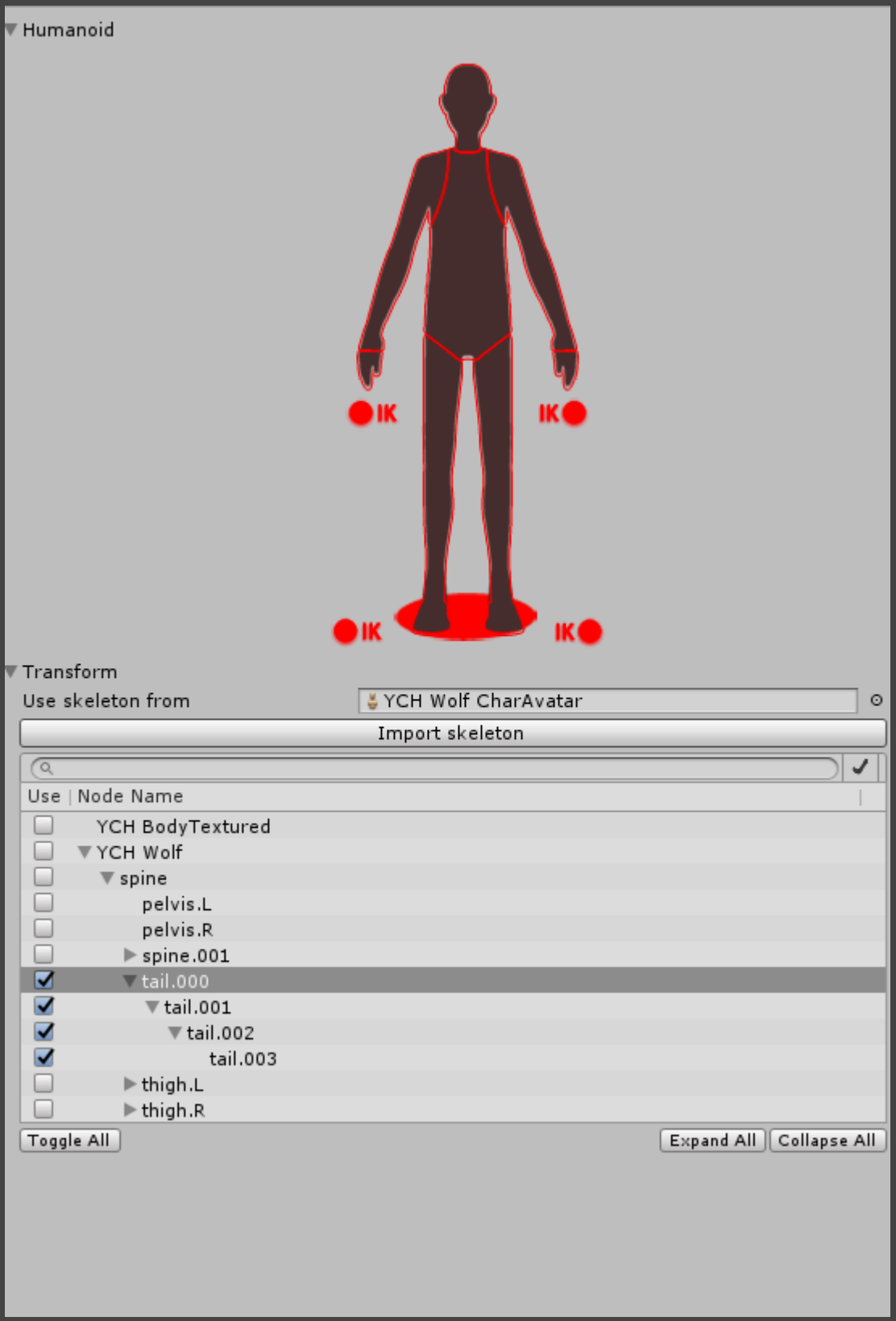


And now, as you scroll through the wheel in the menu, it'll slowly transition from no animation towards your full blendshape, in effect turning it up from 0 to max! MAXIMUM MAW CONTROL

The Gesture Layer

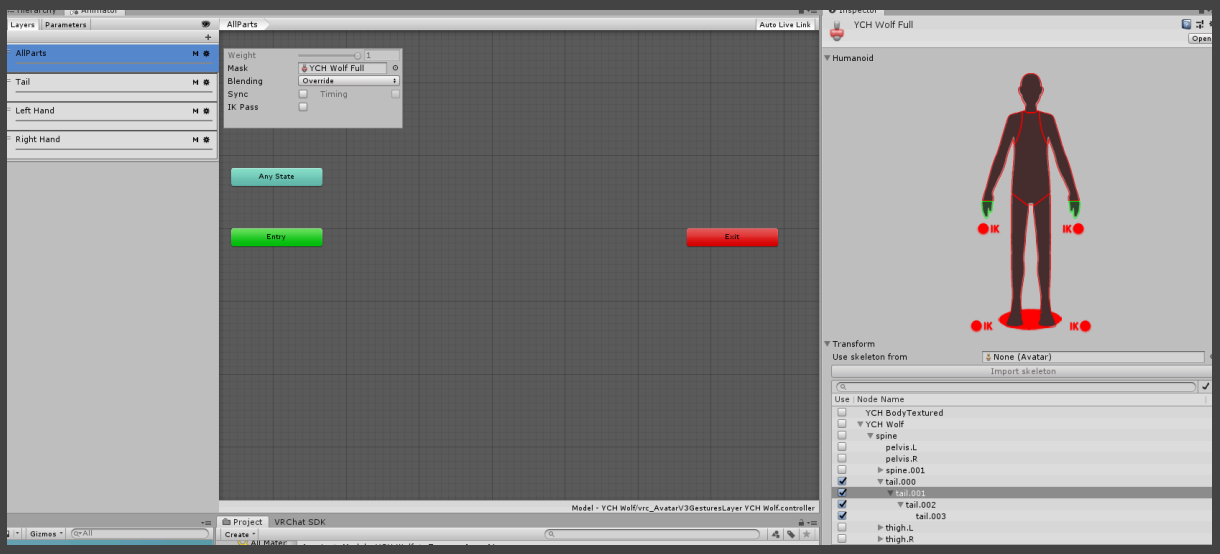
We'll be doing a similar thing for the tail, but we'll be animating by transforming the bones in Unity, and so we'll have to use the Gesture layer. Which also means we'll need to use a mask. This makes sure your animation only affects the specific bones, and doesn't override everything else!


Exit
Create
Show in Explorer
Open
Delete
Rename
Copy Path Alt+Ctrl+C
Open Scene Additive
Import New Asset...
Import Package
Export Package...
Find References In Scene
Select Dependencies
Refresh Ctrl+R
Reimport
Reimport All
Extract From Prefab
Run API Updater...
Update UIElements Schema
Open C# Project
VRChat
Folder
C# Script
Shader
Testing
Playables
Assembly Definition
TextMeshPro
Scene
Prefab Variant
Audio Mixer
Material
Lens Flare
Render Texture
Lightmap Parameters
Custom Render Texture
Sprite Atlas
Sprites
Tile
Animator Controller
Animation
Animator Override Controller
Avatar Mask
Timeline
Physic Material
Physics Material 2D
GUI Skin
Custom Font
Legacy
Brush

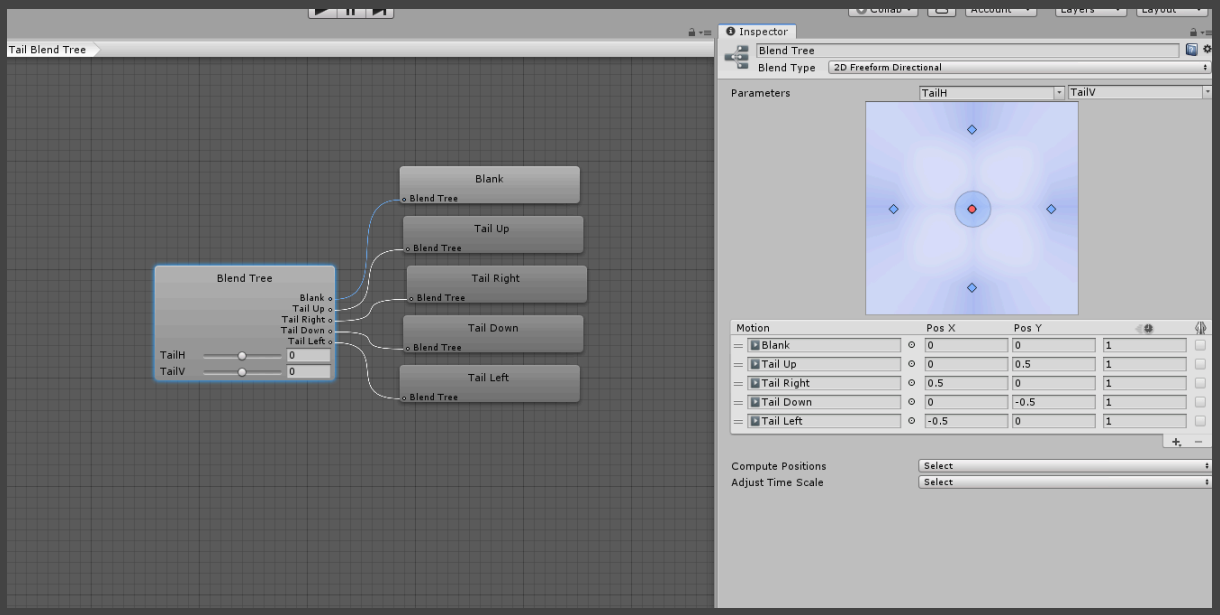


Import your characters skeleton and check the bones that you'll change in this layer.

Also go to the all parts layer and its mask and make a copy to add these in, since the top one is the sum of all layers below it.

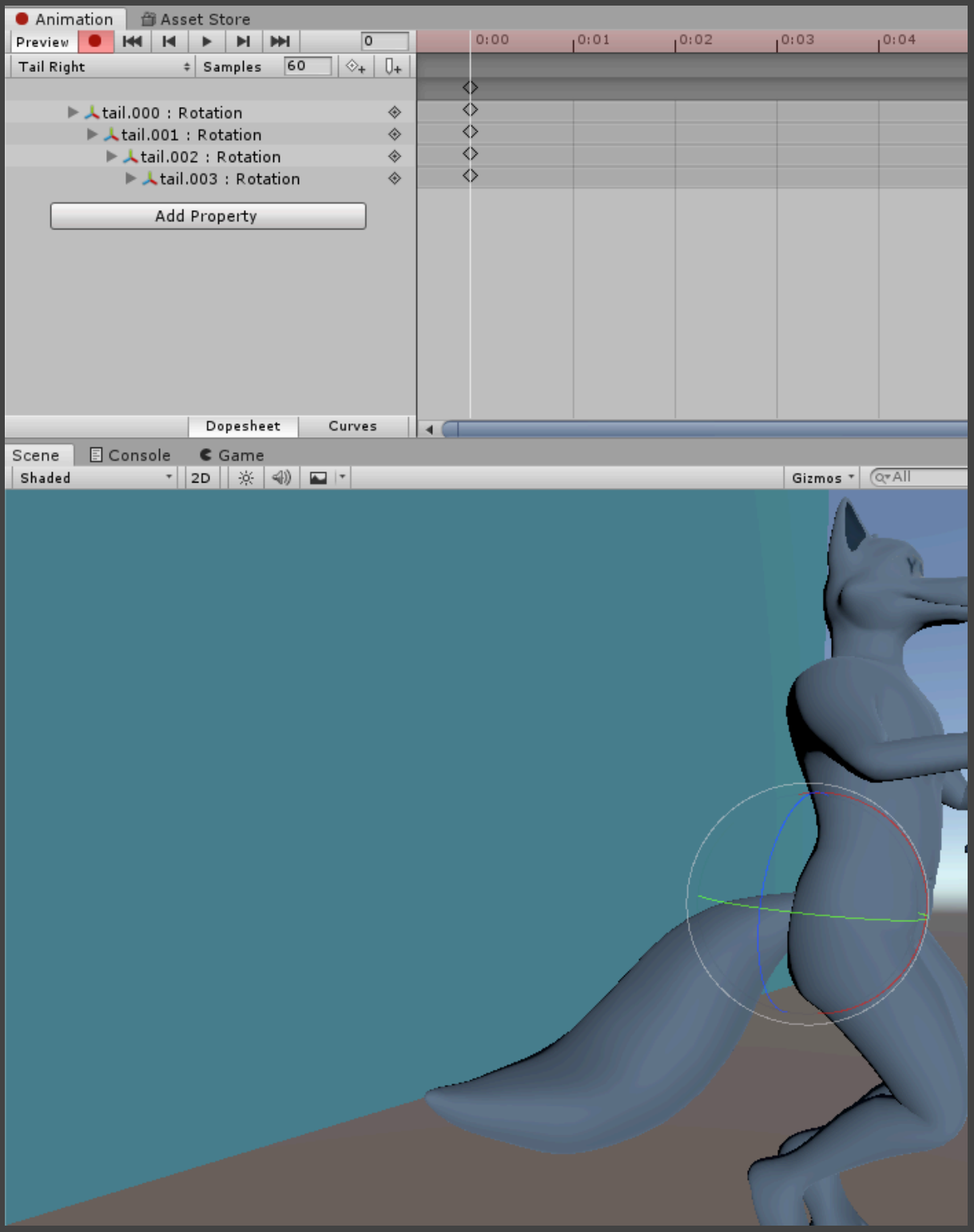


Back to the animator, new layer, new blend tree, new parameters, as we did before, except now we've got two parameters, and a 2D blend tree to manage it:



Don't set the X and Y Positions to 1, or else you'll only get the full animation state when it's perfectly at that point. I tend to use somewhere between 0.5 and 0.75 to give a little bit of "full" leeway.

Add in your animations, done in the animation tab, just swish that tail about!



Double check everything is set up properly, and hit that build button!

And most importantly, have fun~!