

Creating Life03

MacGruber's Tutorial Series

This tutorial goes step-by-step through the process of how the *Life03* demo scene was made. I'm trying to share why I'm doing some things the way the I do. Hopefully you will find this helpful and can apply these ideas to your own VaM scenes. **Note that all screenshots in this tutorial have been made with the 0.3 version of the plugin.**

Making plans

As with all projects, you can progress better when you have a plan. Doesn't have to be too detailed, but you should have a vague idea in mind what you want to achieve. The goal for this scene was to provide a demo for inclusion into the Life plugin package. It was supposed to show off the newly improved "Thrust" feature, without too much distracting other things. I decided this would work best with a sensual bedroom scene were two people have fun in a classic cowgirl position. This scene is very similar to the *Sofa* scene I used to present the original Breathing plugin.

In order to keep the package nice, clean and small we limit ourselves to VaM build-in assets. Except obviously for the scripts included in the package.

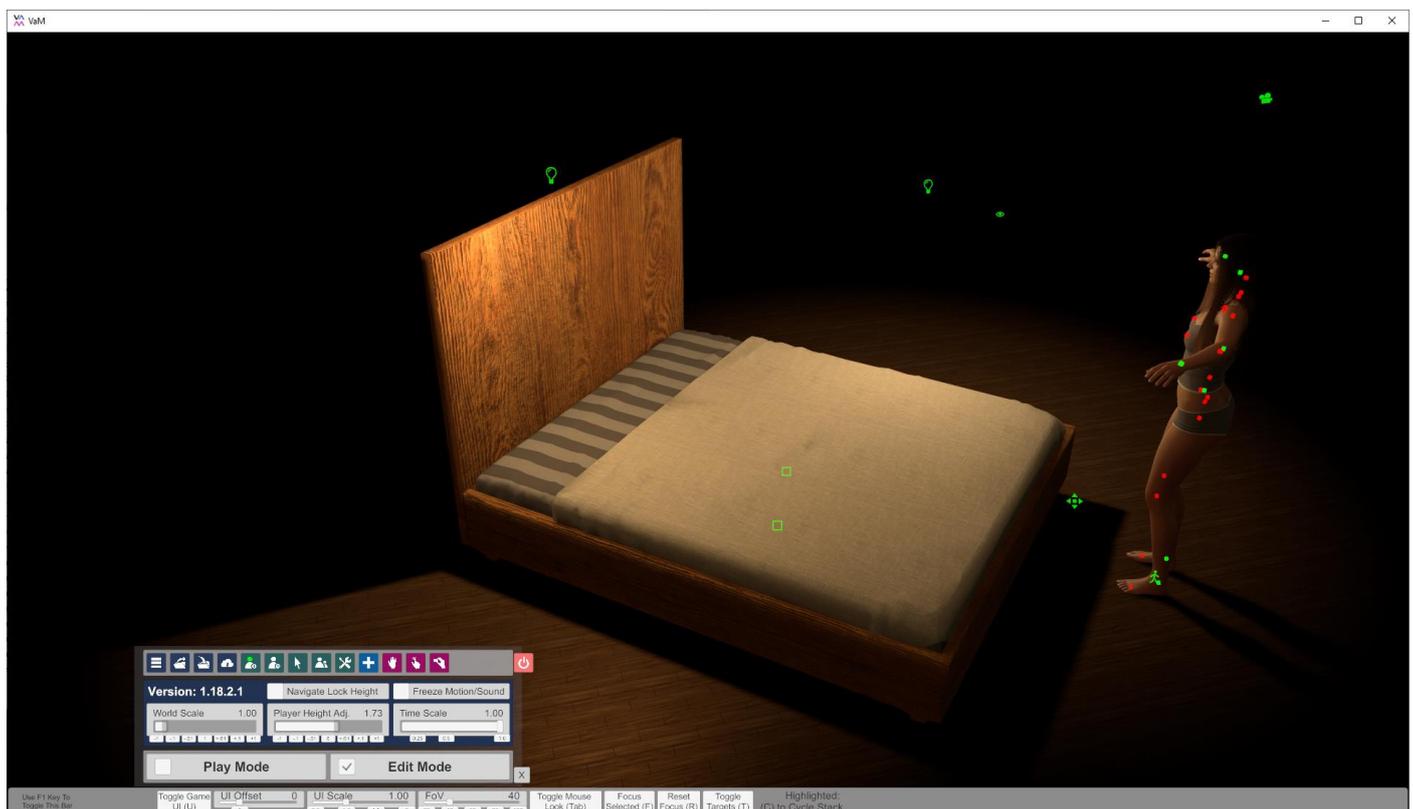
Save the scene!

I strongly recommend regular saving of your scene. VaM does not have an "undo" function, and sometimes a tiny change can ruin everything. I have been saving my scene about 20-30 times during the production of this.

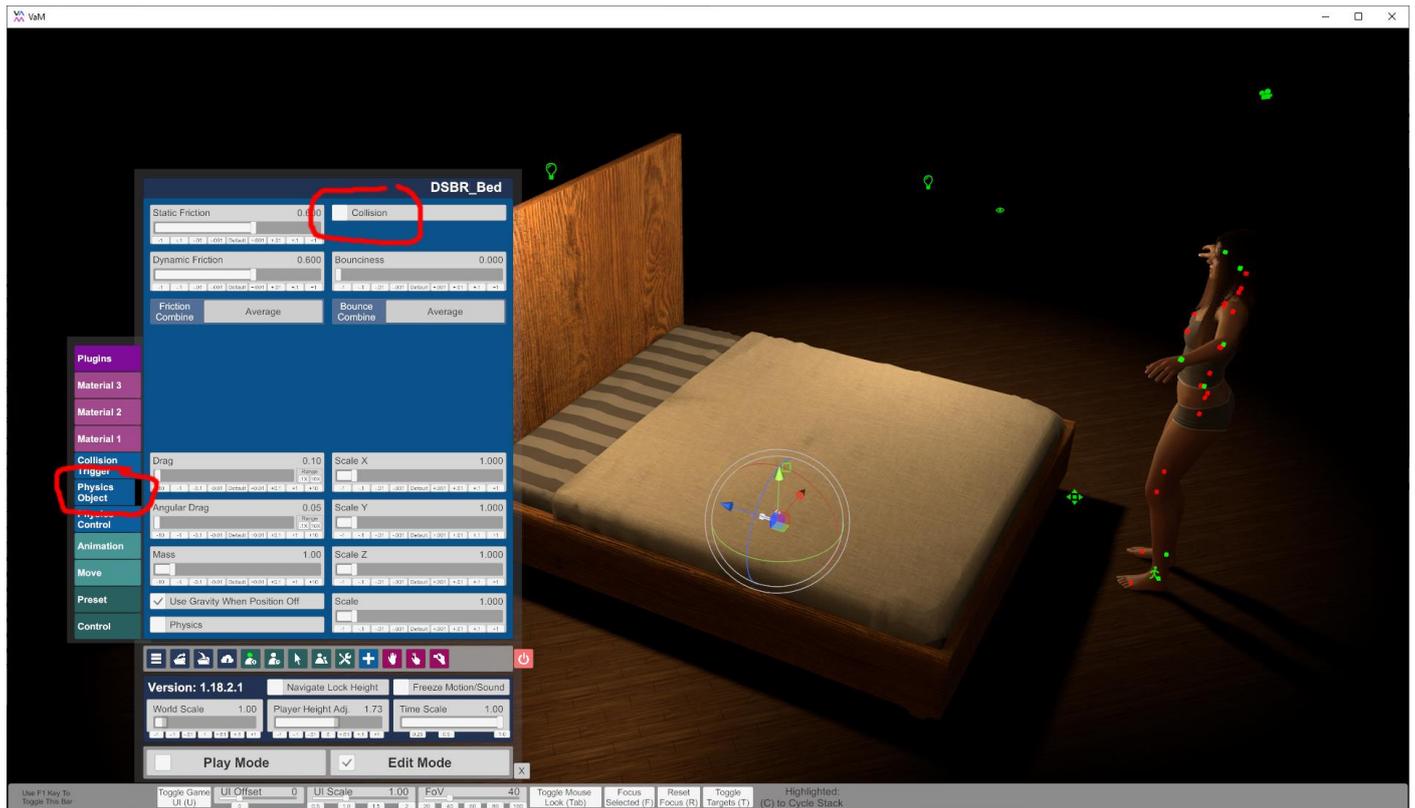
Basic Scene Layout

We start with the basic layout of the scene, adding a *WoodPanel* for the floor and the *DSBR_Bed*. Careful to never place objects near the center of the scene (Position 0/0/0), since that is the location were a new Person will appear when added to the scene. If you have anything with collision at that position and add a new Person atom, your Person might "explode" due to VaM physics trying to solve an impossible situation.

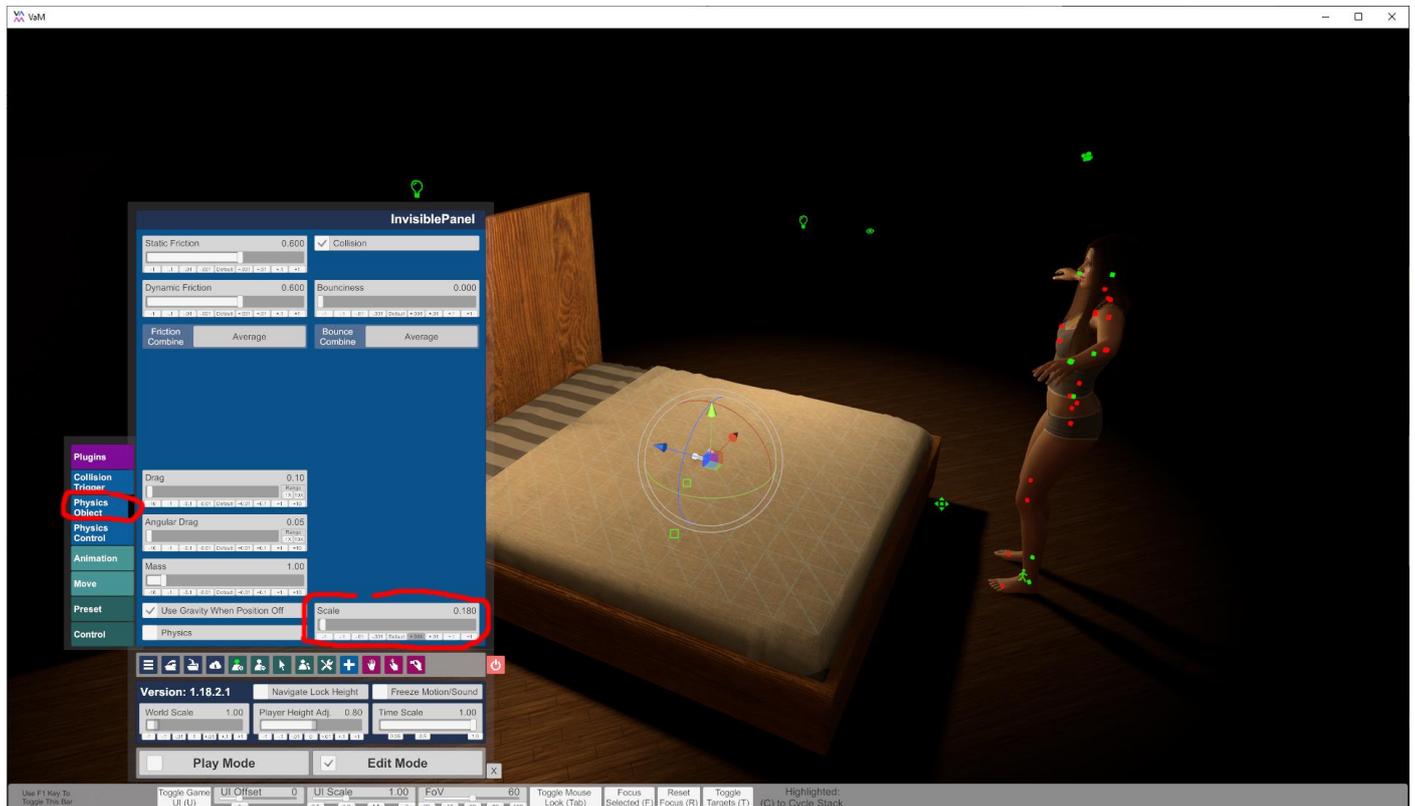
I'm also adding two *InvisibleLight*'s and make them bright enough so we see something while working. This is just for working, we will bother with nice lighting later.



Because of how physics work in VaM, any *Person* placed on the bed would appear to hover by a centimeter or so. To work around that, we deactivate “Collision” on the bed atom:



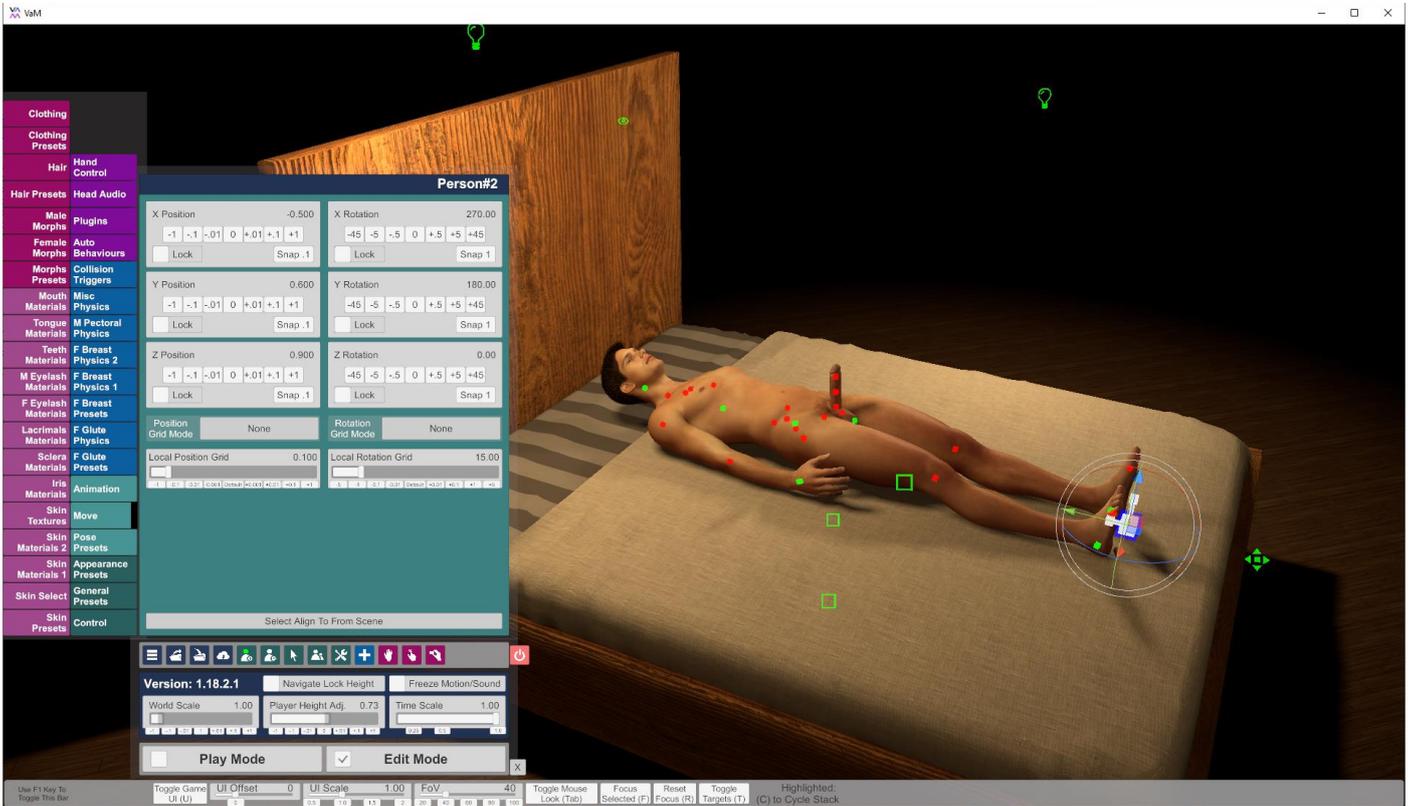
Instead we add an *InvisiblePanel* atom to the scene, scale it down to an appropriate size and position it just below the surface of the bed. This will create a fake impression of characters sinking a bit into the soft bed. Of course, that's not accurate without proper soft-physics of the bed, but it's what we can do easily. Using just a simple *InvisiblePanel* atom for collision will also give us a slight performance gain compared to using the more complex bed geometry. Player with lower-end machines will be thankful.



Character placement

Now let's put some characters into the scene. We start with the guy.

When I position characters, I mostly use the buttons on the "Move" tab to move the *Person* atom as close to the desired position as possible. It's just much faster and precise to move in 1m or 10cm steps and rotate in 45° and 5° steps, even when in VR. Moving the entire *Person* atom instead of the individual controls has the added benefit that the *Person* atom will be close to the characters feet, making it easy to find if players what to make modifications like loading custom looks.



For final positioning, of course, I move controls in VR by hand:

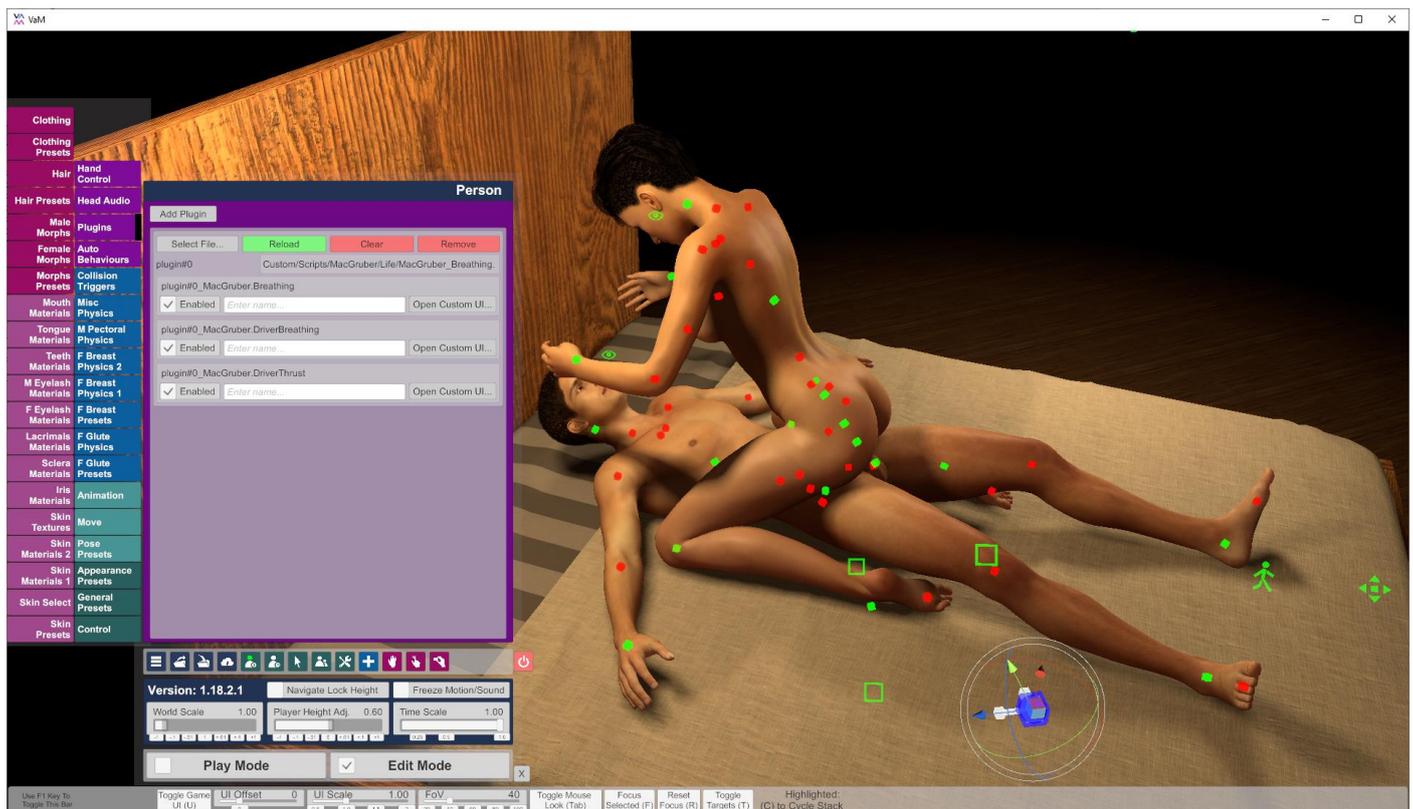


Now we get to adding and placing the girl. Move the girls *Person* atom to make her hover like one meter over the bed and lean her a bit forward. Then pull the feet up and lower the entire atom down to place her on top of the guy. The tricky part is of course the insertion of the penis. In this case it helped to set Position and Rotation of the *penisBaseControl* to "On", and rotate the base a bit. After the penis is inserted, enable Position and Rotation for the other two penis controls as well. This will help it to stay inside during the "action". Give it a manual test run by moving the hip control of the girl to see if you got a good angle. It's an iterative process, you have to tweak a bit until it works well.



Breathing and Drivers!

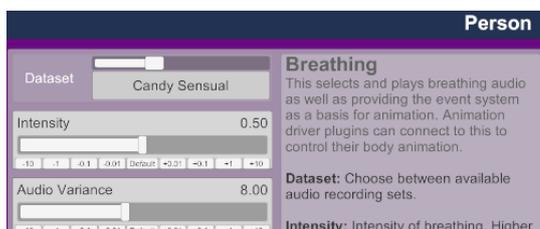
Now let's get some action! We place a `MacGruber_Breathing.csl` (v0.3) or `MacGruber_BreathingThrust.csl` (v0.4+) plugin onto the girl:



This plugin consists of a number of sub-plugins:

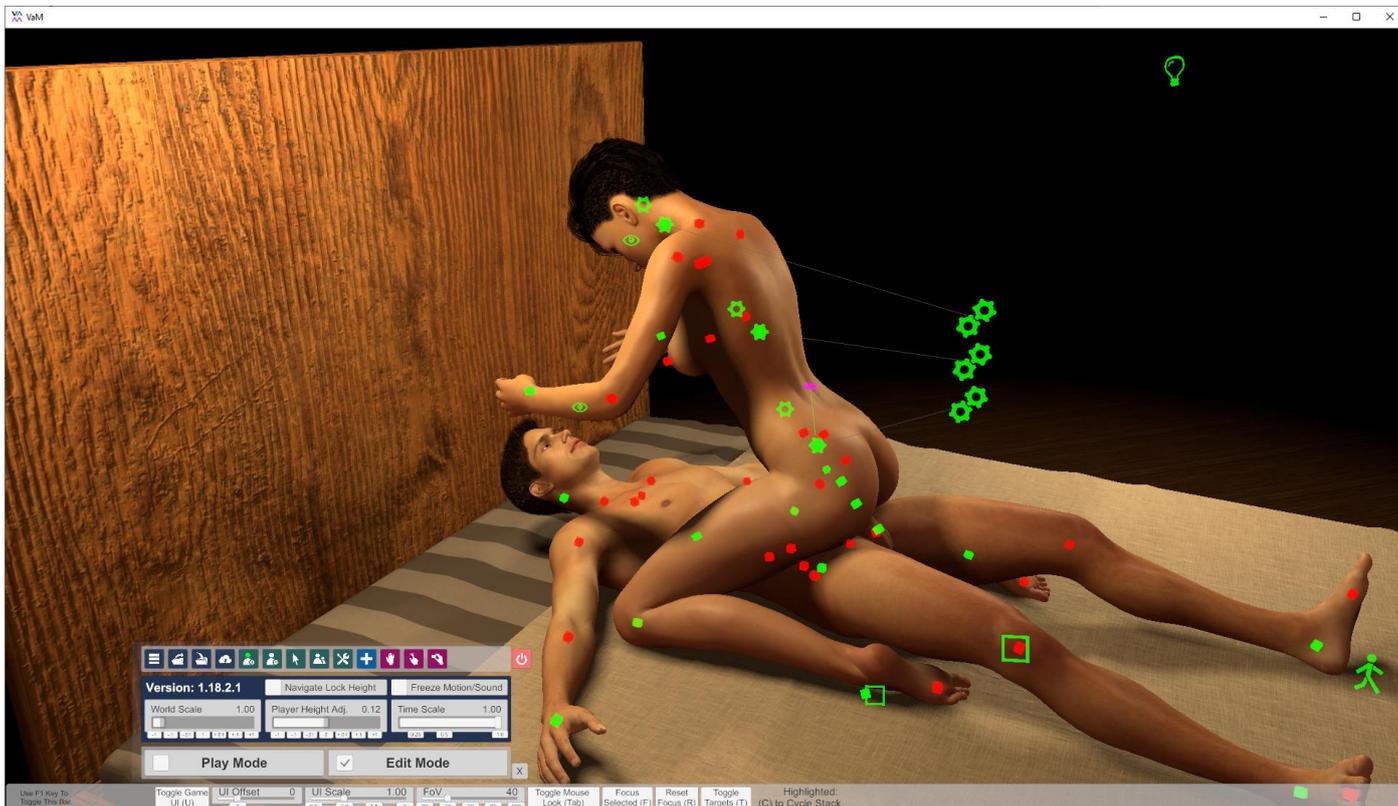
- *Breathing* controls the selection and scheduling of breathing audio files. It's the core of the plugin, providing data for a number of connected "Drivers".
- *DriverBreathing* controls the animation of chest, stomach, nose and mouth in sync with the breathing. You should see the girl breathing already.
- *DriverThrust* allows to move parts of the scene in sync with the breathing. This is not doing anything, yet, since we need to tell it what to do. In this case we mainly want to move the hip of the girl.
- *DriverThrustSFX* (v0.4+) plays additional sound effects like some wet lube, a slap when bodies collide and some light squeaking of the mattress. Setup of this is not covered in this tutorial, however you can check out the Life04 demo scene.

For a scene like this we want a more sexual breathing, so I recommend setting the *Breathing* sub-plugin to use the "Candy Sensual" dataset, "Original Moan" would do fine as well. Setting the intensity so some moderate 0.5 for now. We will get back to this later.

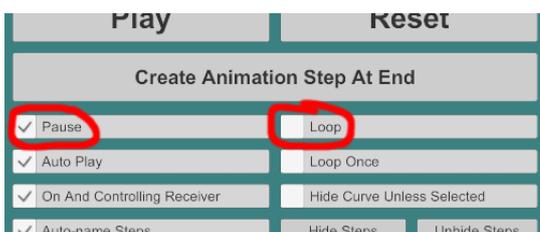


DriverThrust is able to control an arbitrary number of *AnimationPatterns* and/or *VariableTriggers* in the scene. I'm placing three *AnimationPatterns* with two *AnimationSteps* each, providing a range of motion for *Hip*, *Chest* and *Head* controls. You can have more *AnimationSteps*, if you like to create a curved motion. However, in the end you need the 0 position of the animation to be the "inside" position, while 1 is the "outside" position. Any animation beyond 1 is going to be ignored by the plugin, so you might need to tweak the transition time of your *AnimationSteps* to create an animation of length 1. As an alternative you can also use a *VariableTrigger* to remap values to whatever range you need.

As a side note, for a slightly better body position I decided to parent the *Abdomen2* control of the girl to her *Hip*, so they will animate together. We could have used a fourth *AnimationPattern* instead as well.



Each *AnimationPattern* is set to be paused, since the plugin will take over the animation. In this case, with only two *AnimationSteps*, we also want it to be not looping:



Because we want the chest control to be able to rotate for breathing animation, that *AnimationPattern* needs to be set to control the position only. Same goes for the head, since we want to add a *Gaze* plugin later which will do some idle rotation:

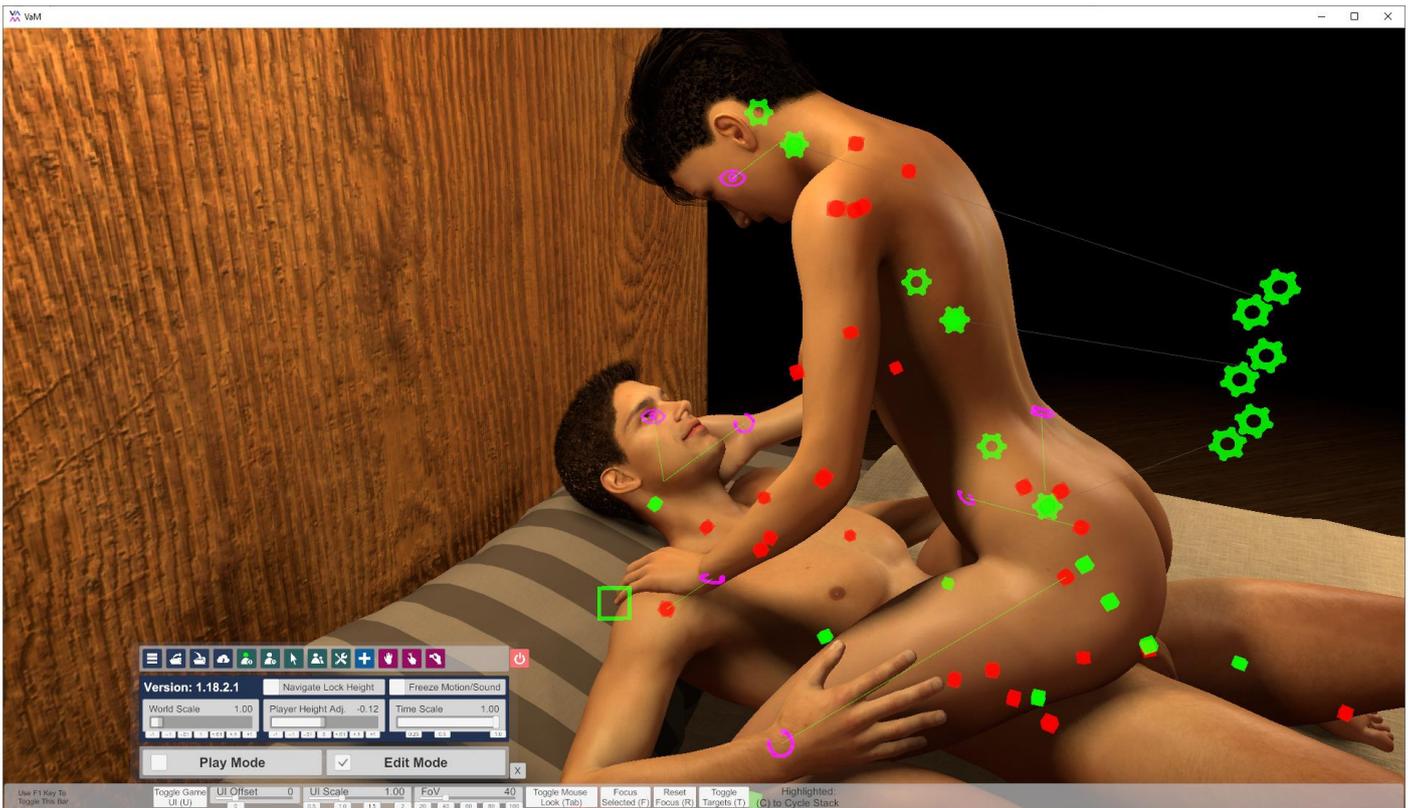


Linking to the *DriverBreathing* plugin happens automatically by atom name, it's just looking for anything named `ATOMNAME#ThrustSOMETHING`. Since our girl atom is just named "Person", we go with three *AnimationPatterns* renamed to: "Person#ThrustHip", "Person#ThrustChest" and "Person#ThrustHead". As soon as you rename the atoms, the plugin will start animating them.

Sensual connection

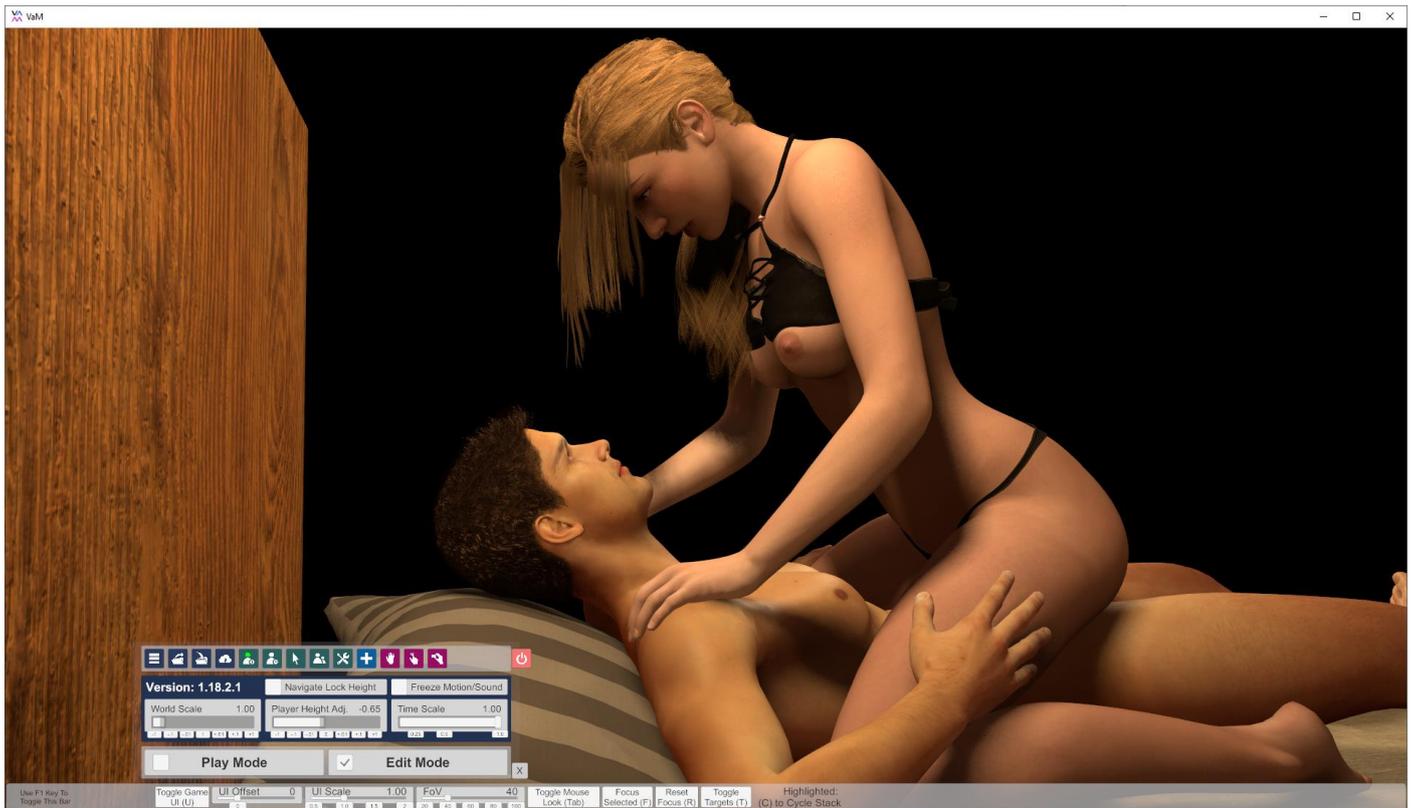
As mentioned in the beginning, I'm aiming for a sensual cowgirl scene. To achieve that, we want the happy couple focused and touching each other. I recommend to disable the *Breathing* plugin for a moment, linking characters to each other is easier when they are holding still. Just go to the girl's "Plugins" tab and disable all the checkboxes.

- The guy gets a *DSBR_BedPillow* as head support, to be able to look better at the girl. Keep in mind that in VR your field of vision is rather limited. If a player would happen to possess the guy, he needs to be able to look properly at the girl.
- Both characters *EyeTargetControl* are moved between the eyes of the respective other character and parented to the "head". I recommend parenting to "head", not "headControl". The latter is where you want the head to be, while the former is where it actually ended up to be.
- For both characters go to the "Auto Behaviors" tab and set "Eyes Look At" to "Target", so they actually look at their *EyeTargetControl* instead of the player.
- Put a *MacGruber_Gaze.cs* plugin on the girl and enable "LookAt EyeTarget" there as well.
- If you like you can additionally put a *MacGruber_Eyes.cs* onto the girl, it makes the character close the eyes for a few seconds every now and then. Just another trick to make this a bit more sensual.
- I rested the left hand of the girl onto the shoulder of the guy and parented it to the shoulder.
- Same with the girl's right hand, its parented to the "head" of the guy. Only the finger tips are actually touching the head, while the palm stays a bit away. This is to avoid the hand being awkwardly *inside* your head when you possess the guy.
- Lastly, I parented the hands of the guy to the thigh controls of the girl. This will make the guys hands move nicely with the girl's motion.



Adding detail to the scene

Time to get some more details into the scene. I start with a nicer look for the girl. Since I want to record a video of this demo later, I need to think about a position for the camera. I choose a position somewhere like here:



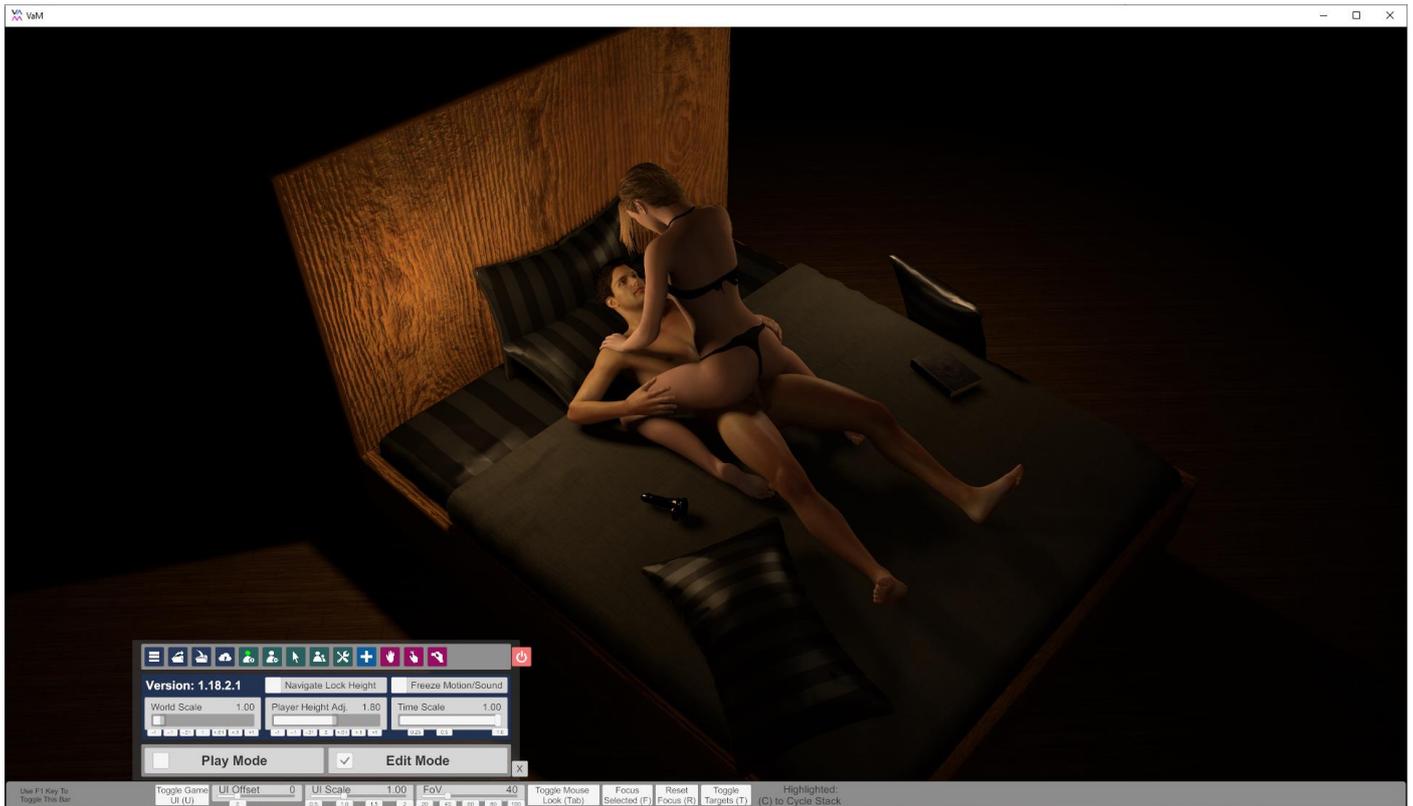
Look: For the basic look of the girl I just go with a mostly default. I know there are glorious looks out there, but this is about the plugin, I don't want a look clutter up the package with skin textures.

Glute Physics: Since the girl's ass is rather wobbly from the plugin movement, I adjusted the "Spring" setting a bit in the "F Glues Physics" tab. A value of 27 looks much better in my opinion.

Hair: Because I want the eyes of both characters to be visible, we need a haircut for the girl that allows this. I could use the old non-physics hair, but that would be boring. However, it was also difficult to find hair that supports all the movement in this scene and still look physically okayish. As you know I ended up using one of Roac22's glorious hair packages.

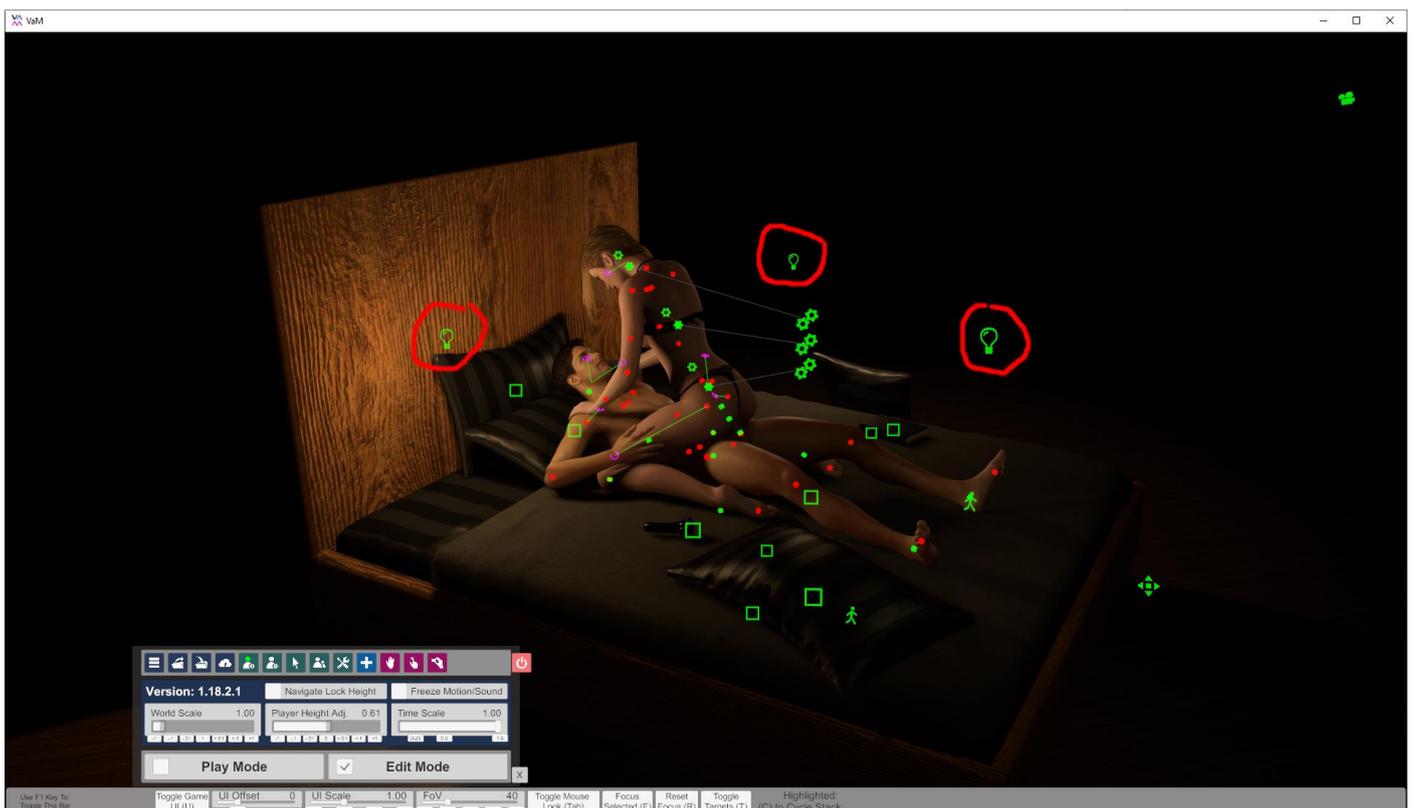
Clothing: In my opinion a scene looks better with more details. That's why my girl is wearing clothing, while not actually letting it cover anything. I really like the default "Heat Up" top and panty for this kind of thing. Of course, colored black/grey to fit the rest of the scene.

Props: To make a scene less boring I always distribute few of the props in the scene. Usually I would load stuff from AssetBundles, but here we restricted ourselves to the default props. Materials of the bed sheets and pillows have been changed to make them look less “default”. Adding a *Dildo* is also nice, telling a little story of its own.



Lighting: For the lighting I go for a classic three-point lighting here. Not just because VaM allows 3 light sources by default, but also because that is actually common artistic thing to do. [Check Wikipedia for more details](#). I placed the lights in a way that they work well from the intended camera perspective, but also when someone should possess the guy. For performance reasons the lowest light, the one on the back, has it’s “Cast Shadows” disabled.

I wanted a rather dark scene, so the global scene lighting settings can stay at their default values.

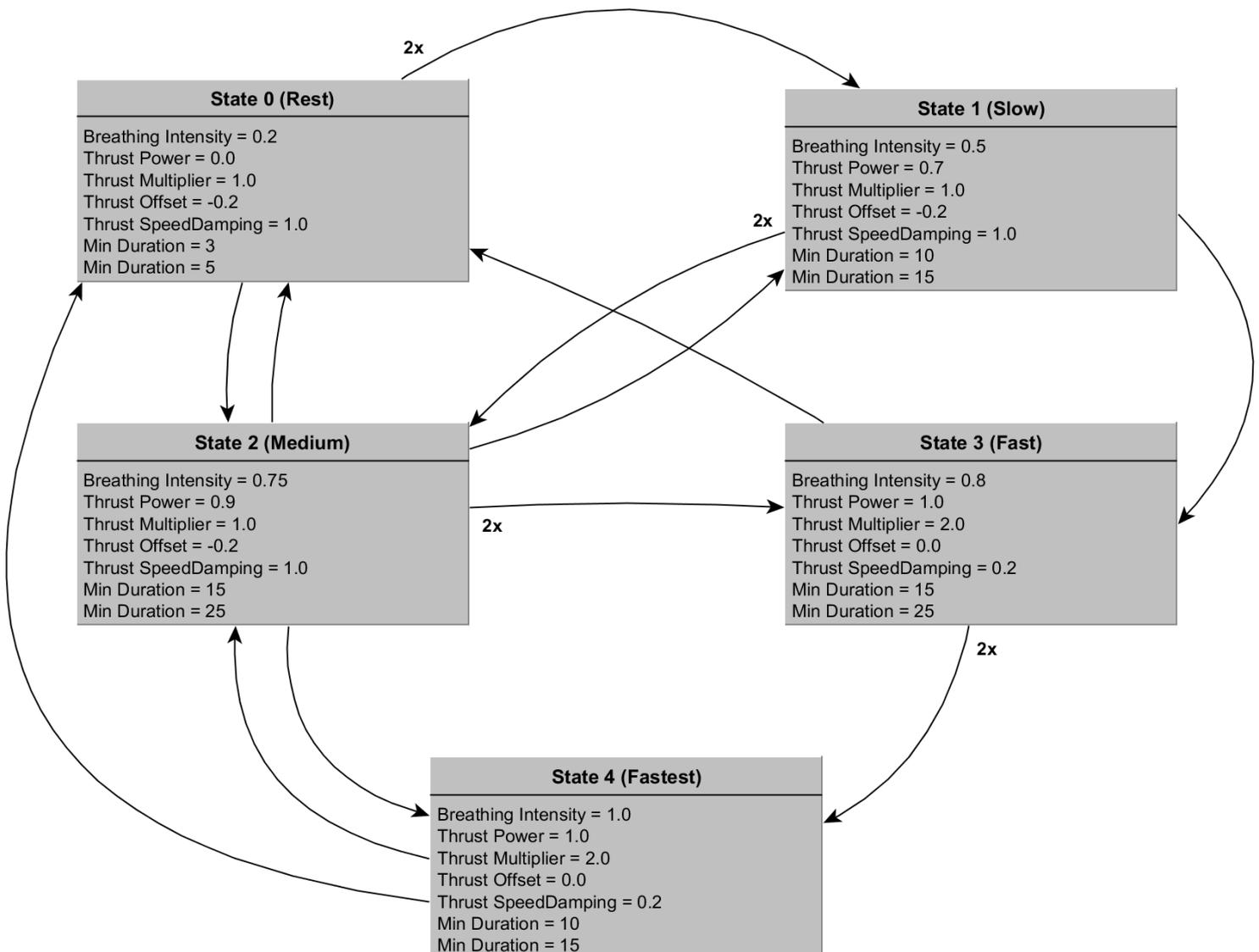


Scene Controller

The big benefit of using the *Breathing* plugin is that you can tweak the values at any time. That means we can change values while the scene is running. So, let's do that! Let's have scene controller script that tweaks values at runtime. It will make the girl faster or slower with some randomness. This will make the scene a bit different every time and it won't appear like just a looped animation playing as there is no repeating pattern. Another effort to make the scene feel more alive.

Conceptually my scene controller is a so-called "state machine" with 5 states. When the scene is loaded the controller starts in state 0, a rest state without thrust motion. This is because I set the "power" slider of the *DriverThrust* plugin to 0 for this state. Also, it sets the breathing intensity to a rather low value. The controller remains in a state for a random duration between "Min Duration" and "Max Duration" seconds. After that time, the controller picks one transition at random from the diagram below. Transitions labeled with "2x" are just there twice, resulting in a double chance to be chosen.

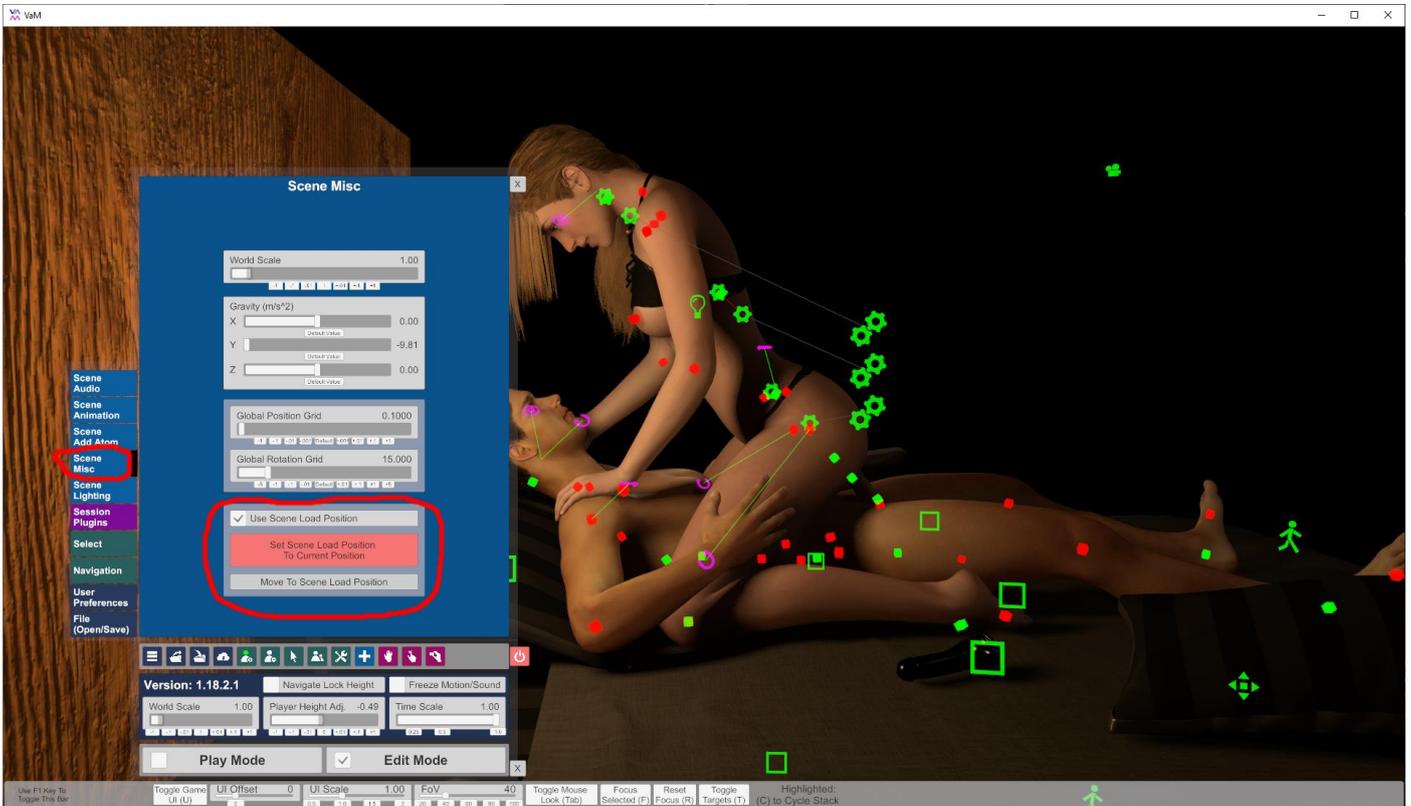
All of this is happening in a simple plugin named *Scene_Life03.cs*, which is also attached to the girl. This is intended to be a scene-specific plugin. If you create your own scene, you likely want to copy and modify the script to allow using your own tweaking values. Non-coders, have no fear! All the state data is just at the end of the script. You can easily tweak values, add more states or change transitions. If you feel up to it, you can of course also add different parameters to tweak.



Preparing for shipping

There are two things I recommend when making the final save on a VaM scene you want to share with others:

First, I recommend to setting a “Scene Load Position”. Usually VaM would use the position the camera was at when you took the thumbnail screenshot. However, the resulting position seems to differ between Desktop and VR mode. Scene Load Position appears to be more reliable. Also, it allows you to shoot the scene thumbnail from a different position.



Second, I recommend all animation to be saved in a position your characters can recover from when loaded later. In this particular scene, the scene controller will start the scene with the “rest state”, meaning the cock will be fully inside her. Because of that the scene should be saved in that position. Otherwise you the body would snap into position when loading the scene later, potentially causing the feared “explosion” of characters. The easiest way here to achieve that is to wait for a slow period in the animation and save the scene (= taking the thumbnail) exactly at a moment the girls is fully in the down position.