

November 2023 Progress Report

A new stable release is available on [GitHub](#)!

Reminder: You can manage your Supporter rewards (including your credits name, if you're eligible) on the [supporter page](#).

Constraints

No progress on the constraint tutorial yet unfortunately, I decided to focus on some quality-of-life changes first:

Constraints can now be created between actors/properties by using drag-and-drop, similar to SFM:

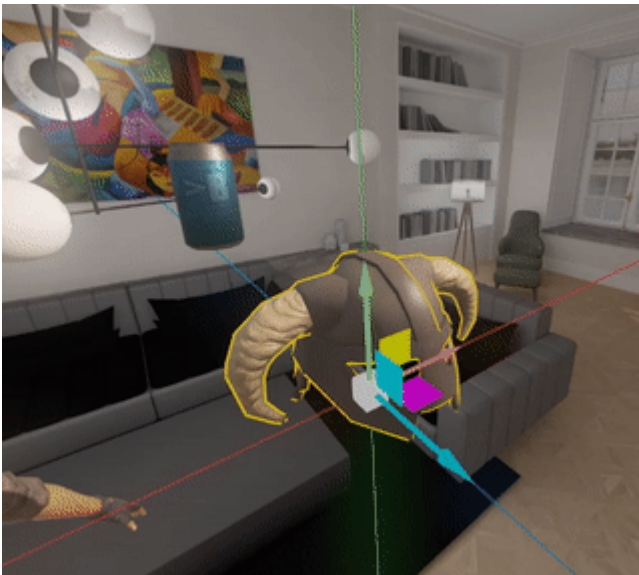
A constraint can be created between any two properties that are compatible (i.e. two vector or rotation-based properties).

I also added several new variants of the child-of constraint:

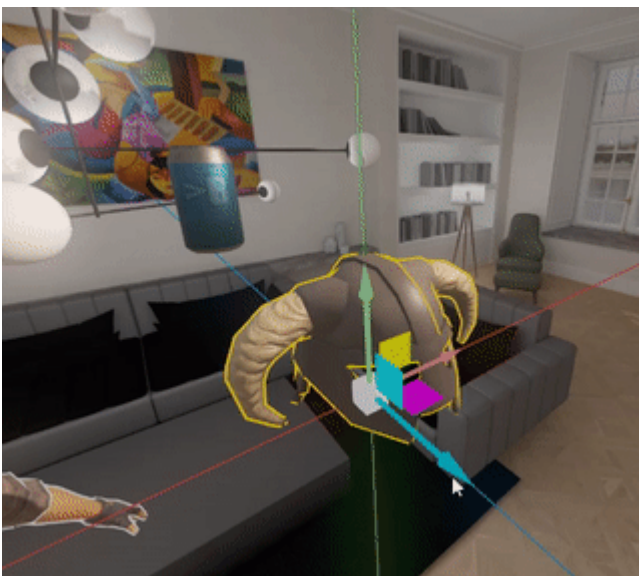
- Regular child-of constraint (equivalent to SFM's locks):



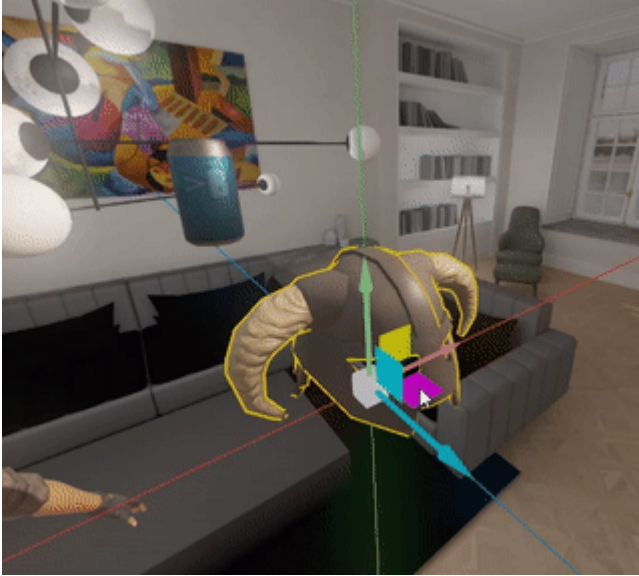
- Positional child-of constraint (Rotation of child remains unaffected):



- Translational child-of constraint (Child is affected by parent position, but not rotation):



- Rotational child-of constraint (Child has the same rotation as the parent, but its position is unaffected):



In addition, if you use drag-and-drop to place an object from the model explorer in the scene, the object will be automatically attached to the object under the cursor, if the two objects are compatible:

For example, if you have a hat that has a matching bone name with an actor in the scene, like in the example above, you can just drag-and-drop it onto the actor and the hat will be attached to the actor automatically using a child-of constraint. You can hold the alt-button to disable this feature.

Please ignore the delayed movement of the hat in the video, that only happens while moving the actor and will not be a problem during playback/rendering.

There's still a few small improvements I want to make to the constraint-system, but I'll get around to the tutorial soon.

Render Previews

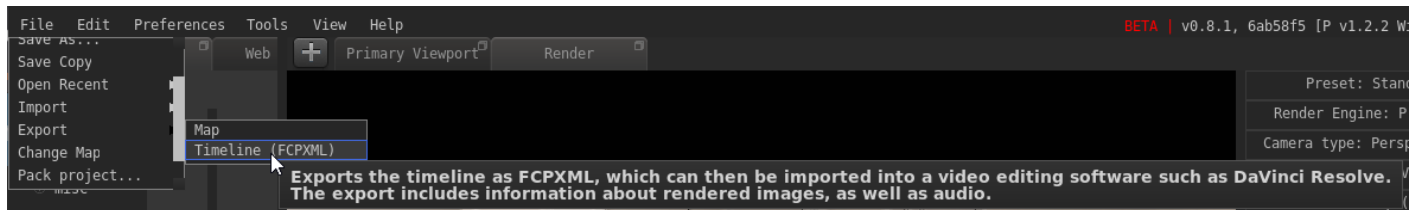
You can now view all previously rendered frames in the render panel:

Previously it would only show whatever frame you rendered last in the current session, but the render panel can now be used as a sort-of video player for all frames that have been rendered for

the project so far. The playback quality is significantly lower than the actual image quality of the rendered frames, but it's only meant as a preview anyway.

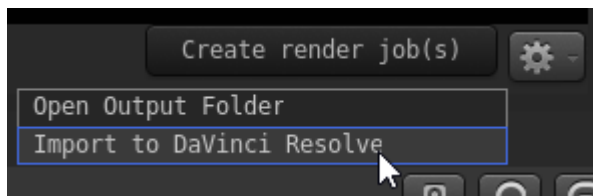
Timeline Export

I've mentioned in my last post that I was working on a timeline export feature for DaVinci Resolve. I decided to built on that, and added a timeline export feature for the FCPXML format, which should be supported by most video editing programs (I've only tested it with DaVinci Resolve, however.):



This should, in theory, allow you to import the rendered image sequences and audio in the PFM timeline into most video editing software. Please let me know if there's issues with it with whatever software you are using.

For DaVinci Resolve, there is also a special import function in the render panel:

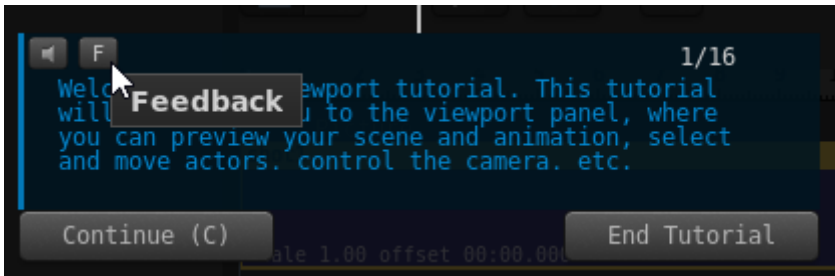


This just makes it a little easier by directly launching DaVinci Resolve (assuming it's installed), then you just have to run a script within DaVinci to import the timeline automatically (as I've already showcased in my last post).

I've updated the render tutorial with new information about the new render preview, as well as the timeline export. You can check it out in the latest release!

Tutorial Feedback

I've enhanced the tutorial system by a direct feedback functionality:



Using this button you can provide direct feedback for a specific slide/tutorial:

Feedback

Please use the text-box below to describe any issues you had with this tutorial slide, if it was unclear what to do, or if you have a suggestion on how to improve this slide. Please limit your message to 200 characters or less.

Enter your message here.

OK Cancel

I will collect any feedback I get and try and use it to improve the tutorials for the future, so please make use of it!

I've also fixed a bunch of AMD-related crash issues, so if you ran into any of those, it should hopefully work now!

Revision #12

Created 2023-11-20 09:55:38 UTC by Silverlan

Updated 2025-01-19 14:13:14 UTC by Silverlan