USD WG - 2023-05-24

Conference call details

https://zoom-lfx.platform.linuxfoundation.org/meeting/96955375634?password=f8dd722d-2a2b-41bb-9fc7-f09aacaf11b2

Agenda

- Sub-working group updates
    - Camera
    - MaterialX
    - Assets
    - Web Visualization
    - Games
- USD proposals roundup
- Paolo Selva & Kimball Thurston: *USD modalities at Weta*

Notes

- Sub-working Group Updates
    - Web (Phil)
        - WebGPU features were enabled in browsers
            - Cleaner pass of webAssembly, simpler setup
            - 64bit support in chrome and ff are behind feature flags
            - Some of the hacks from earlier attempts for web assembly implemetations will be deprecated soon.
    - Assets (Jeremy)
        - Awesome discussion around Weta presentation of assets
            - Validation guidelines
            - Asset structure
    - Camera (mmin)
        - Next monday's meeting might be a little lite because of cinegear expo.
        - Intend to discuss attributes in openEXR
- Games
    - No update
- MaterialX
    - No update

- USD self-assembling thingamajigs (Nick)

    - Mathew Kuroc - sometimes users struggle with model hierarchy.
        - Added some things to help with this.
        - You don't have to keep going up and down the hierarchy
        - Check it out here
            - https://github.com/NVIDIA-Omniverse/USD-proposals/tree/layerSourceInfo/proposals/layerSourceInfo
            - https://github.com/NVIDIA-Omniverse/USD-proposals/tree/assetStructureProfiles/proposals/assetStructureProfiles
- VFX Cafe in Montreal (Alex)
    - <crickets>
- USD Modalities at Weta (Paolo & Kimball)
    - USD Structures - Importance of identifying "intents" - Paolo - want to keep it short
    - Want to try and discuss the intent of structures
    - Ex: What is the best USD structure for rprpreesznting 1M cubes?
        - Can we all agree on the intent and structure among different companies?
            - Some groups within a pipeline may have varying structures depending on function and intent
                - Perhaps enable departments to be able to go back-and-forth
        - Questions:
            - Do we use classes?
            - Where is the material created and assigned?
            - Where do we set Instanceable?
            - How do artists manipulate individual elements, instances and materials?
            - Is it "geo" or "geom" or "mat" or "mtl"? And where are they defined?
            - How many layers are too many layers?
            - <more questions>
    - Hope that others can contribute to the discussions are structures or API's around the structures.
    - Nvidia has some discussions similar to this
    - <graph is presented on Weta workflow of assets>
    - Different asset attributes can exist from different departments
        - Lighters vs Layouts
        - Different departments can update different parts
            - May have different Opinions
    - Convert existing assets/elements/products to USD
        - Exchange data with Clients
            - Agreed intent of "sharing data" made it "easy" to have an agreed data structure
                - Used through Atlas

            - Wasn't perfect
        - Over the years, the question becomes more complex with the varying use-cases among departments
            - The structure of exchanging data was becoming more conflicting with the internal use-case among departments
    - Import/Export data into/from hosts/DCC's for internal use
        - Custom renderers (Manuka, Gazebo) and custom shaders
        - Lots of Weta geometry-procedurals for trees, hair, braids, feathers
            - Atlas was made of custom tools, so allowed for easier conversion to USD
- Provide a new Lighting Workflow
    - Enable Solaris workflows for USD authoring in Lighting
    - Enable blended-workflows with Solaris and Katana
    - The Lighting in Solaris Intent
        - <graph is presented>
        - Model, Hero, and Proxy are varying representations
            - Blocking may be super-polygons
                - And LOD that doesn't share materials, etc.
        - Within Maya, export out USD-ified geometry
            - <graph is presented>
        - The root element is a kind asset element
            - Want to give it an actual name, how we identify things within a scene
        - Abstract frame of class representation
            - Then we add all of these as variants to "purpose frames"
                - Sculpt frames with the names of the purpose
                - Purpose render, purpose proxy
        - "Why all the variants and the representations class?"
            - Switch each purpose to any of the representations independently
                - Load unloaded (payloads off)
                - Everything is variant 'impostor" - load as many variants
                    - Good for far-distance camera, but when getting closer, it becomes a bit useles
                - Control each element independently
                    - Allows for specific toggling
                    - Bake out shaders, auto generated for all lower-res versions
                    - Add a sub-set of variants

        - No need to switch to a variant to see its content
            - Can see all the sub-variants without switching
            - See variants nodes within Solaris, etc.. without switching context
        - Same variant loaded in both purposes could have different opinions
            - Edits applied on the “p[rpose” prims are applied only to the variants loaded in those purposes
            - What you see in the proxy-purpose doesn’t necessarily need to affect what you see in the render-purpose by default
        - Can modify all the modelHero in both purposes by addressing the “representations/modelHero” prim directly
        - Q: is extra prims count coming from the “prepresenations” an actual issues?
            - Yes, but what is the cost of not having?
                - Versus what is the time cost of having to modify various compositions
                - Still asking this
        - <What about the Layout-structure?>
            - <image w/ hierarchy of structure is shown>
            - All assets used in the layout are under an abstract prim called “_class_”
                - All the unique assets are listed in _class_
                - Each “element” of the layout inherits from its own “asset class”
                - Actual asset content loaded in the classes
                    - You can do all your edits within the classes frames
                - This allows for global edits via classes
            - Every assetElement in the layout inherits from a class prim that can drive its instanceable metadata
                - Based on the variants, you can enable or disable the instanceable flag
- Recap: An artist can go on a specific element, with a specific purpose
    - Render that specific element
    - Can apply globally via representations class and edit modelHero variant

            - Can apply on the Class, affects all prim purposes and modelHero variants
        - Enable FX/Layout workflow in Lighting in Solaris
            - FX already in Houdini, can we enable Solaris for them?
            - Could we speed up workflow
        - Rethink how assets/elements/products are published
            - Fast results, but not fully embracing USD features
            - Can we enable sparse edits now?
            - Ex: creature-based opinions merged with lighting opinions?
        - Get USD in the guts of the pipeline in all DCC's
            - Should the data travelling in the pipeline be USD
            - Are all vendors ready with their DCC's?
            - Do not want a "Weta USD"
            - Not stuck with one DCC
            - Maybe we can all come up with a baseline approach, and even agree on the intent
    - That is pretty much the presentation, any questions?
        - <James Pedlingham> you say you create a feature-scene that is its own element, its own layer. How do you relate the assets that were set up in the elements with all the various references so that they are all inheriting from /classes?
            - <Paolo> currently each published layout has a layout file, which is basically all the hierarchy of transforms, along with a library file, which is the class. We are basically relying on different libraries from different shots will be basically fine. We are looking at how much that is an impact. Say two master layouts, with subsections from different layouts, could easily have couple of hundred files. Definitely bring in the same classes. Understood to be expensive. The intent is that the path will be fully-resolved when that layout is brought into the shot.
            - <James> Now I understand that it is built with that one shot layout /hierarchy
            - <Paolo> yes we split everything in chunks
            - <James> Makes sense, thank you
        - <Spiff> Great presentation. It seems like the way you are laying things out as classes is kind of scratching at how we tradoff variant sets. It sort of works out ok with limited (one) variation. If you have an asset that has like, four, variant sets, do you think there would be a need to unroll that combinatorial explosion of things? Or perhaps have a practical limit on how the artists can work with these variants?
            - <Paolo> Yes we don't expose the sub-variants of the assets class. We are going to see if the requirements will . If too

many representations can be flattened, or how they want to visualize the representations. Would that be a problem?

- <Spiff> You could make an argument that it is only the Representation that could be free floating and changing based on workflow (we call it Standin at Pixar). If you want to avoid all modeling variants of a model, you kind of need to know which one was selected
- <Paolo> Yes one example that came up was with the Lighters, someone could render a proxy in an offline shot, maybe using blocking geo that is good enough for occlusion. Having a way to see what new representations were being referenced for a specific pass might be useful in a custom-layer. Maybe we are overusing this to see where everything is accessible, possibly good at the start. We'll see.
- <Shawn Dunn> I thought you cold have chair and chair broken variant
    - <Paolo> this is another name-clashing problems. Internally we use variant and sub-variant as part of the name of an asset. Where Chair is the variant and Chair-Broken would be the subvariant. In this case, there would be different entries in the class
        - We use Variants as way for users to pick different representations
        - Just because there is a representation, that doesn't mean it's loaded.
- <Alex> How much of the intent should be encoded in the USD files? If you try to enable all the intents for all the various artists, would that be too much?
    - <Paolo> that is a good question. Is there a common Intent that we can all agree upon? The Lighting intent may layer a Lighting Intent on top of everything. The benefit is that there are localized opinions. If you start bringing in new Intents, since we use Solaris, others would be aware of it. We went with everything is described "here", and when y ou make a change, then that is a localized intent
- <Christian Lopez Barron> When modifying a model asset in a scene, are there helper nodes? Or do they go directly to the hierarchy?
    - <Paolo> There is a nice feature in Solaris that allows you to go through the various levels. At least in Solaris, you get this for free. There is an option in the viewport that lets you select the class prim rather

than the actual prim. This year we are taking a step back and looking at where we can optimize USD usage. The nice thing about that is that you can have different UI's based on different departments. We might need to provide simplified toolsets for different artists, such as Layout and Lighting
        - <Christian> Are the different opinions for different departments kept?
            - <Paolo> That would be preserved in the Layout representation for whomever is reading the shot content. Pulling the data as a shot content
    - <Alex> We are out of time. Any last minute questions?
    - <Wave> Thanks for sharing, very generous of you guys
- End of Meeting

<from Chat>
Jeremy Cowles to Everyone (May 24, 2023, 1:08 PM)
Proposals mentioned today from @Matthew Kuruc and @Aaron Luk

https://github.com/NVIDIA-Omniverse/USD-proposals/tree/layerSourceInfo/proposals/layerSourceInfo

https://github.com/NVIDIA-Omniverse/USD-proposals/tree/assetStructureProfiles/proposals/assetStructureProfiles

Matthew Kuruc US to Everyone (May 24, 2023, 1:10 PM)
Here is the proposal for self assembling model hierarchy (https://github.com/PixarAnimationStudios/USD-proposals/pull/10)

geffes to Everyone (May 24, 2023, 2:02 PM)
Thanks!

Jessica Wang to Everyone (May 24, 2023, 2:03 PM)
Thank you Paolo! 👋🏻

Action Items

- …