USD Web Visualization

Calendar

Slack Channel

Zoom meeting

USD on the Web prototype can be tested here.

Wednesday, October 4th 2023

- Alan presenting project to perform animations workflows using front-end UI, submitting jobs to back-end server ultimately serving Users back with baked animation.
- Revisiting productization steps of WebAssembly prototype, including actionable steps to submit a first end-to-end workflow building the lib `arch` via Emscripten from OpenUSD 23.11
    - Based on suggestion from Nick, base branch supporting integration can be made from the soon-to-be-released OpenUSD 23.11.
    - Autodesk to submit Dockerfile already used for recent WebGPU prototypes, known to be working (~spring 2023)
    - Iterative work to be done in order to include modifications to `build-usd.py` script and `cmake` files, in order to integrate Emscripten builds.
    - (Other details available on the itemized productization list.)

Wednesday, November 2nd 2022 (11:00AM PT/2:00PM ET/9:00PM GMT)

- Anything interesting to show or discuss?
- Proposed agenda:
    - High-level overview of the productization/maintenance work of the WebAssembly prototype

Wednesday, October 5th 2022 (11:00AM PT/2:00PM ET/9:00PM GMT)

- Anything interesting to show or discuss?
    - Proposals for wider color gamuts on the web
        - https://www.w3.org/TR/css-color-4/
        - https://css-tricks.com/new-css-color-features-preview/
        - https://lea.verou.me/2022/06/releasing-colorjs/
- Proposed agenda:
    - Opportunity to share list of priorities with the Google Chrome team.
    - Go over a state-of-the-union on WebAssembly, SIMD, WebGPU, threading, memory, etc. to establish/coordinate on priorities

Wednesday, September 7th 2022 (11:00AM PT/2:00PM ET/9:00PM GMT)

- **Special guest:** Brent Scannell (Sr. Product Manager at Autodesk, Khronos glTF Working Group)
- Notes:
    - Benefits & challenges of the vendor extensions in glTF
        - Contribution model intended to make it simple for vendors/implementors to prototype/share extensions to glTF
        - Process is rather informal, may eventually evolve to a whitepaper/RFC model to gather feedback from the community ahead of submitting extensions
    - Benefits & challenges of not offering a format "glTF runtime", focusing on model definition instead
        - Great for flexibility, getting started contributing
        - Can be a challenge in ensuring assets behave similarly in different implementor runtimes
    - Details about the glXF proposal
        - Intent is to bring greater interactivity to glTF experiences
        - Supporting scene-level interactions, as well as interaction between assets ("smart assets")
        - Intent is to compose scenes via referencing, using glTF assets, or other scenes
        - No particular prescription about the level of granularity of assets at this time.
        - Intent is not necessarily to dive into "simulation workflows" at this time (e.g. car collisions & deformations), but research in this space is welcome
    - Commonalities between USD & glTF communities
        - Topics about where the "authoring runtime" ends and where the "consumer runtime" beings
            - e.g. where would a hypothetical Unreal/Unity engine meet an embedded hypothetical USD/glTF runtime.
        - Challenges about asset validation, asset structure, best practices
            - Established test suite for validating conformity of glTF assets against expected behavior
            - Challenge in agreeing which renderer is the "most accurate" when conforming across vendors
        - glTF has rich transmission, last-mile delivery format ("The JPG of 3D assets")
          USD has rich composition workflows ("The PSD of 3D assets")
          Opportunities for communities to avoid solving the same problems each on their own, and meet where the formats overlap.
- Invitation:

- Online webinar on Delivering Interactive Experiences with glTF, September 27th, 2022

Wednesday, August 17th 2022 (11:00AM PT/2:00PM ET/9:00PM GMT)

Proposed agenda:

- Post-SIGGRAPH 2022 regroup
    - Aligning with Assets Working Group regarding validation, JavaScript-to-WASM bindings for USD, etc.

~~Wednesday, June 15th 2022 (11:00AM PT/2:00PM ET/9:00PM GMT)~~ Wednesday, July 13th 2022 (11:00AM PT/2:00PM ET/9:00PM GMT)

Note: Proposed agenda rescheduled to next meeting due to audience particularly interested in USDz validation unable to attend.

- Anything interesting to show?
- Proposed agenda:
    - From previous conversations:
        - USDz support can require significant effort for validating assets, and is a manual process.
        - Proposed action item: combine efforts to create a validation tool, if issues are difficult to raise to vendors/implementers.
    - Potential questions:
        - Have the issues resolved at the source (🙂)
        - Enhance existing `usdchecker` vs. develop a different tool?
        - Which runtime versions should be supported/specified?
        - etc.

Wednesday, May 18th 2022 (11:00AM PT/2:00PM ET/9:00PM GMT)

- Follow-up on conversations from previous session about comparing glTF and USD for visualization
    - Eric Chadwick provided a description the Wayfair workflow:
        - Relying on 3ds Max and Maya to author and design content, using mostly VRay materials
        - Baking down textures and geometry to later package assets in USDz format
        - Strong overlap with parallel conversations about LoDs, texture/mesh compression also covered in main Working Group and *Ga mes* sub-working group
        - Regarding interactive experiences (mobile/AR/VR/XR), fast loading time, rendering-optimized content is a must
            - Users/downstream consumers are unlikely to stand with their devices while waiting for content to load
            - Experience may be jarring if content suddenly appears in the world, or partially loads
- Early conversation about combining efforts regarding asset compliance/validation for Apple's USDz format
    - Avoiding reinventing the wheel by sharing tools/scripts to flag potential known issues with Apple's interpretation of the format
    - Perhaps enhancing the `usdchecker` tool to support more options, or take inspiration from the tool and build a separate/specialized one

Wednesday, April 20th 2022 (11:00AM PT/2:00PM ET/9:00PM GMT)

- Touch point on the principal focus of the initiative:
    - Common agreement that the focus of the project is not to compete or dethrone glTF as the transmission format for assets and content on the web.
    - glTF has specific targets, use cases and built-in optimizations specifically targeting transmission and "last-mile" delivery of assets, which is less of a concern for authoring formats such as USD.
        - Nick Porcino mentioned intent to revisit his *Last mile vs. interchange* article, to update the content based on latest industry development
    - It may be worth revisiting some of the existing documents published to state this more publicly, as this none-competitive aspect may otherwise not be clear to the audience outside the USD Working Group.
    - Potential future work may include publishing a "best practices" guide regarding publishing USDz assets for the Web, covering topics such as asset complexity/density, texture optimizations, known limitations, etc.
- Alexander Schwank shared updates:
    - From the MaterialX Working Group, about Wayfair sharing their experience transitioning from a glTF pipeline towards a USD one:
        - see group meeting notes: April 7, 2022
    - From Intel, about the Sponza scene, available in glTF, USD and other formats:
        - https://www.intel.com/content/www/us/en/developer/topic-technology/graphics-research/samples.html
        - https://www.intel.com/content/www/us/en/newsroom/opinion/intel-graphics-step-up-research.html#gs.xfzxkf
- Potential groups worth reaching out to regarding USD "runtime" features:
    - Sketchfab
    - Shopify
- Potential Questions for the larger USD group:
    - Does Autodesk envision any specific use cases for WebAssembly? Any gaps, limitations with the current prototype?
        - Jerran Schmidt- While data visualization is one component, I did mention briefly in the HgiWebGPU presentation that web-based USD content authoring applications are a particular use case for this. (*Note: I no longer represent Autodesk on this)*
    - Does Pixar see any potential openings for platforms other than X86/64-bit architectures?

Wednesday, March 23rd, 2022 (11:00AM PT/2:00PM ET/9:00PM GMT)

- Demonstration of an early prototype of HdStorm built to WebAssembly

- Demonstration of an early prototype of USD WebAssembly running within A-Frame.io :
    - A-Frame.io USD Wasm.mp4
- Recent noteworthy USD WebAssembly updates:
    - Allow dropping of USD/USDZ files on page to load them
- Project proposals to bring back to the larger USD Working Group, either for standalone value or for inclusion within other projects:
    - Asynchronous HTTP(S) ArResolver

Wednesday, November 17, 2021 (2:00PM PT/5:00PM ET)

- Video Conference Link

Wednesday, October 20, 2021 (2:00PM PT/5:00PM ET)

- Low traffic
- Pitch for a topic of next meeting about glTF & USD
- Discussion on Materials (MaterialX, aiSurface in browser)
- Update on Autodesk & legal dept.

Wednesday, September 22, 2021 (2:00PM PT/5:00PM ET)

- Post discussion on <model> tag

Wednesday, August 25, 2021 (2:00PM PT/5:00PM ET)

- Demo of assets viewing and interaction in Autodesk Forge in Chrome
    - Rendering, selection, and scene-inspection
    - A lot of work has been done on bindings
    - Custom changes to WebASM build of TBB

- ~~Video Conference Link~~

Wednesday, July 14, 2021 (2:00PM PT/5:00PM ET)

- Conversion from **USDZ** into blender and out to **GLTF** is a great short term solution, but...
- Worth doing test for conversion of animated models over time.
- Makes a compelling case for a native **Sdf** plug-in for **GLTF** import/export.
- Past initial mesh import, would **USDSkel** and **GLTF** bi-directional conversion be lossless?
- **Draco plugin** may need to support native **Draco** files better, or does it matter in a separate Sdf plugin?
- **KTX** as native **Hio** plugin would likely be mandatory, does **Hio** support unpacking direct to GPU texture via **Hgi**? (Brian from AMD hints at a yes)
- Is **MaterialX** and its advances (especially in tackling WebASM) going to be the way forward for all shading interchange in USD.
- **OCIO** and color accuracy again came up as important, but too large a topic compared to getting a better bare-bones implementation anyone can use.
- ~~Video Conference Link~~

Wednesday, June 2, 2021 (3:00PM PT/6:00PM ET)

- Autodesk showed USD import into Forge. Translates to Forge-native geometry.
- Autodesk also working on a WebGPU Hydra delegate that does PBR shading.
- A lot of overlap between the delegate above and Filament (i.e. SPIR-V tools running in browser)
- TBB remains an issue that everyone is keen to solve.
- Philippe Sawicki working on an MVP list for JS-bindings.

- Dhruv Govil is interested in what can be leveraged from ASWF initiative to auto-gen Rust bindings.

- ~~Video Conference Link~~

Wednesday, May 5, 2021 (3:00PM PT/6:00PM ET)

- ~~Video Conference Link~~

JavaScript bindings, need a real strategy on how to tackle them.

Static Initialization macros don't work well in WebASM, but being explicit about what is used for dead stripping to occur might be better for download times.

See what changes have landed in /dev that may help reduce reliance on WebAsm-TBB and its CORS issues.

Github issue for PRs

Wednesday, April 7, 2021

- ~~Video Conference Link~~

Working example of WebASM build of USD with a Hydra delegate for Filament

- https://github.com/marsupial/USD/tree/HdFilament
- Working on macOS & Firefox-87.0

Ideally GitHub Pages version at https://marsupial.github.io/USD/extras/imaging/examples/hdFilament would be a live demo, but

- due to the lack of 2 http headers WebASM using threads will fail to load
- a local server.py can be used to serve the branch locally instead
- Colored Quad, Sphere, and Cube are USDA. Gold monkey rotating in circle is USDZ. PBR monkey spinning around its center is glTF.
- HdFilament shader generation is minimal and **displayColor** is used for primitives. Textures on **monkey.usdz** are not loading correctly.
- USDA loaded can be edited in the HTML: https://github.com/marsupial/USD/blob/HdFilament/extras/imaging/examples/hdFilament/index.html#L119

Wednesday, March 10, 2021

☑ Roman Zulak

☑ Philippe Sawicki

☑ Alexander Schwank

☑ Dhruv Govil

☑ Alex Gerveshi

☑ Nick Porcino

☑ JT Nelson

Goal is to have USD WebAsm branch and a WebAsm build of Filament Hydra Delegate by next meeting (April 7).

- Gain a better understanding of performance and limitations loading arbitrary assets and materials.
- Let a larger audience see the size of changes needed to support the platform.
- If the changes are too large for Pixar to mainline, what options are there?
- Provide a use case for Filament that could help push HdFilament code into Google's repo.
- Initially may be a private repo/branch not totally accessible to the world.

JavaScript bindings to USD.

- Roman has started porting a ui-handle implementation to WebGL that would be simplified with re-use of Gf/math library.
- What is the cost (in terms of source code) for additional bindings.

Python support via browser (à la Pyodide)?

- Is there any gain in having existing Python scripts run in a browser?
- May be too expensive of a runtime environment.

Plugins and external dependencies for a WebASM build:

- http(s) ArResolver would be very useful once everything starts working.
- OpenSubdiv is probably a good candidate for a next-step.
- Is OpenVDB support in WebASM via NanoVDB a possibility (or useful)?
- Qt WebASM is interesting, but a little too unstable right now.