USD WG - 2024-10-09

Conference call details

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

Agenda

- Call for volunteer to take meeting notes
- Sub-working group updates
    - Camera
    - MaterialX
    - Games
    - Web
    - Assets
- Sunya Boonyatera: *OpenUSD Python Bindings update*
- Open discussion: *Opportunities to make USD adoption in DCCs easier?*

Notes

- Web Group <Phil Sawicki>
    - Met last week, about issues mentioned over the years
        - What does it mean for interactive, what bits to store
        - Game logic
            - Hovering over assets
            - Scenes vs Application
                - Transport logic + information?
        - Thought gathering
    - Thanks to everyone who joined
        - 3D pointers, but an application might not expose pointer/mouse information
    - NVidia crafted with the help of the learning team some USD content
        - Might help people who are just getting into USD
        - Built on top of the Jupyter Notebooks created during the span of the Web WG
        - Please check out or spread the word to folks who are interested in learning USD
            - https://www.nvidia.com/en-us/learn/learning-path/openusd/
- Camera WG <Michael Min>
    - Anders Langlund gave an overview on some of his work for OpenUSD & physCam

- Proposals <Nick Porcino>
    - Extended Unicode identifier proposals
        - Extended with medial hyphens?
            - People want to know
            - If we revisit this, would it count as re-medial hyphens?
    - Nat on has some considerations around text identifiers
        - https://github.com/PixarAnimationStudios/OpenUSD-proposals/issues/73

- OpenUSD Python Bindings <Sunya Boonyatera>
    - Work on removing boost::python
        - Remove OpenUSD's last dependency on boost
        - Approach: Add pxr_boost::python into OpenUSD
            - Embedded boost::python, with all boat usage removed
        - Done as a couple of days ago
        - If you open USD now, it does not rely directly on boost
        - Pxr_boost::python is enabled in 24.11
        - Usd-core 24.11 PyPI package will be using pxr_boost::python
    - NO MORE BOOST
        - Exceptions
        - Build_usd.py doesn't build boost
            - …except if using OpenImageIO or OpenVDB
        - USD build through cmake itself doesn't require boost
            - …unless using OpenVDB
        - UNLESS you REALLY REALLY need boost
            - Can revert to previous use of boost::python via:
                - "--bost-python" option for build_usd.py
                - "=DPXR_USD_BPPST+PYTHON=ON" when running cmake
            - Deprecated
                - This is for testing and transition purposes and will be removed in future release
            - First or second release of next year (25.x)
        - Impact on client code
            - No impact if you're just using OpenUSD's Python bindings by themselves
            - C++ code wrapped to Python via boost:python that deals with OpenUSD types (e.g. USD.Stage) must be updated to use pxr_boost::python
            - Schema libraries must be regenerated to update their Python bindings
        - Nerdy details
            - 451 includes of 130 boost headers removed
            - Across the board compile time improvements
        - Questions?

- <Arun Rao> Will this be a separate package in OpenUSD?
  - <Sunya> exactly, Is a library within OpenUSD
- <Alex Schwank> What's the plan for the rollover?
  - <Sunya> I'll be keeping track of the boost::python library
    - As new Python releases come out, are we going to have to fix up our code accommodate? Probably yes
    - Will probably need help from the community to help testing against new versions of Python
- <Sunya> That's it for me, thanks!
- Relighted / Godot
  - Open discussion: *Opportunities to make USD adoption in DCCs easier?*
    - Godot feedback
      - The USD library is significantly larger than Godot itself, so it's unlikely to be merged directly into the engine.
      - We are interested in eventually supporting a USD plugin/extension
      - Without widespread interest, the community or individual stakeholders need to implement and advocate for this.
    - <Nick Porcino> Is the size presumption based on looking at the build, but not accounting for all the included dependencies like tbb?
      - Would push back personally
      - <Alex> Build is 132 MB, and USD is about 50Mb
      - <Sunya> Is that with USD Core
        - <Gauillame LaForge> Yes it is not including Core
      - <Alex> Couple of questions here
        - What exactly are they looking at in terms of build size presumption
        - What exactly do they need?
        - Need documentation that says this is what you need, what you don't need
      - <Nick> I'm looking at 100Mb with various dependencies
        - Not disputing that the size isn't a concern, but whether or not its bigger than Godot…?
      - <Jean-Silas Moore> And Godot builds fast.
    - <Alex> Next question is more along the lines of
      - If a DCC developer comes to us, and they want to integrate USD into their app, should it be done through the plug-in architecture or native support?
      - <Nick> Godot is based on C++, so retracting my earlier statement
        - I think there are a bunch of things we need to check before we start recommending things

  - <Alex> Well, considering Godot is one thing, but what do we tell the general community of developers? Is the import/export plug-in strategy better, or go native?
  - <Wave> Don’t know Godot that way, but does it have “USD ideas” in it? Does it have referencing, layering, etc..? In other words, does it already align well with USD philosophies and it would be a more-natural integration? Or is it something that will only work through plug-in strategy?
    - Something we’ve been discussing with the Blender community for years
    - How does Godot consider scene graphs?
  - <Nick> Their import structure is based on Assimp, which is a heritage importer that has considerations for every asset type except USD. Very old architecture.
    - Omniverse has, for the most part, a game engine that is communicating via scene indices via Hydra so that their flow architecture doesn’t align strictly to USD architecture but allows for workflow to USD. Seems to match closest to how the gaming community is considering using USD.
  - <Levi Biasco> A lot of us are trying,
  - <Nick> Omniverse is probably a more common example to consider
  - <Alex> So we can ask things like what is your scene graph like, does it align well with USD? Perhaps put together a help document
    - So people can walk through it and figure out what the best solution would be for them
    - <Nick raises a digit, which seems to be a thumbs up> Gets a thumbs up for me, seems like the best suggestion
- Godot doesn’t see a widespread industry adoption of USD in the gaming community
  - Need someone to make a strong case for USD else they will not add it
- <Michael Min> I’m struggling to understand the
- <Peter Sicialano> That is something along the lines of what I’m perceiving (coming from the small-studios perspective).
- <Levi Biasco> Small studios don’t necessarily share the same concerns that the ‘larger’ gaming studios have
- <Alex> We should revisit the conversation with Godot to get clarity on the concern about USD integration/adoption
  - Maybe get help from the community to help write plugins, etc..

- <Scott Geffert> I’m still driving, but I’m finding this conversation very intriguing.
    - I’d love it if there’s a way to get involved with these conversations about moving these USD assets across different workflows and projects.
        - It’s a chicken-egg conversation
        - Might need just one person to step in
        - If there’s anything we can do, any entry point, would be glad to help
- <Nick> Was it specifically something about Godot, or the larger issue
- <Alex> It’s about the larger issue of USD usage and integration
    - Was brought up after a presentation from Godot at SIGGRAPH and a conversation thereafter
    - Represents the larger consideration of USD usage and integration
- <Andrew Kaufman> Share the same concerns.
    - Give me the minimal runtime requirements, I’m a developer
- <Nick> There’s a difference between the 30=pages on docstrings, versus the minimal usage considerations
- <Sunya> Certainly there are things like generating only schemas that you need
    - Feels like the distribution size issues became more prevalent
- <Alex> Could someone file a GitHub issue to track this issue?
    - <Andrew Kaufman> Sure , I’ll create one there
- <Alex> Paul did you have more to say about this issue?
- <Paul Molodowitch> The issues discussed are probably shared amongst gaming developers in general (build sizes). You could make a version of USD with standard C++ pointers that didn’t have all these groundwork pieces that might be easier to integrate into an existing engine. Just a thought.
- <Alex> It’s definitely worth considering and discussing.
    - We’ll continue the discussions with the Godot games team next week
    - Will start a questionnaire doc

- <Alex> We might have a topic for next week from Pixar. But feel free to reach out if there’s anything else you’d like to discuss
- <Alex> Thanks for joining, see you everybody

END OF MEETING

<from Zoom chat>

JT Nelson to Everyone (Oct 9, 2024, 1:02 PM)
Do we have a fun fact for before we start?

I like those, lol

Phil Sawicki (NVIDIA Corporation) to Everyone (Oct 9, 2024, 1:02 PM)
I wasn't on vacation and I also barley know what is going on 🙂

JT Nelson to Everyone (Oct 9, 2024, 1:10 PM)
https://www.nvidia.com/en-us/learn/learning-path/openusd/

Phil Sawicki (NVIDIA Corporation) (Oct 9, 2024, 1:11 PM)
@Michael Min (Adobe Inc.) Thanks JT for the link!

Alexander Schwank (Apple Inc.) to Everyone (Oct 9, 2024, 1:11 PM)

- We had a good chat about Payloads that came up from a question about selectively loading sibling payloads
- There is a big emphasis on the viewport with payloads and there is some interest in opting for draw mode over payload deferring.
- Potential activation implications between references and payloads in resolvers and there is a dependency on your file format.
- Tests during the meeting showed this may be addressed already but further tests are needed.
- Better performance had been seen in instanceable references over payloads.
- Kind issue warnings are apparent.
- Some tools help to diagnose: Houdini's Scene Doctor, NIM usdValidate
- Usd's built in checker does not check kind validity.
- Usd Author time is better but relies on tools or user education.
- Light instancing on Assets
- Failed tests for light linking and instanced lights.
- Mark Tucker suggested this is something renderer vendors have to figure out.

https://docs.google.com/document/d/1PKwrYzcmcQuQMraHtRnIDA-YqGoWcAR34rE0dATDfXU/edit

Nick Porcino (Pixar Animation Studios) to Everyone (Oct 9, 2024, 1:14 PM)
https://github.com/PixarAnimationStudios/OpenUSD-proposals/issues/73

Francois Devic (Eidos-Montréal) to Everyone (Oct 9, 2024, 1:17 PM)
This will help the adoption of USD for some game studio as Boost is Evil (for game studio)

> Guillaume Laforge (Autodesk) (Oct 9, 2024, 1:21 PM)
> Hopefully those game studios don't use OpenVDB ;)
> Francois Devic (Eidos-Montréal) (Oct 9, 2024, 1:21 PM)
> Hahaha chut :D

Arun Rao to Everyone (Oct 9, 2024, 1:22 PM)
Will pxr:boost_python come as an independent package in the distribution?

> Levi Biasco (Sony Group Corporation) (Oct 9, 2024, 1:23 PM)
> It does seem to be its own library under external
> https://github.com/PixarAnimationStudios/OpenUSD/blob/848956c81043b6402124f5411588dcaf41963e17/pxr/external/boost/CMakeLists.txt

JT Nelson to Everyone (Oct 9, 2024, 1:26 PM)
And I think there's a repo discussion

Sunya Boonyatera (Pixar Animation Studios) (Oct 9, 2024, 1:27 PM)
Cool — I poked at OpenVDB's code a bit and on the surface it looks like the boost usage is pretty minimal

Sunya Boonyatera (Pixar Animation Studios) to Everyone (Oct 9, 2024, 1:27 PM)
If it was even possible for them to move their boost usage into their implementations and out of the public headers that'd be a great step forward too

Nick Porcino (Pixar Animation Studios) to Everyone (Oct 9, 2024, 1:27 PM)
That statistic is no longer true now that boost isn't there :p

Guillaume Laforge (Autodesk) to Everyone (Oct 9, 2024, 1:28 PM)
We ship USD libs in Bifrost without Python and it is ~=180Mb

Matthew Kuruc to Everyone (Oct 9, 2024, 1:29 PM)
Does it include boost headers?

Guillaume Laforge (Autodesk) to Everyone (Oct 9, 2024, 1:30 PM)
Actually yes, we ship hydra libs too!

Levi Biasco (Sony Group Corporation) to Everyone (Oct 9, 2024, 1:30 PM)
Yeah I believe monolithic builds of USD are between 30 and 50 MB

Guillaume Laforge (Autodesk) to Everyone (Oct 9, 2024, 1:30 PM)
Without Hydra it is indeed way smaller

Nick Porcino (Pixar Animation Studios) to Everyone (Oct 9, 2024, 1:32 PM)
godot is in fact c++ based, I double checked my statement

Matthew Kuruc to Everyone (Oct 9, 2024, 1:33 PM)
FWIW, my notes tell me that that a C++ only, no imaging build of OpenUSD went from 184MB => 11MB once boost was no longer required.

Peter Siciliano to Everyone (Oct 9, 2024, 1:35 PM)
We have been working on a plugin approach for USD in runtime for UE… We are leveraging stuff like this: https://github.com/lighttransport/tinyusdz/pull/192

Jean-Silas Moor to Everyone (Oct 9, 2024, 1:36 PM)
https://github.com/assimp/assimp

Guillaume Laforge (Autodesk) (Oct 9, 2024, 1:38 PM)
Using this Assimp library to import USD assets would sounds weird. Better to leverage USD file formats plugin 🙂.

You to Everyone (Oct 9, 2024, 1:37 PM)
I have to say it. do we need to Wait for Godot?

Ben Chung-Hoon (NVIDIA Corporation) to Everyone (Oct 9, 2024, 1:39 PM)
Folks can build and extend omniverse now:
https://github.com/NVIDIA-Omniverse/kit-app-template

JT Nelson to Everyone (Oct 9, 2024, 1:26 PM)
From attending OpenVDB TSC meetings, they have also been looking at and reducing dependency on boost. It should be in their minutes over the past couple years off and on.

Sunya Boonyatera (Pixar Animation Studios) (Oct 9, 2024, 1:40 PM)
@Tyler Furby (Wabi Foundation) Thanks for the link!

Andrew Kaufman (NVIDIA Corporation) to Everyone (Oct 9, 2024, 1:40 PM)
Re the distro size, it seems like once boost is removed the schemas might start to dominate? Just looking at usdGeom, it seems a bit bulky to me. Does a runtime actually require all that description text?

```
> du -h lib/usd/usdGeom/
132K	lib/usd/usdGeom/resources/usdGeom
```

```
392K    lib/usd/usdGeom/resources
396K    lib/usd/usdGeom/
```

Levi Biasco (Sony Group Corporation) (Oct 9, 2024, 1:41 PM)
Can schema files be USDC? Also you definitely don't need to include the description text, though you'd need to build something bespoke to remove it from a build

Nick Porcino (Pixar Animation Studios) (Oct 9, 2024, 1:43 PM)
usdc loads a lot faster !!
if we're talking about runtime/game, descriptive text no, if a dcc yes....

Guillaume Laforge (Autodesk) (Oct 9, 2024, 1:43 PM)
All the schemas in one single used sounds like the smaller we could get!

“Usdc”

JT Nelson to Everyone (Oct 9, 2024, 1:45 PM)
USD import through ASIMP is in O3DE's development branch now.
https://github.com/o3de/o3de/pull/18361

Nick Porcino (Pixar Animation Studios) (Oct 9, 2024, 1:46 PM)
oh interesting!!!! was not aware

Ashwin Bhat (Autodesk) (Oct 9, 2024, 1:48 PM)
Isn’t that based on tinyusd?

JT Nelson to Everyone (Oct 9, 2024, 1:46 PM)
They/we(SCB/PasOS) are discussing long term strategies and will be having a meeting dedicated to it soon. I will post the meeting link in the Slack channel.

Paul Molodowitch to Everyone (Oct 9, 2024, 1:54 PM)
I do feel that there's some legitimate concerns here that shouldn't be brushed aside. USD wasn't originally designed to be something that is easily integrated into other projects. It's a slice of an entire development ecosystem / codebase - it has it's own vector libraries, it's own string libraries, it's own pointers, etc, etc. If you start a new project from the ground up and adopt all the same structures as USD uses, then this isn't an issue, but if you have an existing codebase, and you want to introduce USD into it's core runtime, there's going to be a lot of work of work to either swap out your existing structures for USDs, or make translators, etc.

JT Nelson to Everyone (Oct 9, 2024, 1:57 PM)
From our O3DE Discord:

Gene [Amazon] — 08/05/2024 11:39 AM

Asset Importer [added USD support](https://github.com/assimp/assimp/pull/5628) I'm working on a pull request to update our AssImp 3rdParty package. Here's a single usdz biplane in Editor

Nick Porcino (Pixar Animation Studios) to Everyone (Oct 9, 2024, 1:58 PM)
I see they are using tinyusdz for import.

Action Items

- ...

Questions

- …