

ASWF / * ACADEMY
SOFTWARE
FOUNDATION

open
Source
days'22

ASWF / * ACADEMY
SOFTWARE
FOUNDATION

open
Source
days^{'22}

MaterialX / glTF Update

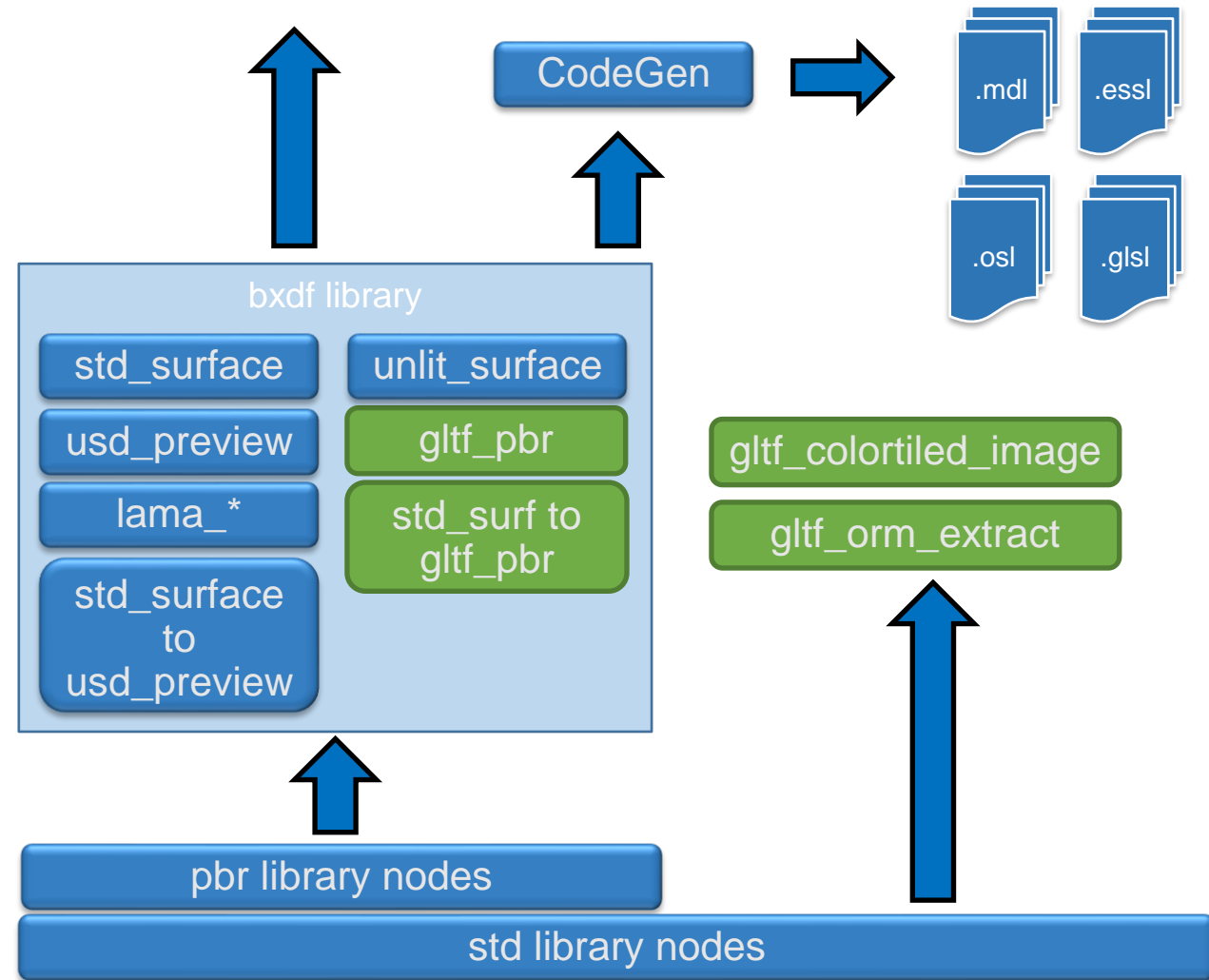
Bernard Kwok (Khronos/MaterialX) and Pablo Delgado (Enscape)

Outline

- glTF
 - glTF Materials in the MaterialX Ecosystem
 - glTF Interoperability
- Web Updates
 - Assets and Visualization
 - Rendering and Validation
- What's Next

Introducing glTF to MaterialX

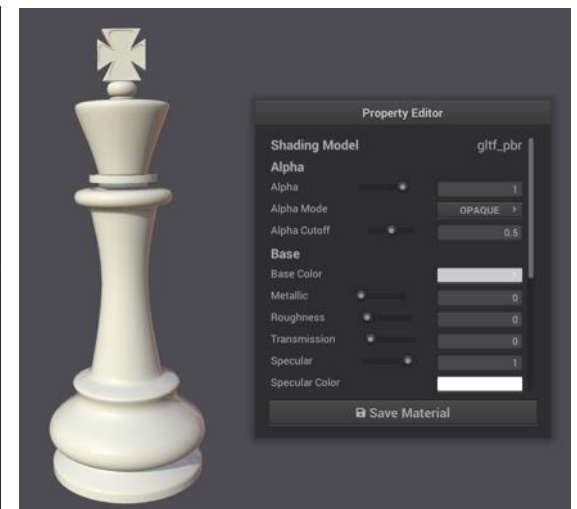
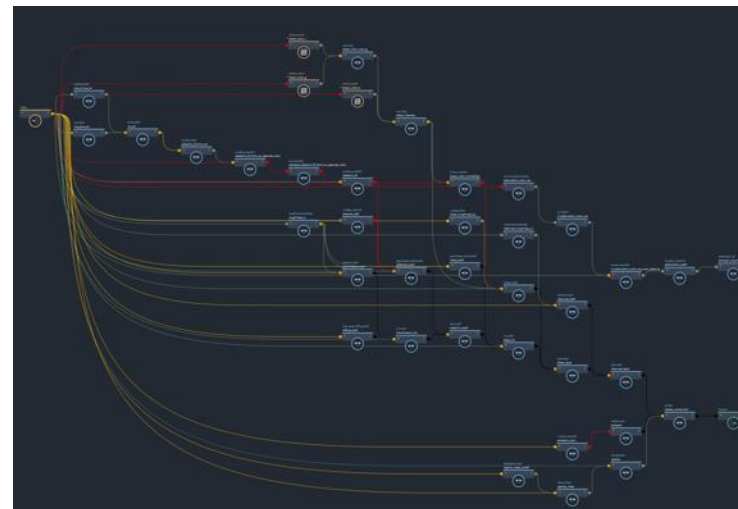
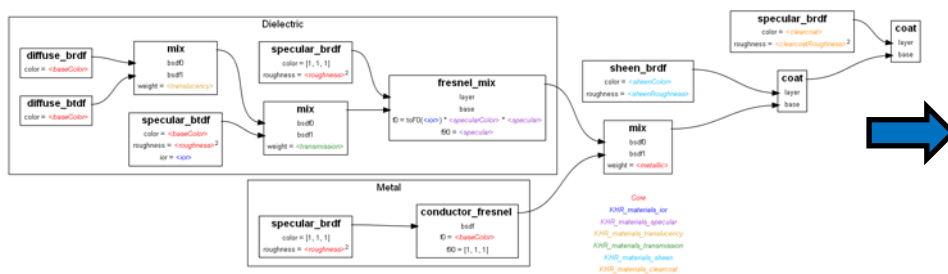
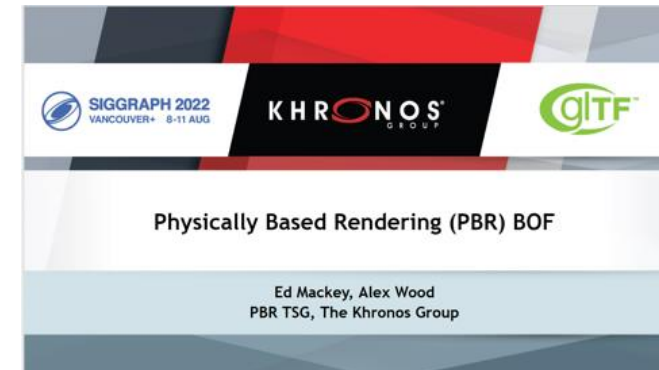
- MaterialX [v1.38.4](#)
- New **gltf_pbr** shading model definition available (from Tobias Häußler)
- Building blocks: standardized node libraries
- In progress:
 - Shader translation graphs
 - Pattern definitions
 - Colored image
 - Image channel extract and combine.



The glTF PBR Shading Model

- KHR transmission, specular, ior, sheen, clearcoat, volume, emissive strength support.
- Iridescence, diffuse transmission to come.
- Help drive pbr improvements (e.g., thin-walled material support)

- Details: [Khronos Siggraph 2022 BOF](#)



Left: PBR Layering (Courtesy Tobias Häußler). Middle: MaterialX nodegraph, (Courtesy Nicolas Savva, Autodesk). Right: definition and UI (from MaterialXView). Chessboard assets by Moeen Sayed and Mujtaba Sayed for SideFX.

glTF PBR Support

- Use definition like any other shading model
 - Version, limits, UI hints metadata
 - Arbitrary nodegraph inclusion
 - Definition creation
 - Materials and assignments



“King “ from chessboard assets by Moeen Sayed and Mujtaba Sayed for SideFX.

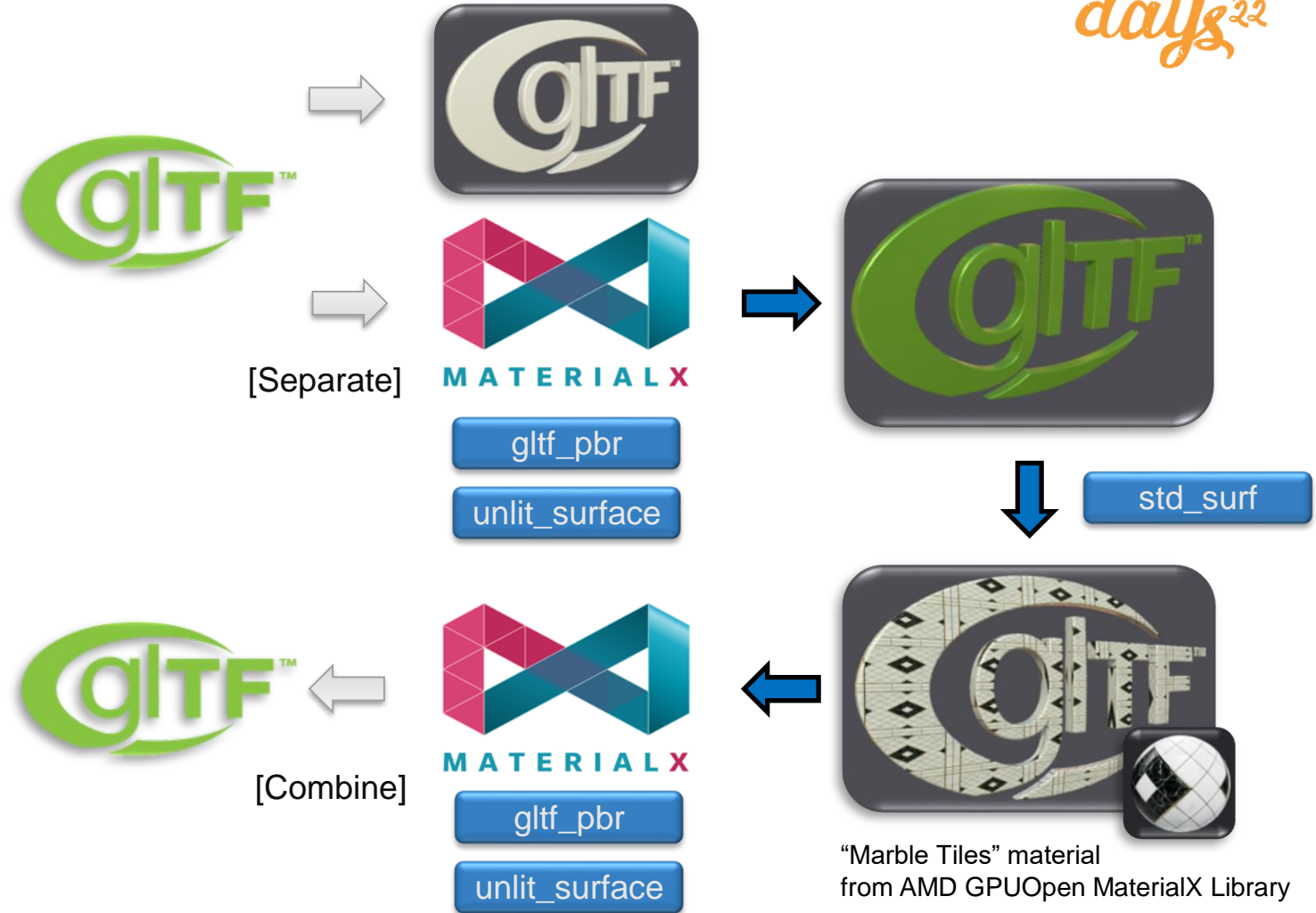
- Multiple backend shader generators (GLSL, ESSL, OSL, MDL etc.)



“boombox” sample model from Khronos. MDL sample viewer from NVIDIA. MaterialX example courtesy Ashwin Bhat, Autodesk)

MaterialX Workflows

- MaterialX workflows: authoring / lookdev / rendering
- *Separation* of materials, assignments, bindings
- Optional glTF translation
- Considerations:
 - Value resolves
 - UV Transform spaces
 - Path conventions
 - Binary resources
 - Units, color management



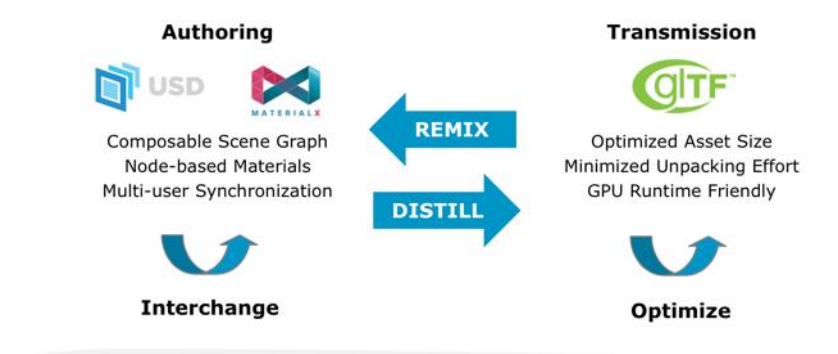
Remapping / Distillation Workflow



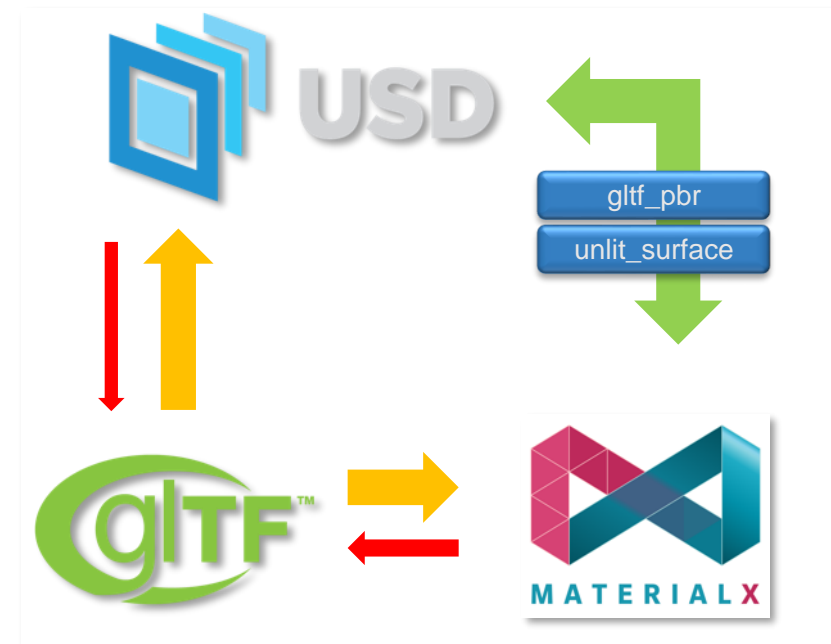
Rendered using: MaterialXView (left 3 courtesy Christian Robles, Autodesk) and [Dassault Enterprise PBR Sample Renderer Demo Viewer](#) (right-most). Model and textures for “King” from chessboard assets by Moeen Sayed and Mujtaba Sayed for SideFX.

USD / MaterialX / glTF Ecosystem

- **gltf_pbr** and unlit available in USD 22.08
- glTF's role, responsibilities, and workflows under "Metaverse" umbrella of discussions
- For authoring: USD / MaterialX *interoperability*
- For transmission: glTF *interoperation*

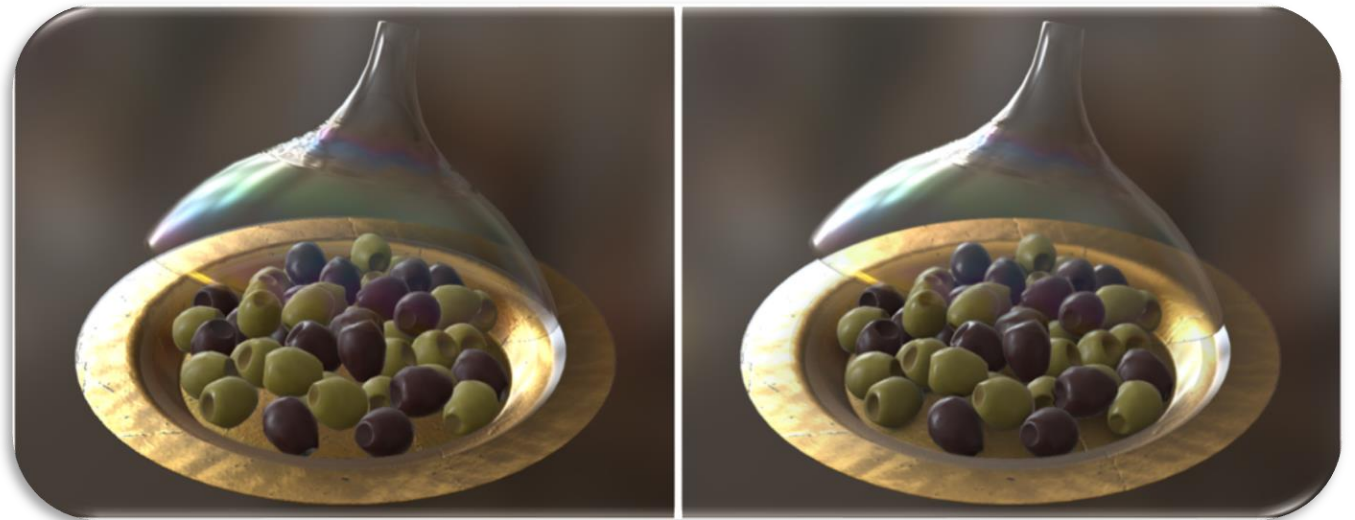
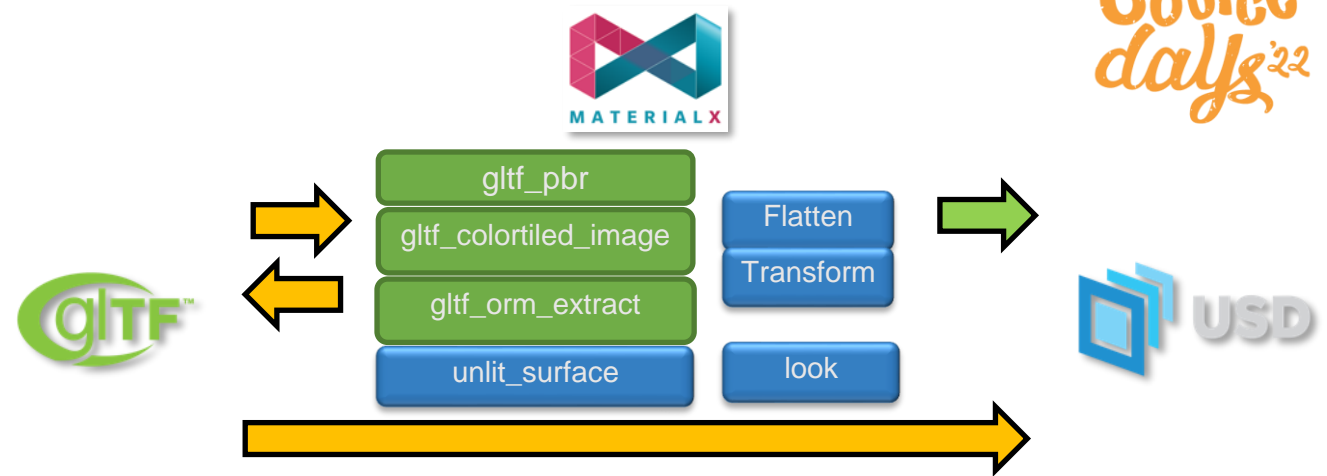


[From: 2022 Metaverse Standards Forum](#)



glTF Interoperation

- MaterialX gltf_pbr to glTF import / export in progress. ([Khronos fork](#), [guc](#))
 - Material variants, unlit to come
- [guc](#): glTF to USD+MaterialX converter via MaterialX graph creation or direct to USD.
- Standardize target MaterialX graph and utilities in Khronos or core MaterialX



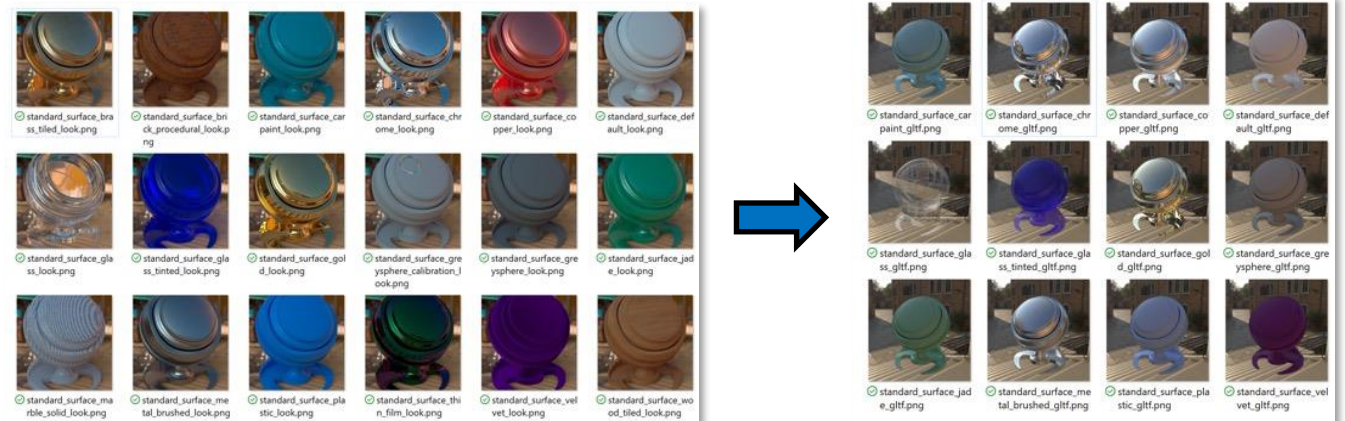
USD+MaterialX in hdStorm

glTF Sample Viewer

Wayfair's Iridescent Dish with Olives asset ([CC BY](#)). Example courtesy Pablo Delgado.

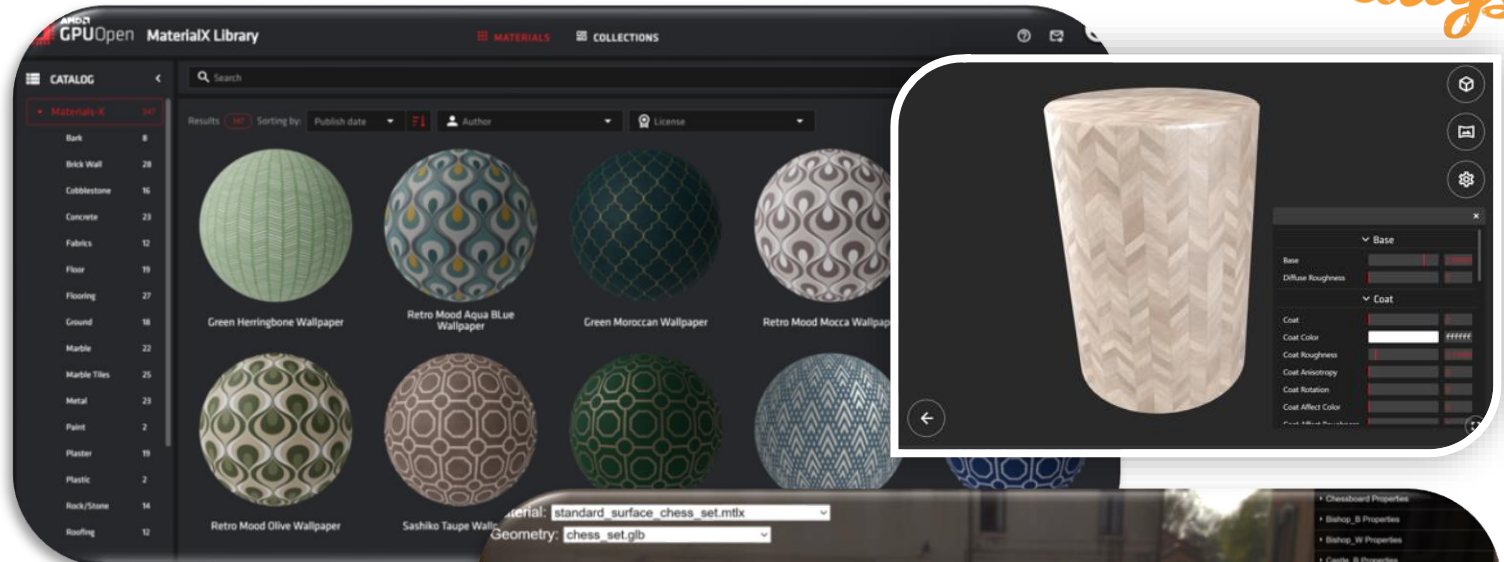
Shader Generation and Rendering Validation

- *glslangValidator* for ESSL and Vulkan shader validation
- Extend test assets (e.g. [glTF Sample Library](#) or [Cesium Examples](#) etc.)
- Open Source Reference Rasterizers and Path tracers
 - glTF for OSL "testrender"

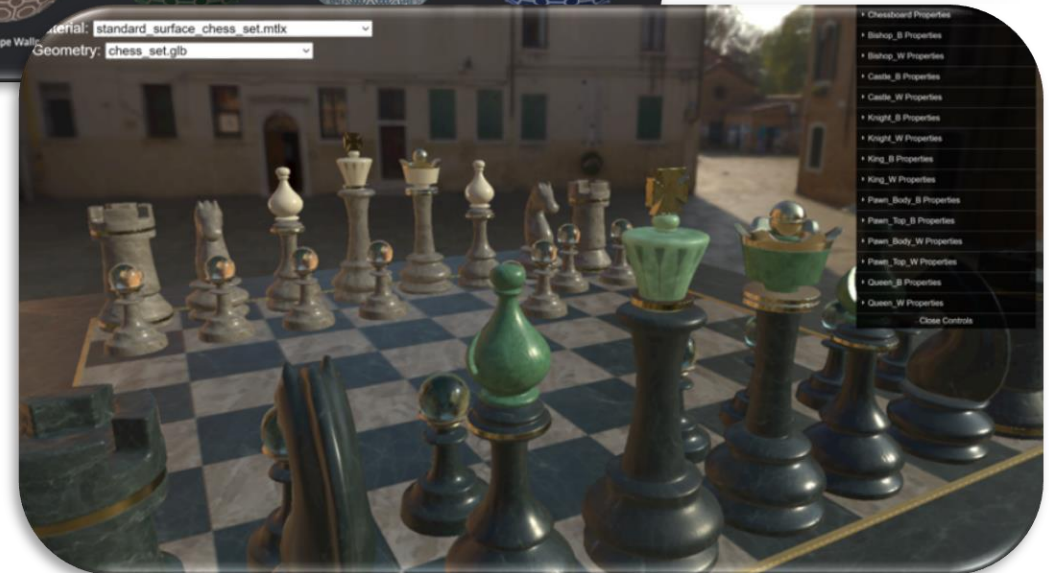


Assets and Visualization

- Desktop Viewer
 - glTF geometry support
- [MaterialX Web Viewer](#)
 - Editability and Performance
 - Assignment support
- [AMD GPUOpen MaterialX Web Viewer](#)
 - gltf_pbr materials
- Packaging: npm registry



AMD GPUOpen MaterialX library.



Chessboard assets by Moeen Sayed and Mujtaba Sayed for SideFX.

Summary and Future

- glTF material model in MaterialX ecosystem
 - Current and future extensions
- Complete export workflow
 - Translation graphs, baking of procedural patterns
- Validation via open-source reference path tracers
 - Desktop: OSL “testrender”
 - Web: Dassault Renderer, other?
- Improve real-time performance / configurability
 - Shader generation variants
 - Stream requirements
- Extend transmission formats
 - Standard shading models and shader node graph support ?
 - Meta data: Color management, real world units ?

Acknowledgements

Alex Wood: AGI	Jonathan Stone: LucasFilm	Pablo Delgado: Enscape
Ashwin Bhat: Autodesk	Lutz Kettner: NVIDIA	Tobias Häußler: Dassault Systèmes
Brian Savory: AMD	Magnus Petterson: IKEA	
Christian Robles: Autodesk	Moeen Sayed: Side Effects	
Doug Smythe: ILM	Mujtaba Sayed: Side Effects	
Ed Mackey: AGI	Nicolas Savva: Autodesk	
Emmett Lalish: Google	Niklas Harrysson: Lumiere	
Eric Chadwick: Wayfair		

