

# open Source days'23

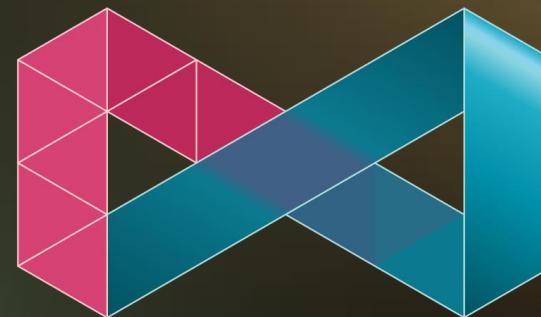
#ASWF

/\* ACADEMY SOFTWARE FOUNDATION

open  
Source  
days'23

/\* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series



MATERIAL X

August 2<sup>nd</sup>, 2023

#ASWF

/\* ACADEMY SOFTWARE FOUNDATION

The OpenPBR Surface Shading Model – Guido Quaroni, Adrien Herubel, et al

MaterialX in USD and Hydra – Karen Lucknavalai, Pixar USD

LookdevX in Maya – Nikola Milosevic, Autodesk

The Stråla MaterialX Editor – Magnus Pettersson, IKEA

Updates on MaterialX and MDL – Kai Rohmer, NVIDIA

RenderMan and MaterialX – Mark Manca, Pixar RenderMan

The QuiltiX Graph Editor – Manuel Köster, Richard Frangenberg

open  
Source  
days'23

/\* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series  
**OpenPBR**

August 2<sup>nd</sup> 2023

#ASWF

/\* ACADEMY SOFTWARE FOUNDATION

# OpenPBR - Introduction



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Artists leverage creative applications from different vendors
- Look differences across apps is a major complaint
- "Innovate" and "Simplify" to put 3D artists forward
- Subproject of MaterialX within the ASWF

# Announcing OpenPBR

A new material specification created by the teams at



This physically based shading model is being developed to offer creatives a more artist-friendly bridge between different software applications.

The new shading model will be a subproject of MaterialX within the Academy Software Foundation.

ASWF /\* ACADEMY  
SOFTWARE  
FOUNDATION





The following side-by-side renders are from a scene we affectionately called "Shader Playground."

They are a  
**work in progress.**

Every 3D application has its differences but both art & tech teams are already seeing the benefits of OpenPBR working similarly inside **Autodesk Arnold and the Adobe proprietary renderer.**



OpenPBR inside Maya + Arnold



OpenPBR inside Adobe proprietary renderer





OpenPBR inside Maya + Arnold



OpenPBR inside Adobe proprietary renderer



A  
ARN

OpenPBR inside Maya + Arnold



OpenPBR inside Adobe proprietary renderer

S

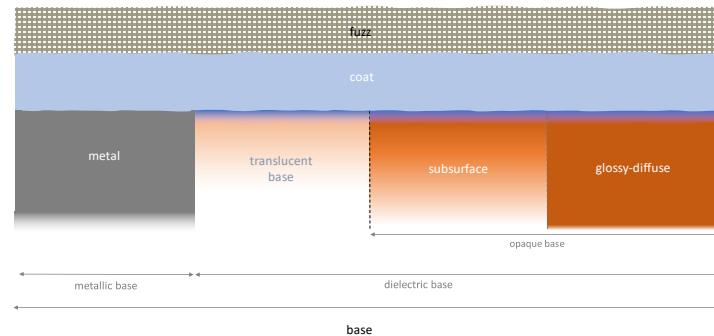
# OpenPBR – Technical Overview



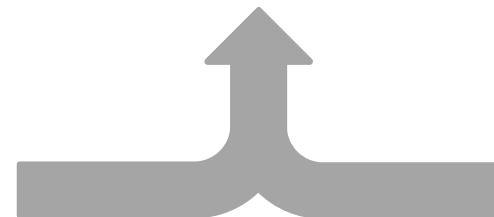
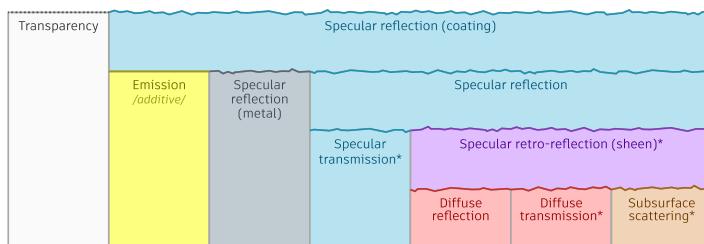
/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- A physically based surface model based on **Autodesk Standard Surface** and **Adobe Standard Material**
- An evolution of two production-proven models

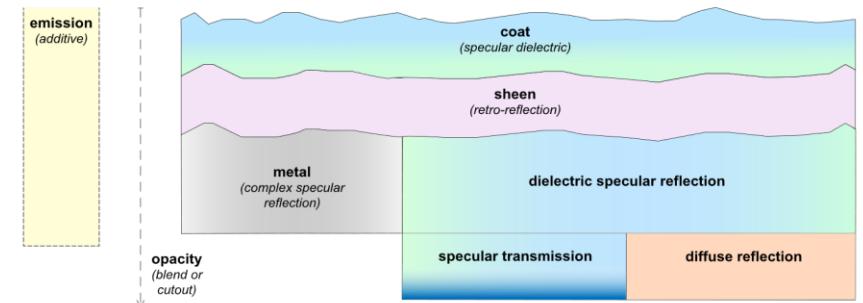
OpenPBR Surface



Autodesk Standard Surface



Adobe Standard Material



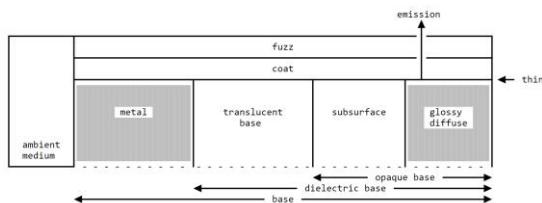
# OpenPBR – Technical Overview

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

## Principled and physically based layering specification

### Formal structure (implementation agnostic)



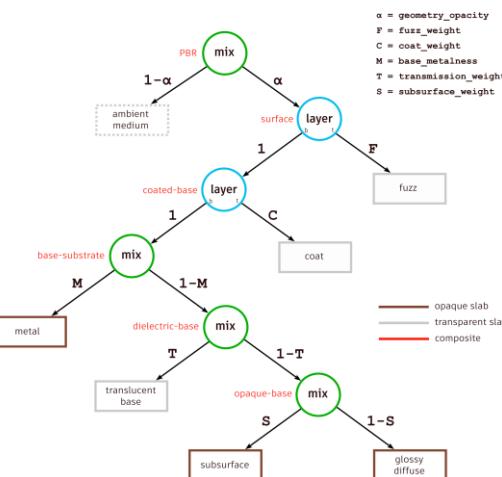
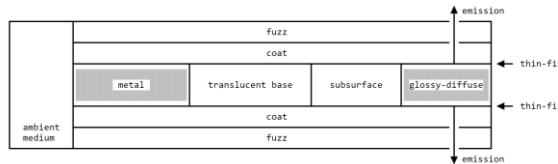
To summarize the formal structure, this consists of the following slabs:

$$\begin{aligned} S_{\text{ambient-medium}} &= \text{Slab}(0) \\ S_{\text{fuzz}} &= \text{Slab}(f_{\text{fuzz}}, V_{\text{fuzz}}) \\ S_{\text{coat}} &= \text{Slab}(f_{\text{coat}}, V_{\text{coat}}) \\ S_{\text{metal}} &= \text{Slab}(f_{\text{conductor}}) \\ S_{\text{translucent-base}} &= \text{Slab}(f_{\text{dielectric}}, V_{\infty}^{\text{dielectric}}) \\ S_{\text{subsurface}} &= \text{Slab}(f_{\text{dielectric}}, V_{\infty}^{\text{subsurface}}) \\ S_{\text{glossy-diffuse}} &= \text{Slab}(f_{\text{glossy-diffuse}}) \end{aligned}$$

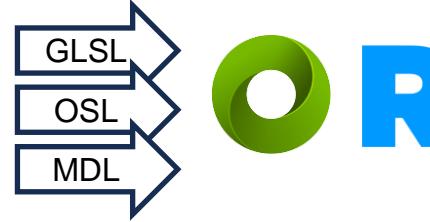
Which are composed to build the material structure, denoted PBR below, as follows:

$$\begin{aligned} M_{\text{PBR}} &= \text{mix}(S_{\text{ambient-medium}}, M_{\text{surface}}, \alpha) \\ M_{\text{surface}} &= \text{layer}(M_{\text{coated-base}}, S_{\text{fuzz}}, F) \\ M_{\text{coated-base}} &= \text{layer}(M_{\text{base-substrate}}, S_{\text{coat}}, C) \\ M_{\text{base-substrate}} &= \text{mix}(M_{\text{dielectric-base}}, S_{\text{metal}}, M) \\ M_{\text{dielectric-base}} &= \text{mix}(M_{\text{opaque-base}}, S_{\text{translucent-base}}, T) \\ M_{\text{opaque-base}} &= \text{mix}(S_{\text{glossy-diffuse}}, S_{\text{subsurface}}, S) \end{aligned}$$

where  $\alpha = \text{geometry\_opacity}$   
where  $F = \text{fuzz\_weight}$   
where  $C = \text{coat\_weight}$   
where  $M = \text{base\_metalness}$   
where  $T = \text{transmission\_weight}$   
where  $S = \text{subsurface\_weight}$



implementers



### Particular implementations

$$\begin{aligned} f_{\text{PBR}} &= \text{lerp}(f_{\text{transparent}}, f_{\text{surface}}, \alpha), \\ f_{\text{surface}} &= F f_{\text{fuzz}} + \text{lerp}(1, 1 - \bar{E}[f_{\text{fuzz}}], F) f_{\text{coated-base}}, \\ f_{\text{coated-base}} &= C f_{\text{coat}} + \text{lerp}(1, T_{\text{coat}}(1 - E[f_{\text{coat}}]), C) f_{\text{base-substrate}}, \\ L_e &= \text{lerp}(1, T_{\text{coat}}, C) E, \\ f_{\text{base-substrate}} &= \text{lerp}(f_{\text{dielectric-base}}, f_{\text{conductor}}, M), \\ f_{\text{dielectric-base}} &= f_{\text{specular}}^R + (1 - E[f_{\text{specular}}^R]) f_{\text{dielectric-base}}^T, \\ f_{\text{dielectric-base}}^T &= \text{lerp}(\text{lerp}(f_{\text{diffuse}}, f_{\text{SSS}}, S), f_{\text{specular}}^T, T). \end{aligned}$$

# OpenPBR – Technical Overview

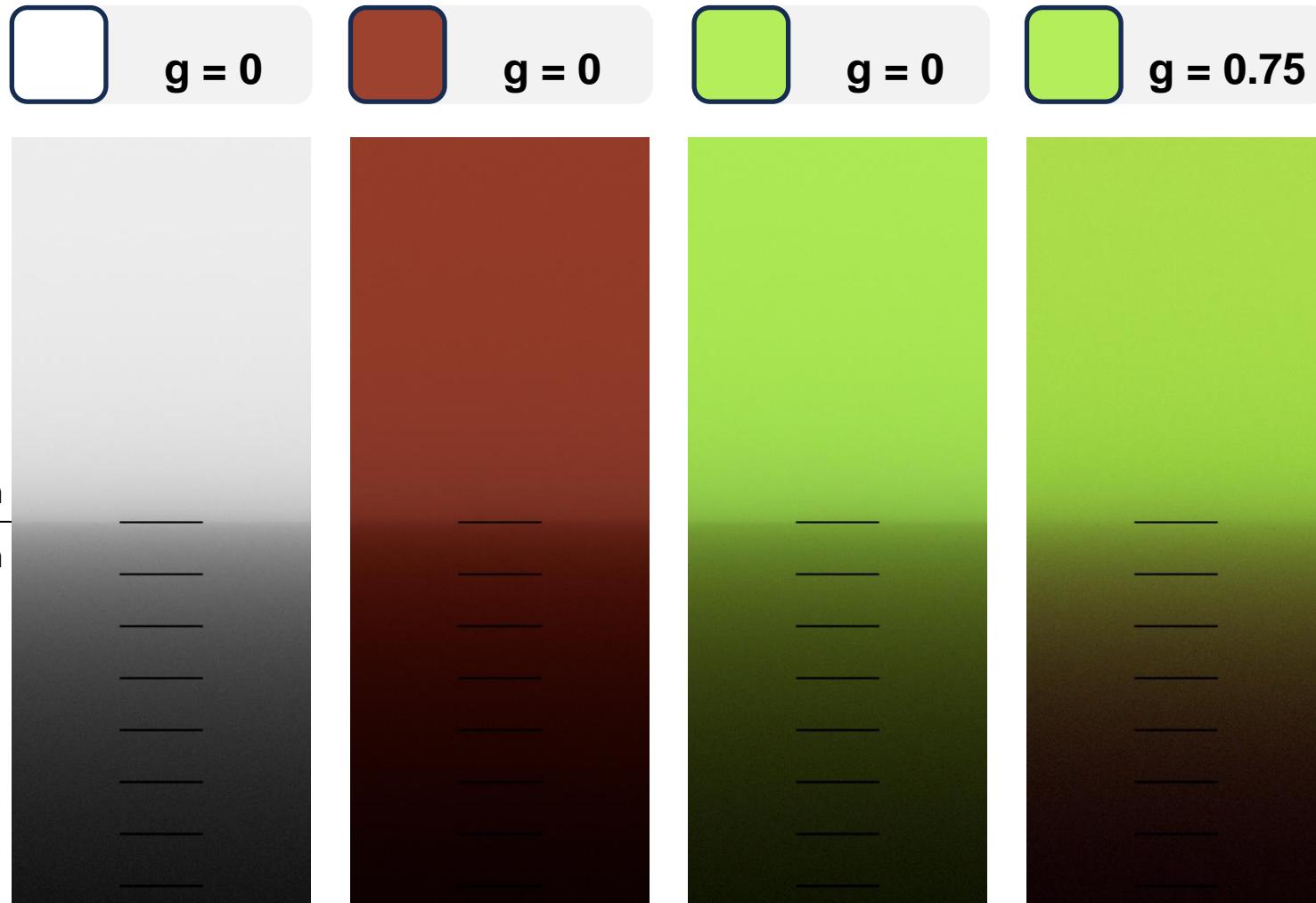
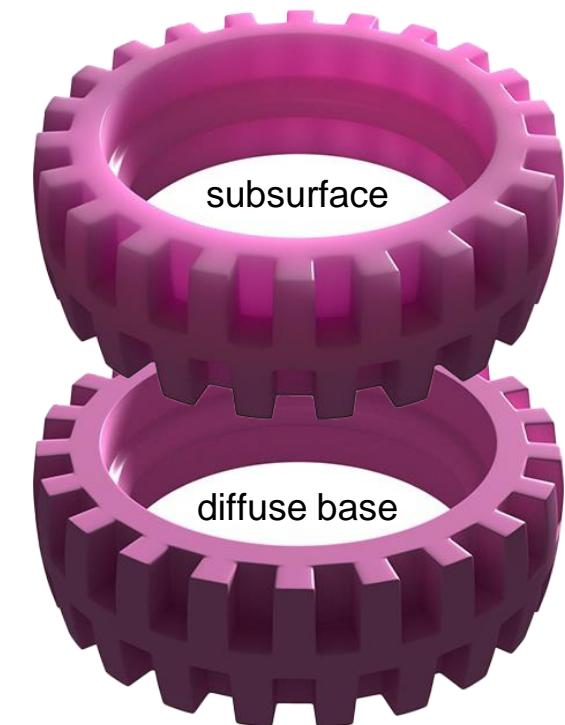


/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

## Commonalities and differences

Dedicated SSS component

Inspired by Standard Surface



# OpenPBR – Technical Overview

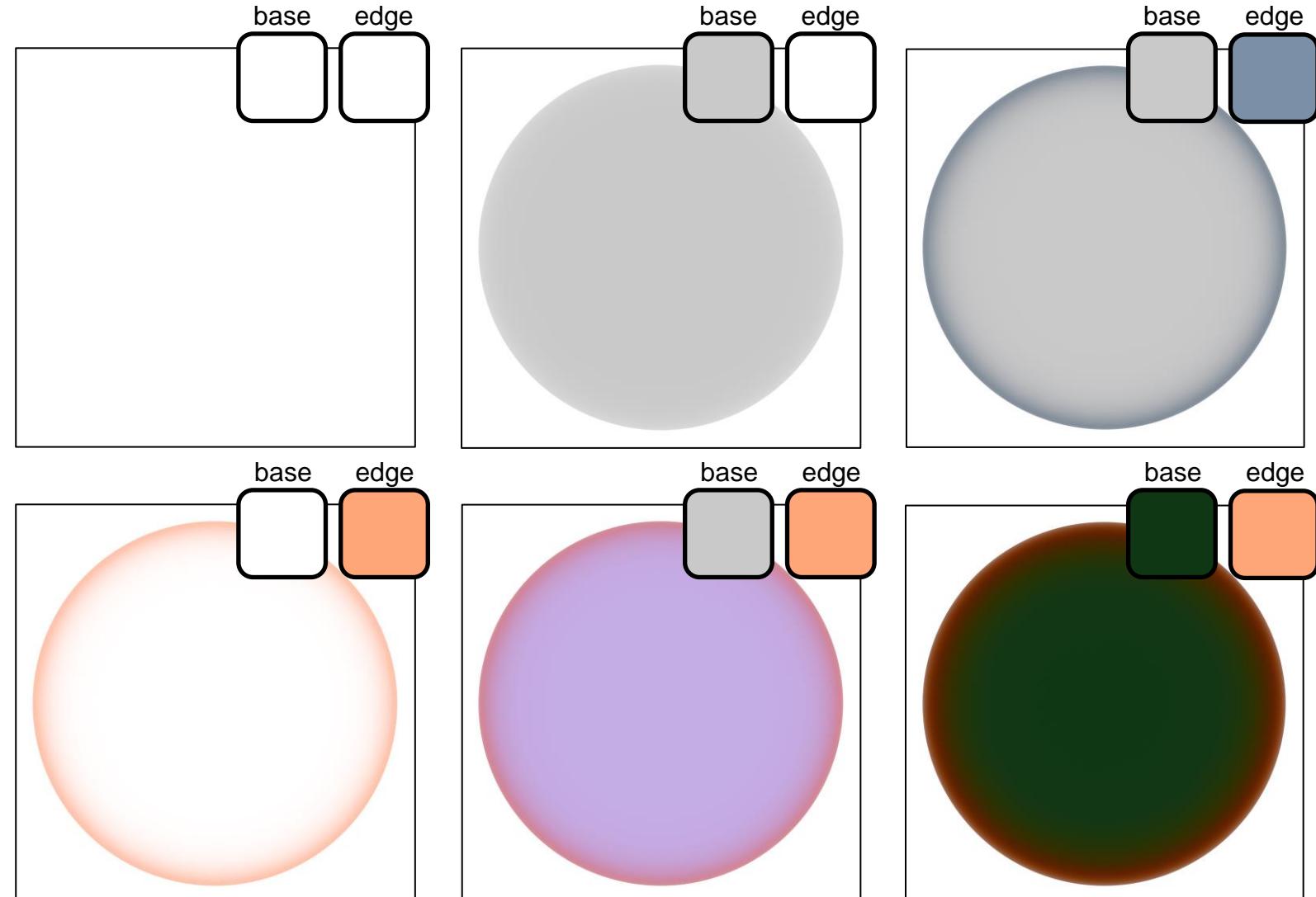
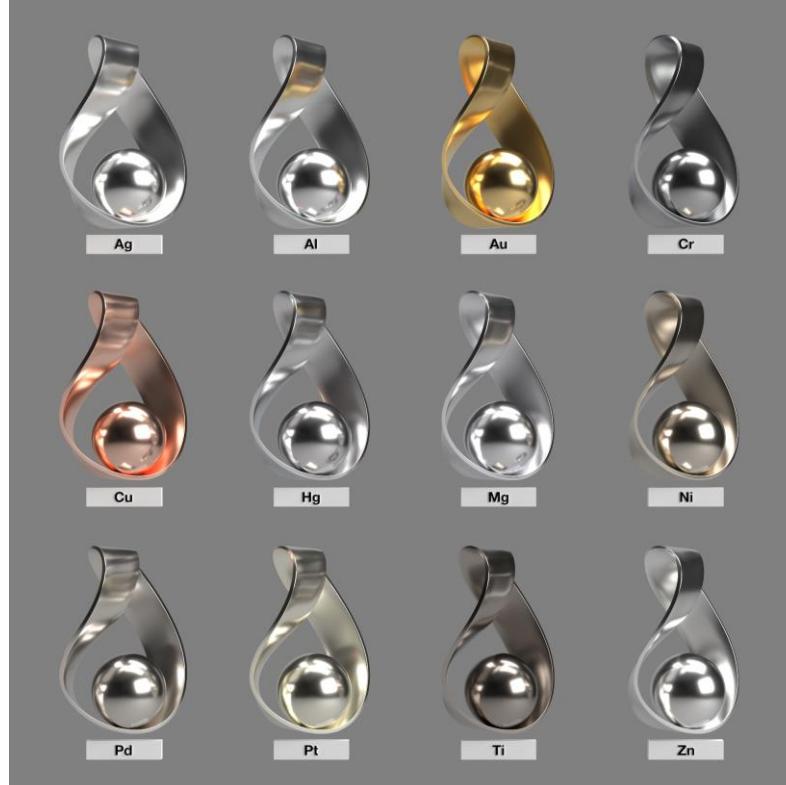


/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

## Commonalities and differences

F82-tint metal reflectivity

Inspired by Adobe Standard Material



# OpenPBR – Technical Overview

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

## Commonalities and differences

Sheen/Fuzz on top of coat

Different from both original models

base

base + coat

base + coat + fuzz



# OpenPBR - Conclusion



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Great collaboration and good spirit
- An open standard under MaterialX
- Early reviewing by third parties
- Integration in products
- Next steps & future initiatives
- Reach out and discuss at SIGGRAPH

**Autodesk Booth Wed 9<sup>th</sup> & Thu 10<sup>th</sup> at 11am**



Chris Rydalch - SideFX

# OpenPBR - Contributors



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Zap Andersson
- Paul Edmondson
- Julien Guertault
- Adrien Herubel
- Alan King
- Peter Kutz
- Andréa Machizaud
- Jamie Portsmouth
- Frédéric Servant



open  
Source  
days'23

/\* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

MaterialX in USD/Hydra

Karen Lucknavalai, Pixar

Aug 2, 2023

#ASWF

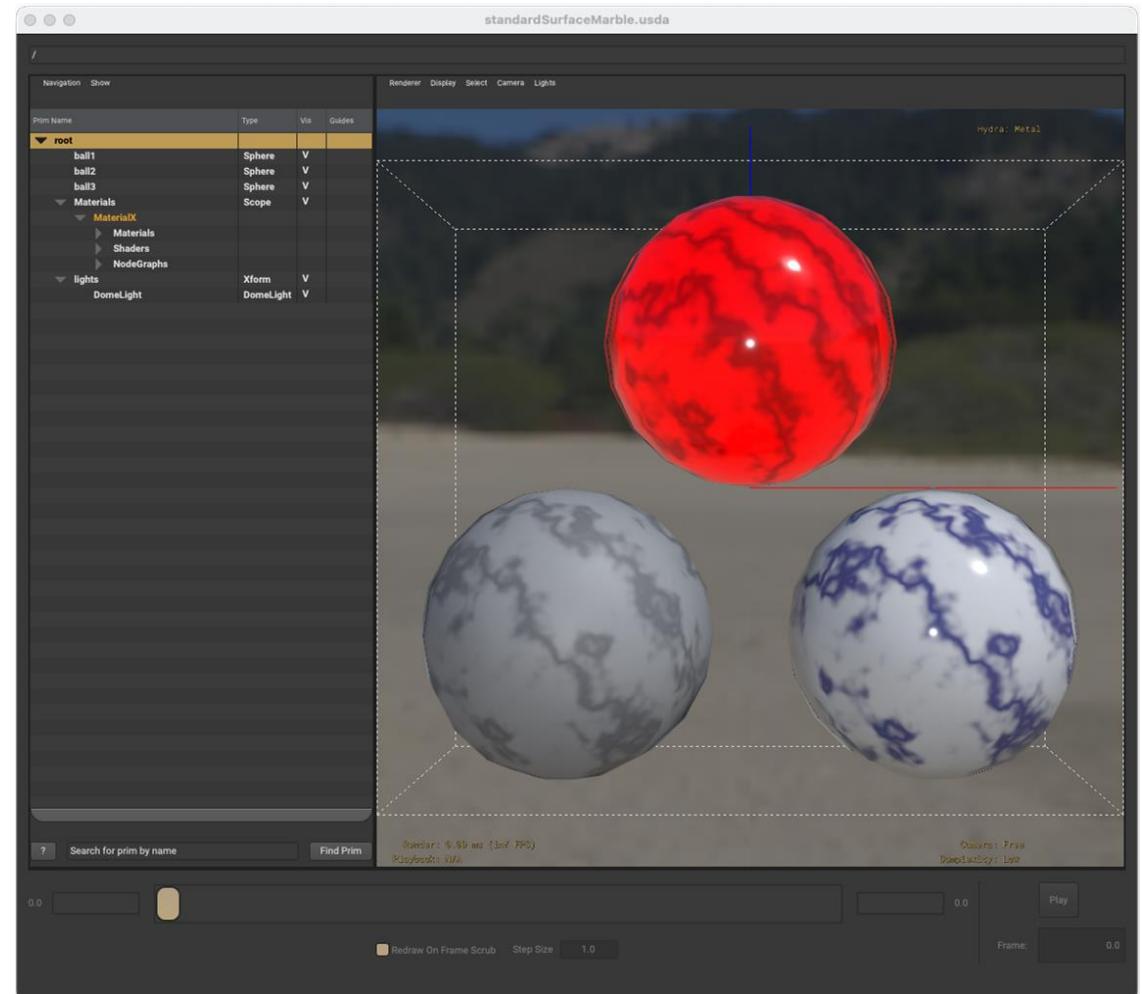
/\* ACADEMY SOFTWARE FOUNDATION

# MaterialX in USD/Hydra - Updates

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

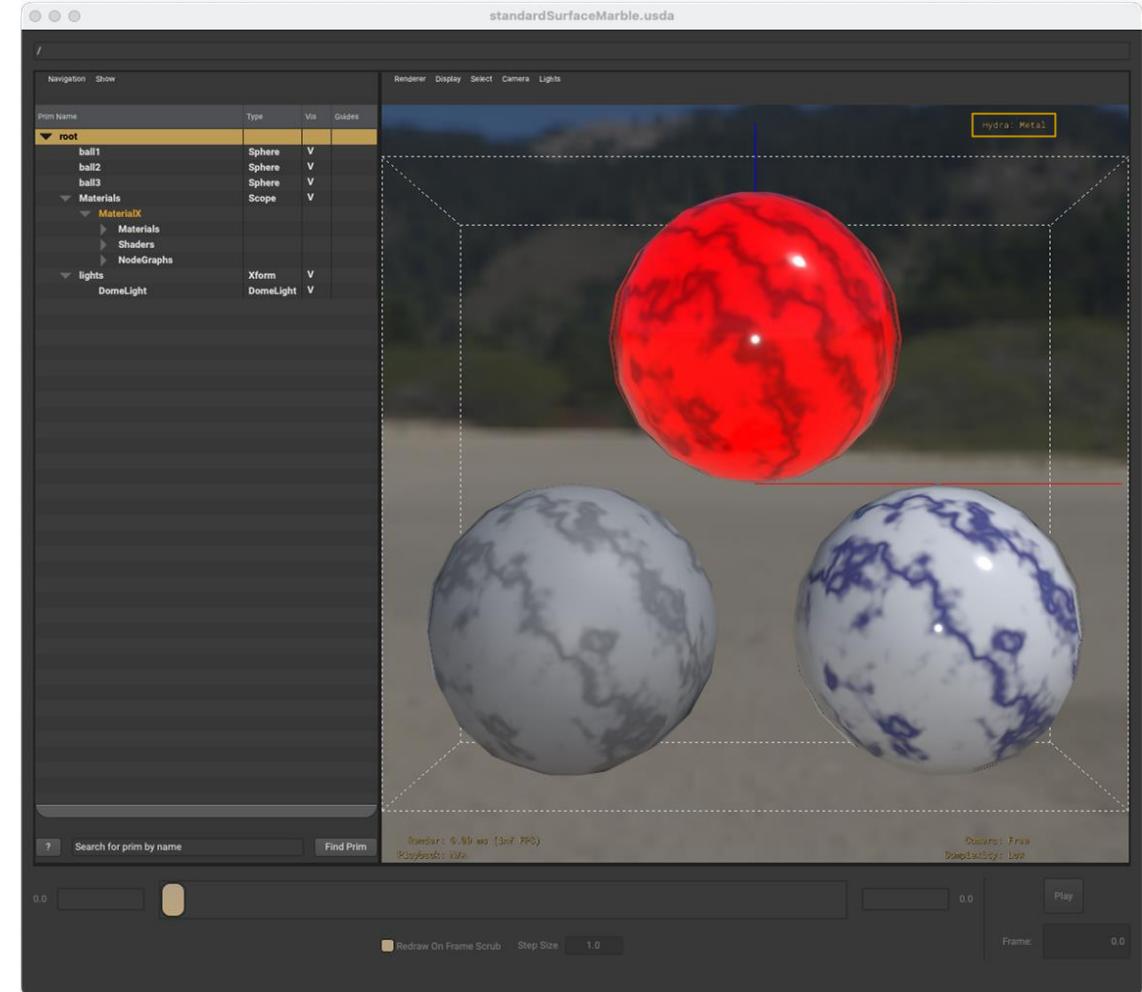
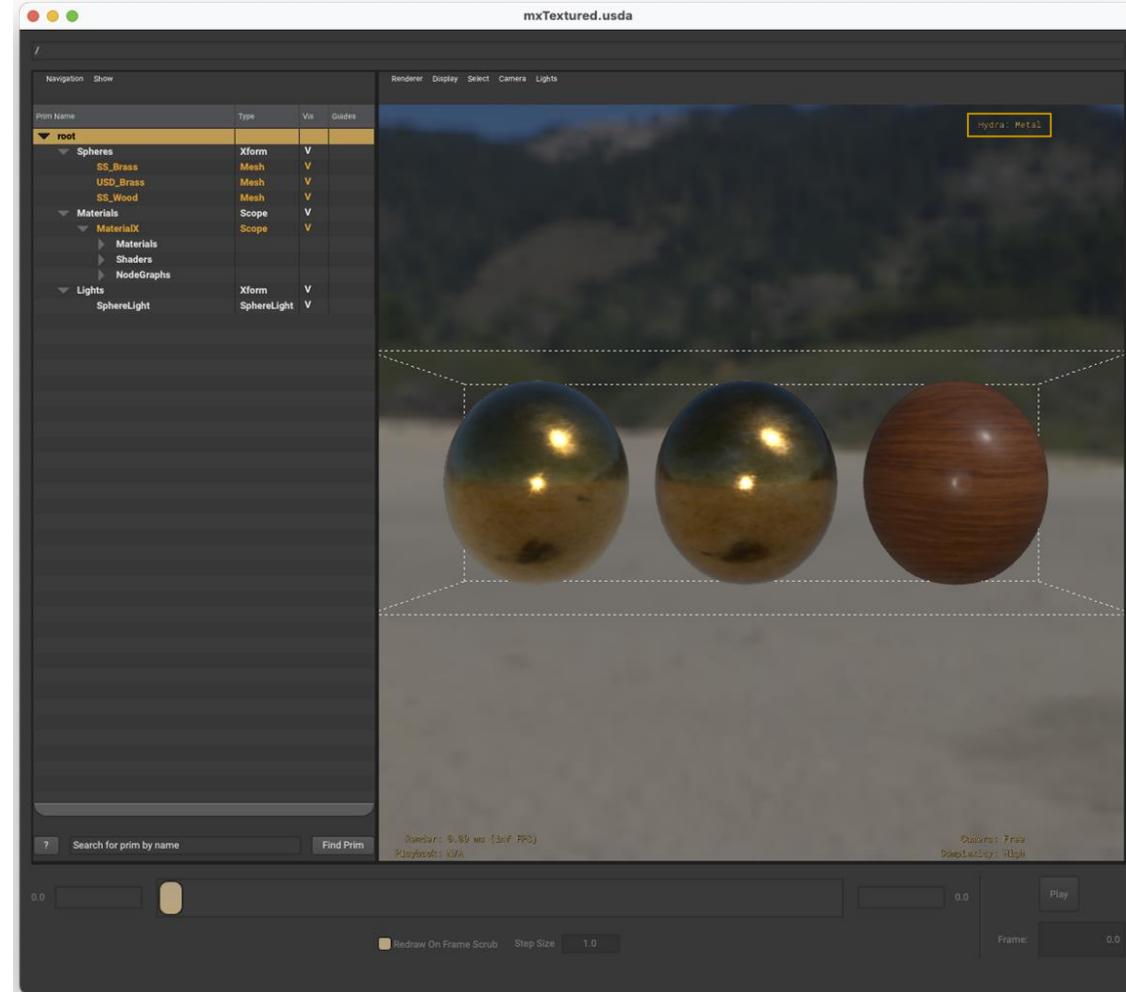
- MaterialX enabled by default in USD
- Support up to MaterialX version 1.38.7



# MaterialX in USD/Hydra - Updates

open  
Source  
days '23

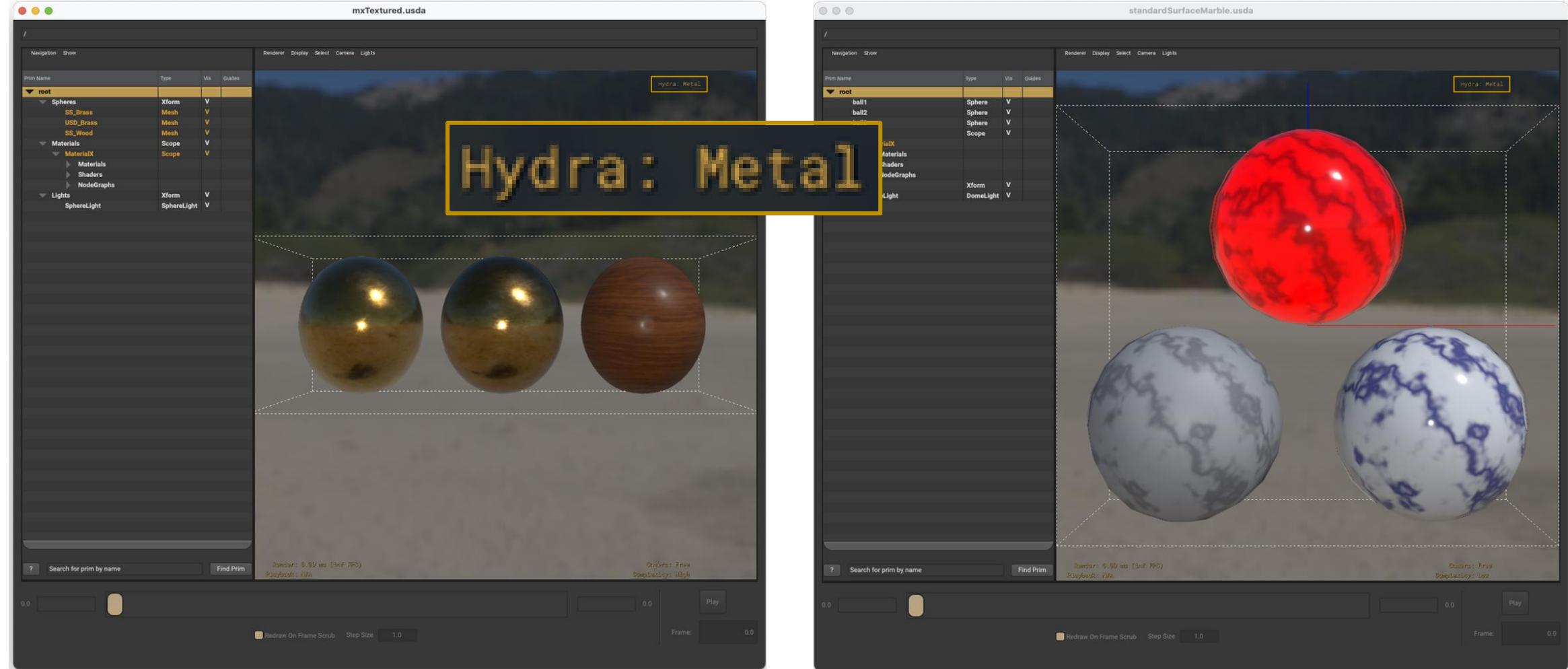
/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF



# MaterialX in USD/Hydra - Updates

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF



# MaterialX in USD/Hydra - Updates



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

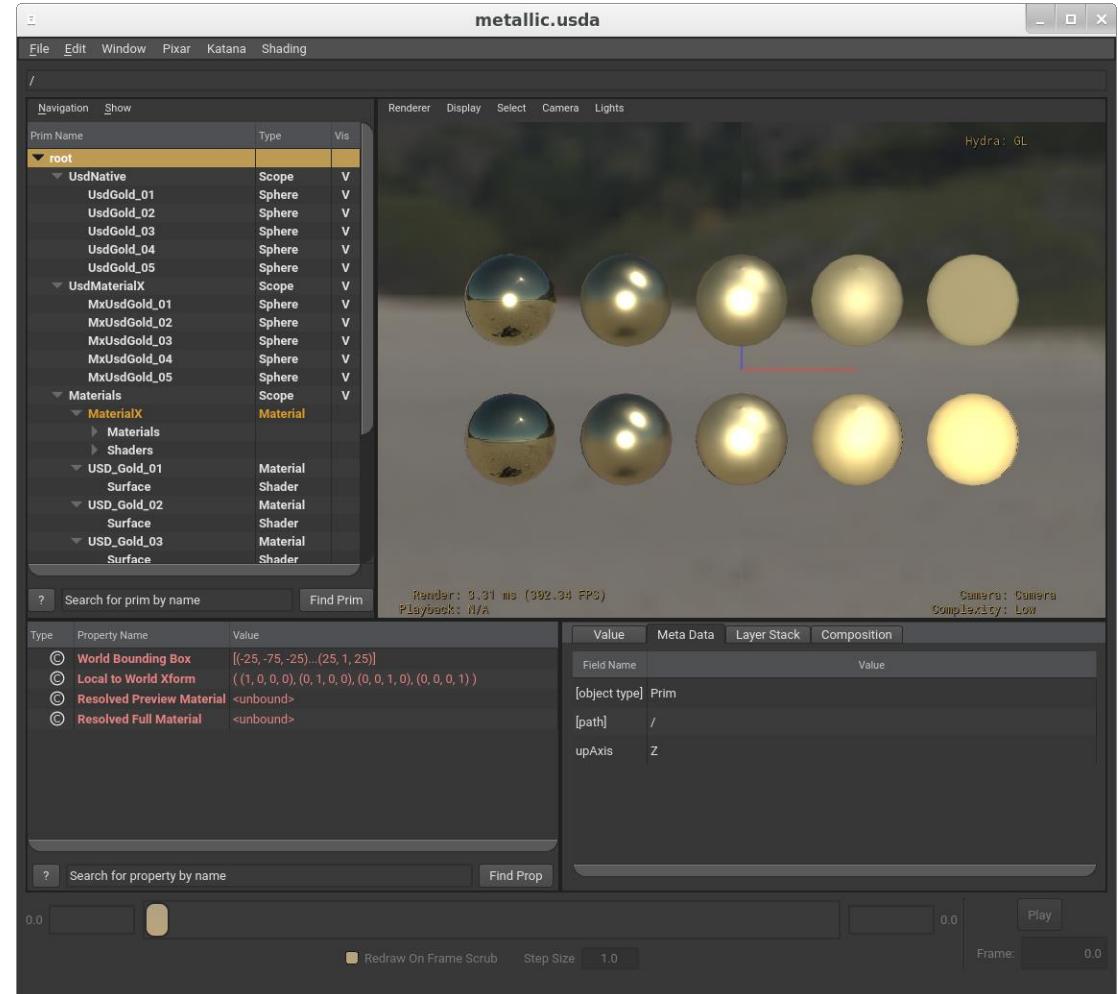
- More flexibility in how materials can be defined
  - Node placement wrt nodegraphs
  - Nodegraph input and interface connections
  - Material names
- Wider support for Custom Nodes
- Other Improvements
  - Can include relative paths
  - Handling string array processing
  - More metadata is parsed into the SDR properties

# MaterialX in USD/Hydra - Updates

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Aligning MaterialX implementation of USDPreviewSurface Materials
- Opensourced and extended the imaging tests

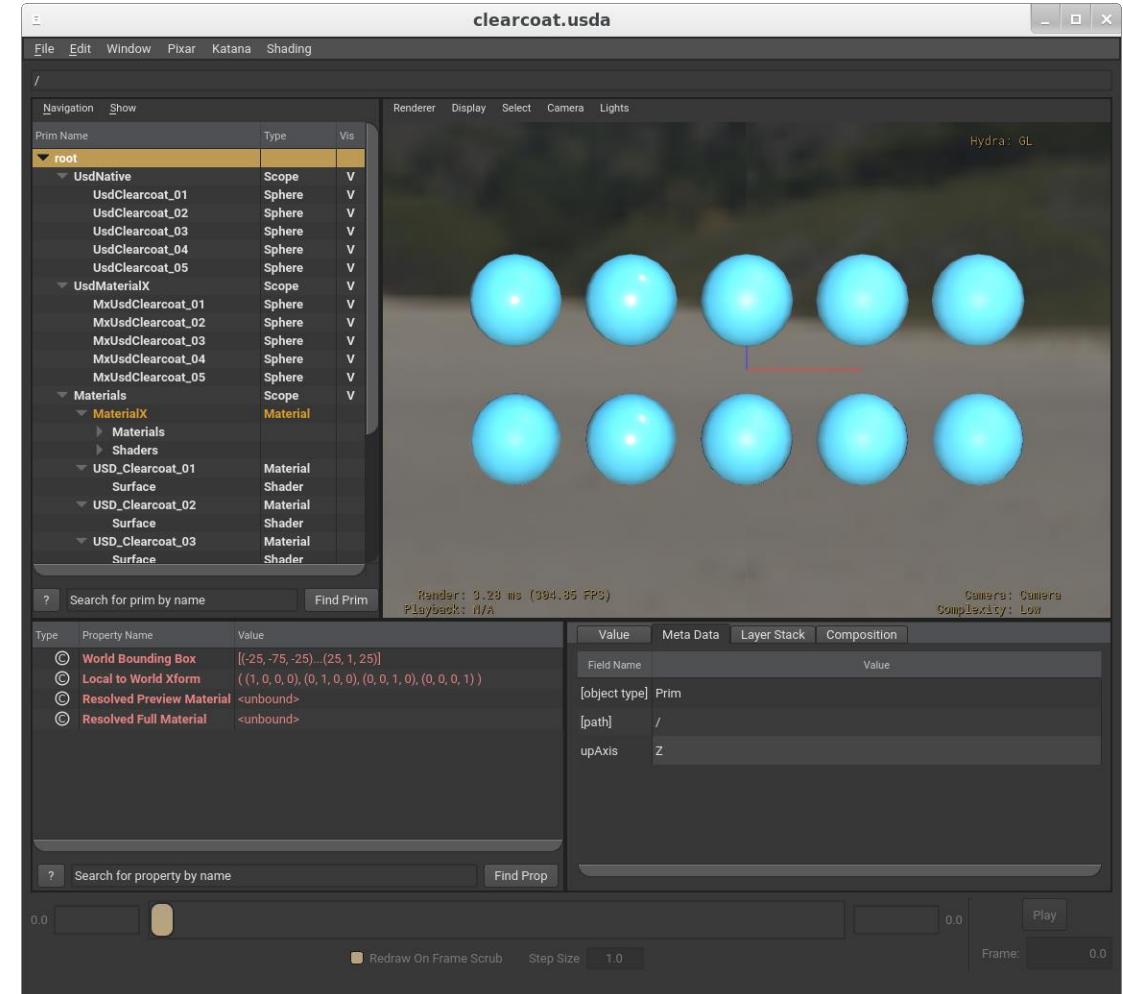


# MaterialX in USD/Hydra - Updates

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Aligning MaterialX implementation of USDPreviewSurface Materials
- Opensourced and extended the imaging tests

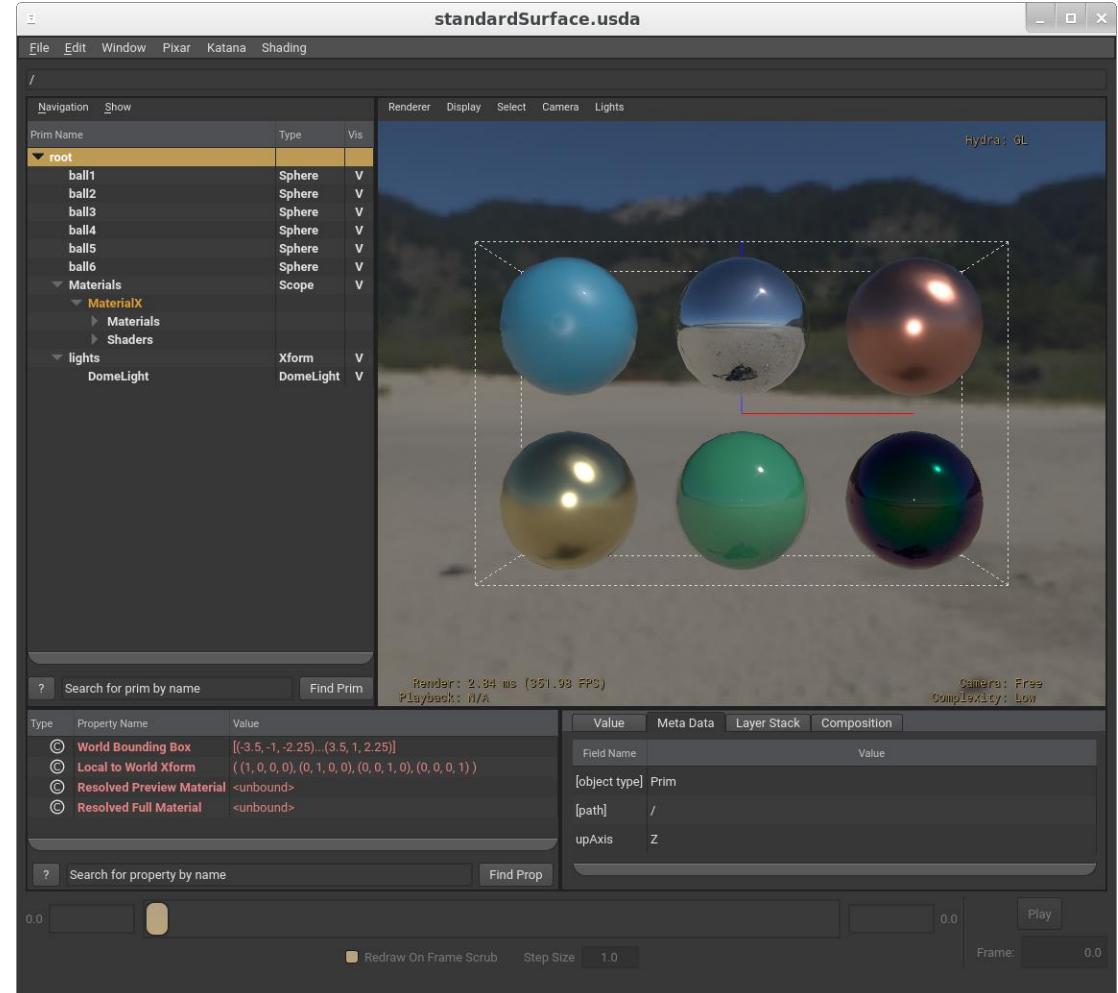


# MaterialX in USD/Hydra - Updates

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Distant light support
- geomprop/primvar values in Storm
- Initial material caching added to Storm



# MaterialX in USD/Hydra - HdPrman

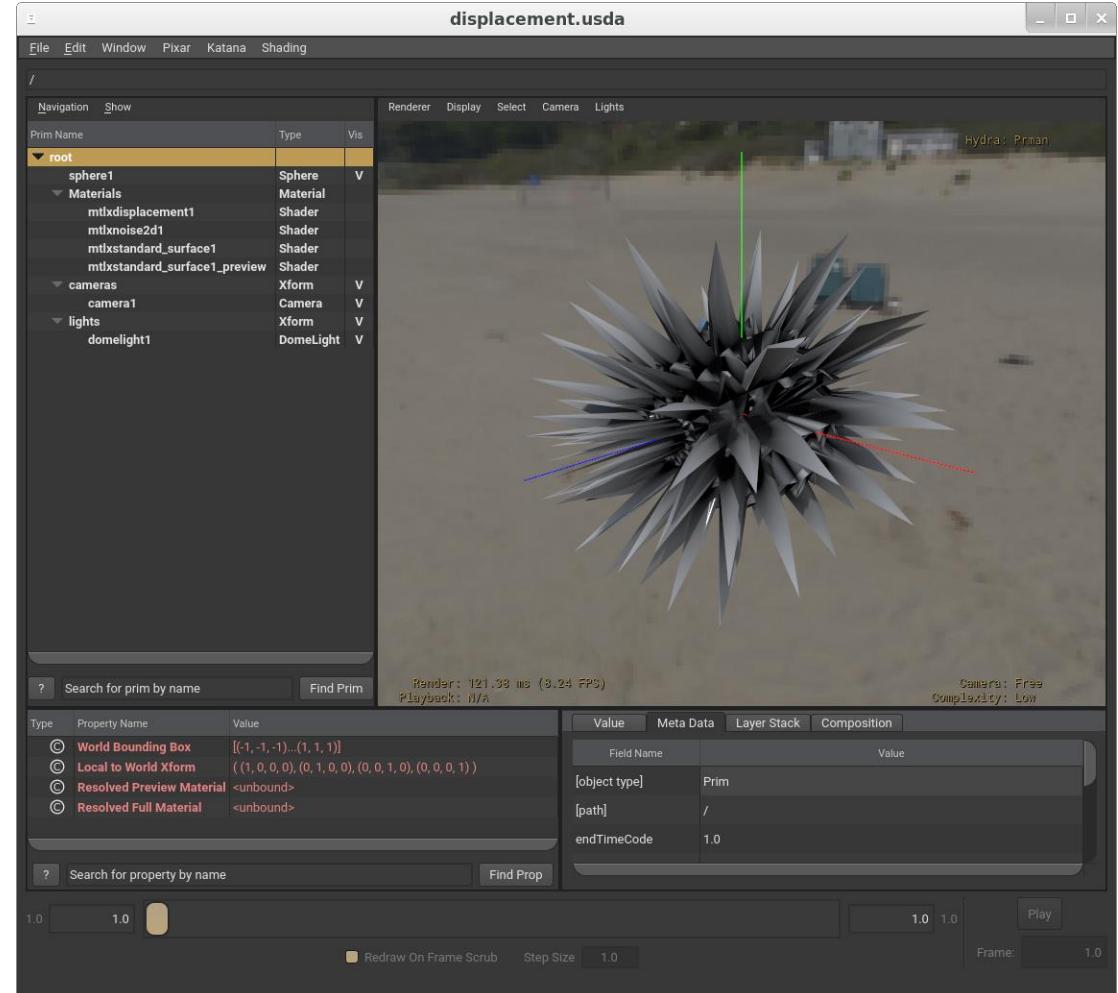
open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

## Support for Displacement

Fixes for:

- Texture coordinate names
- Normal Maps



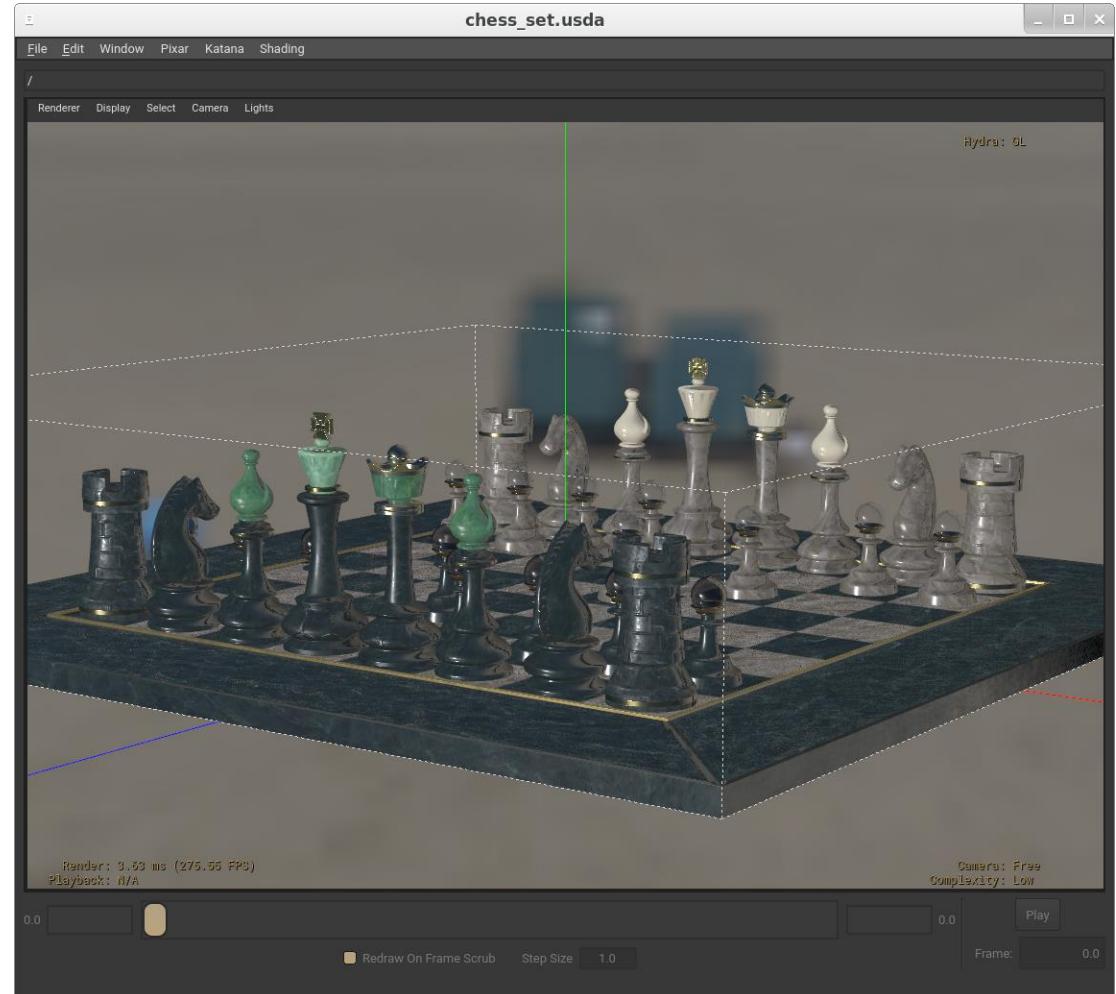
# MaterialX in USD/Hydra - Future Work

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

## Future Work:

- Normal Map issues in Storm
- MaterialX Colorspace handling in Hydra (Goal: 23.11)

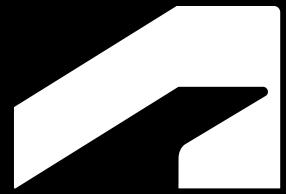




# Thank you

More information at the  
USD, Hydra BOF  
at SIGGRAPH:

Tuesday Aug 8, 2pm - 4pm



# LOOKDEV

## Agnostic Material Editor

MaterialX Town-Hall

Siggraph 2023

Nikola Milosevic  
Product Manager

Orn Gunnarsson  
Software Manager

# LookdevX | Released in Maya 2024

- Agnostic Material Authoring tool
  - Native USD material authoring
    - Resolving USD Material Story
  - Native MaterialX authoring (in progress)
  - Arnold Integrated
  - UFE layer used for DCC connection
    - Enabling DCC portability



# Release 2024 | LookdevX

## Delivered Key Features

- Material manipulation from LookdevX, Outliner ,VP and Attribute Editor
- Direct Material Assignment
- Arnold support
- MaterialX Viewport support
- Color Managed LookdevX Graph
- Material Authoring workflow
- USD Material Scope control
- Multi-selection actions
- Node & Graph Duplication
- Undo - Redo
- Enums support AE/PE
- And more.....



# Considering | [LookdevX](#)

## Key Features

- Publishing workflows
- Presets workflow
- Icon Shelf
- Searching Nodes and/or Attributes
- Dimming and coloring Nodes and Noodles
- Expose Custom Attribute control on Node
- Material Swatches
- Create node from Port
- Deep Attribute Promotion
- Ramp node
- Automation tools



# Next | LookdevX

- Hydra implementation support
  - Material graphs can be accurately represented in Storm and Arnold delegate
- MaterialX Maya native support
- 3ds Max Implementation
- Procedural Material Binding – USD
  - Late material binding enabled by Bifrost-USD



AUTODESK Arnold

open  
Source  
days'23

/\* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

Stråla MaterialX Editor

2023-08-02

#ASWF

/\* ACADEMY SOFTWARE FOUNDATION

# Agenda



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- MaterialX at IKEA
- Technical Design

# Technical Design



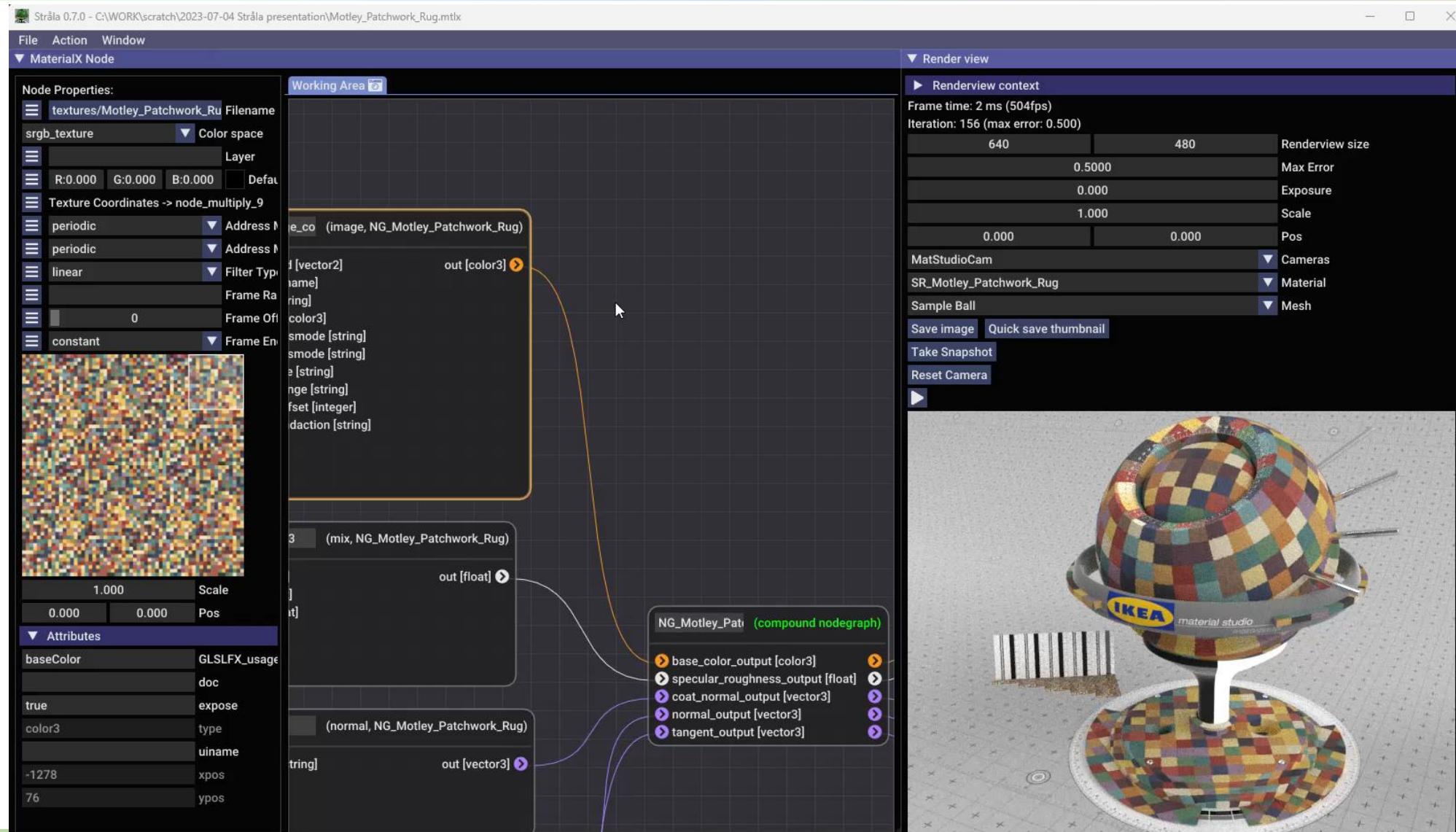
/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- C++
- [Dear ImGui](#) for interface
- [ImGui Node Editor](#) for displaying networks
- Uses MaterialX OSL backend for code generation
- OIIO for texture handling
- Modified OSL testrender for previews

# Node editing

open  
Source  
days '23

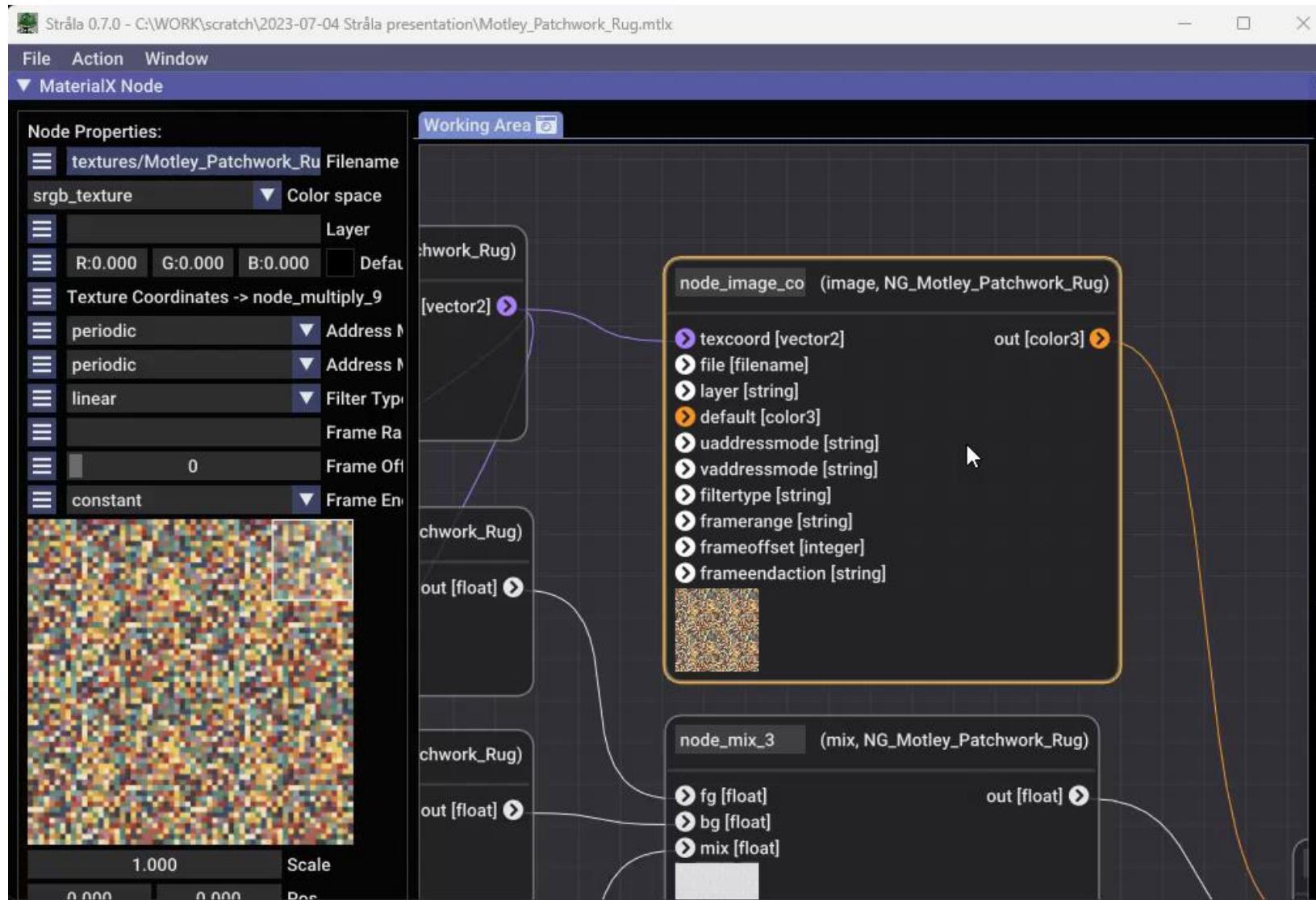
/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF



# Swatches and texture handling

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF



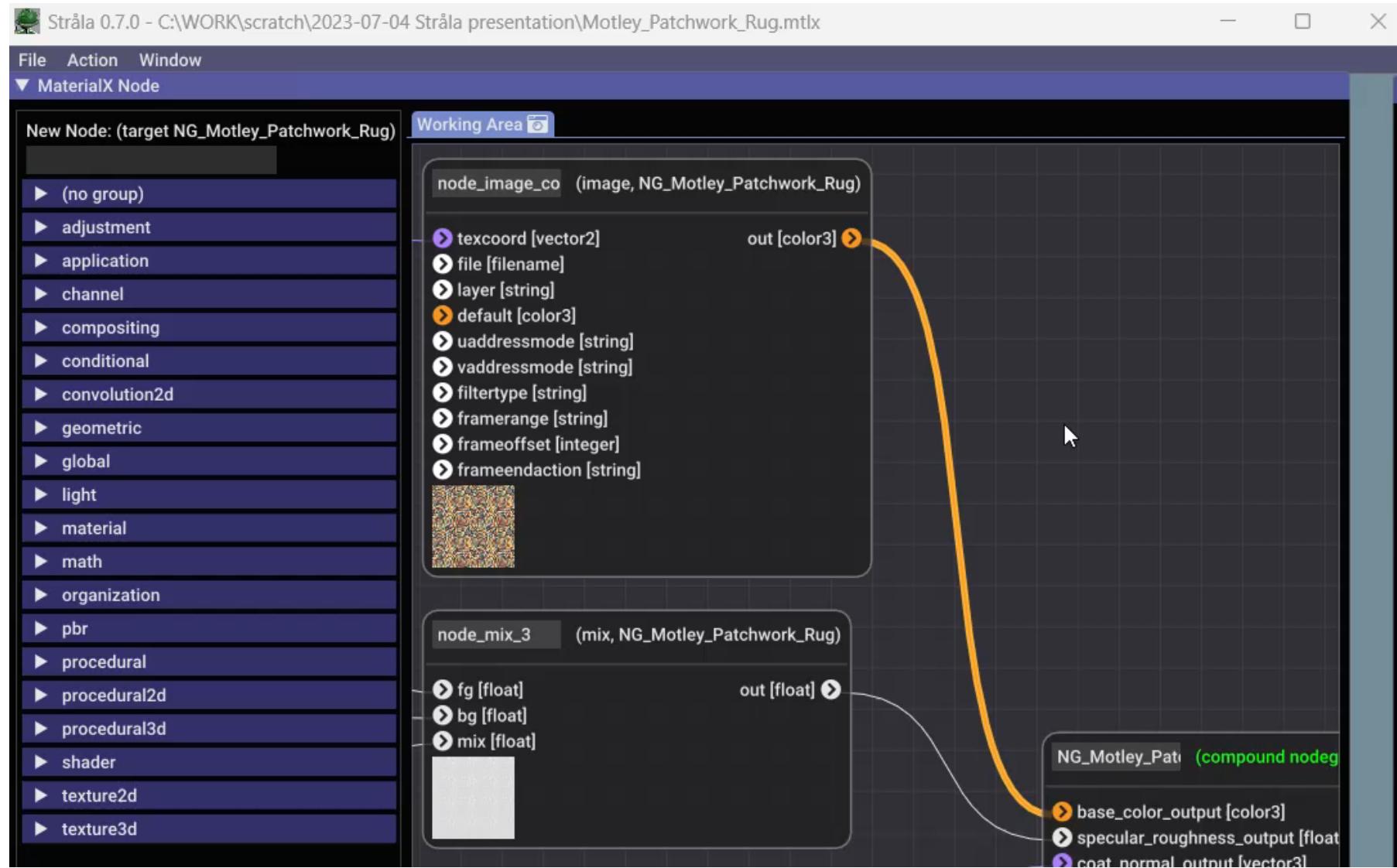
Material from AMD  
material library

# Preview render



open  
Source  
days '23

# MaterialX to gltf



open  
Source  
dayz'23

open  
Source  
days'23

/\* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

NVIDIA Updates on MaterialX and MDL

August 2<sup>nd</sup>, 2023

Kai Rohmer  
Senior Software Engineer  
NVIDIA

#ASWF

# MaterialX and ShaderGen

## MaterialX

An open standard for network-based CG object looks originally developed by Lucasfilm

<https://www.materialx.org>

<https://github.com/AcademySoftwareFoundation/MaterialX>

## MaterialX Physically-Based Shading Nodes

<https://www.materialx.org/assets/MaterialX.v1.38.PBRSpec.pdf>

## ShaderGen

Transforms the MaterialX descriptions into executable code  
Contribution by Autodesk

<https://github.com/AcademySoftwareFoundation/MaterialX/blob/main/documents/DeveloperGuide/ShaderGeneration.md>



Chessboard assets by  
Moeen Sayed and  
Mujtaba Sayed for SideFX.

# NVIDIA® MDL Language and SDK

## NVIDIA Material Definition Language (MDL)

Domain-specific language to define PBR materials

Declarative components to compose material graphs

Procedural texturing functions to drive material inputs

## SDK / Compiler / Backends

Translation to HLSL, GLSL, PTX, x86, ARM, LLVM-IR

Interface to sample and evaluate materials

Independent of the lighting and rendering algorithms

## Open Source (BSD 3-clause license)

<https://github.com/NVIDIA/MDL-SDK>



NVIDIA vMaterials v2.2  
1800+ Materials - FREE TO USE

# MaterialXGenMdl

## Library for MDL Code Generation

Official part since MaterialX 1.38.0

<https://github.com/AcademySoftwareFoundation/MaterialX/tree/main/source/MaterialXGenMdl>

Joint development from



Used for example in



**GitHub**

The screenshot displays the GitHub interface for the `materialx/MaterialX` repository. Key features shown include:

- Code:** Main branch, 16 tags.
- Issues:** 6 open issues.
- Pull requests:** 1 pending pull request.
- Actions:** CI status and logs.
- Projects:** Project board and tasks.
- Security:** Vulnerability analysis.
- Insights:** Performance metrics and code quality reports.
- About:** MaterialX is described as an open standard for rich material and look-development content between applications and renderers.
- Releases:** Version 1.38.0 (Latest) released 23 days ago, with 15 releases in total.
- Packages:** No packages published.
- Contributors:** 18 contributors listed.
- Languages:** C++ (89.7%), Python (3.6%), GLSL (2.5%), CMake (2.1%), C (1.5%), Objective-C (0.6%).

# MaterialXGenMdl released with 1.38.7

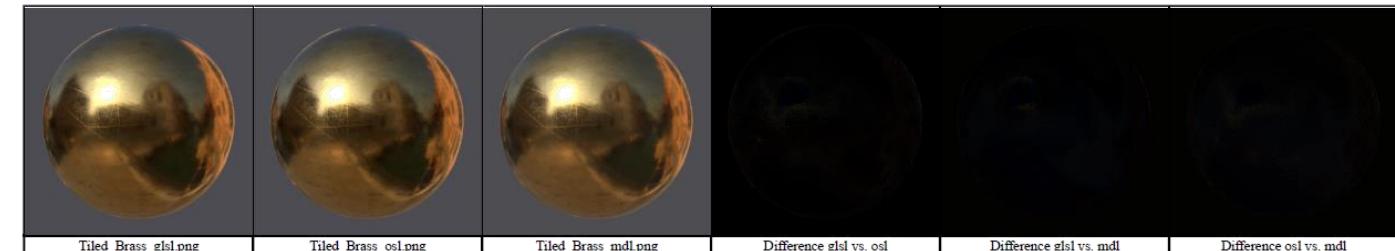
Better integration in the MaterialX Test Suite

MDL DXR Example Renderer matches the MaterialXTest scene

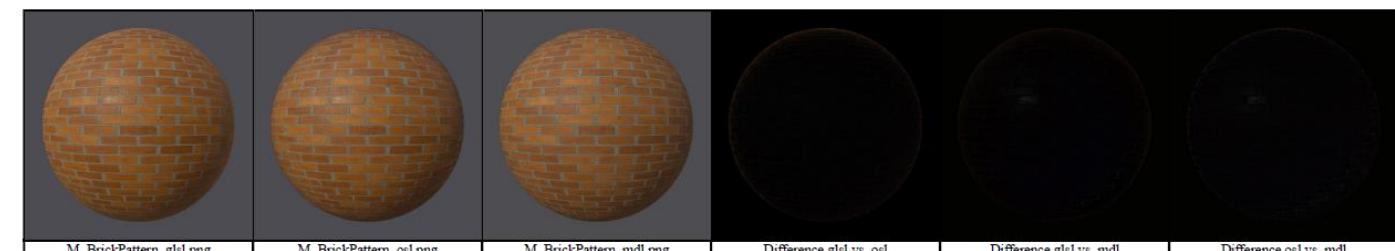
added 3-way comparison to the *tests\_to\_html.py* script

gsl (in: E:/github/MaterialX/build) vs osl (in: E:/github/MaterialX/build) vs mdl (in: E:/github/MaterialX/build)

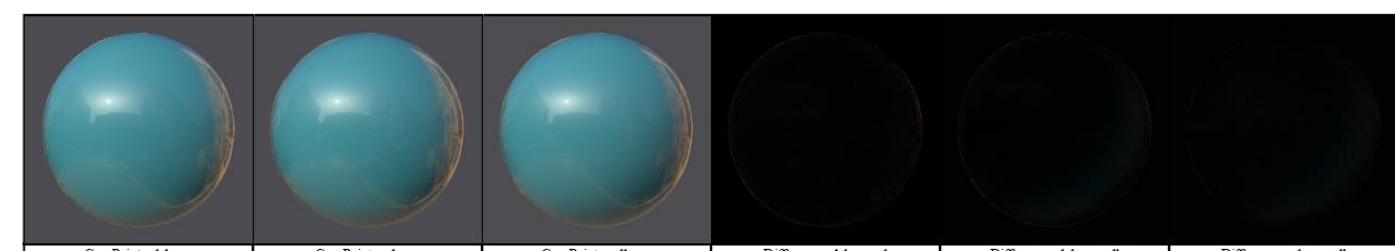
E:\github\MaterialX\build\bin\resources\Materials\Examples\StandardSurface\standard\_surface\_brass\_tiled:



E:\github\MaterialX\build\bin\resources\Materials\Examples\StandardSurface\standard\_surface\_brick\_procedural:



E:\github\MaterialX\build\bin\resources\Materials\Examples\StandardSurface\standard\_surface\_carpaint:



GLSL

OSL

MDL

GLSL  
vs. OSL

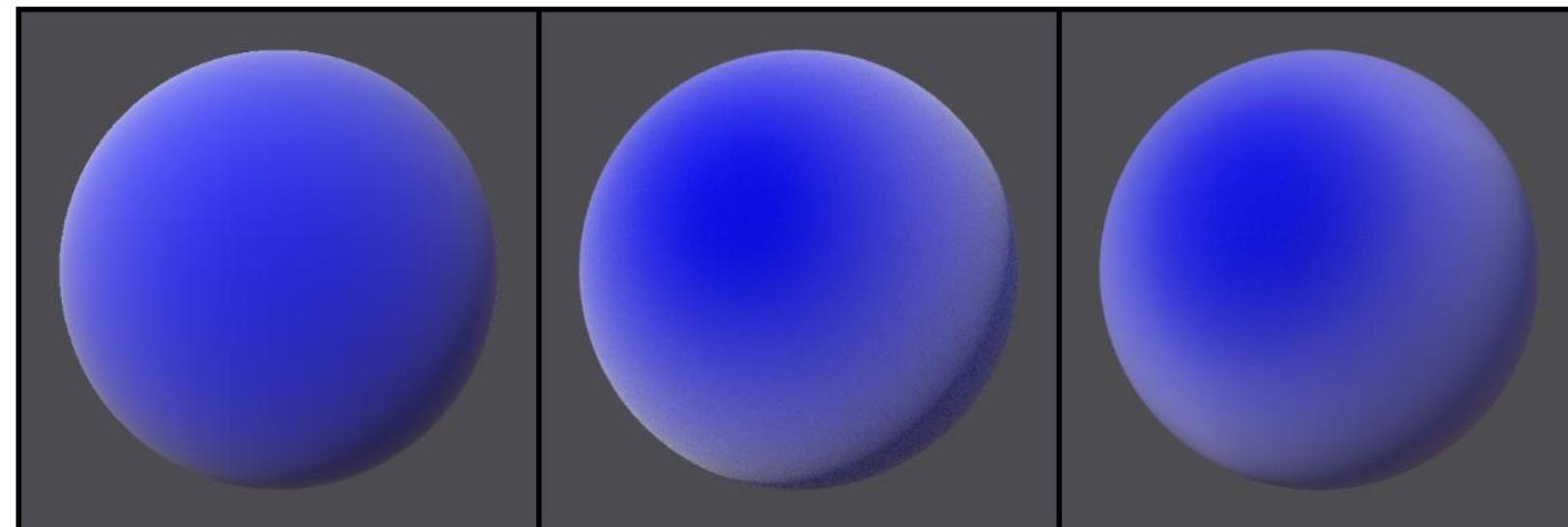
MDL  
vs. GLSL

MDL  
vs. OSL

# MaterialXGenMdl released with 1.38.7

## Full Sheen BSDF Support

Sheen can be layered over arbitrary nodes



GLSL  
(approximation only)

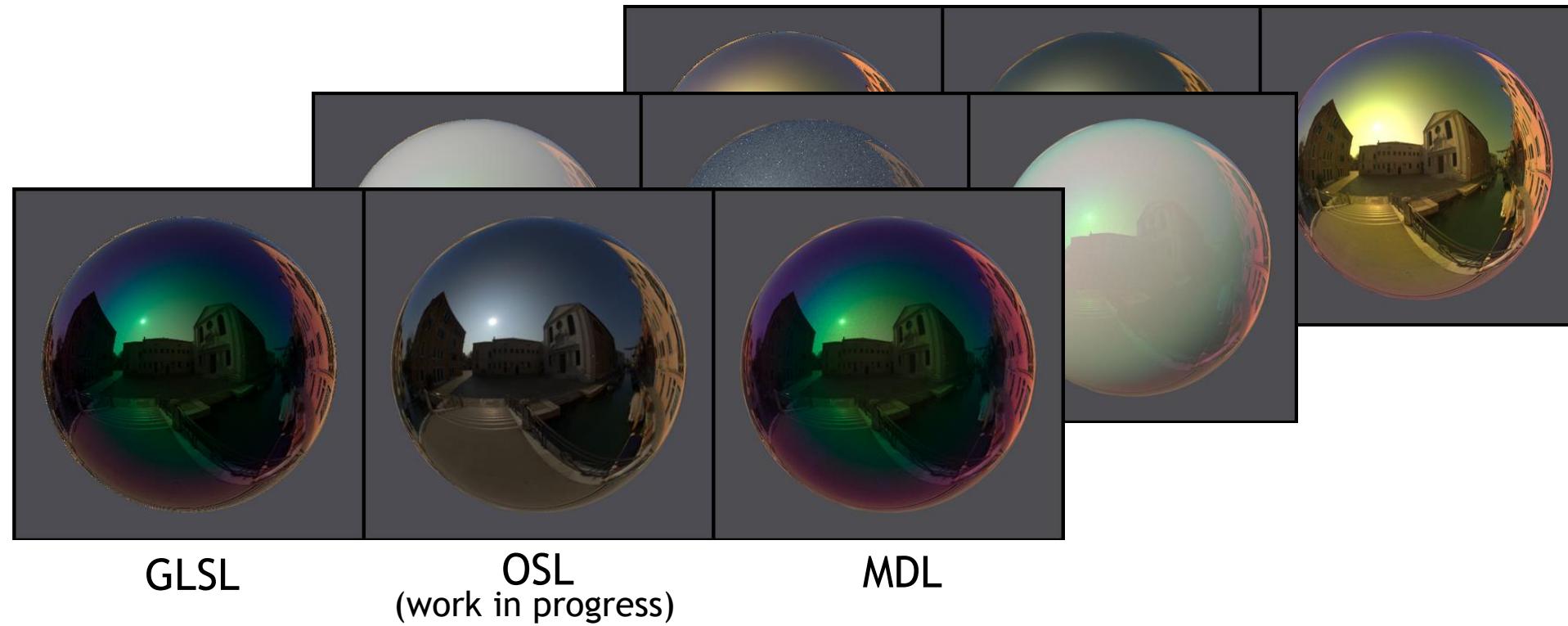
OSL

MDL

# MaterialXGenMdl released with 1.38.7

## Proper Support for the Thin-Film BSDF

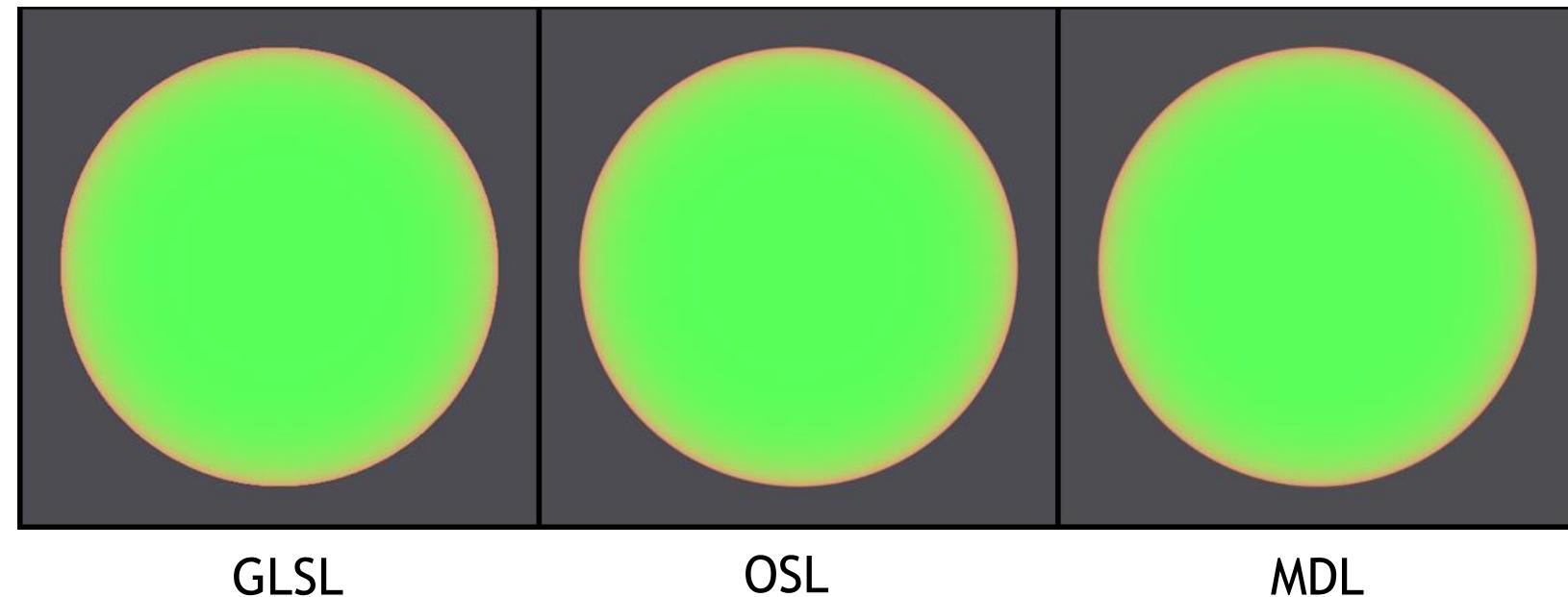
over dielectrics, conductors, and the generalized Schlick



# MaterialXGenMdl released with 1.38.7

New Generalized Schlick EDF

Coordinated specification between MDL and MaterialX workgroup



# MaterialXGenMdl released with 1.38.7

## Volume Emission

### Full Support for Add BSDF, EDF, and VDF Nodes

Implemented by new MDL *unbounded\_mix*

Note, can violate energy conservation laws!

## Minor Updates and Fixes

Support for swizzles on custom types (used in UsdPreviewSurface)

Handle structures that have Surface Shader fields

Resource resolution on application side for more flexibility

Improved generalized Schlick BSDF

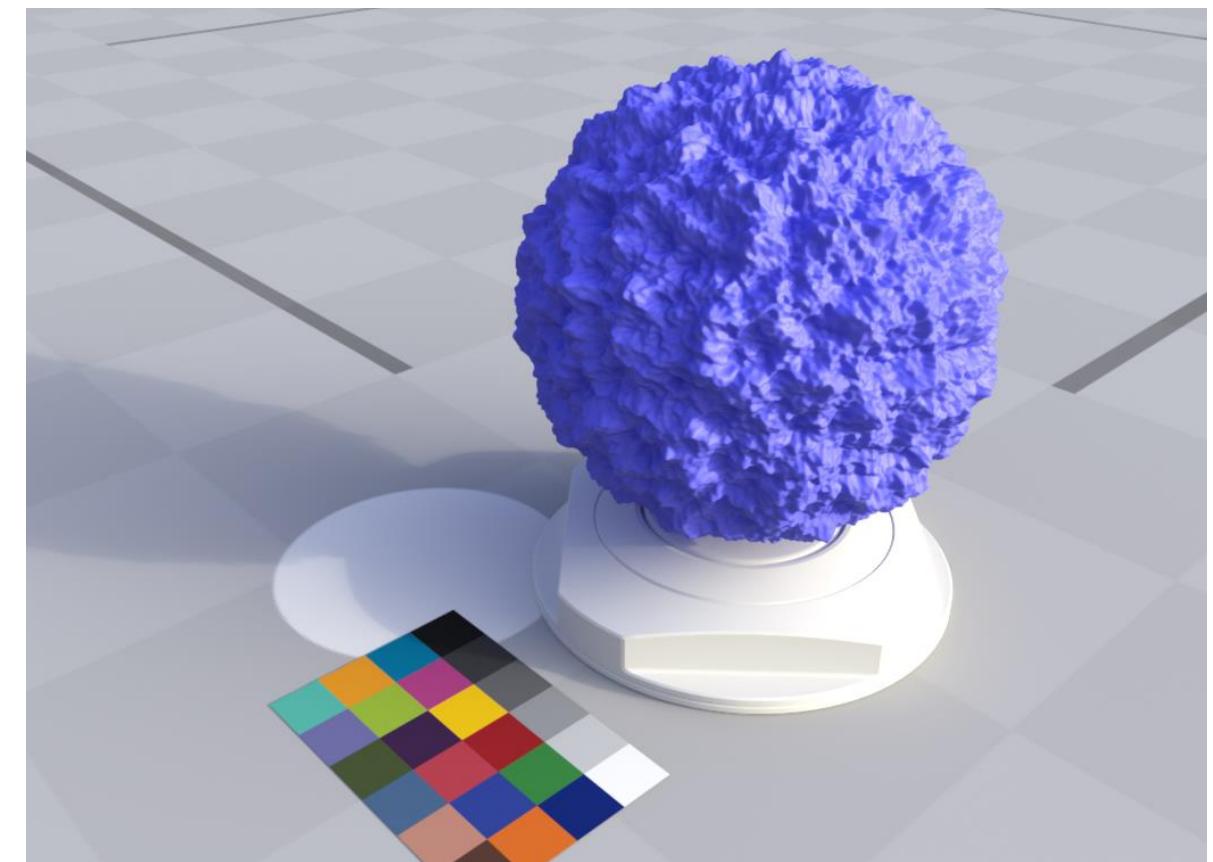
# MaterialXGenMdl In Progress

## Displacements

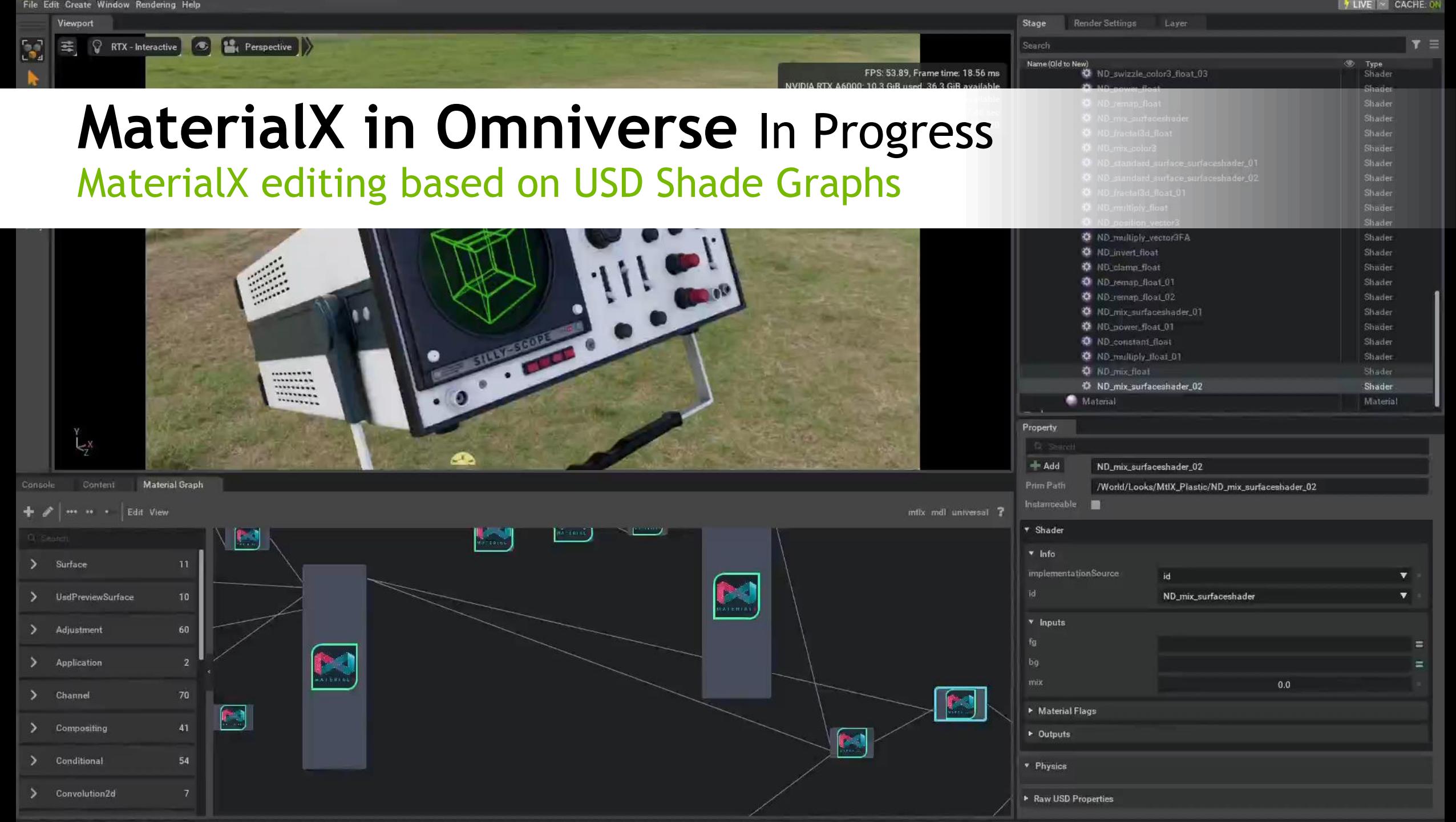
Already in the github main branch

## Versioning

Add MDL ShaderGen option to select the target MDL version 1.6, 1.7, or 1.8



Rendered in NVIDIA Iray

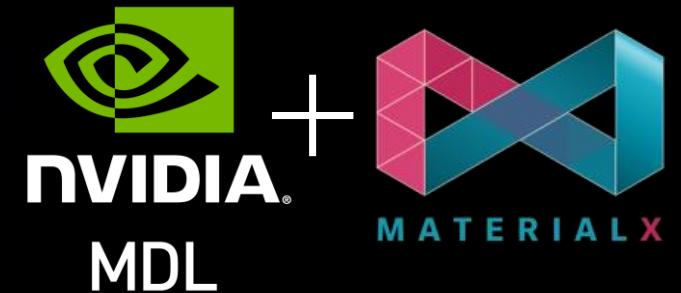


open  
Source  
days'23



#ASWF

Thank you



Rendered in NVIDIA Omniverse

open  
Source  
days'23

/\* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

RenderMan and MaterialX

Mark Manca

August 2, 2023

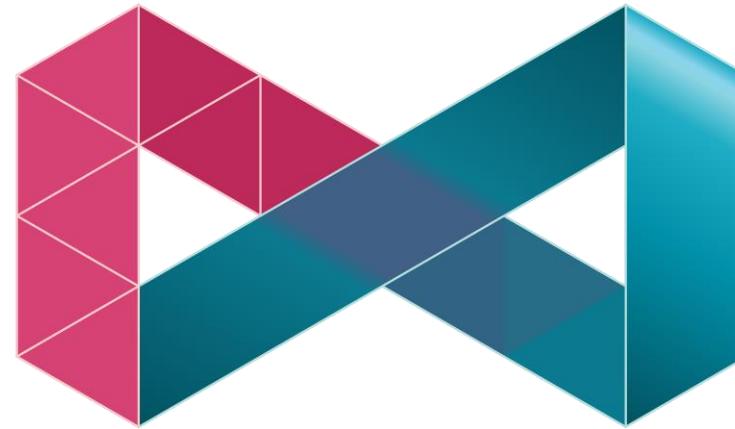
#ASWF

/\* ACADEMY SOFTWARE FOUNDATION

# The path to create once, use anywhere



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF



## MATERIALX

- Final renders for Film/Streaming
- Previews in GL for Animators and others in different DCCs
- LED walls/game engine driven env
- Theme parks
- Digital backlot/Legacy content
- Interchange w/other studios



RENDERMAN



# MaterialX and Future RenderMan



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Build support for the full expressivity of MaterialX
  - Down to the OSL primitive closure level
  - Support arbitrary composition of these “lobe primitives”
- Continue innovating with ILM on MaterialX Lama
- Building it all in XPU

open  
Source  
days'23

/\* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

Q U I L T I X

Manuel Köster & Richard Frangenberg

2nd of August, 2023

#ASWF

/\* ACADEMY SOFTWARE FOUNDATION

# What is QUILTIX



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Material Node Editor based on



- Export and Import Materials
- Live preview of your shaded Assets based on USD's Hydra

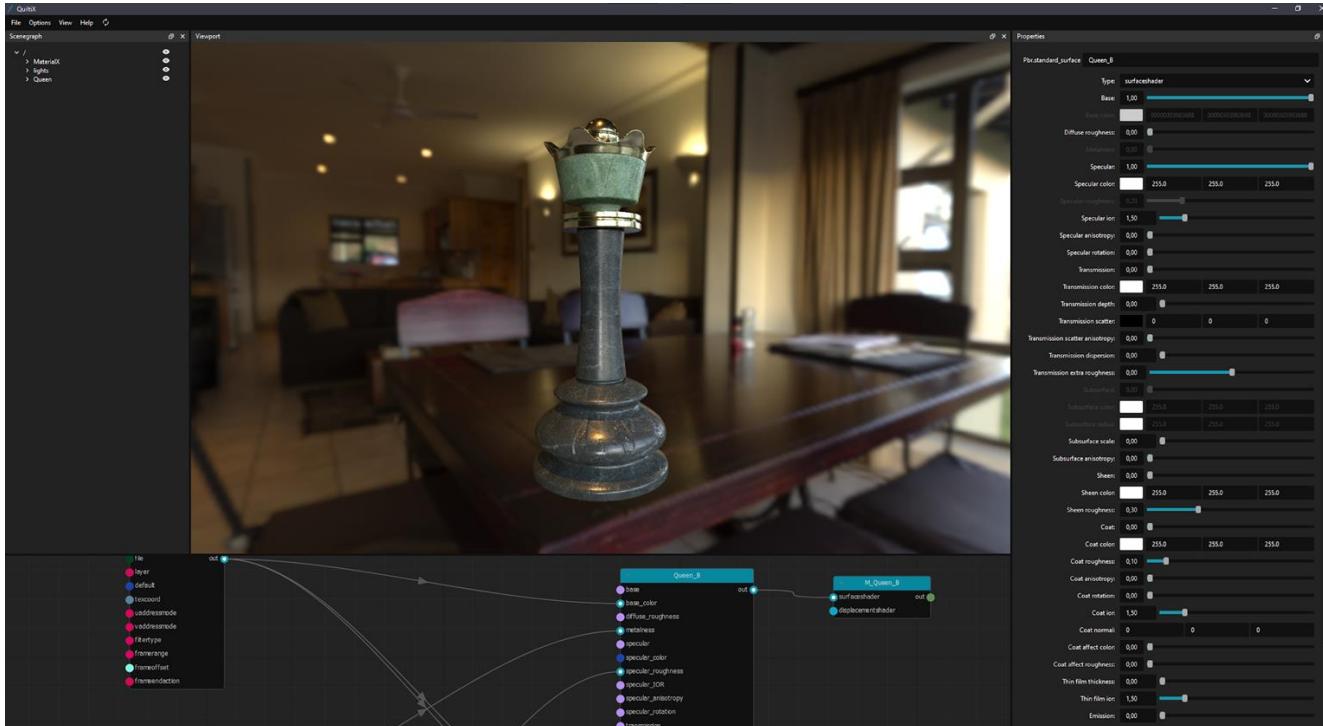
# Why did we create



open  
Source  
days<sup>23</sup>

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Artist friendly
- Hydra renderer integration for production ready look development
- Easily integrate-able into DCCs and Pipelines
- ❤️ Open Source ❤️

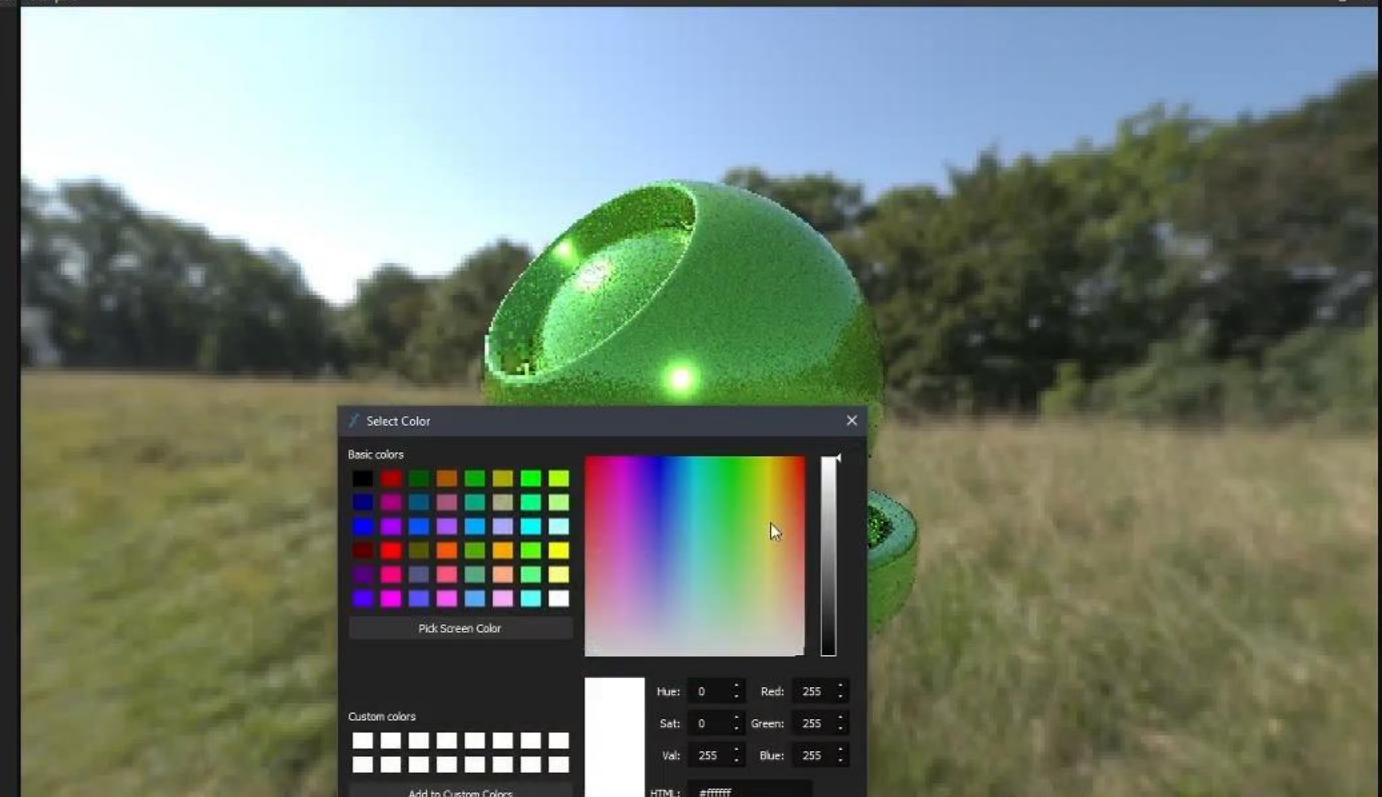


File Options View Help

Scenegraph

- /
- > lights
- > MaterialX
- > Calibration\_Mesh
- > Preview\_Mesh

Viewport



Properties

Pbr.Standard_surface	Standard_surface
Type:	surface shader
Base:	0,62
Base color:	[Color Swatch] 1,0 1,0 1,0
Diffuse roughness:	0,00
Metalness:	0,50
Specular:	1,00
Specular color:	[Color Swatch] 1,0 1,0 1,0
Specular roughness:	0,20
Specular ior:	1,50
Specular anisotropy:	0,00
Specular rotation:	0,00
Transmission:	0,00
Transmission color:	[Color Swatch] 1,0 1,0 1,0
Transmission depth:	0,00
Transmission scatter:	0 0 0
Transmission scatter anisotropy:	0,00
Transmission dispersion:	0,00
Transmission extra roughness:	0,00
Subsurface:	0,00
Subsurface color:	[Color Swatch] 1,0 1,0 1,0
Subsurface radius:	1,0 1,0 1,0
Subsurface scale:	1,00
Subsurface anisotropy:	0,00
Sheen:	1,00
Sheen color:	[Color Swatch] 1,0 0,039216 0,039216
Sheen roughness:	0,30
Coat:	1,00
Coat color:	[Color Swatch] 0,227451 1,0 0,090196
Coat roughness:	0,10
Coat anisotropy:	0,00
Coat rotation:	0,00
Coat ior:	1,50
Coat normal:	0 0 0
Coat affect color:	0,00
Coat affect roughness:	0,00
Thin film thickness:	0,00
Thin film ior:	1,50



Graph is valid.

File Options View Help

Scenegraph

/  
  > lights  
  > Calibration\_Mesh  
  > Preview\_Mesh  
  > MaterialX

Viewport



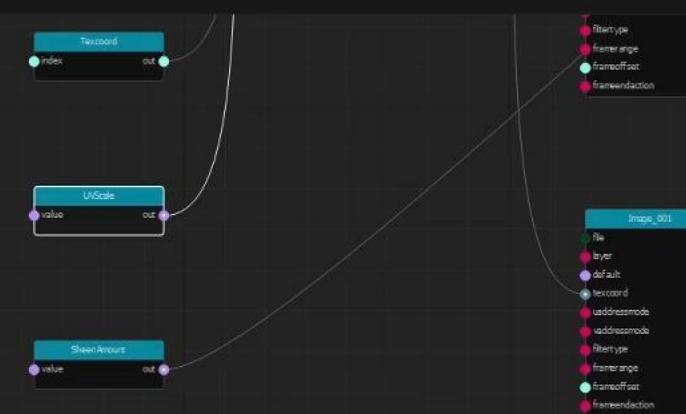
Properties

Procedural.Constant UVScale

Type: float

Value: 7,90

Node Graph NG



Graph is valid.

File Options View Help

Scenegraph

/  
  > lights  
  > Calibration\_Mesh  
  > Preview\_Mesh  
  > MaterialX

Viewport

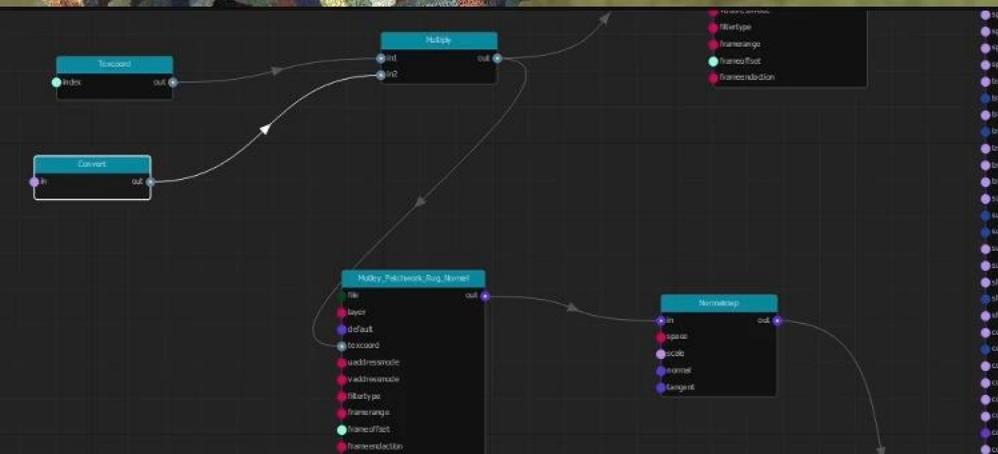


Properties

Channel.Convert Convert

Type: float\_vector2

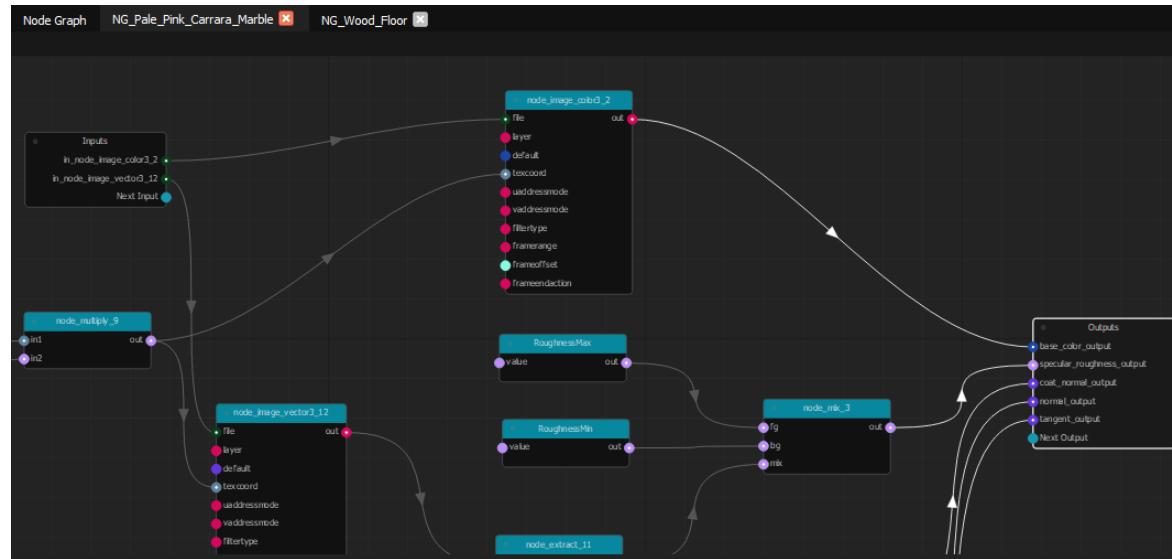
In: 3.43



Graph is valid.

# Nodegraphs

- Organize nodes in subgraphs
- Expose relevant parameters
- Reusable nodegraphs
- Edit node definitions



File Options View Help ⌂

Scenegraph

- ✓ /
- > lights
- > Calibration\_Mesh
- > Preview\_Mesh
- > MaterialIX

Viewport

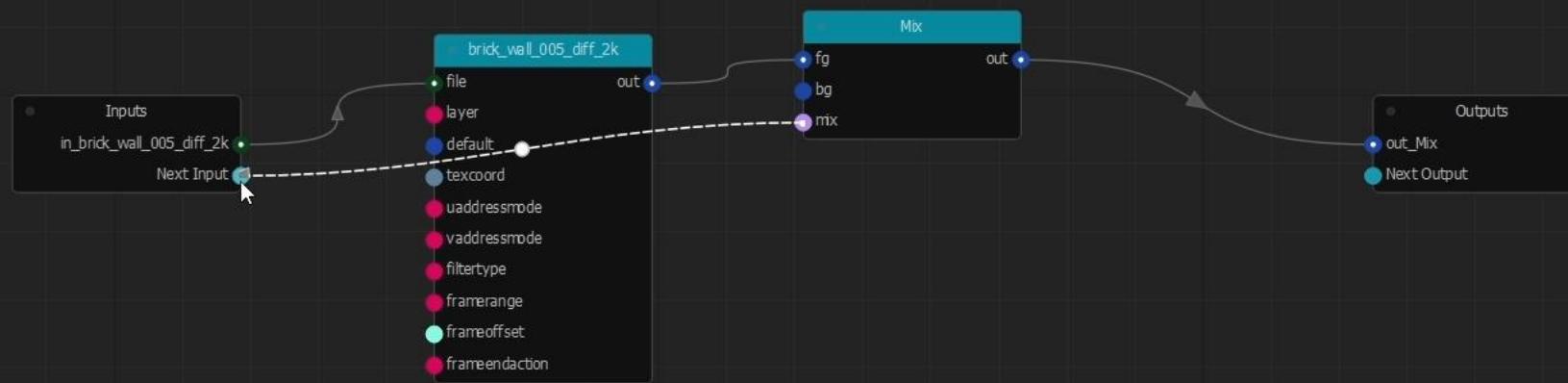


Properties

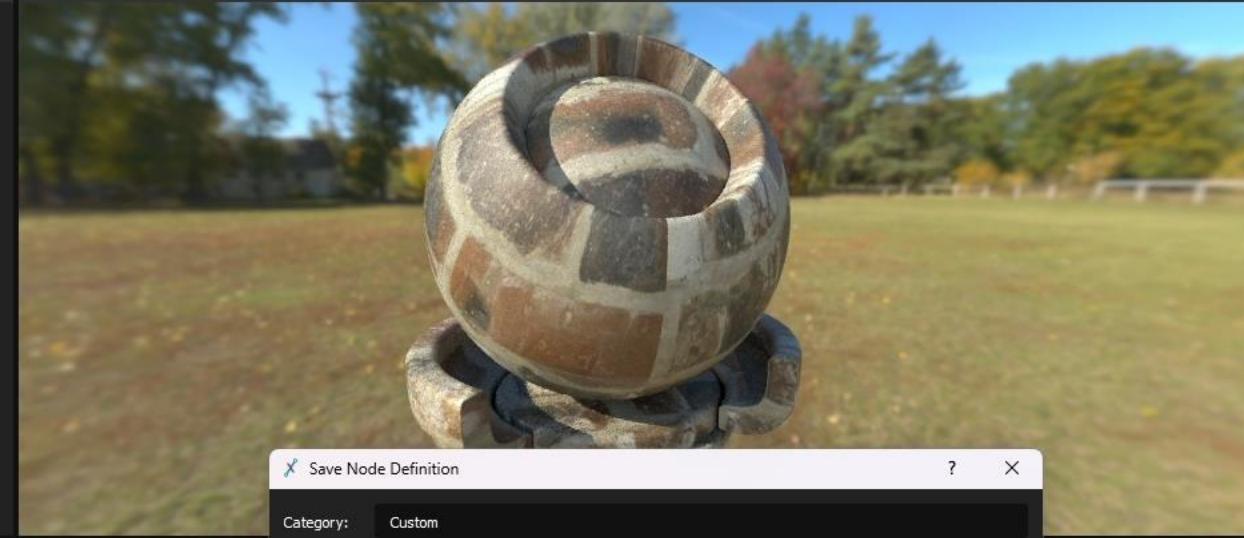
Node Graph

Nodegraph ✎

• Nodegraph



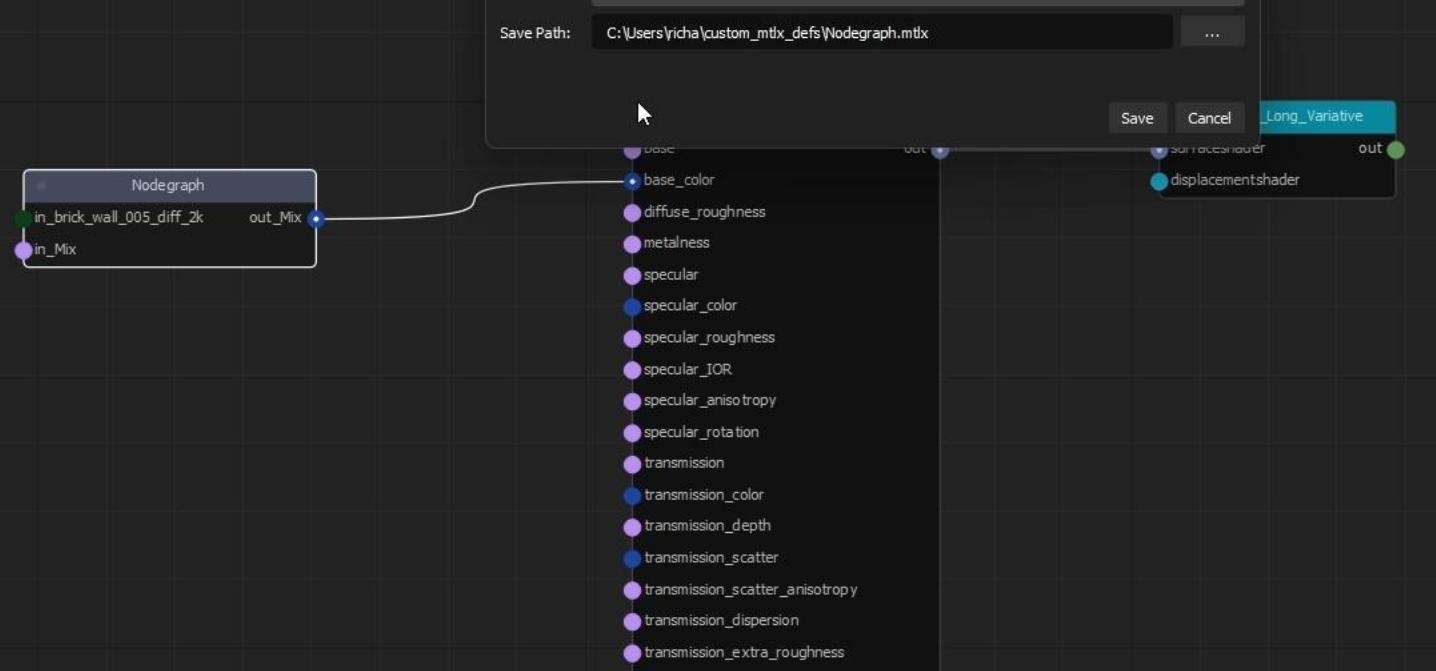
/  
  > lights  
  > Calibration\_Mesh  
  > Preview\_Mesh  
  > MaterialX



Other.Nodegraph Nodegraph

In\_brick\_wall\_005\_diff\_2k: yHaven/texture/brick\_wall\_005\_diff\_2k.jpg

In\_mix: 1,32

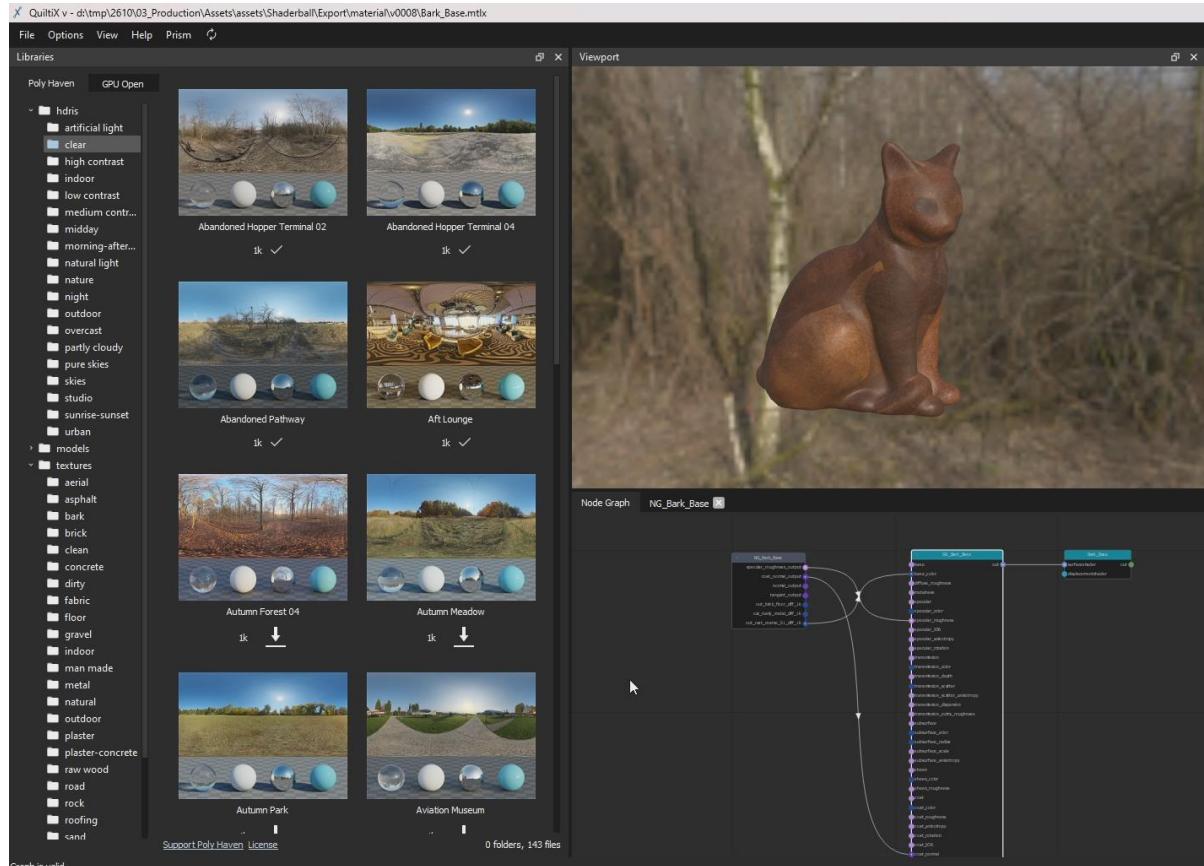


# Pipeline Integration

open  
Source  
days '23

/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Python/Pyside allows easy pipeline integration
- Replaceable viewer/scenegraph widgets
- Custom publish processes



QuiltiX v - d:\tmp\2610\03\_Production\Assets\assets\Shaderball\Export\material\v0008\Bark\_Base.mtlx

File Options View Help Prism

Libraries

Poly Haven GPU Open

hdri  
artificial light  
clear  
high contrast  
indoor  
low contrast  
medium contr...  
midday  
morning-after...  
natural light  
nature  
night  
outdoor  
overcast  
partly cloudy  
pure skies  
skies  
studio  
sunrise-sunset  
urban

models

textures  
aerial  
asphalt  
bark  
brick  
clean  
concrete  
dirty  
fabric  
floor  
gravel  
indoor  
man made  
metal  
natural  
outdoor  
plaster  
plaster-concrete  
raw wood  
road  
rock  
roofing  
sand

Aerial Beach 02 1k JPG Aerial Beach 03 1k JPG

Aerial Grass Rock 1k JPG Aerial Ground Rock 1k JPG

Aerial Mud 1 1k JPG Aerial Rocks 01 1k JPG

Aerial Rocks 02 1k JPG Aerial Rocks 04 1k JPG

Support Poly Haven License

0 folders, 18 files

Viewport

Properties

Pbr.Standard\_surface SR\_Bark\_Base

Type: surface shader

Base: 0,80

Base color: 1920929 1920929 1920929

Diffuse roughness: 0,00

Metalness: 0,00

Specular: 1,00

Specular color: 0 0 0

Specular roughness: 0,20

Specular ior: 1,50

Specular anisotropy: 0,00

Specular rotation: 0,00

Transmission: 0,00

Transmission color: 1.0 1.0 1.0

Transmission depth: 0,00

Transmission scatter: 0 0 0

Transmission scatter anisotropy: 0,00

Transmission dispersion: 0,00

Transmission extra roughness: 0,00

Subsurface: 0,00

Subsurface color: 1.0 1.0 1.0

Subsurface radius: 1.0 1.0 1.0

Subsurface scale: 1,00

Subsurface anisotropy: 0,00

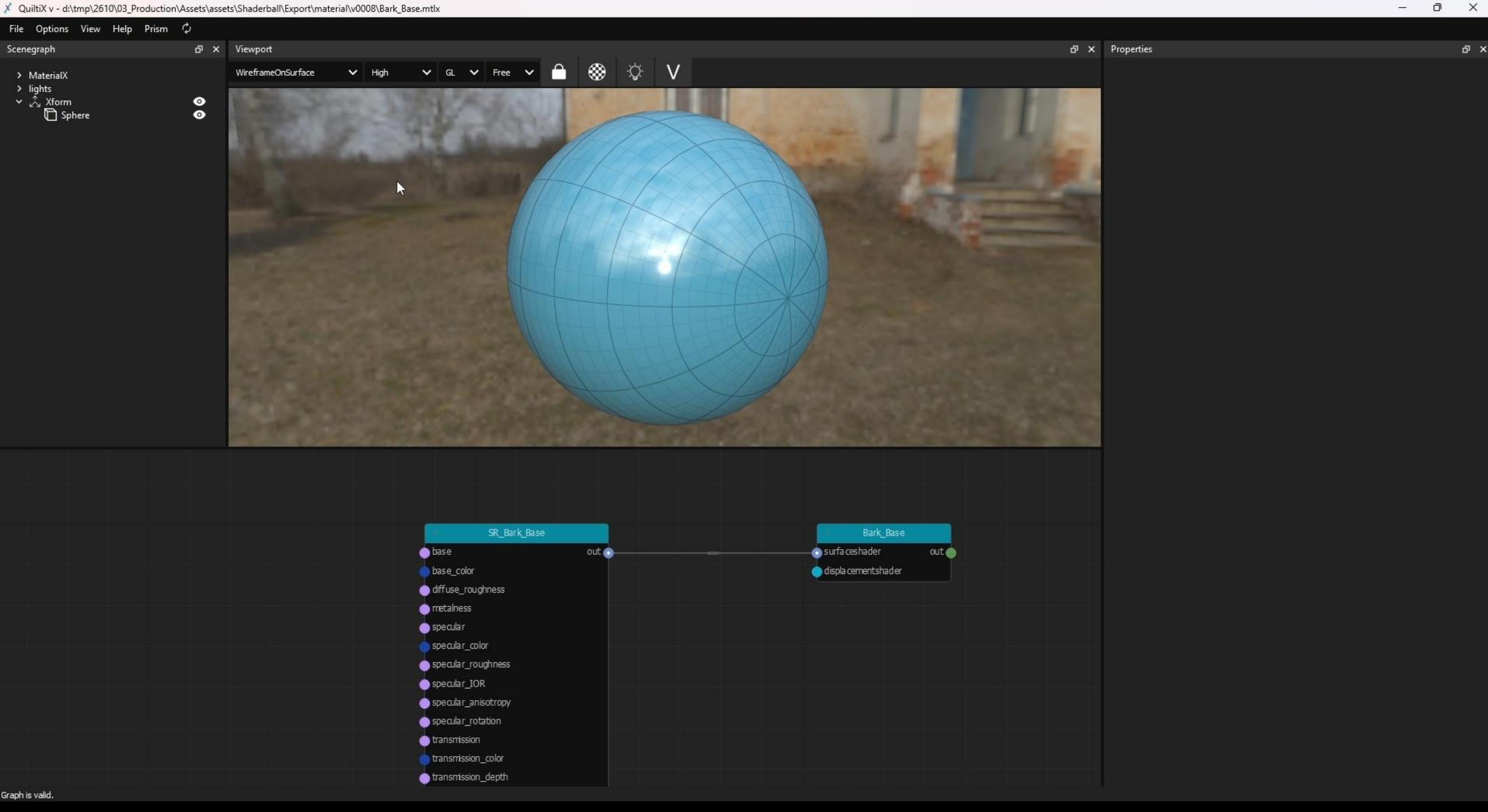
Sheen: 0,00

Sheen color: 1.0 1.0 1.0

Sheen roughness: 0,30

Node Graph NG\_Bark\_Base

Graph is valid.



# Pipeline Friendly



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF

- Highly decoupled and extensible
- Customizable through environment variables
- BYO MaterialX/USD (if you like)
- Open source (Apache)



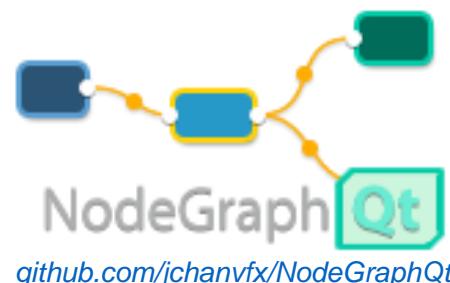
# Tech stack



/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF



MATERIAL X



[github.com/jchanvfx/NodeGraphQt](https://github.com/jchanvfx/NodeGraphQt)



# Compatibility

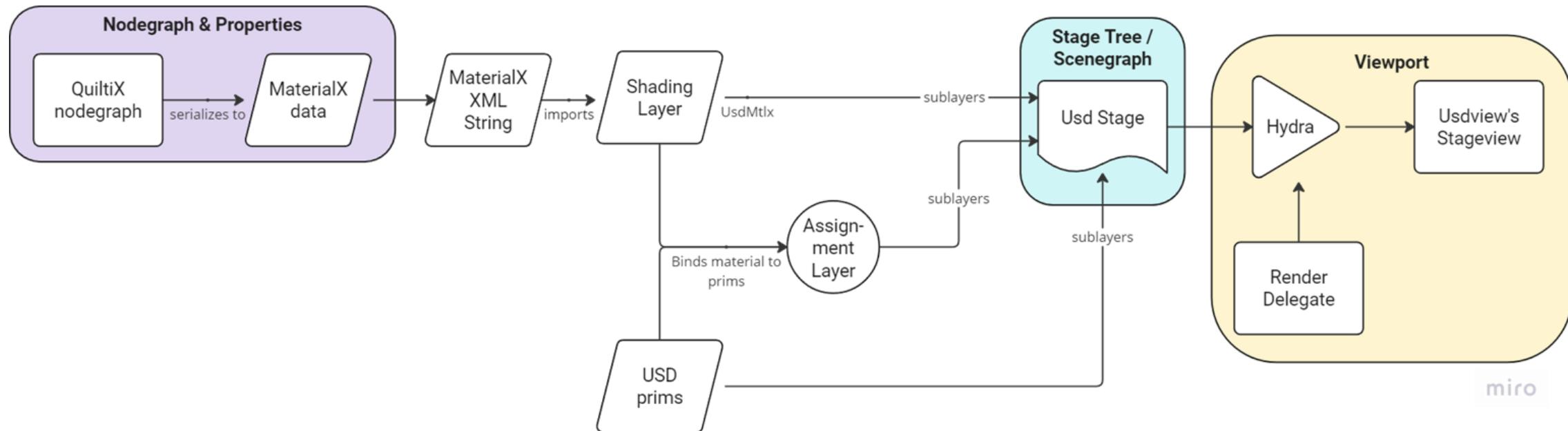
- OpenUSD 22.08 - MaterialX 1.38.3 (+ Arnold)
- Experimental: OpenUSD 23.08 - MaterialX 1.38.7
- Karma: OpenUSD 22.05 (Houdini 19.5)
- Other USD/MaterialX versions (untested)



# How does QUILTiX work

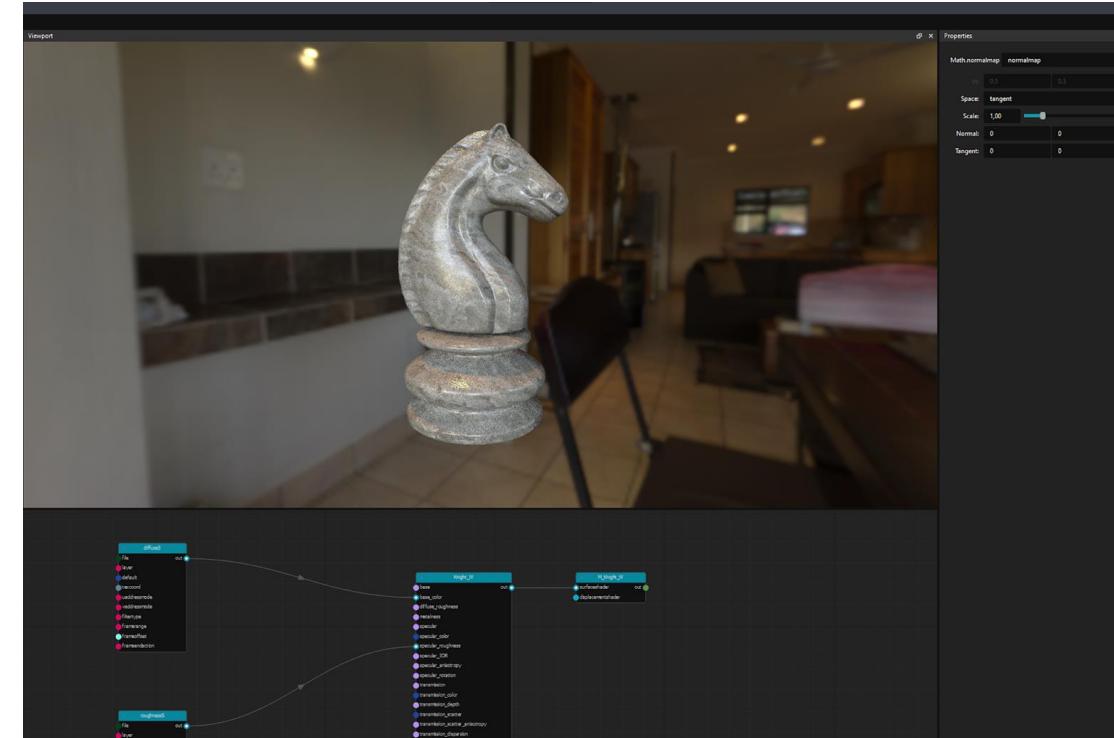
open  
Source  
days '23

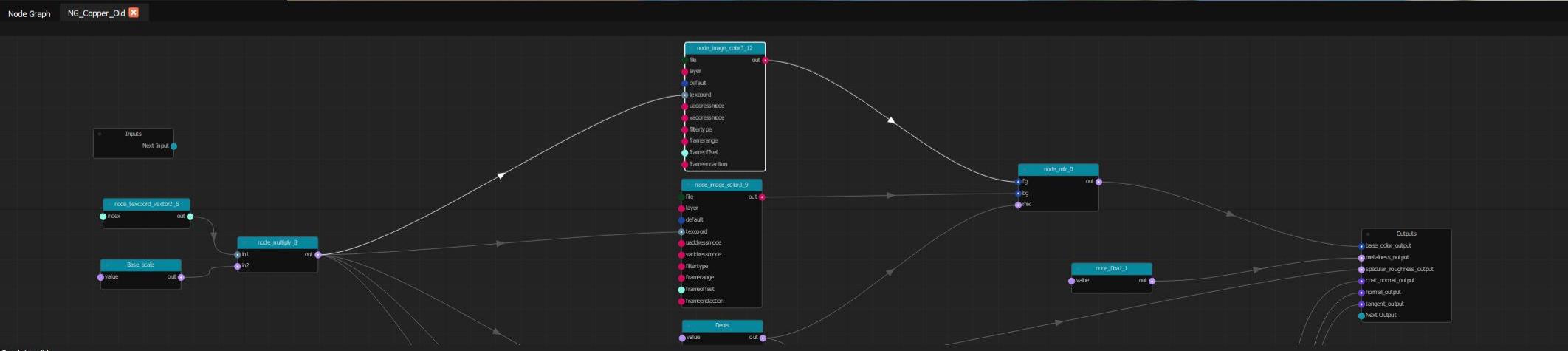
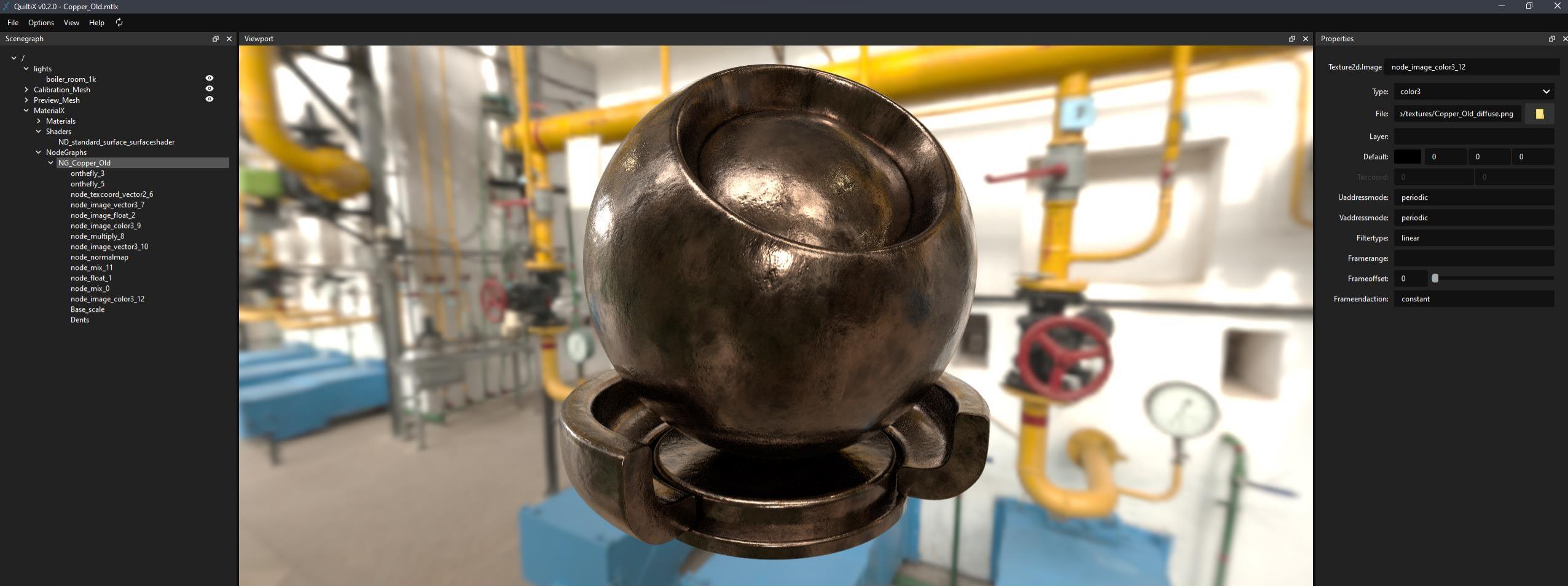
/\* ACADEMY  
SOFTWARE  
FOUNDATION  
#ASWF



# Next steps

- CI/CD
  - Testing & Linting
- Community collaboration
- Deepen NodeGraphQt collab
- Features
  - Renderer settings
  - Support more MaterialX features (like parameter folders)
  - Colorspace support
- And hopefully much more





QuiltX v0.2.0 - TH\_Fabric\_Leather.mtlx

File Options View Help

Scenegraph

Viewport

Properties

Pbr.Standard\_surface SR\_TH\_Fabric\_Leather

Type: surfaceshader

Base: 0,80

Base color: 1920929 1920929 1920929

Diffuse roughness: 0,00

Metalness: 0,00

Specular: 1,00

Specular color: 1.0 1.0 1.0

Specular roughness: 0,20

Specular ior: 1,50

Specular anisotropy: 0,00

Specular rotation: 0,00

Transmission: 0,00

Transmission color: 1.0 1.0 1.0

Transmission depth: 0,00

Transmission scatter: 0 0 0

Transmission scatter anisotropy: 0,00

Transmission dispersion: 0,00

Transmission extra roughness: 0,00

Subsurface: 0,00

Subsurface color: 1.0 1.0 1.0

Subsurface radius: 1.0 1.0 1.0

Subsurface scale: 1,00

Subsurface anisotropy: 0,00

Sheen: 0,00

Sheen color: 1.0 1.0 1.0

Sheen roughness: 0,30

NG\_TH\_Fabric\_Leather

SR\_TH\_Fabric\_Leather

TH\_Fabric\_Leather

Graph is valid.

```
graph LR; NG[NG_TH_Fabric_Leather] -- "base_color_output" --> SR[SR_TH_Fabric_Leather]; NG -- "specular_roughness_output" --> SR; NG -- "coat_normal_output" --> SR; NG -- "normal_output" --> SR; NG -- "tangent_output" --> SR; SR -- "base" --> TH[TH_Fabric_Leather]; SR -- "diffuse_roughness" --> TH; SR -- "metalness" --> TH; SR -- "specular" --> TH; SR -- "specular_color" --> TH; SR -- "specular_roughness" --> TH; SR -- "specular_IOR" --> TH; SR -- "specular_anisotropy" --> TH; SR -- "specular_rotation" --> TH; SR -- "transmission" --> TH; SR -- "transmission_color" --> TH; SR -- "transmission_depth" --> TH; SR -- "transmission_scatter" --> TH; SR -- "transmission_scatter_anisotropy" --> TH; SR -- "transmission_dispersion" --> TH; SR -- "transmission_extra_roughness" --> TH; SR -- "subsurface" --> TH;
```

File Options View Help ⌘

Scenegraph

/  
 > lights  
 > Calibration\_Mesh  
 > Preview\_Mesh  
 > MaterialX  
 > Materials  
 > TH\_Castle\_Brick\_Broken  
 ND\_standard\_surface\_surfaceshader  
 > NG\_TH\_Castle\_Brick\_Broken  
 onthefly\_2  
 onthefly\_4  
 node\_texcoord\_vector2\_5  
 node\_multiply\_6  
 node\_image\_vector3\_7  
 node\_normalmap  
 node\_image\_color3\_0  
 node\_mix\_1  
 node\_image\_vector3\_9  
 node\_extract\_8  
 uv  
 RoughnessHigh  
 RoughnessLow  
 > Shaders  
 > NodeGraphs

Viewport



Properties

Texture2d.Image node\_image\_color3\_0

Type: color3

File: b/textures/TH\_Castle\_Brick\_Broken\_baseColor.png

Layer:

Default: 0 0 0 0

Texcoord: 0 0

Uaddressmode: periodic

Vaddressmode: periodic

Filtertype: linear

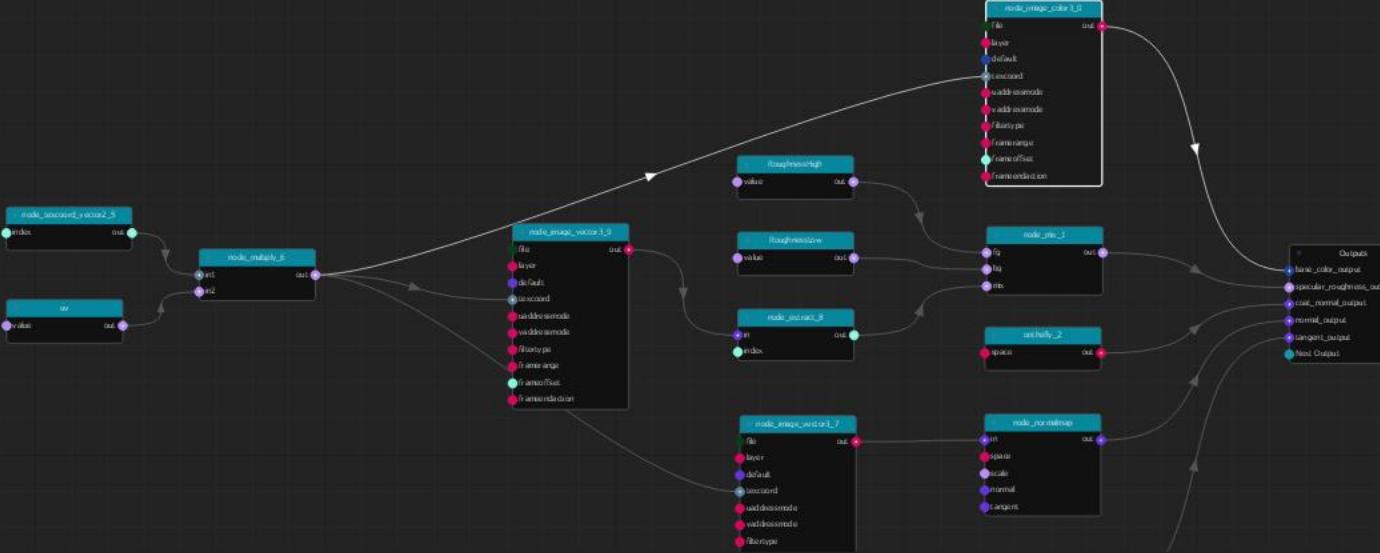
Framerange:

Frameoffset: 0

Frameendaction: constant

Node Graph

NG\_TH\_Castle\_Brick\_Broken



Graph is valid.

## Properties

Pbr.Standard\_surface SR\_TH\_Metal\_Plate

Type: surfaceshader

Base: 0,80

Base color: 011920929 011920929 011920929

Diffuse roughness: 0,00

Metalness: 0,00

Specular: 1,00

Specular color: 1.0 1.0 1.0

Specular roughness: 0,20

Specular ior: 1,50

Specular anisotropy: 0,00

Specular rotation: 0,00

Transmission: 0,00

Transmission color: 1.0 1.0 1.0

Transmission depth: 0,00

Transmission scatter: 0 0 0

Transmission scatter anisotropy: 0,00

Transmission dispersion: 0,00

Transmission extra roughness: 0,00

Subsurface: 0,00

Subsurface color: 1.0 1.0 1.0

Subsurface radius: 1.0 1.0 1.0

Subsurface scale: 1,00

Subsurface anisotropy: 0,00

Sheen: 0,00

Sheen color: 1.0 1.0 1.0

Sheen roughness: 0,30

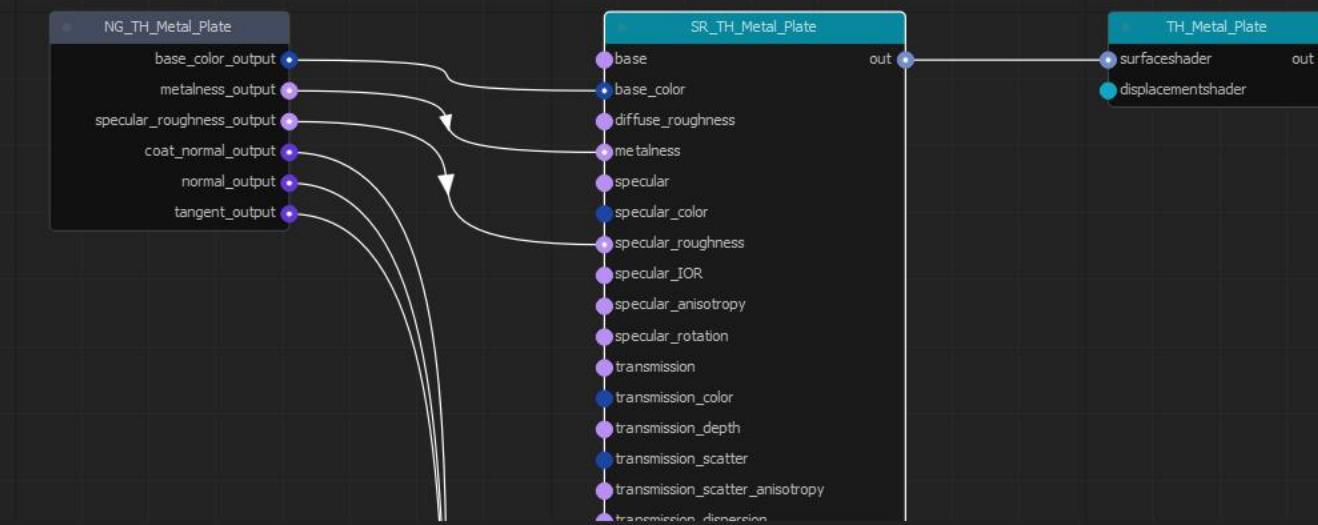
## Scenegraph

```

< />
> lights
> Calibration_Mesh
> Preview_Mesh
> MaterialX
  > Materials
    > TH_Metal_Plate
      ND_standard_surface_surfaceshader
      > NG_TH_Metal_Plate
  > Shaders
    ND_standard_surface_surfaceshader
  > NodeGraphs
    > NG_TH_Metal_Plate

```

## Viewport



open  
Source  
days'23

# Thanks to

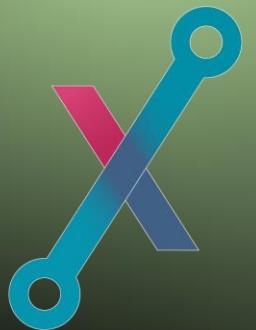


#ASWF

/\* ACADEMY SOFTWARE FOUNDATION

open  
Source  
days'23

Try



out today!

[github.com/PrismPipeline/QuiltiX](https://github.com/PrismPipeline/QuiltiX)  
[pypi.org/project/QuiltiX](https://pypi.org/project/QuiltiX)



#ASWF

/\* ACADEMY SOFTWARE FOUNDATION



Thank You!  
Questions?