

open Source days^{'23}

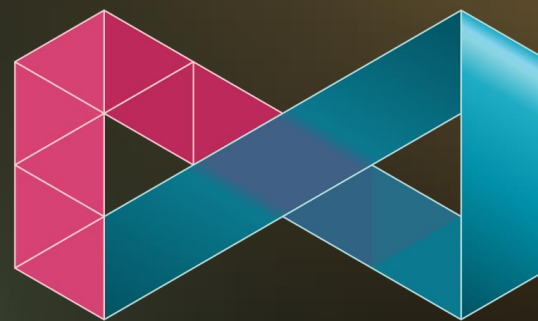
#ASWF

/* ACADEMY SOFTWARE FOUNDATION

open
Source
days²³

/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series



M A T E R I A L X

August 2nd, 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
days²³

/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

OpenPBR

August 2nd 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

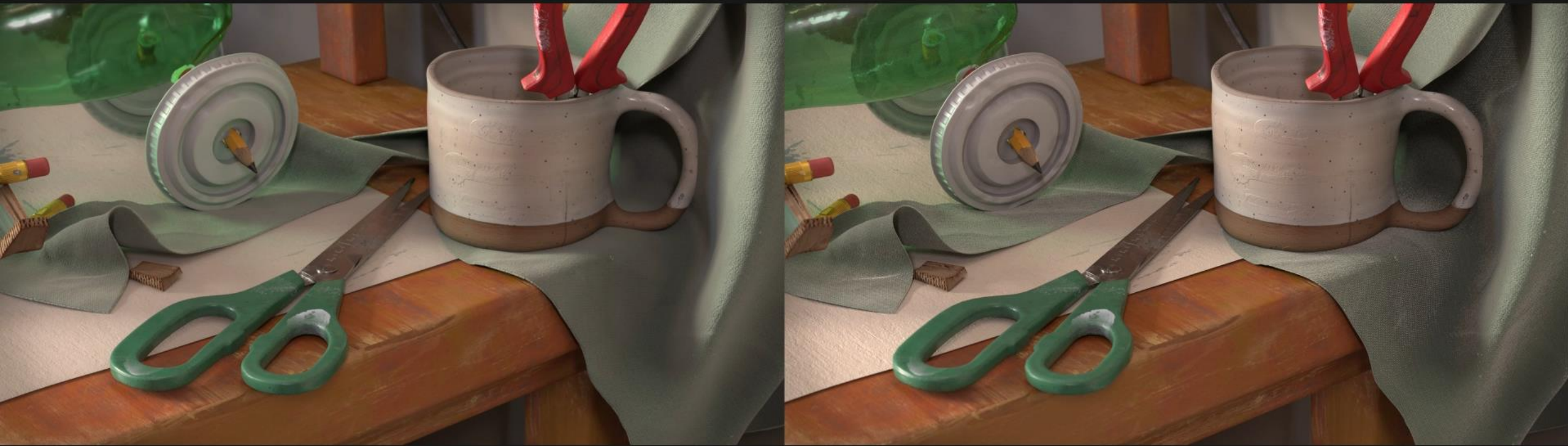


OpenPBR inside Adobe proprietary renderer





OpenPBR inside Maya + Arnold



OpenPBR inside Adobe proprietary renderer





OpenPBR inside Maya + Arnold



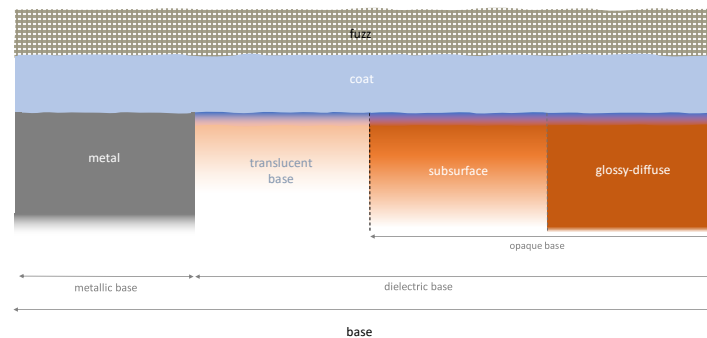
OpenPBR inside Adobe proprietary renderer



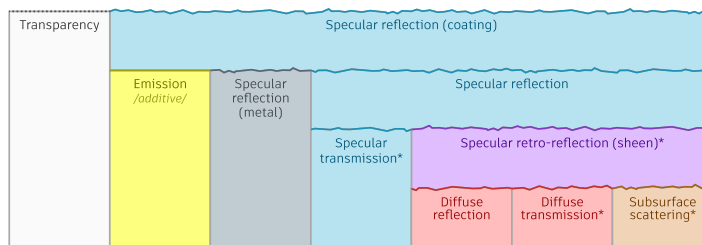
OpenPBR – Technical Overview

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

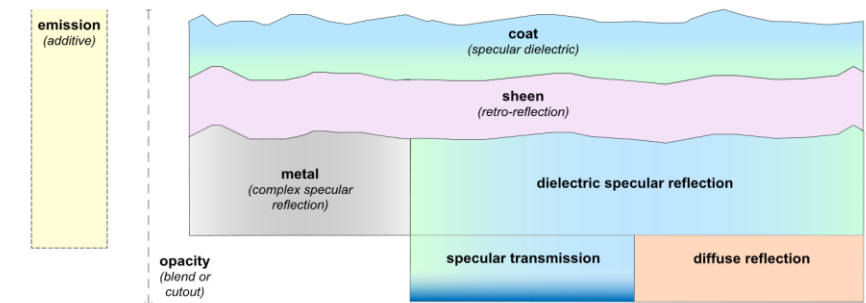
OpenPBR Surface



Autodesk Standard Surface



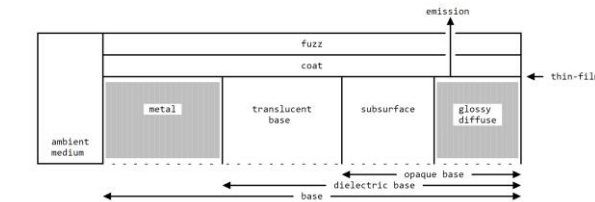
Adobe Standard Material



OpenPBR – Technical Overview

Principled and physically based layering specification

Formal structure (implementation agnostic)



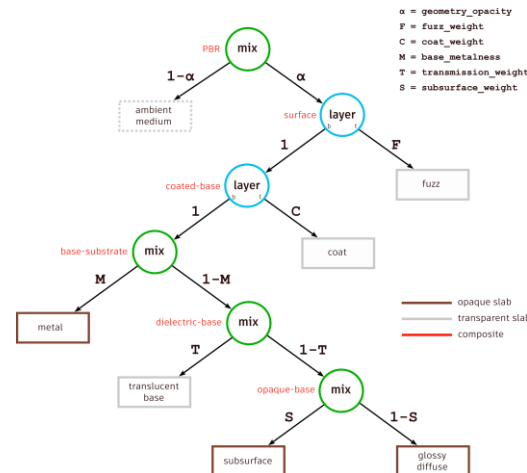
To summarize the formal structure, this consists of the following slabs:

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

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

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

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

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



OpenPBR – Technical Overview

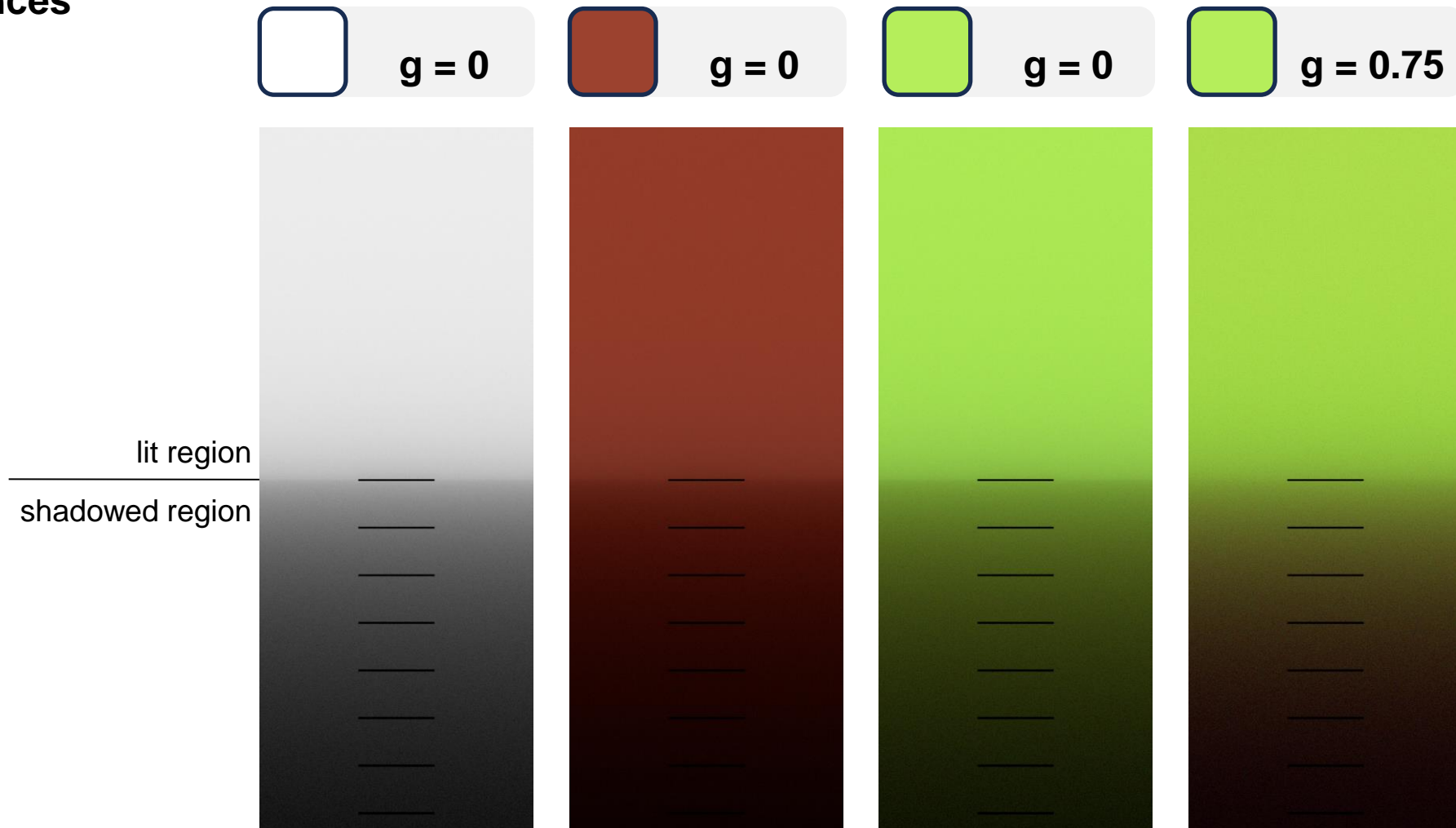
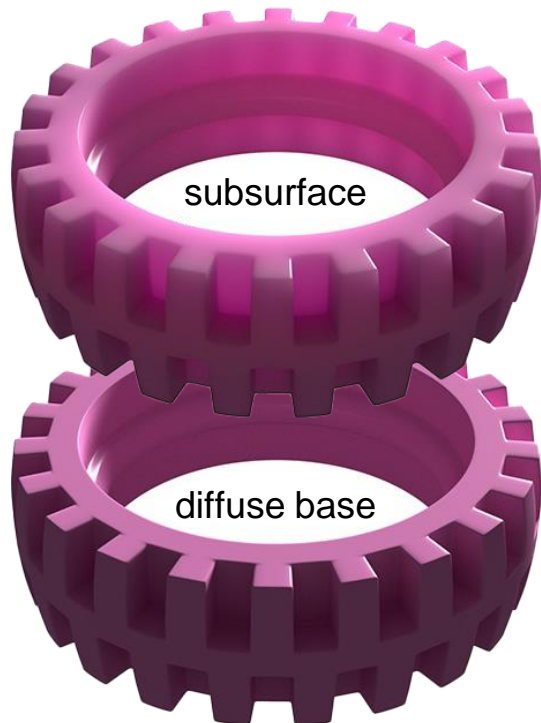


/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

Commonalities and differences

Dedicated SSS component

Inspired by Standard Surface

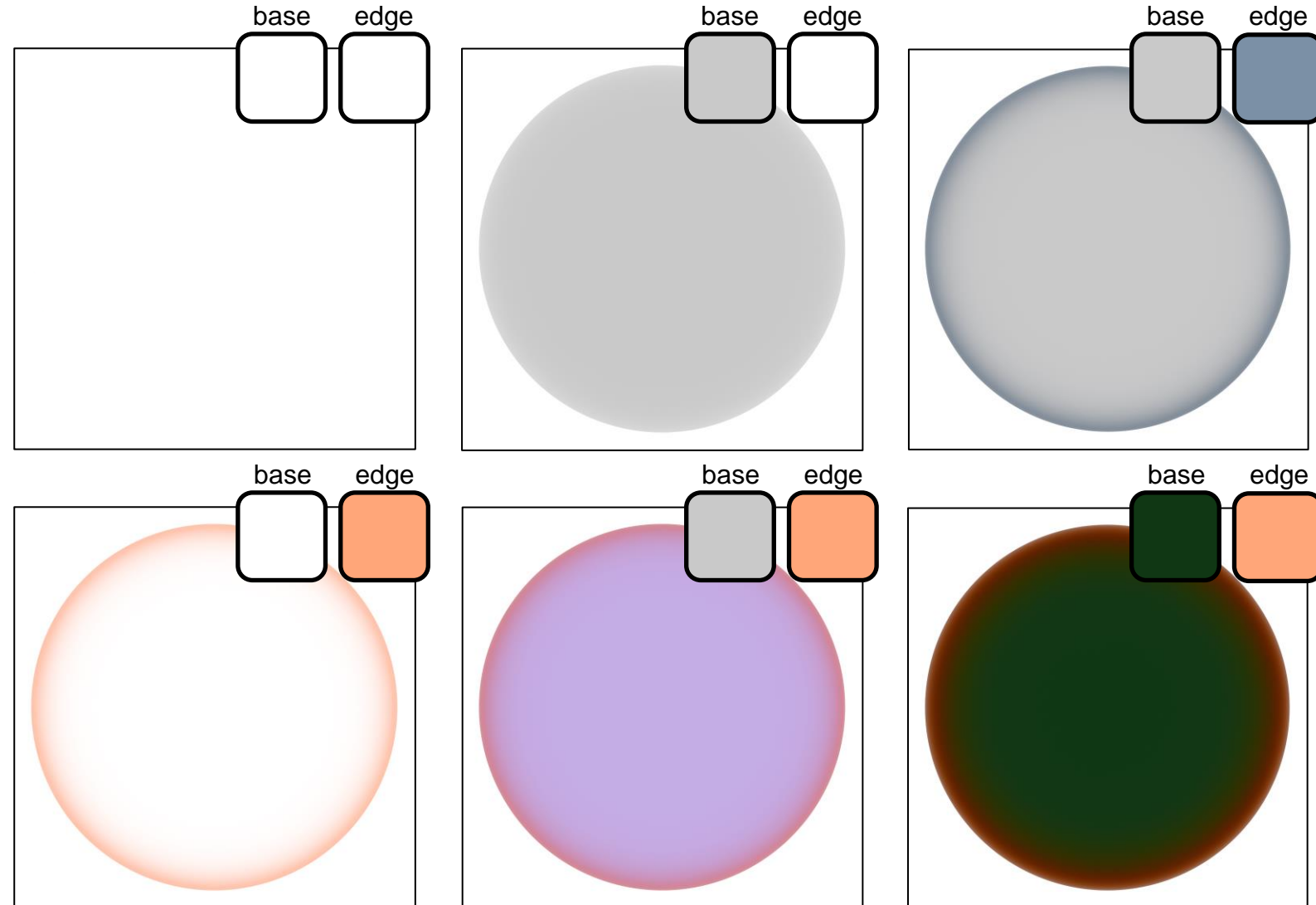
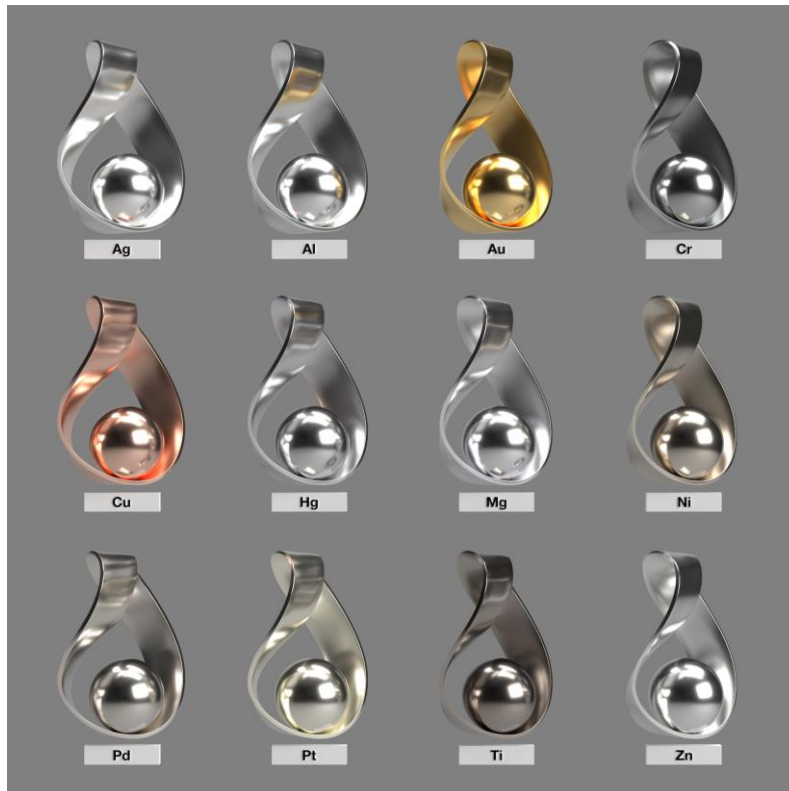


OpenPBR – Technical Overview

Commonalities and differences

F82-tint metal reflectivity

Inspired by Adobe Standard Material



OpenPBR – Technical Overview



/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

Commonalities and differences

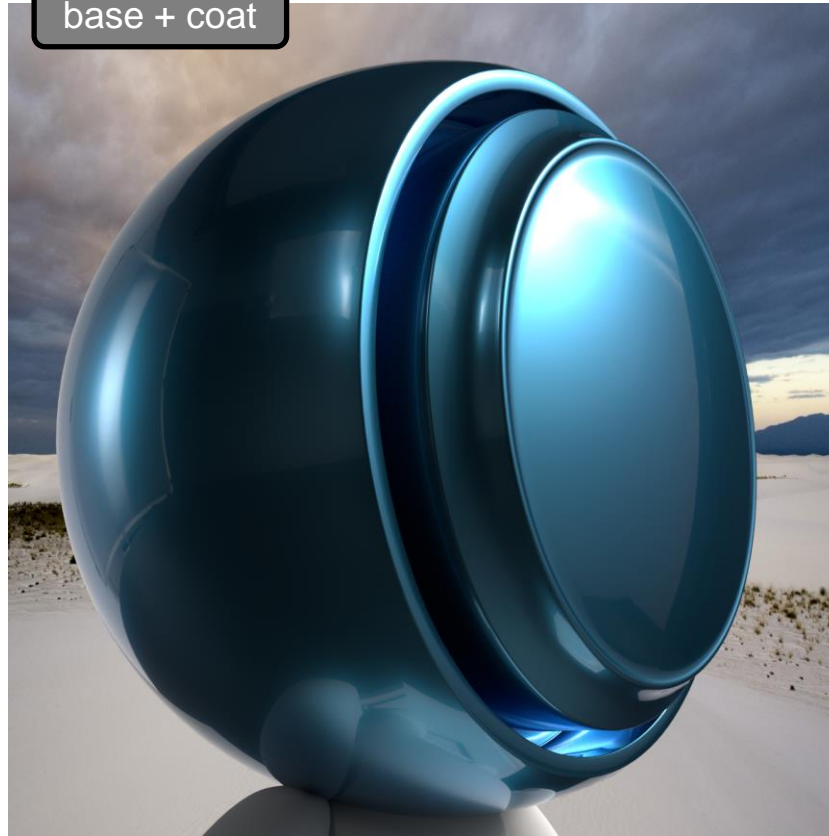
Sheen/Fuzz on top of coat

Different from both original models

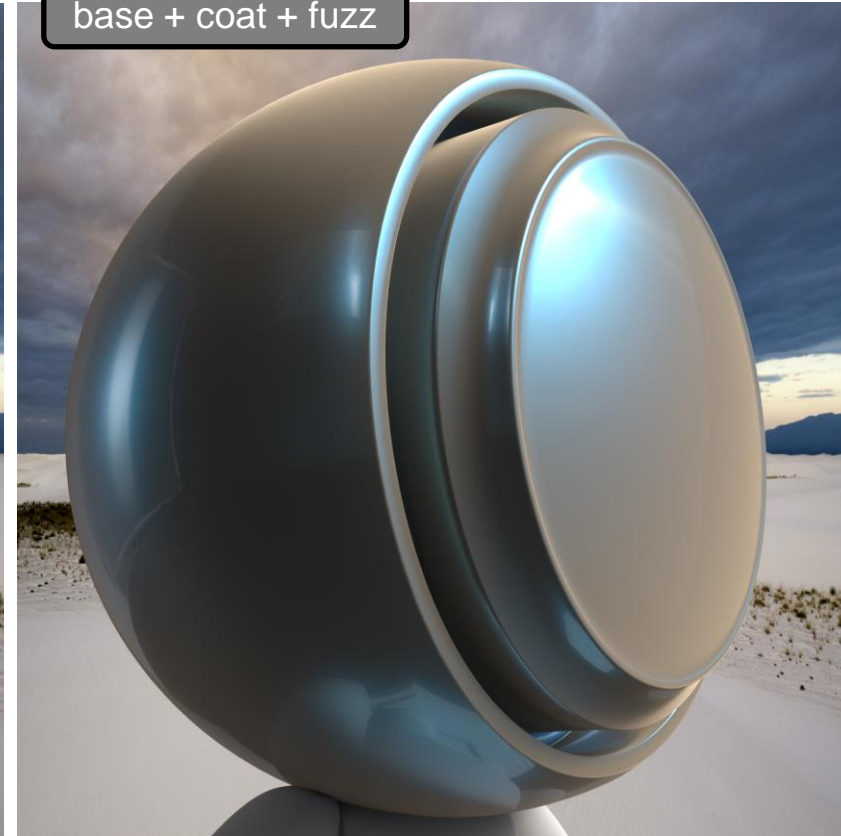
base



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



Chris Rydalch - SideFX

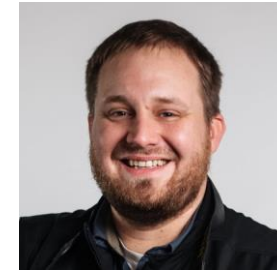
Autodesk Booth Wed 9th & Thu 10th at 11am

OpenPBR - Contributors



/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

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





/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

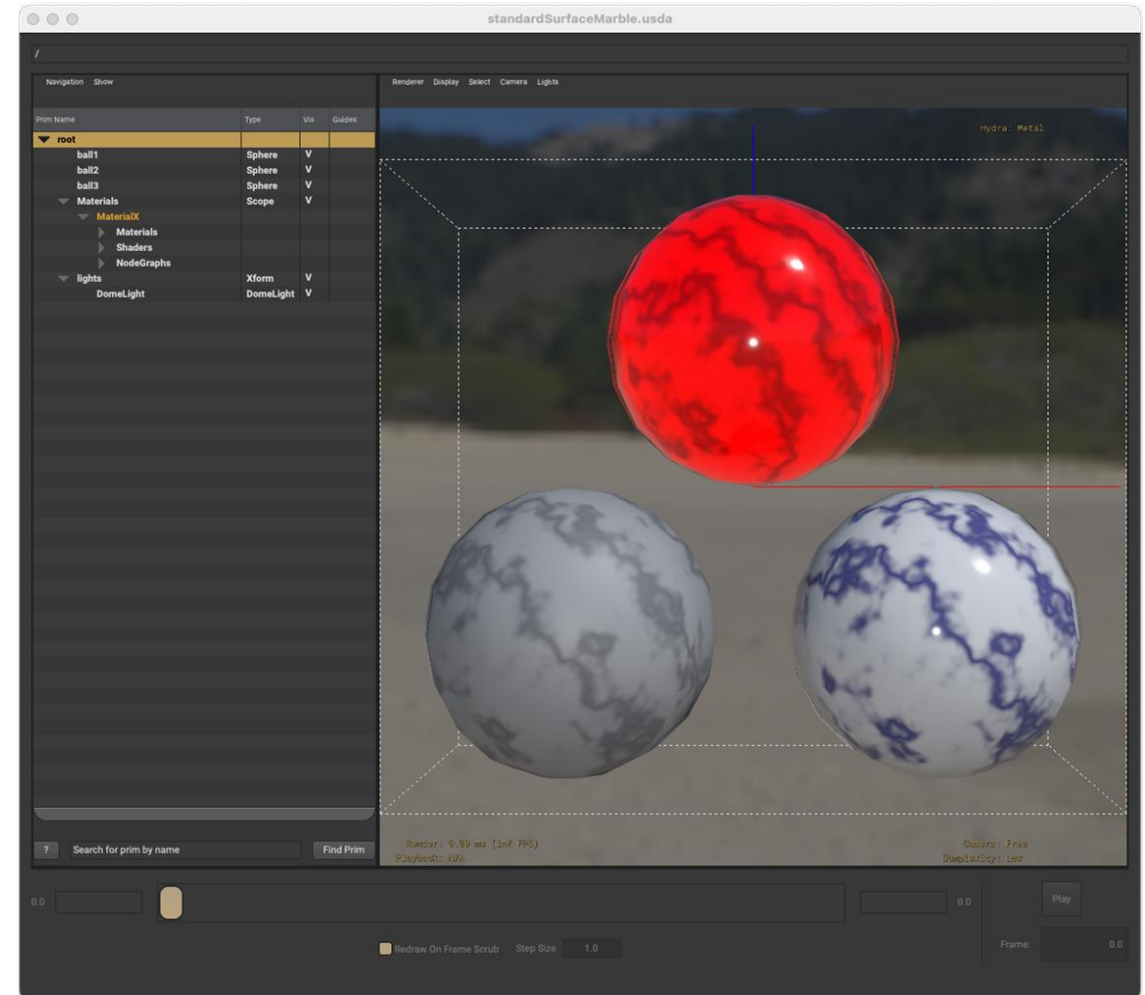
MaterialX in USD/Hydra

Karen Lucknavalai, Pixar

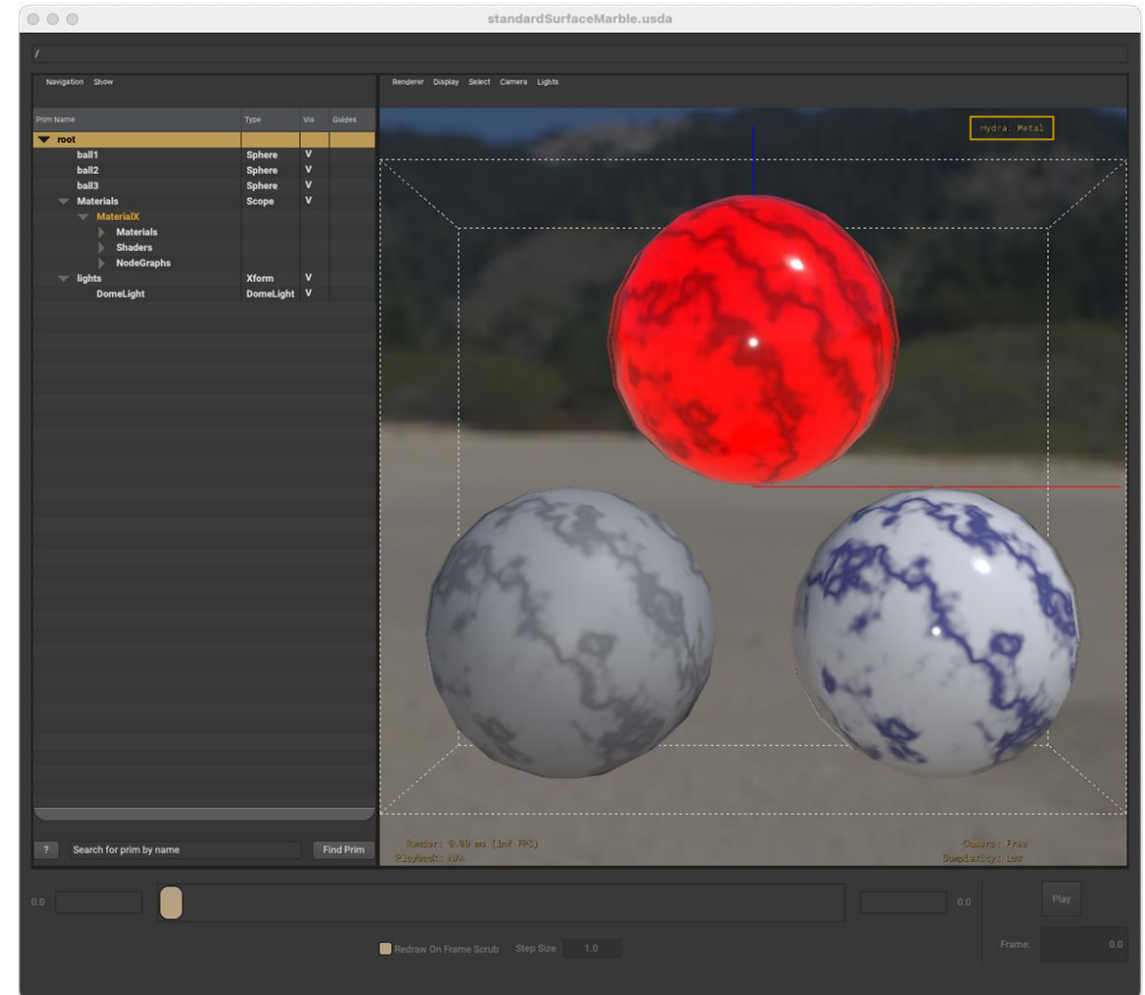
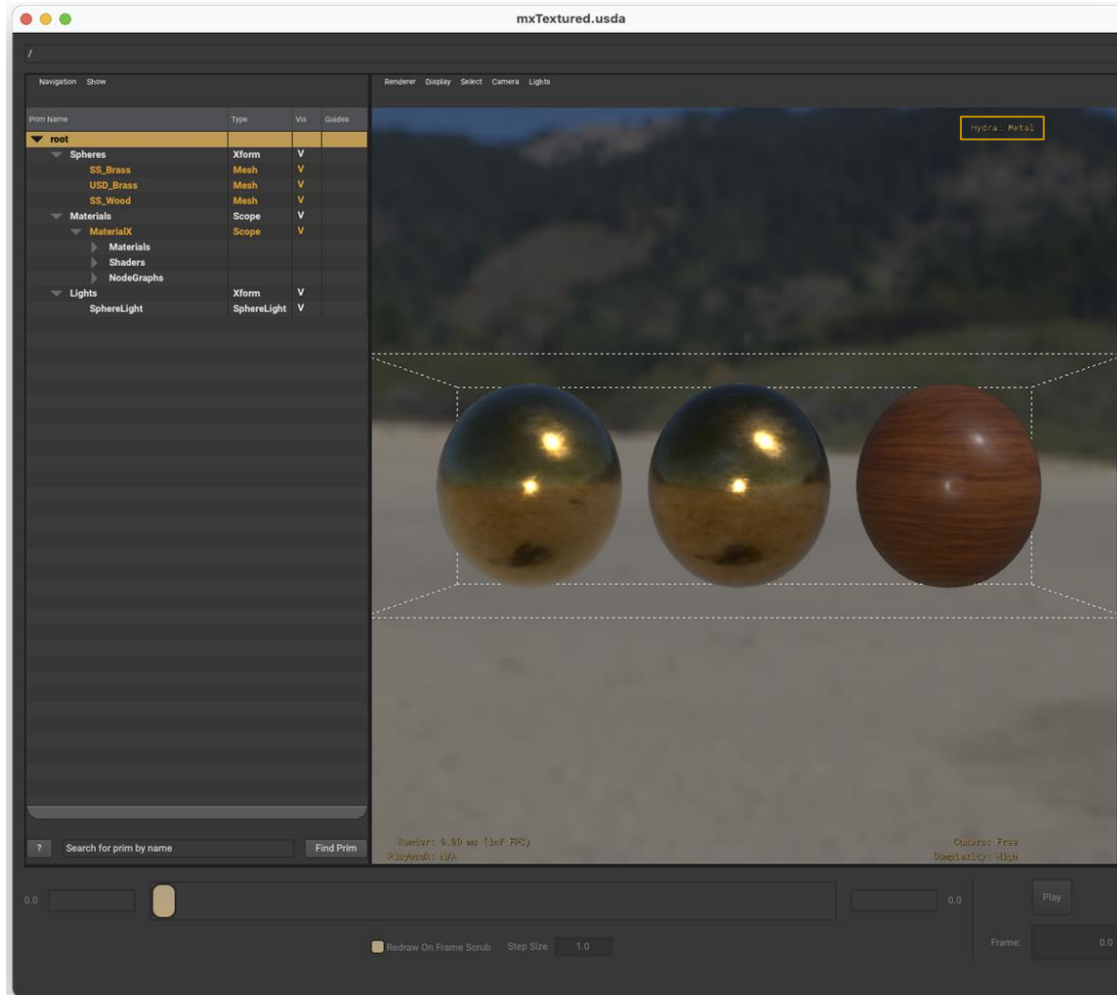
Aug 2, 2023

MaterialX in USD/Hydra - Updates

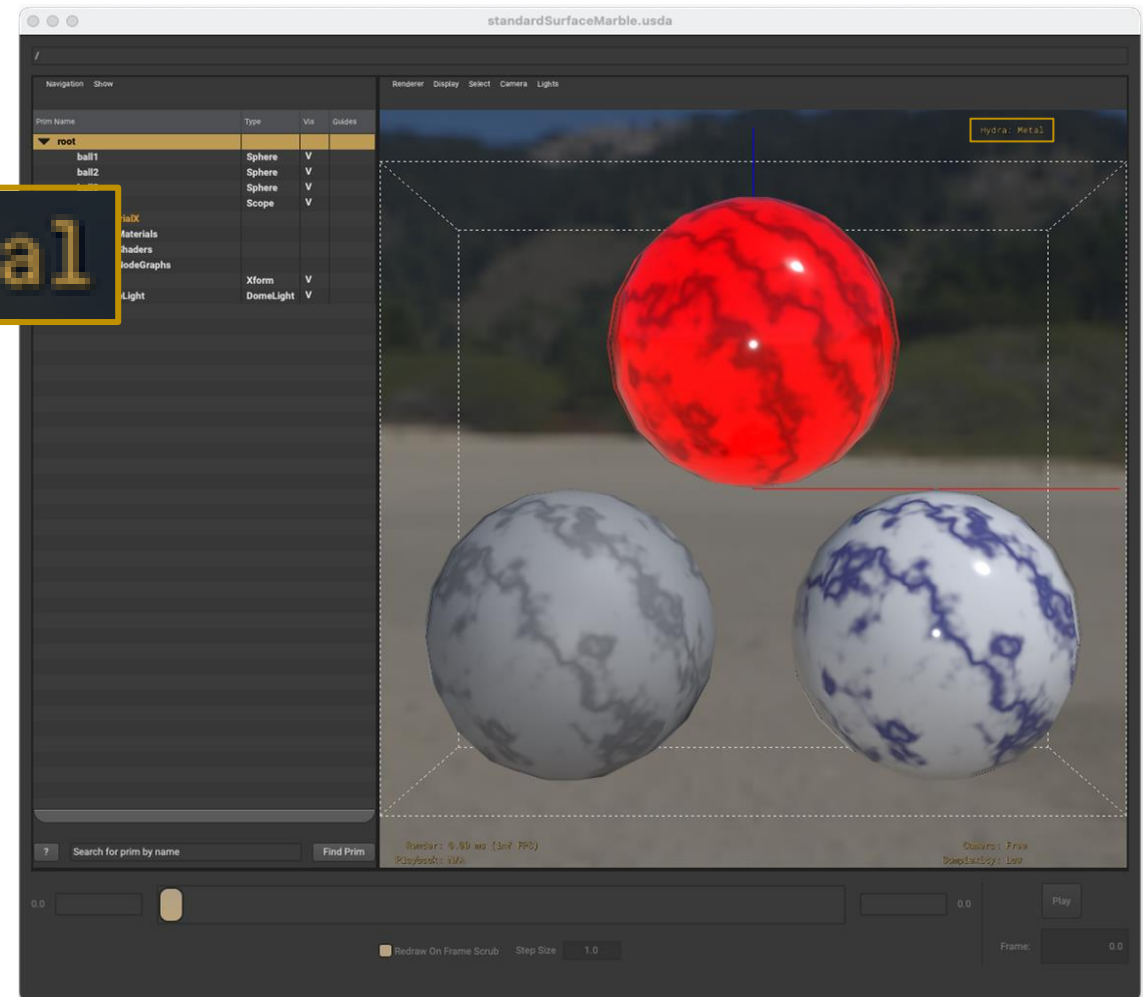
- MaterialX enabled by default in USD
- Support up to MaterialX version 1.38.7



MaterialX in USD/Hydra - Updates



MaterialX in USD/Hydra - Updates



MaterialX in USD/Hydra - Updates

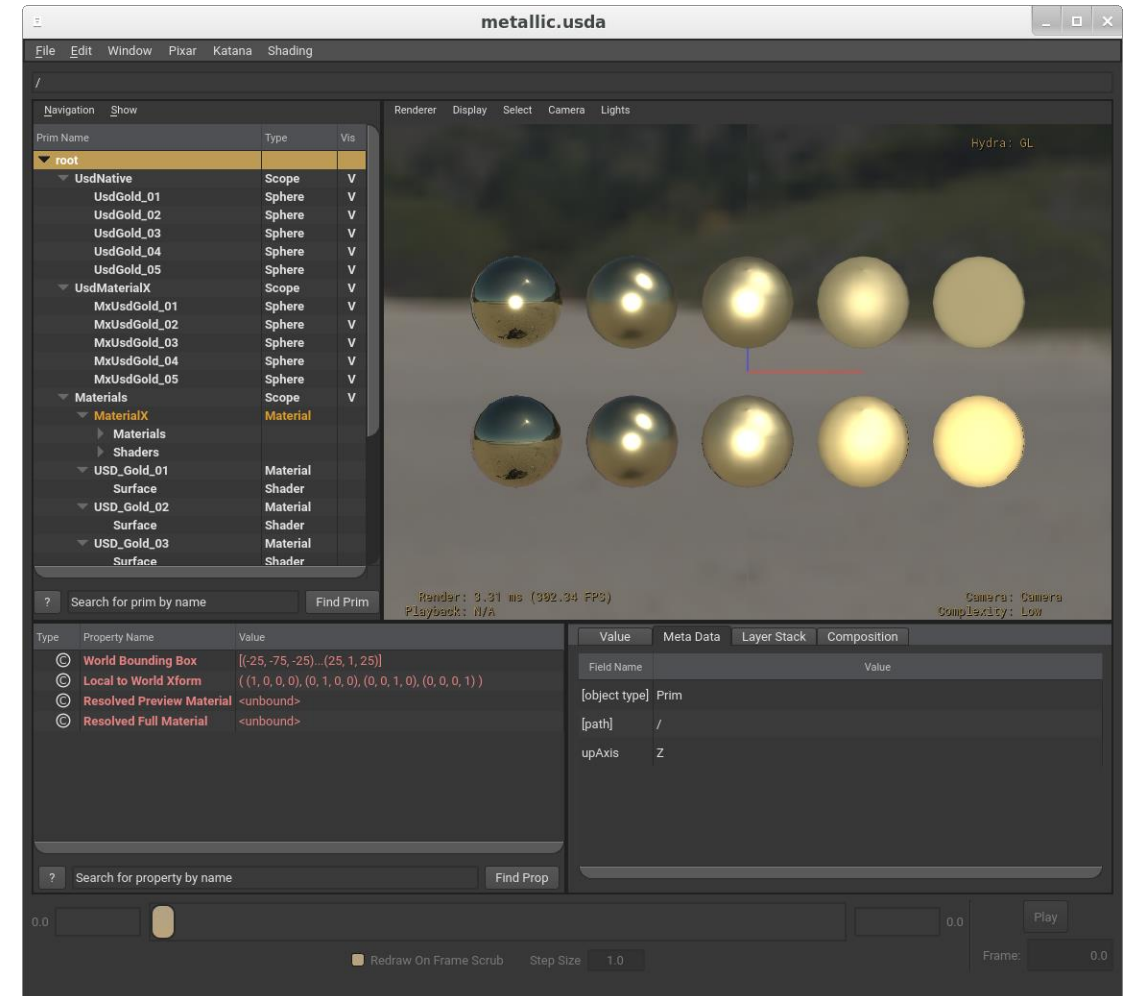


/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

- 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

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

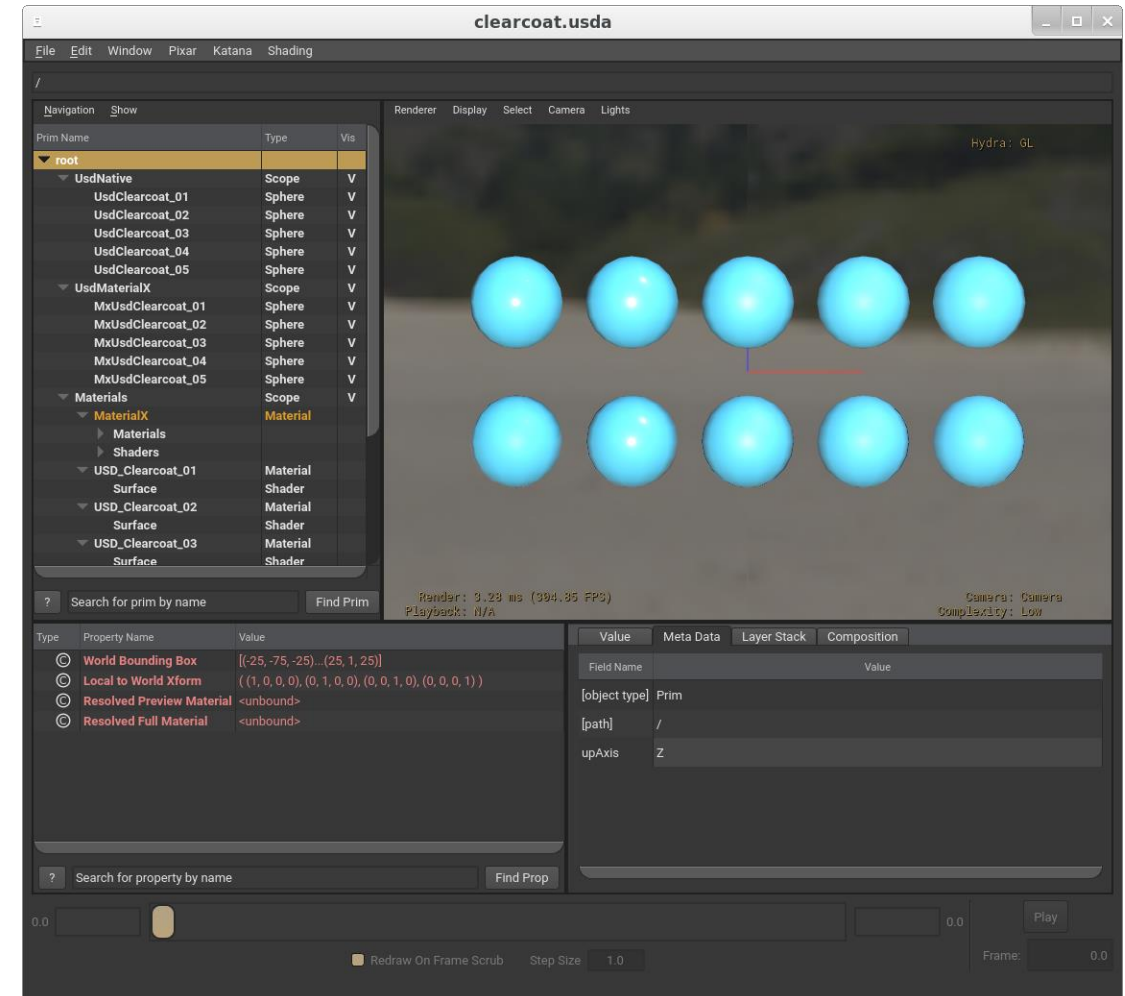


MaterialX in USD/Hydra - Updates



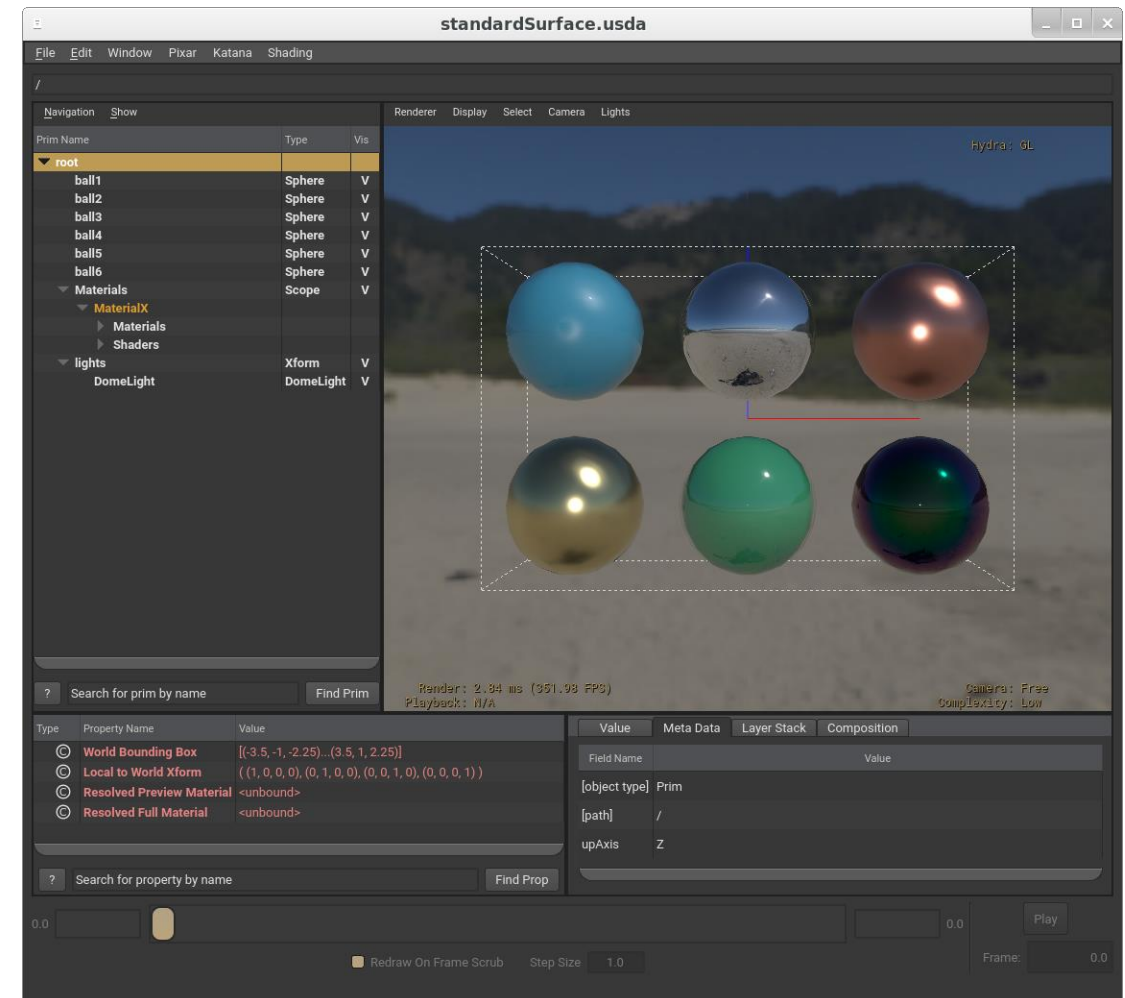
/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

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



MaterialX in USD/Hydra - Updates

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



MaterialX in USD/Hydra - HdPrman

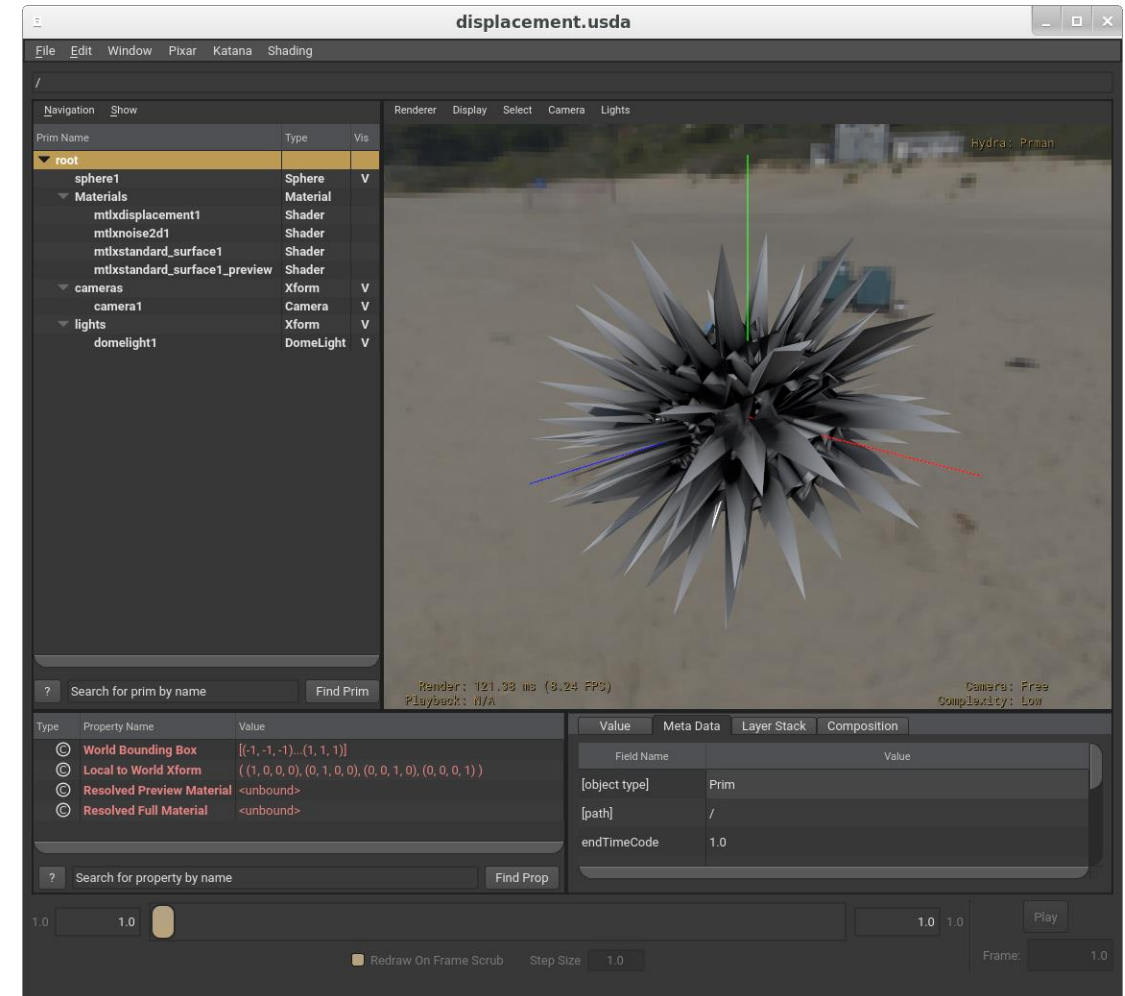


/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

Support for Displacement

Fixes for:

- Texture coordinate names
- Normal Maps



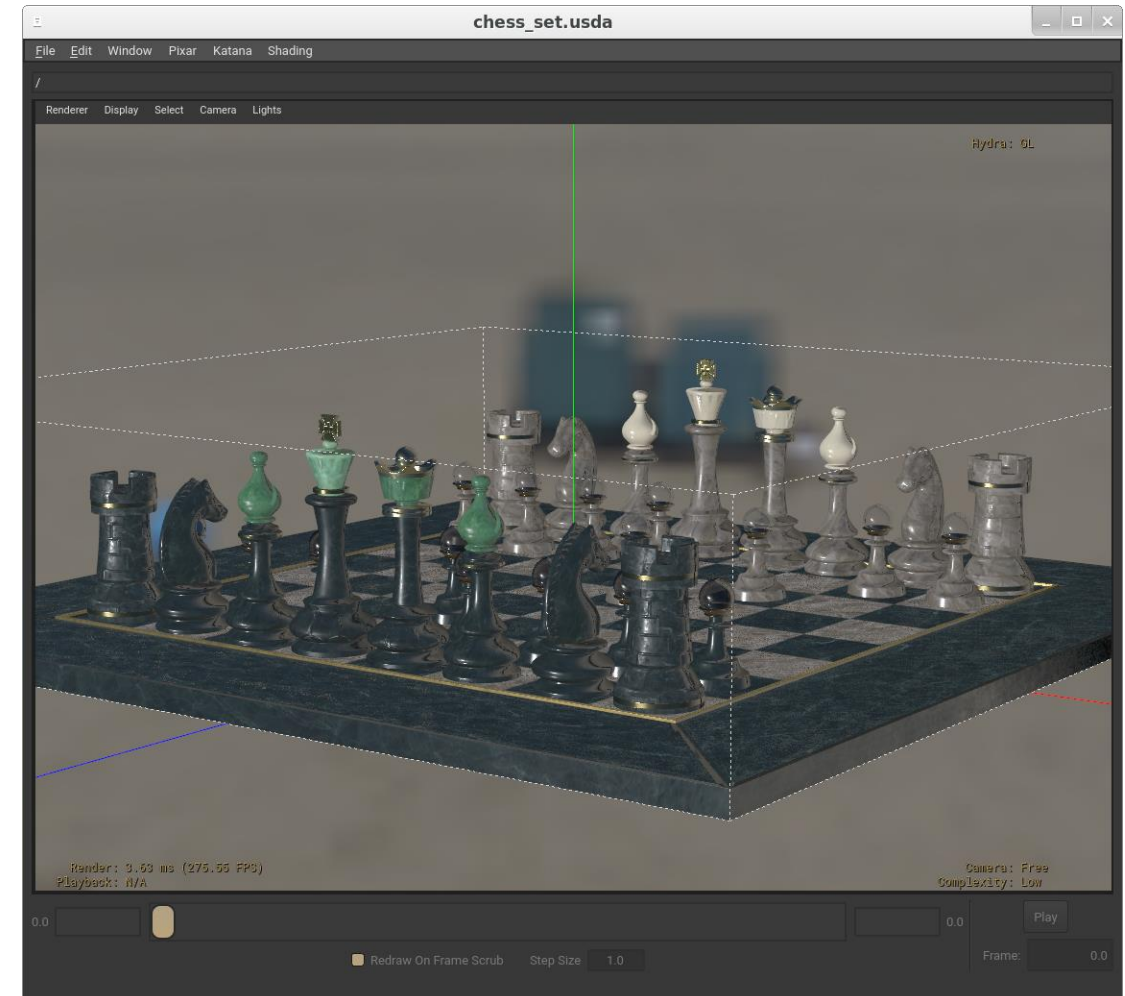
MaterialX in USD/Hydra - Future Work



/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

Future Work:

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

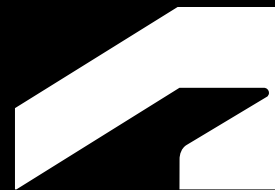




Thank you

**More information at the
USD, Hydra BOF
at SIGGRAPH:**

Tuesday Aug 8, 2pm - 4pm



LOOKDEVX

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 Biforst-USD



open
Source
days²³

/* 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

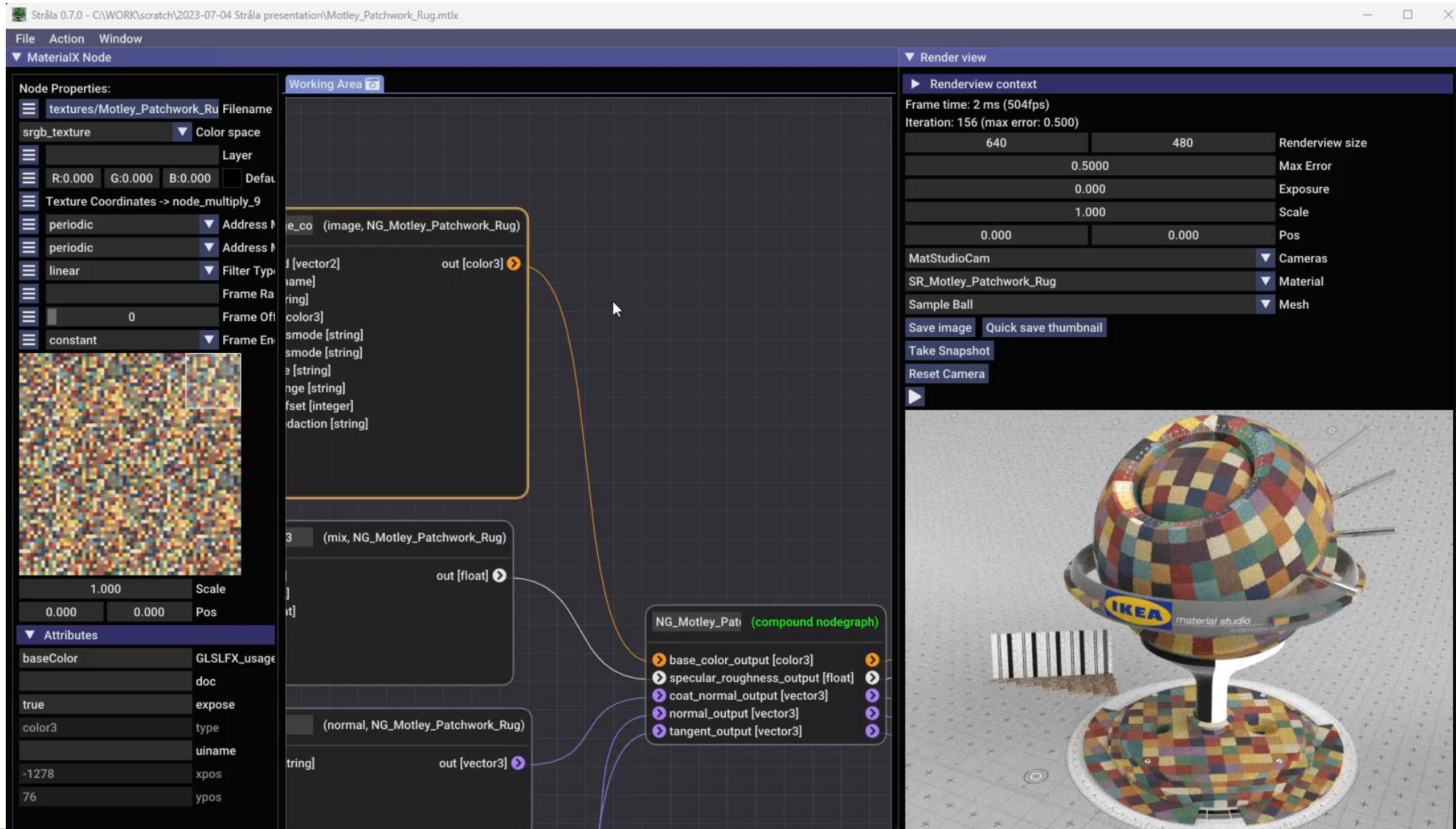
Technical Design



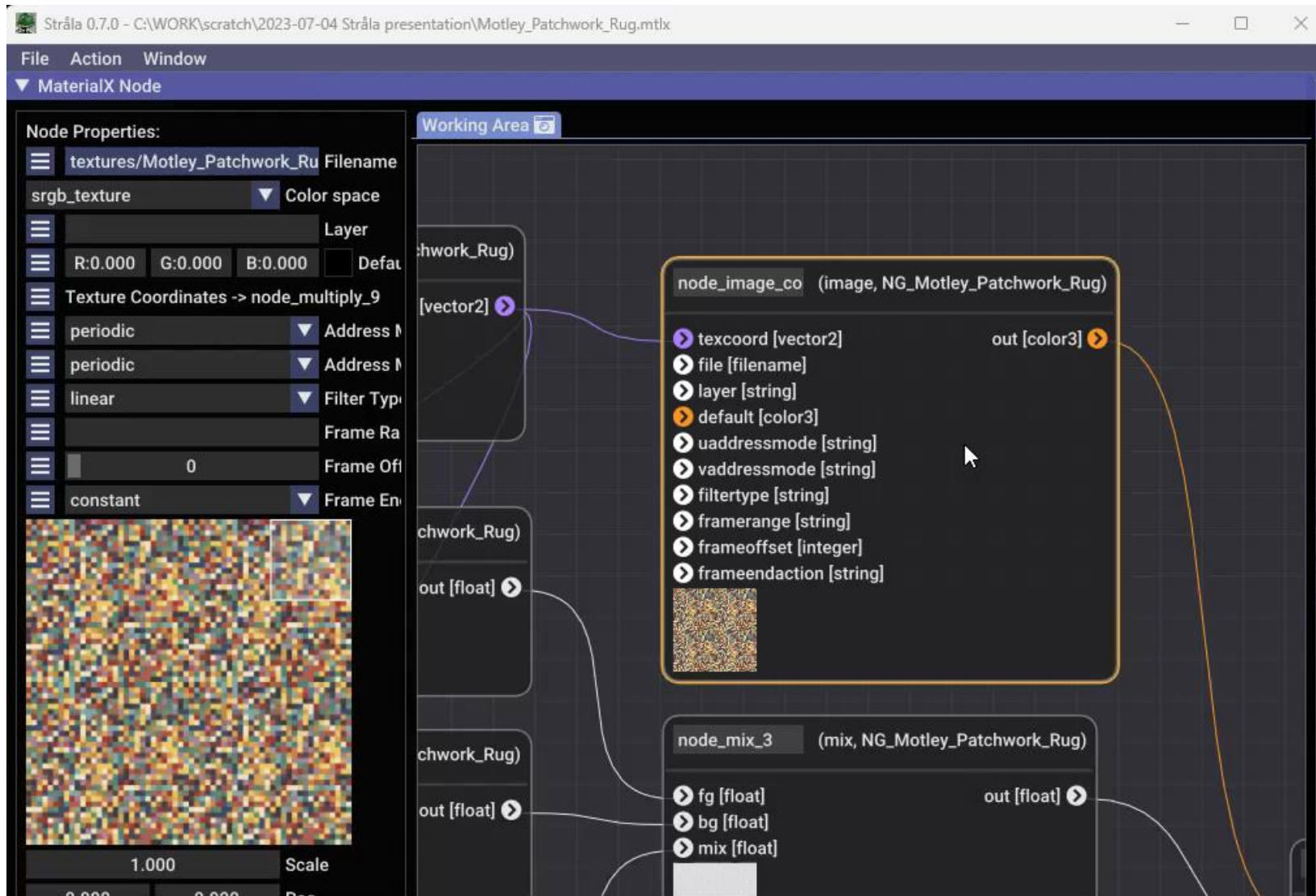
/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

- 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

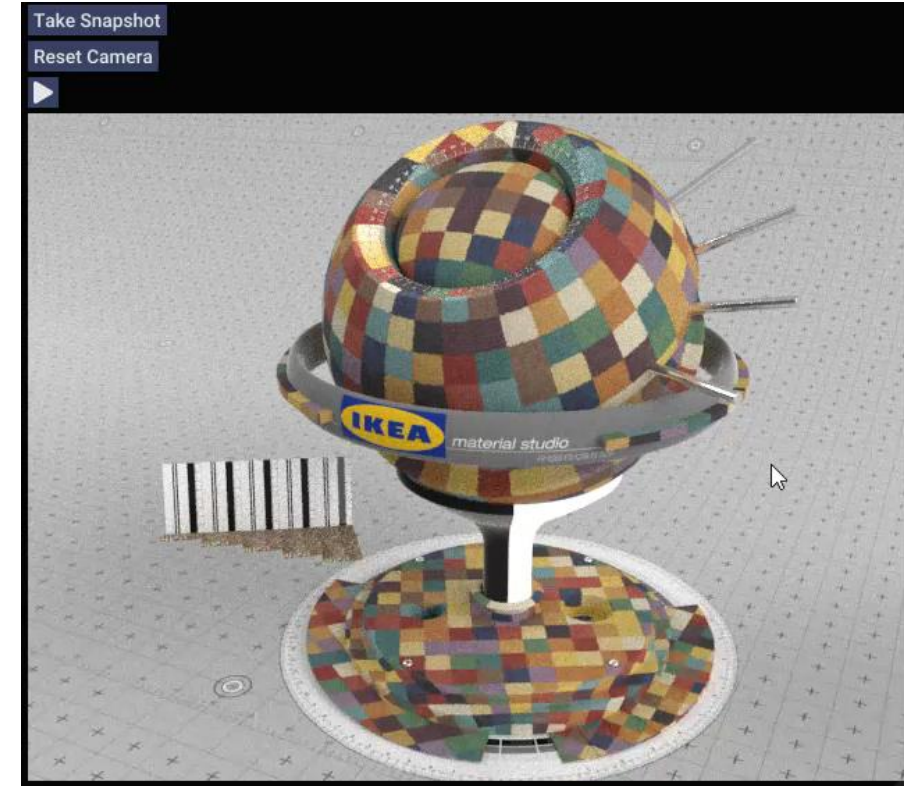


Swatches and texture handling

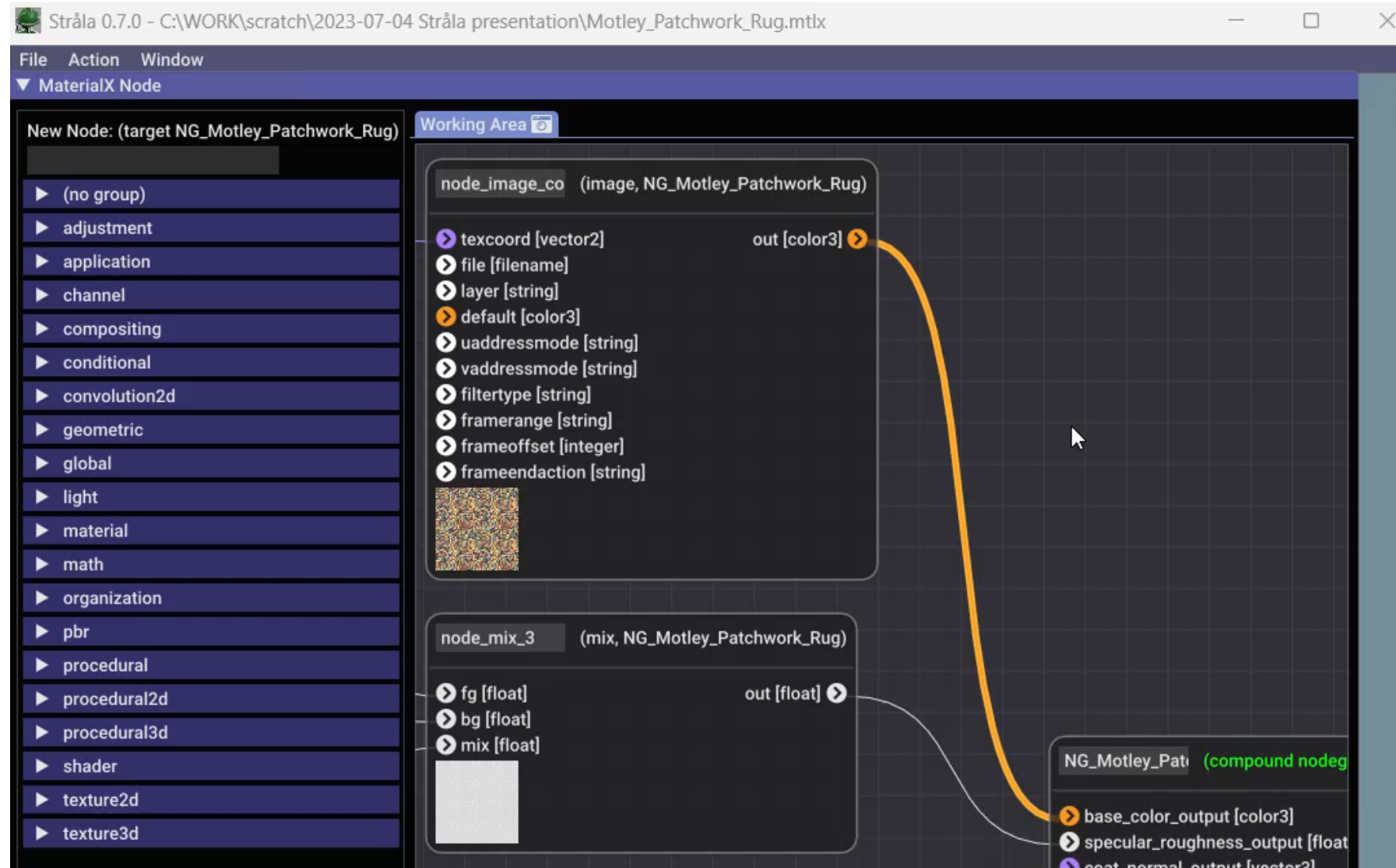


Material from AMD
material library

Preview render



MaterialX to gltf



open
Source
days²³

/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

NVIDIA Updates on MaterialX and MDL

August 2nd, 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>



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

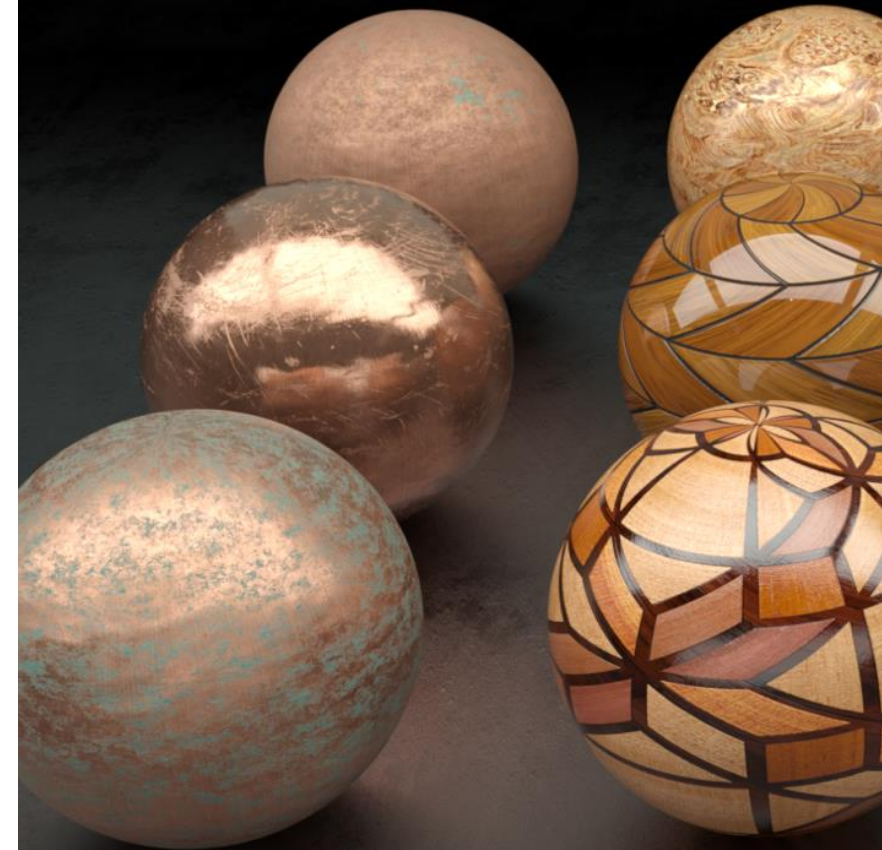
Translation to HLSL, GLSL, PTX, x86, ARM, LLVM-IR

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

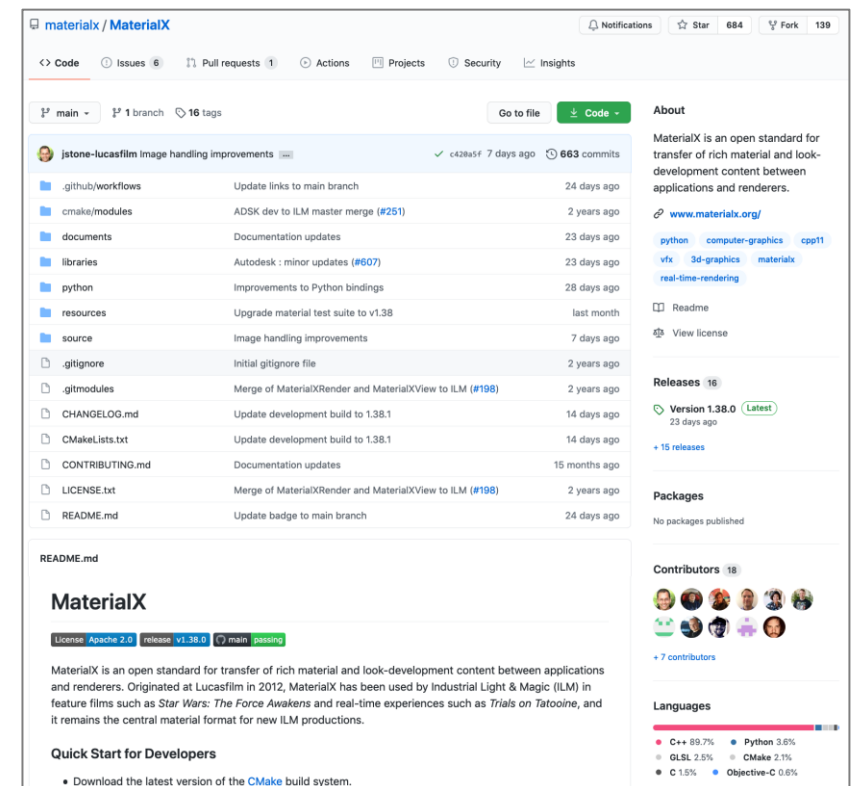
Official part since MaterialX 1.38.0

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

Joint development from



Used for example in



MaterialXGenMdl released with 1.38.7

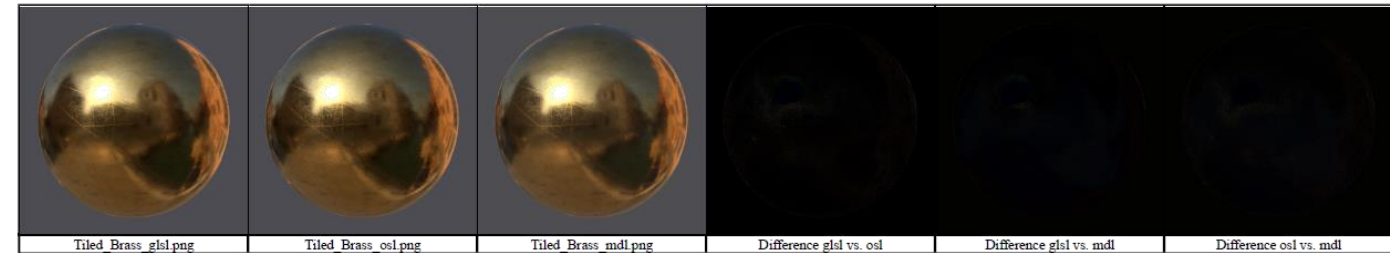
Better integration in the MaterialX Test Suite

MDL DXR Example Renderer matches the MaterialXTest scene

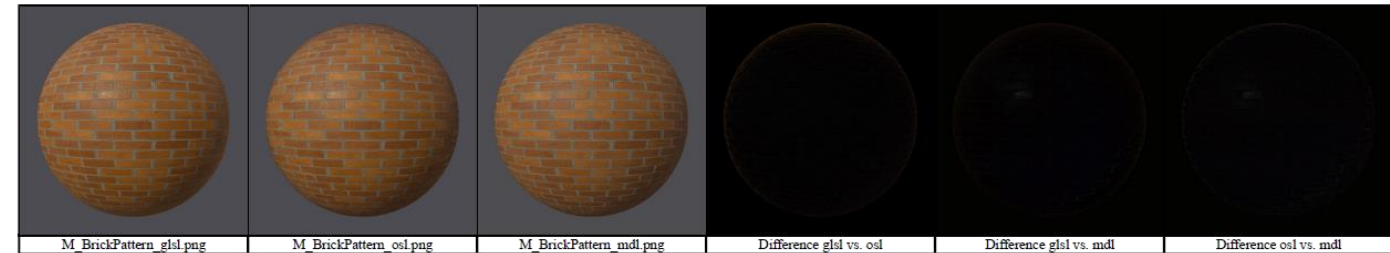
added 3-way comparison to the *tests_to_html.py* script

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

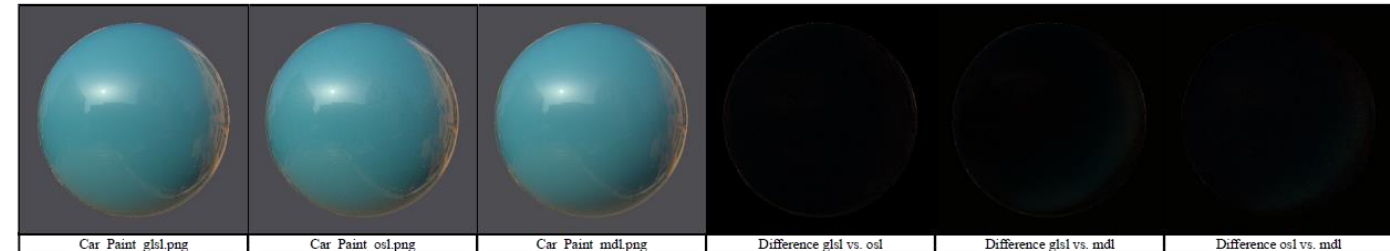
E:/github/MaterialX/build/bin/resources/Materials/Examples/StandardSurface/standard_surface_brass_tiled:



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

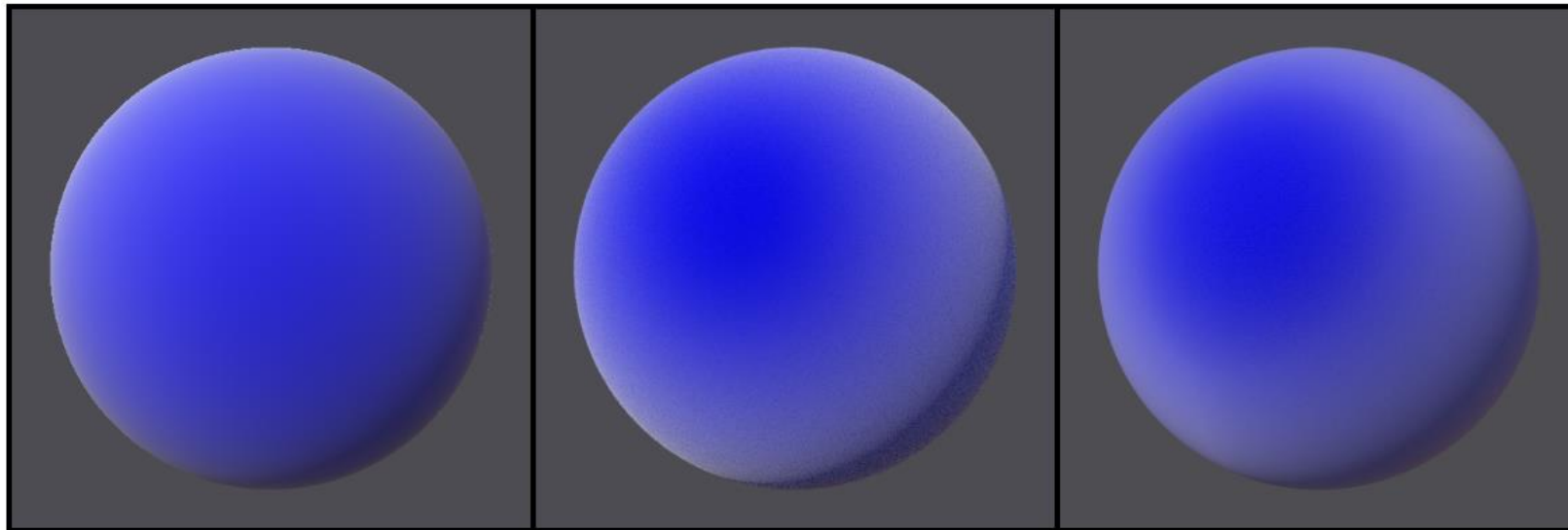
MDL
vs. GLSL

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



GLSL

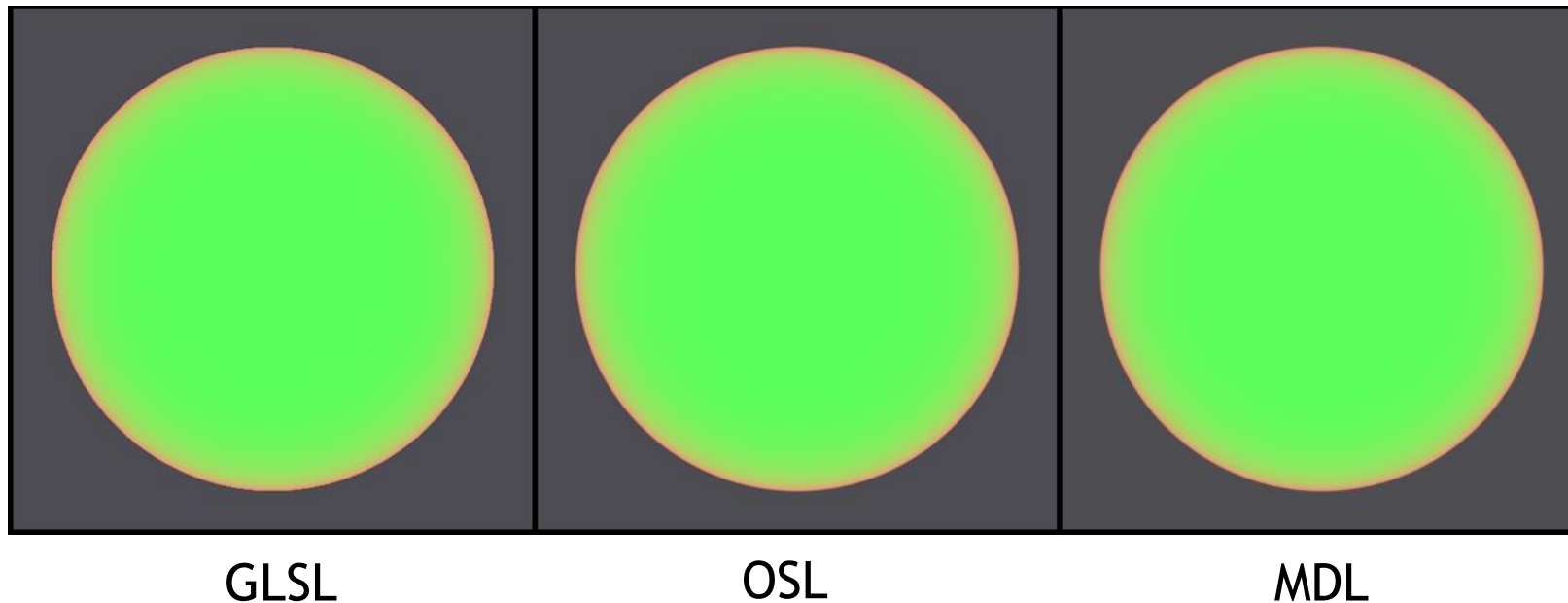
OSL
(work in progress)

MDL

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

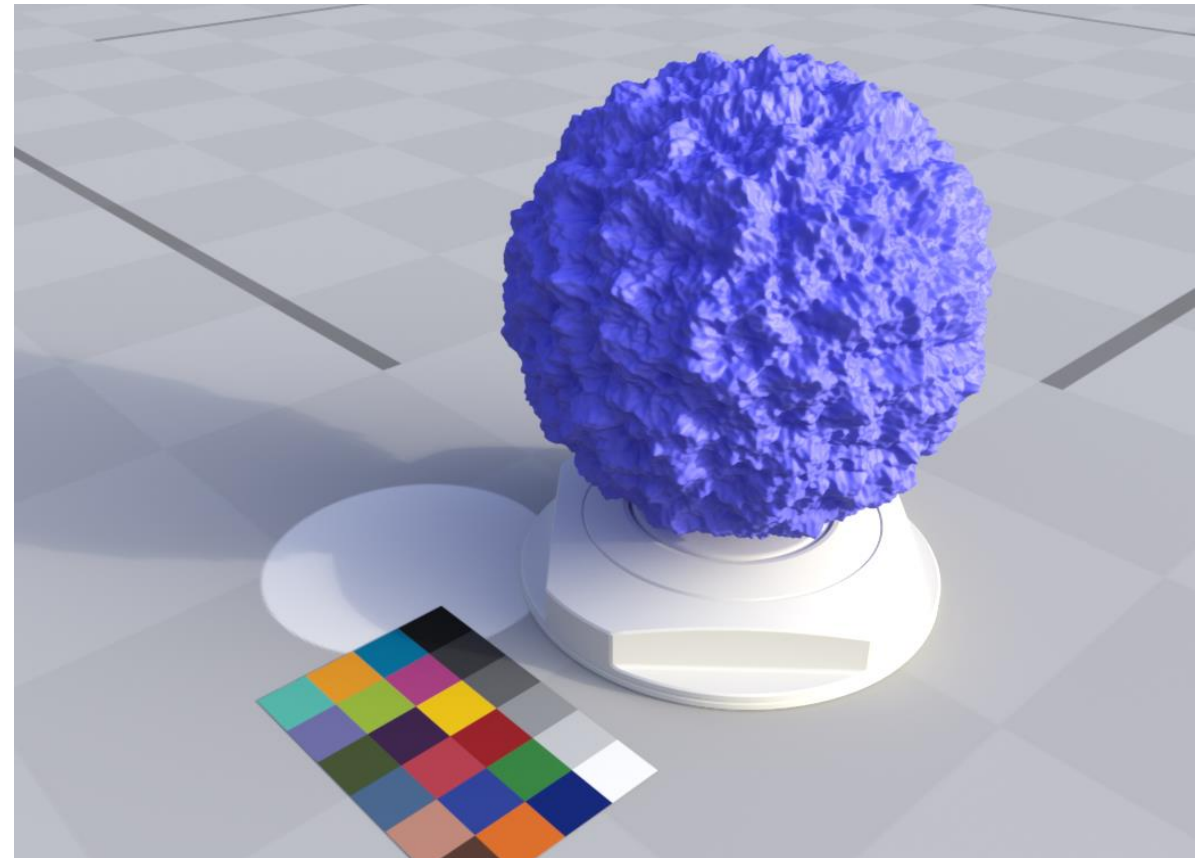
MaterialXGenMdl In Progress

Displacements

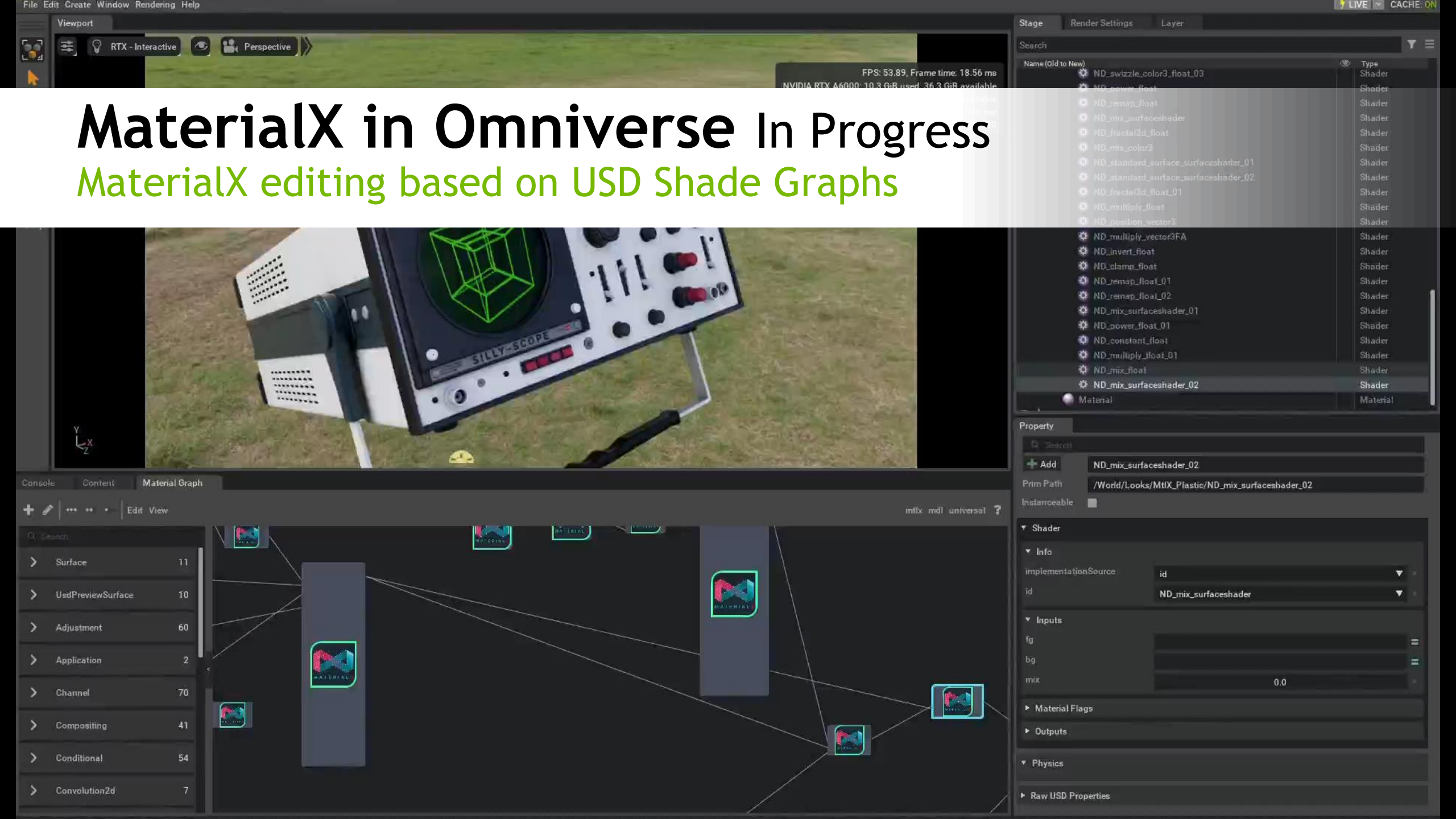
Already in the github main branch

Versioning

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



Rendered in NVIDIA Iray



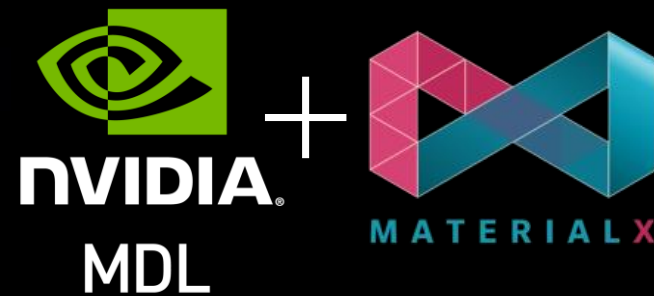
MaterialX in Omniverse In Progress

MaterialX editing based on USD Shade Graphs

open
Source
days²³



Thank you



#ASWF

Rendered in NVIDIA Omniverse

open
Source
days²³

/* 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



M A T E R I A L X

- 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





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
days²³

/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

QUILTIX 

Manuel Köster & Richard Frangenberg

2nd of August, 2023

#ASWF

/* ACADEMY SOFTWARE FOUNDATION

What is QUILTIX



/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

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

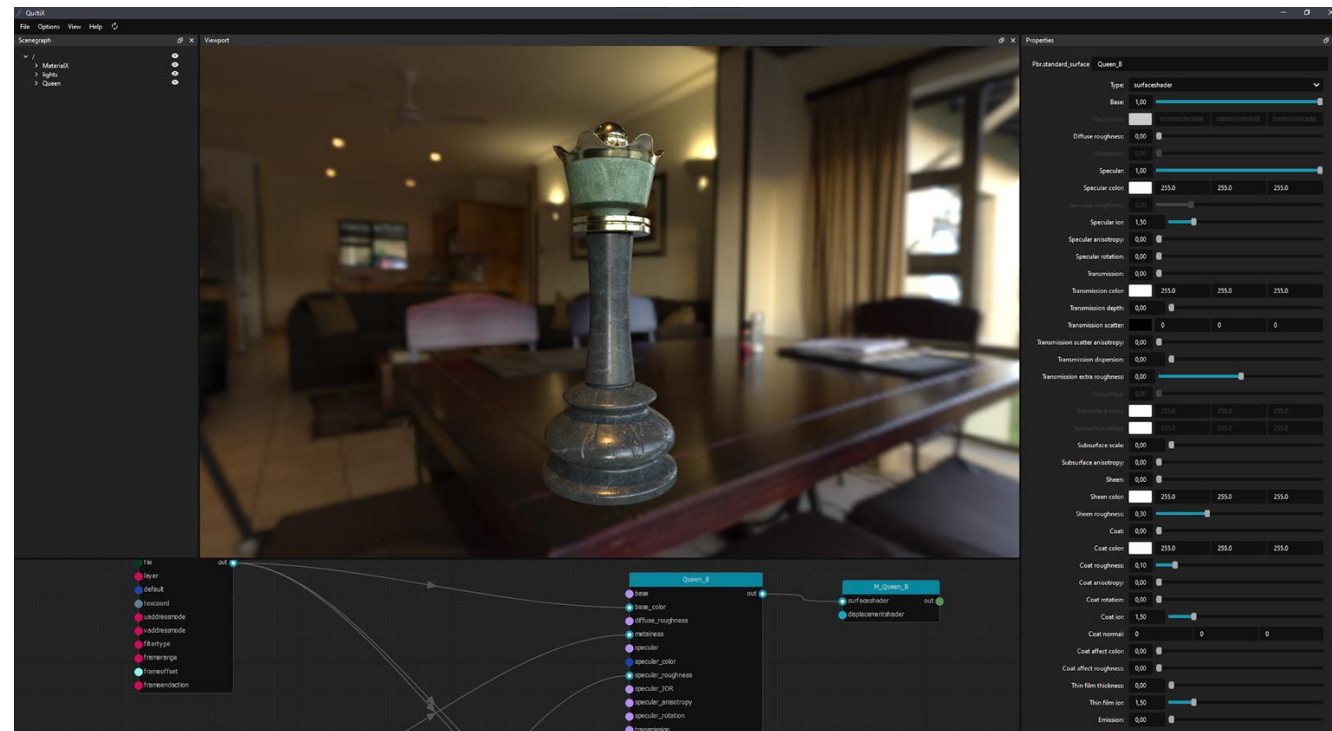
Why did we create



open
Source
days²³

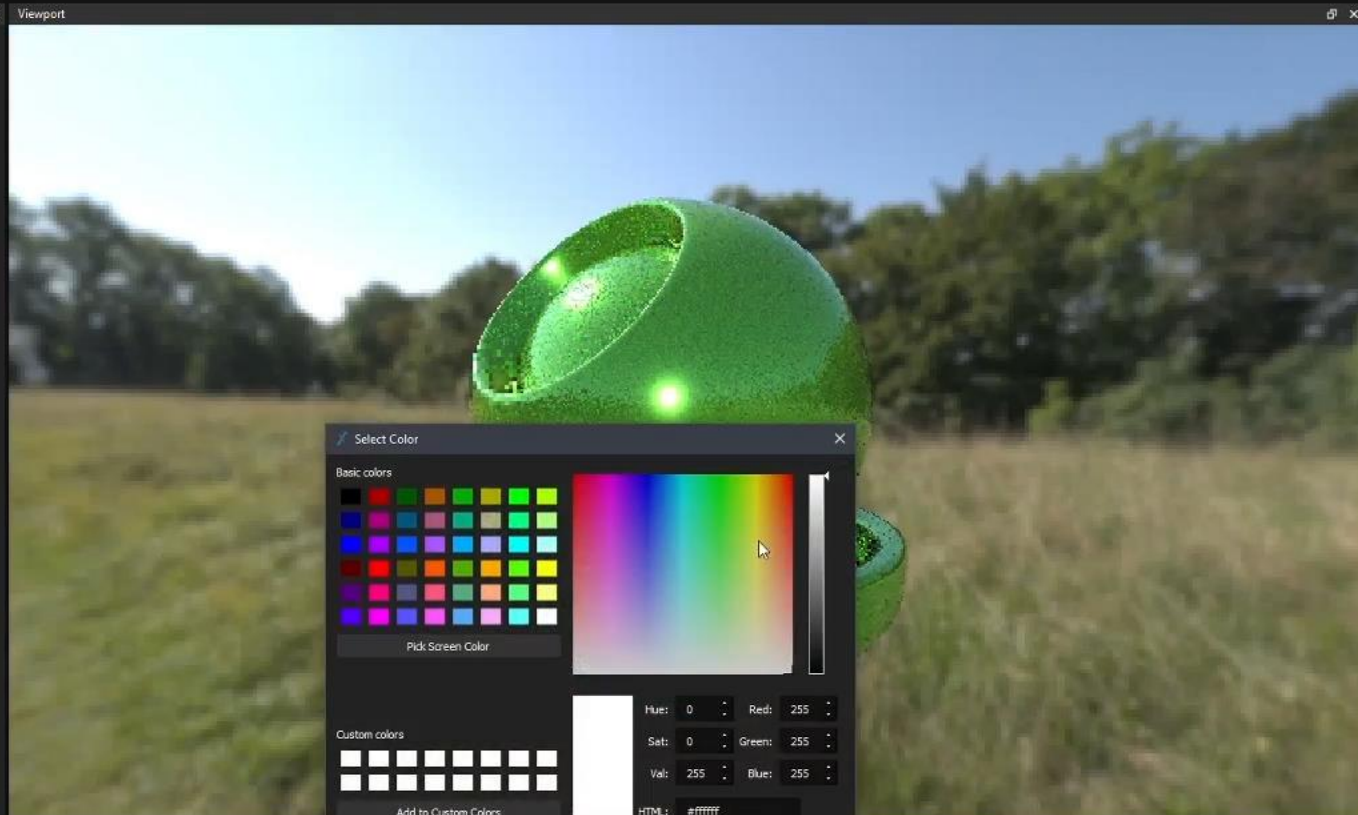
/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

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



Scenegraph

- /
- > lights
- > MaterialX
- > Calibration_Mesh
- > Preview_Mesh



Select Color

Basic colors

Pick Screen Color

Custom colors

Add to Custom Colors

Hue: 0 Red: 255
Sat: 0 Green: 255
Val: 255 Blue: 255
HTML: #ffffff

OK Cancel



Properties

Pbr.Standard_surface Standard_surface

Type: surfaceshader

Base: 0,62

Base color: 1.0 1.0 1.0

Diffuse roughness: 0,00

Metalness: 0,50

Specular: 1,00

Specular color: 1.0 1.0 1.0

Specular roughness: 0,20

Specular ion: 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: 1,00

Sheen color: 1.0 0.039216 0.039216

Sheen roughness: 0,30

Coat: 1,00

Coat color: 3.227451 1.0 0.090196

Coat roughness: 0,10

Coat anisotropy: 0,00

Coat rotation: 0,00

Coat ion: 1,50

Coat normal: 0 0 0

Coat affect color: 0,00

Coat affect roughness: 0,00

Thin film thickness: 0,00

Thin film ion: 1,50

Scenegraph

- > /
- > lights
- > Calibration_Mesh
- > Preview_Mesh
- > MaterialX



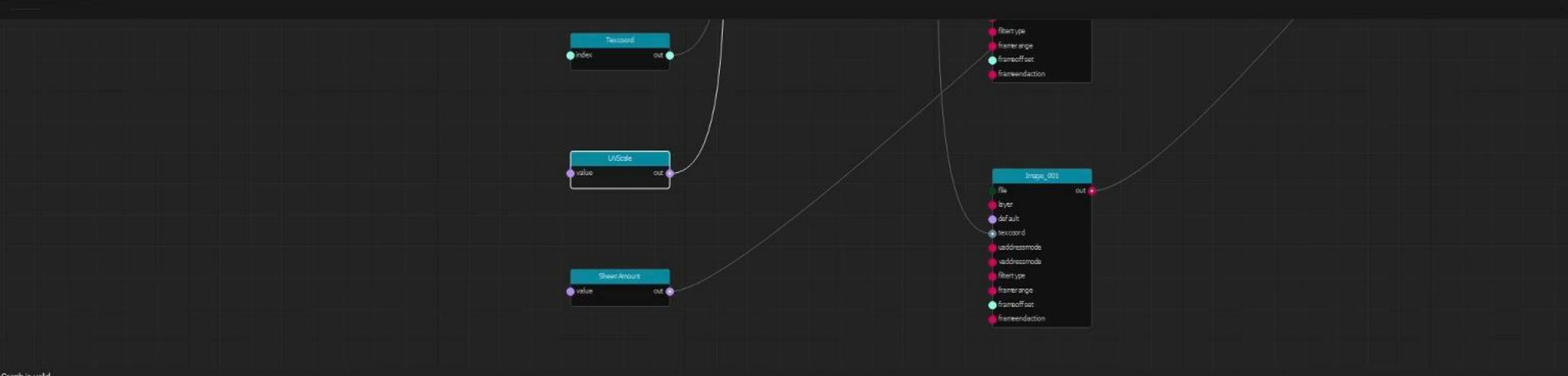
Properties

Procedural.Constant UVScale

Type: float

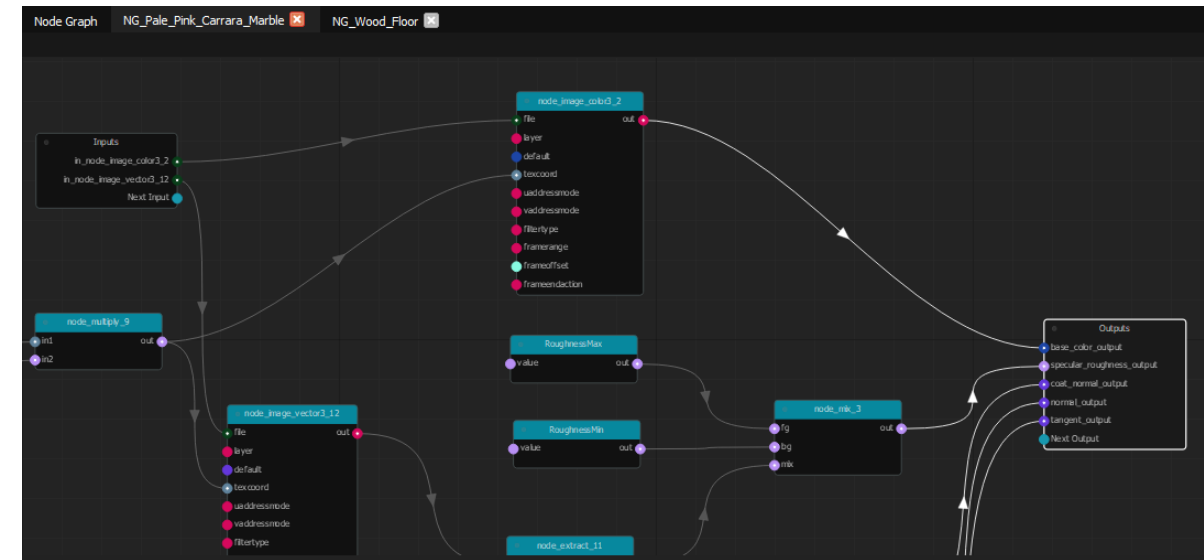
Value: 7,90

Node Graph NG

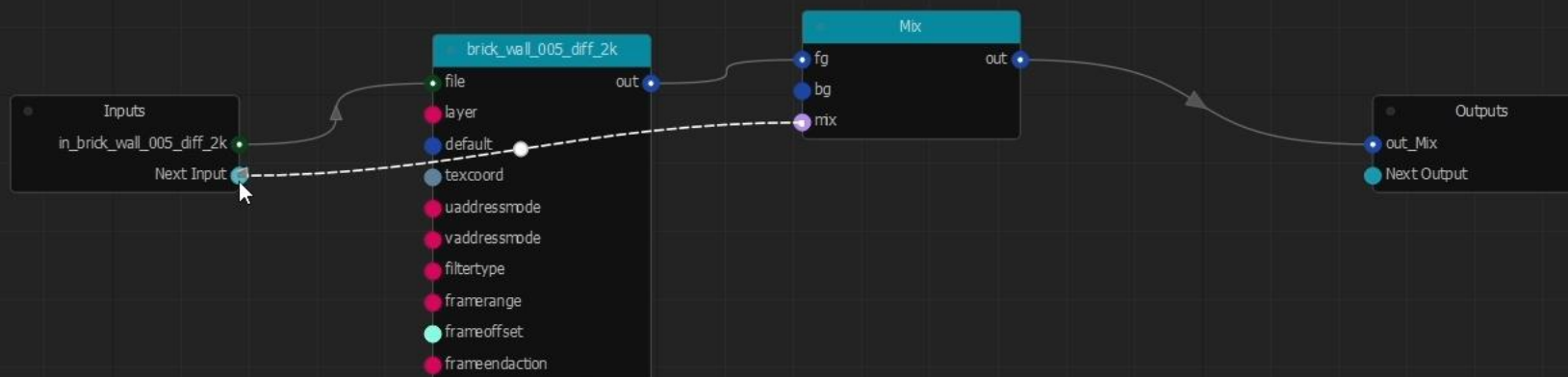


Nodegraphs

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



- > /
- > lights
- > Calibration_Mesh
- > Preview_Mesh
- > MaterialX



- > /
- > 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



Save Node Definition

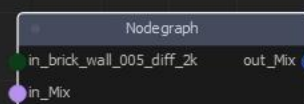
Category: Custom

Name: myStone

Save Path: C:\Users\yicha\custom_mtlx_defs\nodegraph.mtlx

Save Cancel

Node Graph Nodegraph



- base_color
- diffuse_roughness
- metalness
- specular
- specular_color
- specular_roughness
- specular_IOR
- specular_anisotropy
- specular_rotation
- transmission
- transmission_color
- transmission_depth
- transmission_scatter
- transmission_scatter_anisotropy
- transmission_dispersion
- transmission_extra_roughness

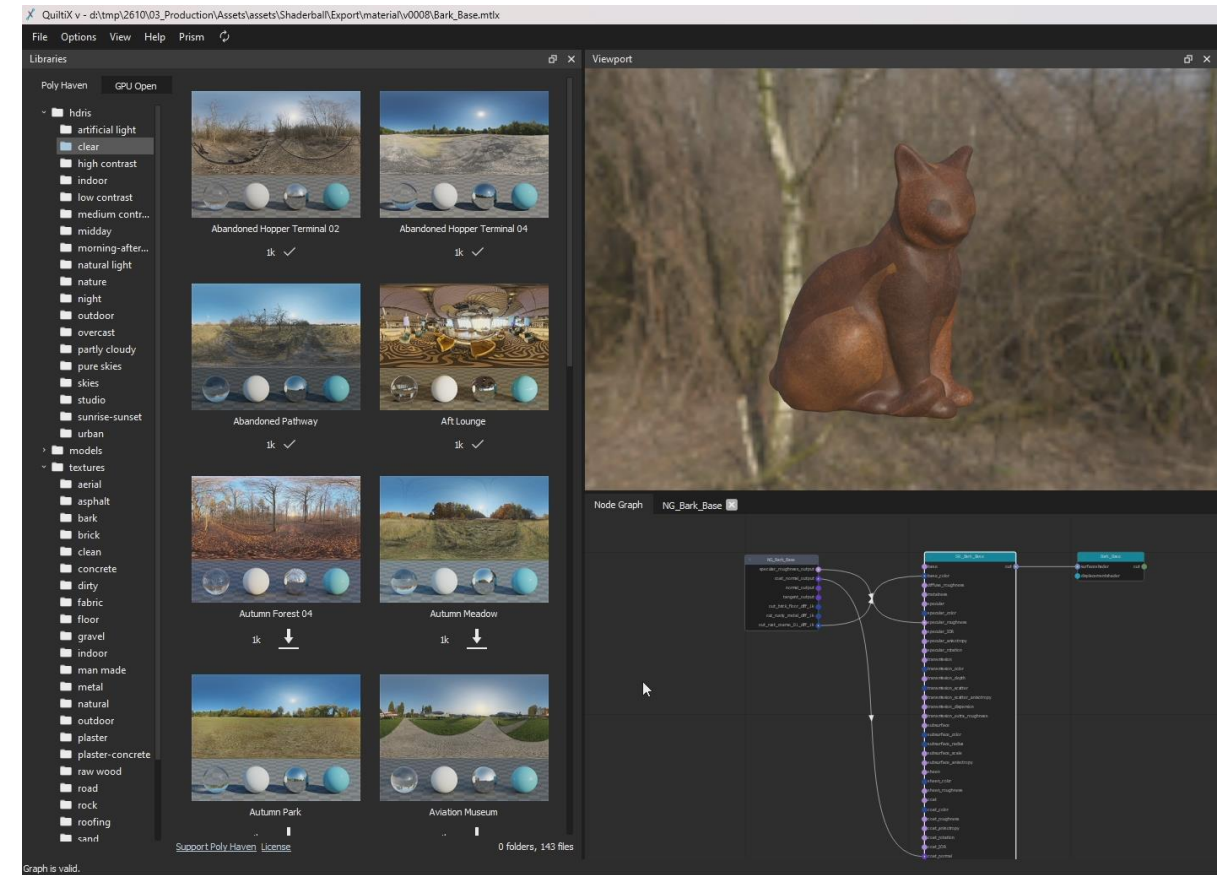
_Long_Variative

Pipeline Integration



/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

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



- hdris
 - 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 Grass Rock

1k JPG



Aerial Mud 1

1k JPG



Aerial Rocks 02

1k JPG



Aerial Beach 03

1k JPG



Aerial Ground Rock

1k JPG



Aerial Rocks 01

1k JPG

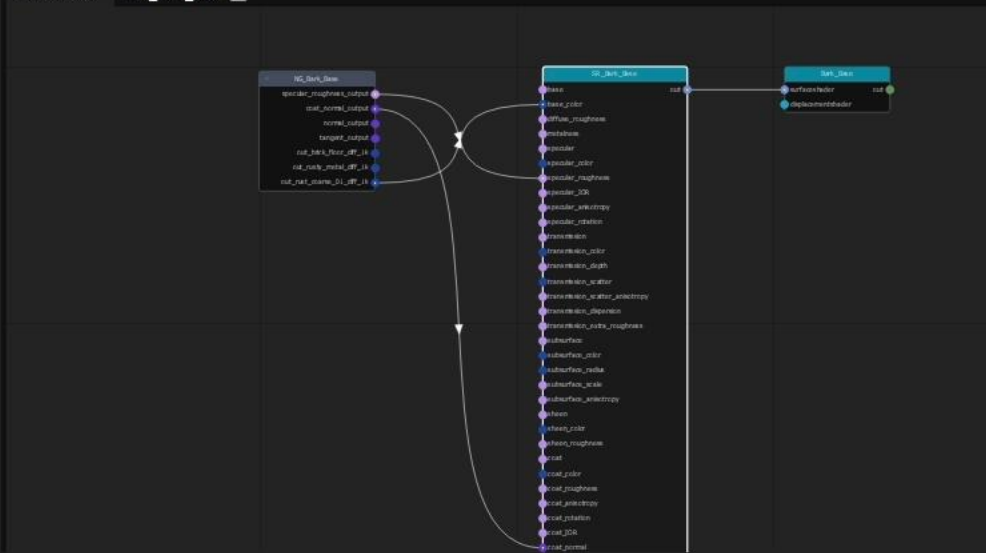


Aerial Rocks 04

1k JPG

[Support Poly Haven](#) [License](#)

0 folders, 18 files



Pbr.Standard_surface SR_Bark_Base

Type: **surfaceshader**

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

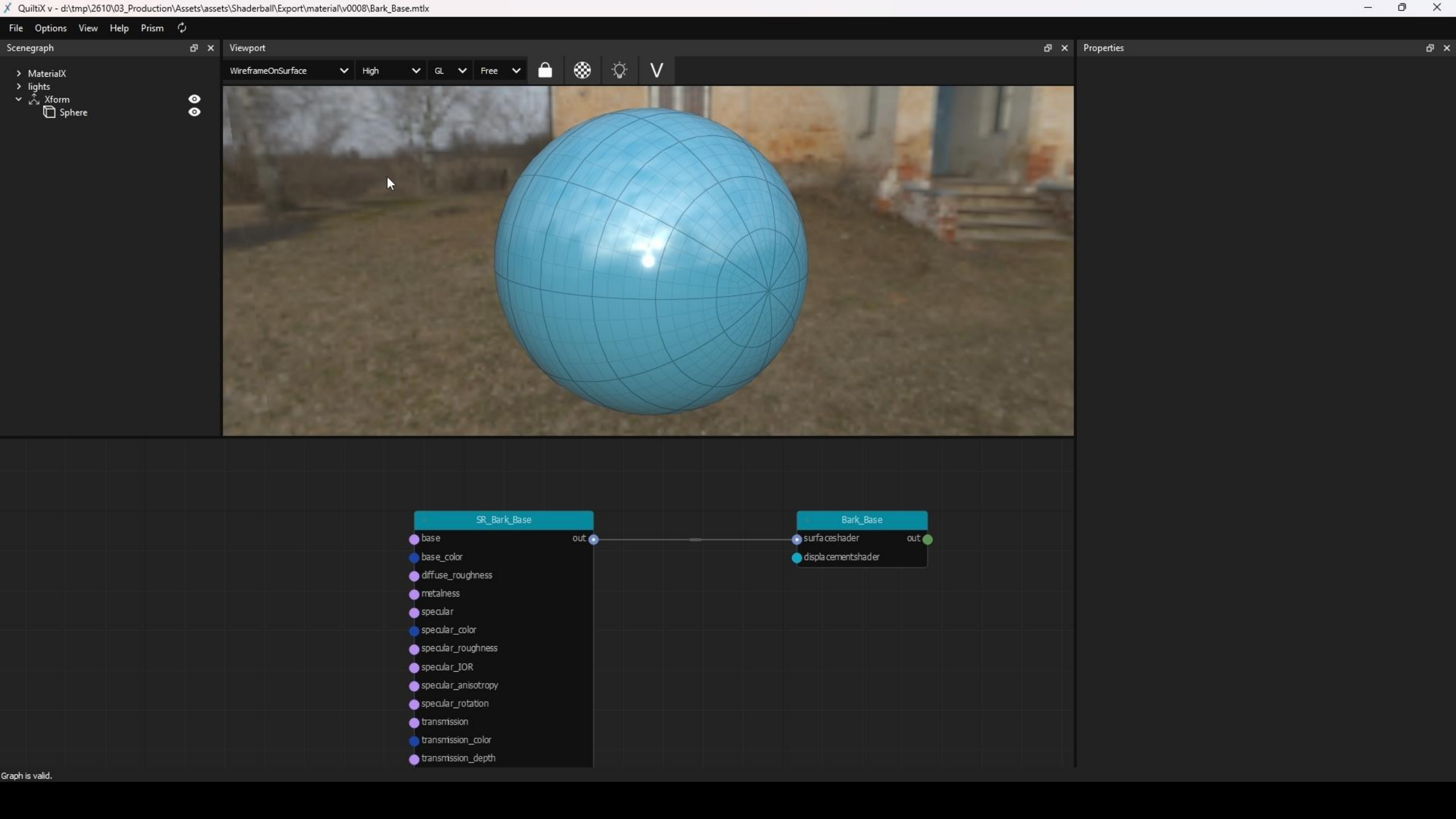
Subsurface scale: 1,00

Subsurface anisotropy: 0,00

Sheen: 0,00

Sheen color: 1.0 1.0 1.0

Sheen roughness: 0,30



Pipeline Friendly

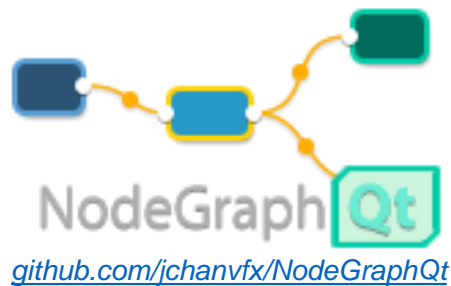


/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

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



Tech stack



Compatibility

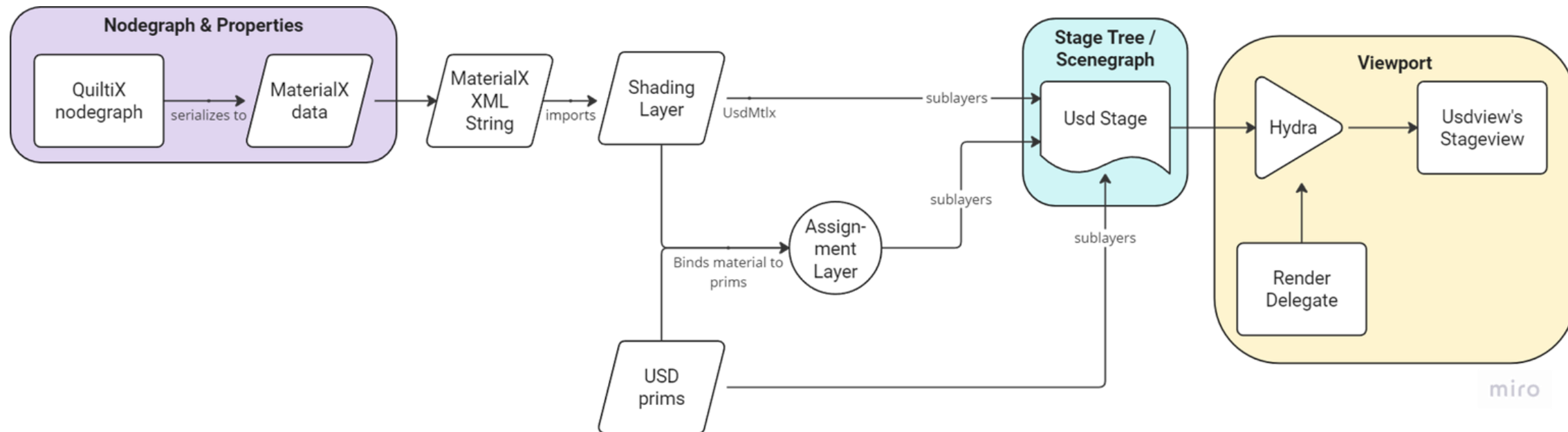


/* ACADEMY
SOFTWARE
FOUNDATION
#ASWF

- 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)

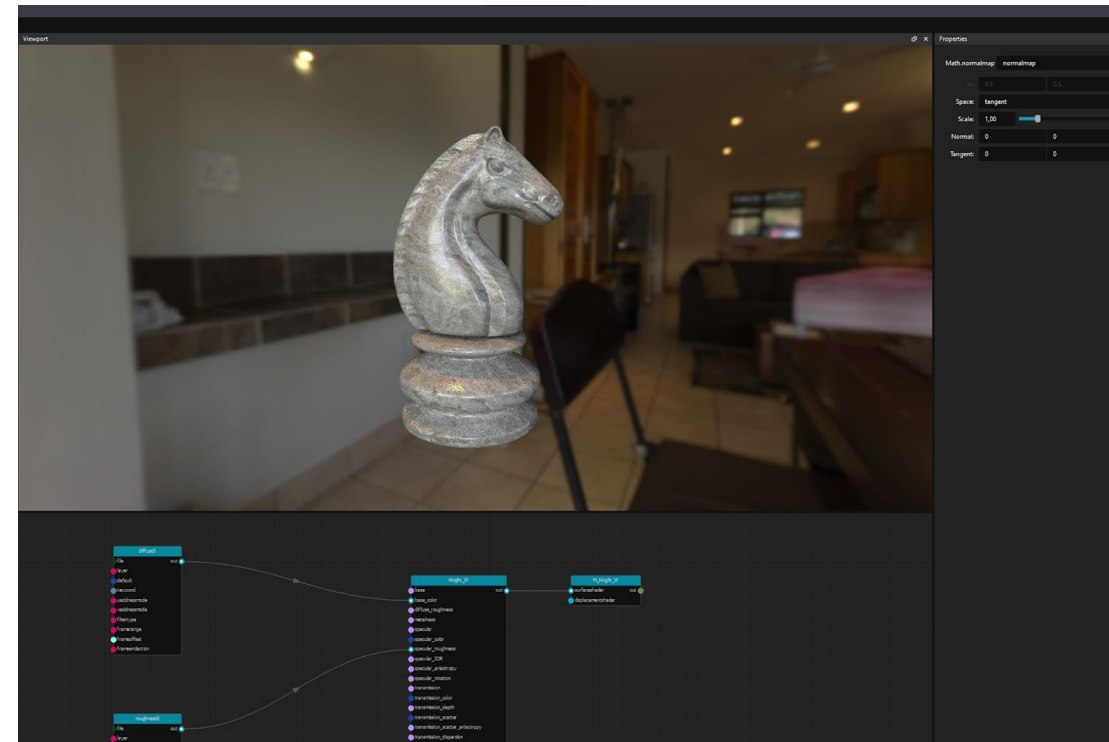


How does QUILTI^X work




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





Node Graph NG_Copper_Old 



- lights
- Calibration_Mesh
- Preview_Mesh
- MaterialX
 - Materials
 - TH_Fabric_Leather
 - Shaders
 - ND_standard_surface_surfaceshader
 - NodeGraphs
 - NG_TH_Fabric_Leather



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

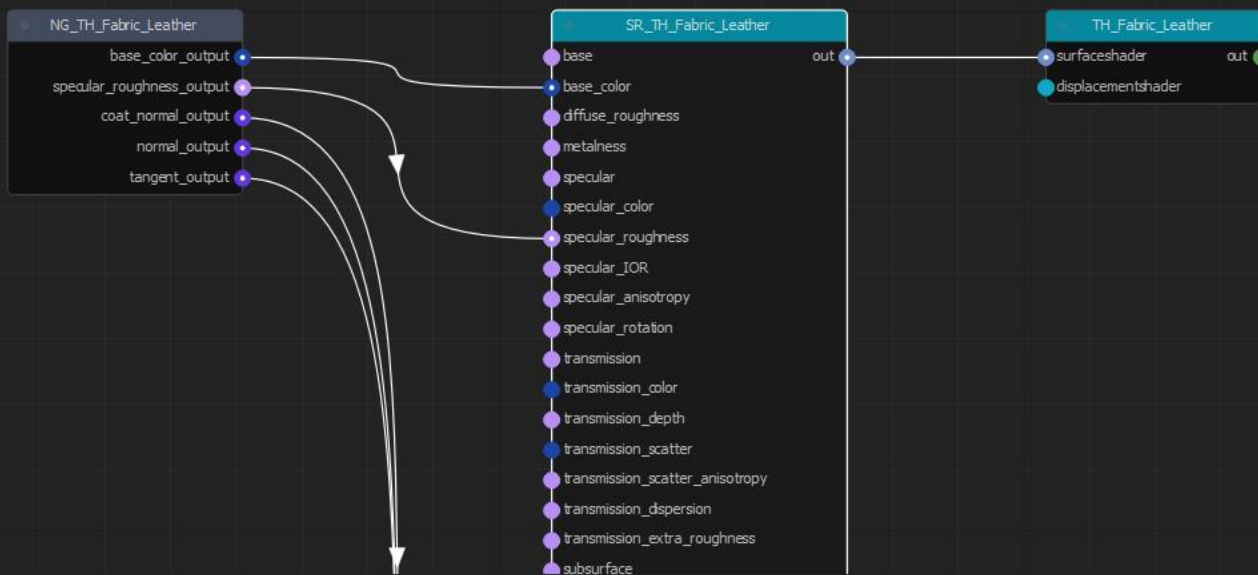
Subsurface scale: 1,00

Subsurface anisotropy: 0,00

Sheen: 0,00

Sheen color: 1.0 1.0 1.0

Sheen roughness: 0,30





Graph is valid.

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

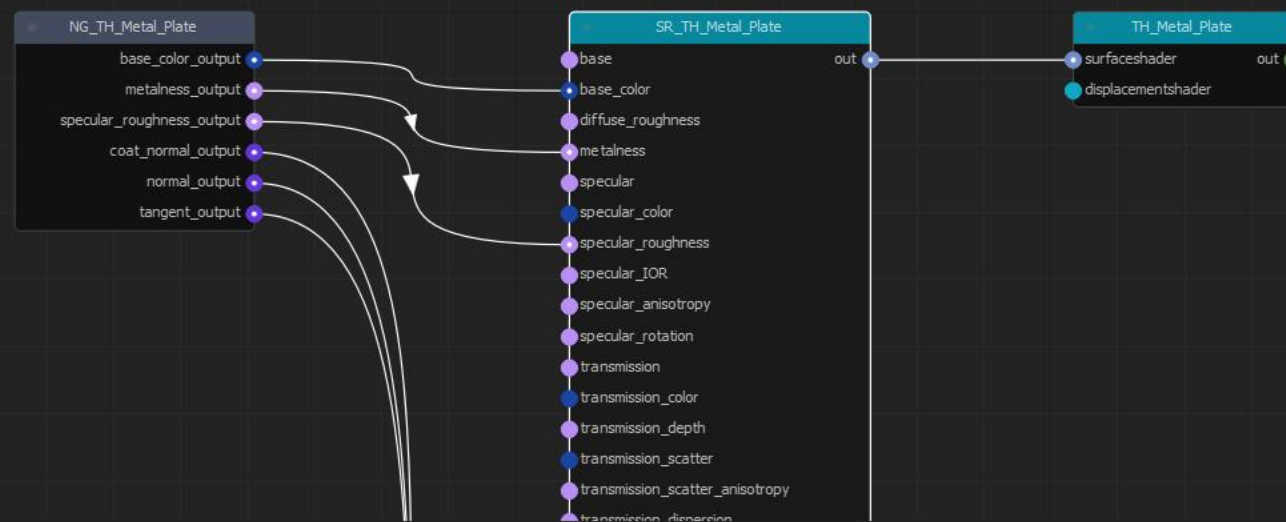
Sheen: 0,00

Sheen color: 1.0 1.0 1.0

Sheen roughness: 0,30

```

/
> 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
    
```



open
Source
days²³

Thanks to



#ASWF

/* ACADEMY SOFTWARE FOUNDATION



Try




out today!

github.com/PrismPipeline/QuiltiX

pypi.org/project/QuiltiX





Thank You!
Questions?