REALISTIC AGING OF MATERIALS IN COMPUTER GRAPHICS

A Thesis

by

MEGAN LEA WALKER

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Ann McNamara
Committee Members,	Carol LaFayette
	Tracy Hammond
Head of Department,	Tim McLaughlin

August 2015

Major Subject: Visualization

ABSTRACT

One of the most challenging tasks in Computer Graphics (CG) is depicting the accurate appearance of aging and weathered materials. This thesis examines the physical aging process of materials and translates that information into data that can be applied to CG materials resulting in a new prototype system for simulating realistic aging and weathering of CG materials. This new system will enable artists to quickly and accurately generate materials with a realistic appearance of aging and weathering. The resulting user interface generated from this system allows artists to create a variety of realistic and customized layered materials which offers a wide array of complexities. This prototype was then implemented into a studio setting which helped speed up the production process for material generation.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

1. MOTIVATION AND INTRODUCTION

The manner in which a physical material ages can vary widely across different materials. Materials can react with a variety of other components including air, water and even human contact [9]. Factors such as dust or dirt accumulation can also result in drastic changes to the appearance of a material. Changes are also impacted by the degree of exposure to the element. Prior research has established how different materials weather.

To recreate the appearance of realistic materials in CG, the artist must first have a solid understanding of how materials react to the real-world elements that cause them to age. Only with a clear understanding of this process can the artist begin to apply suitable values to CG materials. Artists generally spend a significant amount of time gathering reference materials representative of the material they wish to recreate digitally. Studying the real world material, often under different illumination conditions, and at various stages of aging, allows the artist to better understand the behavior they need to capture in a digital format and apply it to the shader. A shader is simply some computer code that governs the appropriate color and other characteristics of a material.

2. STATEMENT OF INTENT

To alleviate this burden of recreating photoreal materials for an artist, a better method was created to enable the artist to create the realistic appearance of materials using physically correct real world aging and weathering data. The goal of this thesis is create such an artist-centered system. This system allows the artist to create a fully customizable layered shader. The artist begins with a simple base shader. They can add layers to emulate levels of aging, weathering and other considerations including dust, dirt, smudges and rust. The capabilities of the new system are showcased by creating a large volume of materials with varying age and weathering with shaders from the new system.

3. BACKGROUND/LITERATURE REVIEW

3.1 Physically Based Shading

To achieve believable materials, physically based shading should be taken into account. Adopting a physically approach has several benefits including providing consistent results to lighting under multiple lighting scenarios and also the shading model is more intuitive due to the fact that it is based off of real world values and contains fewer parameters. Some of these parameters include [19]:

- Albedo (diffuse color)
- Specular
- Roughness (Smoothness, shininess, or glossiness)

3.1.1 Albedo

Albedo can be characterized as the color of the object, an example would be: A red delicious apple is red. We can think of this as a simple color attribute and therefore should not contain any additional lighting information including ambient occlusion, shadows and reflections [21]. Lighting will be provided from the light sources provided for rendering purposes. Albedo values can range from a sRGB value of 32 (charcoal) to 243 (fresh snow) in a physically based system, it is important to note that metallic objects have a diffuse albedo of black [20].

3.1.2 Specular

Specular can be described as the fresnel reflectance at normal incidence. The specular values of materials are based on the index of refraction. For nonmetals sRGB values range from 43-65 while metallic objects can range from 186-255 in a

physically based system. For practical purposes, a default value for nonmetals can be set to 59. Specular values is how a material can be defined for a shader, since these are measured values [21]. For depicting a single material, the specular value should be set to a constant value and should not be mapped to a texture. According to the given data, we can expect for nonmetal values to be very low, while metallic materials match the color of the given metal.

3.1.3 Roughness/Glossiness

Roughness or glossiness is an important attribute for physically based shading that defines how rough a surface is. This attribute defines how blurry the reflections are, the rougher the surface is, the more blurry the reflections are. The less rough the surface is the more the reflections appear sharp [21]. This attribute can be used to achieve breakup in reflections by mapping a texture to it. Artists can paint scratches and speckles in this map to achieve their desired look in the reflections of the materials.

3.2 Aging of Materials

3.2.1 Patina

Several simulations have been computed which illustrate specific weathering and aging in CG. One particular area of interest for illustrating weathering is patina. Patina is a chemical mechanism of weathering typically found on metallic materials. The patina can be characterized as causing visible changes to the color and reflectance of a material [9]. Dorsey and Hanrahan demonstrate a specific approach for simulating patinas on metallic materials in CG [8]. A series of procedural operators are used to apply patinas to a layered structure where the surfaces are represented as homogeneous layers. These layers represent a base copper, tarnish, and patina. This demonstration illustrates the appearance of metals over time as they are exposed to

the atmosphere or treated chemically.

Further CG simulations have been performed to demonstrate patina on objects buried underground for an extended amount of time. Chang and Shih used L-systems to simulate the patina process while still taking the environment map and local geometry into their calculations [6]. Their system much like Dorsey and Hanrahan consisted of 3 layers: a base layer, a tarnish layer, and an upper layer - green patina. They utilize a solid texture surrounding the object to simulate the role of soil in the corrosion process.

Patina is an important weathering mechanism that is taken into account in this thesis. In addition to this specific characteristic, other aging and weathering processes are demonstrated in this method. Each weathering process reacts to one another appropriately and are illustrated in an accurate manner.

3.2.2 Rust

Another corrosive weathering reaction is rust. To simulate rust on metals due to the effects of seawater, Chang and Shih expanded on their L-system patina simulation [7]. An L-System is described as the following [23]:

- A finite set of letters is defined.
- The letters are arranged in arbitrary length sequences to form strings.
- Each letter is assigned a rewriting rule.

The length of the string can grow quickly after only a few rewrites. Below is an example of an L-System being applied:

- Defined is the finite set of letters $\boldsymbol{a}$ and $\boldsymbol{b}$
- Rules are assigned to $\boldsymbol{a}$ and $\boldsymbol{b}$. $\boldsymbol{a} = \boldsymbol{ab}$ and $\boldsymbol{b} = \boldsymbol{a}$.

- The occurance of the letter ***a*** in a string is to be replaced by the sequence ***ab***.
- The occurance of the letter ***b*** in a string is to be replaced by ***a***.
- Substitution of letters takes place in parallel across the entire string. Results are illustrated below:

a

ab

aba

abaab

abaababa

abaababaabaab

This same concept was applied to simulating rust over a surface. The corrosion changes the immediate environment while the rust distribution is determined by the environment. Their model consists of four layers: pits, crack, blisters and an enhanced environment model. The enhanced environment model is used to simulate the interaction between water and the metallic surfaces which involves the effects of seawater.

The method described in this thesis takes the geometry of the asset into account while the artist will be able to determine the level of weathering the object has undergone. The artist can determine where the weathering and aging can occur if they choose.

3.2.3 Impacts

Impacts can be defined as a mechanical mechanism of aging. Typical forms of mechanical process involve compaction like dents and scratches. Some causes of these characteristics are dropping of objects or being hit by other objects [9]. Paquette et

al. focuses on impacts due to repetitive hits from a dierent object over a period of time [26]. Their simulation involves a dynamic object chosen by the user being hit at a static object. The surface mesh deforms according to the individual impacts by the tool. The importance of simulating scratches and small dents are utilized to produce accurate results. In contrast to Paquette's approach, impacts will be handled by the shader. The shader will demonstrate the material difference scratches can cause.

3.3 Material Libraries and Material Editors

Layering materials is an important method to consider for creating accurate material responses. For example, Disney's material designer allows an artist to choose materials from the library to add to the shader [5].

The artist paints a mask to blend between the multiple layers of the shader. Artists can add spatially varying shader modules generally represented by texture maps and/or procedural approaches. Parameters of the shader correspond with these shader modules that the artist can manipulate to get the desired look they want. These can then be layered on top of one another to create complex layered materials. This method allows for quick layering of different materials while providing a simple workflow for the artist. In this thesis the amount of aging is taken into account instead of just considering one level of aging as the Disney method does.

The game company Ready at Dawn created their own gaming engine which utilized physically based shading as well as layering different materials to achieve complex results [24]. Ready at Dawn's material editor allows the artist to build up composite material layers quickly and generate complex materials with minimal effort. The artist can use pre-existing masks or paint their own custom masks in a painting package such as Mari [12]. The material editor offers various parameters that can be adjusted for further customization, like additive surface, scaling and

per-pass contribution. The prototype developed in this thesis provides similar customization opportunities as well as provides several other parameters that would affect particular parts of the geometry itself including edges and crevices.

There are a multitude of material libraries online usually comprised of materials created by a number of different users [18]. There can be a wide array of materials to choose from ranging from accurate depiction of materials to completely unrealistic materials. Searching for exact materials can be difficult due to the amount of entries. The user can usually filter down the material type they want however, the quality of the material can range variably. As far as customizing the materials given, there is typically not a feature that provides these capabilities. The user would have to download the material from the online database, and make their decisions in the shader itself, making this approach not optimal for fast coverage.

There are several 3D packages that improve texture painting and shading workflows. Such software like Substance Painter allows the user to paint materials on a single mesh [2]. The user will receive physically accurate material responses in real time to ease and allow for efficient workflows. Substance Painter also gives the user the ability to paint with particle brushes to add additional wear utilizing some of their brushes. The current implementation of Substance Painter is limited to only one UDIM shell and can only generate texture resolutions of 4K. A UDIM is an automatic UV offset system that assigns an image onto a specific UV tile, allowing the artist to use multiple lower resolution texture maps for neighboring surfaces, which produces a higher resolution result without having to utilize ultra high-resolution images [22].

The Substance Painter workflow can also only be utilized on a single mesh at a time. In the method developed in this thesis, multiple meshes are taken into account and allows for multiple UDIMs to be included at the same time. This allows for

efficient coverage of entire environments without having to load in individual assets one at a time.

4. METHODOLOGY

4.1 Evaluating the Shaders During Development

To evaluate materials during shader development a test scene was modeled and lit to accurately showcase the resulting shaders. The scene required the following characteristics:

- The models needed to be simple.
- The models needed to demonstrate how a material would respond on different surfaces.
- The scene needed to be lit appropriately to evaluate the shader response.

A spherical material ball was modeled initially. When objects are flat, the reflections tend to flatten out as well which can make it difficult to evaluate shader behavior without rotating a camera. Round geometry is ideal for showcasing shader work as the viewer gets a sense of reflections and specular response when the reflections wrap around the sphere. Figure 4.1 shows the difference the same material and lighting can have on different meshes.

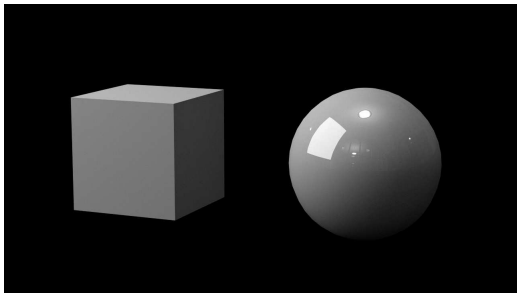


Figure 4.1: Lighting effect with same shader, but different geometry.

The initial geometry was adjusted to include crevices and edges to support the evaluation of the shader. A flat base was also modeled into this scene to help evaluate the shader on a flat surface.

A ground plane with a grid pattern mapped to it was added to the scene to give cast shadows and help provide reference for the artist. Figure 4.2 illustrates the resulting models in the scene modeled in Maya.

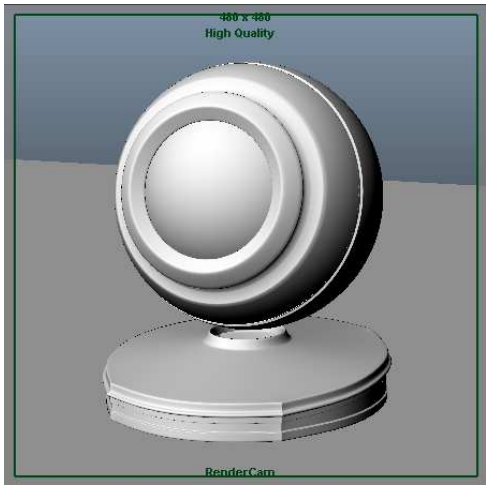


Figure 4.2: Screenshot of scene models in Maya.

To light the scene, an environment sphere was created by using a VRayLightDome containing a texture map that determines the amount of light coming from the direction on the virtual dome [17]. This particular light dome utilizes an interior HDR map to light our scene which also worked with the renderer, Vray. HDRI, or high dynamic range imaging, is an image file that records the actual color and dynamic range of the original scene the image is taken from rather than limiting the gamut and dynamic range of the monitor or other reproduction media [4]. This lighting technique provided the following for shader evaluation:

- Contrast in reflections.
- Highlight areas.
- Shadow areas.
- No added color contribution.

Physically based shading and lighting was used with the renderer Vray and resulted with the render in figure 4.3:

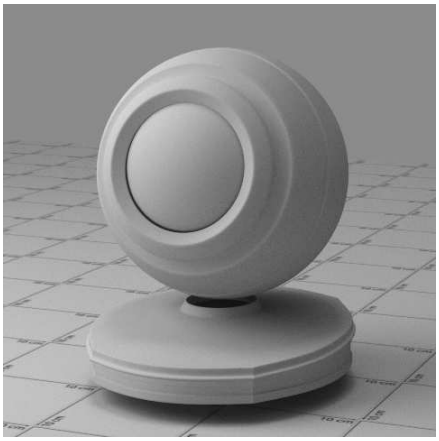


Figure 4.3: Render of the scene with lighting in Maya using Vray.

4.2 Creating Initial Materials

Materials in this method are physically based and the parameters are achieved by the following consideration:

- Albedo will be set according to figure 4.4 derived from researched data [13].
- Specular values will be set according to figure 4.5 and derived from researched data [14][19].

- Glossiness will be determined by artistic eye.
- Bump mapping will be determined by artistic eye.

Texture maps were used in cases to achieve surface variation either for:

- Color variation.
- Glossiness breakup.
- Bump mapping.

These materials are refined adequately for the user to get a wide variety of options through layer customization and were built in Maya and applied to the material ball created from section 4.1.

4.2.1 Nonmetal Materials

The materials in figure 4.6 were created to represent nonmetal objects. These looks were achieved by applying the following:

- Appropriate albedo values were selected and in some cases mapped to a texture according to figure 4.4.
- Specular value is determined from the following acquired data in figure 4.5.
- The shininess of the material is determined by glossiness.
- Textures maps were used to achieve specular breakup.
- Bump mapping is utilized to achieve specular breakup.

These materials provide the user with a wide variety of materials, which allows for flexibility in generating different materials.

Material	Albedo (Linear)	Albedo (sRGB)
Asphalt Fresh	0.04	59
Asphalt Worn	0.15	108
Brick (lowest)	0.3	148
Brick (highest)	0.5	186
Cement	0.55	194
Charcoal	0.04	59
Concrete Old	0.3	148
Concrete New	0.5	186
Gravel (lowest)	0.18	117
Gravel (highest)	0.72	220
Limestone (lowest)	0.3	148
Limestone (highest)	0.45	177
Paint (lowest)	0.2	123
Paint (highest)	0.35	158
Paper White	0.75	224
Rock	0.3	148
Sand (lowest)	0.2	123
Sand (highest)	0.3	148
Slate	0.2	123
Soil Average	0.3	148
Soil Dry Clay	0.23	131
Terracotta tile	0.28	143
Tree bark (oak)	0.1	90

Figure 4.4: Albedo chart for nonmetal materials derived from research.

Material	IOR	F0 (Linear)	F0 (sRGB)
Asphalt	1.635	0.058	70
Brick	1.44	0.033	54
Cement	1.68	0.064	73
Chalk	1.51	0.041	60
Glass	1.51714	0.042	60
Leather	1.54	0.045	62
Paint	1.49	0.039	58
Paper	1.557	0.047	64
Plastic	1.46	0.035	56
Polystyrene	1.55	0.047	63
Rock (low reflectance)	1.4	0.028	50
Rock (high reflectance)	2	0.111	94
Rubber (low reflectance)	1.33	0.02	43
Rubber (high reflectance)	1.5191	0.042	61
Wood polished	1.55	0.047	63
Tree bark	1.4	0.028	50

Figure 4.5: Specular chart for nonmetal materials derived from research.

Figure 4.6: Initial nonmetal materials generated from researched data.

4.2.2 Metal Materials

The following materials in figure 4.8 were created to represent metal materials. To achieve the looks of these, the following attributes were utilized:

- Diffuse always has no contribution.
- Color of the material is driven by specular color and according to figure 4.7.
- The shininess of the material is determined by glossiness.
- Texture maps were used to achieve specular variation.
- Bump mapping is utilized to achieve specular break up.

Material	sRGB
Silver	0.971519, 0.959915, 0.915324
Aluminium	0.913183, 0.921494, 0.924524
Gold	1.000000, 0.765557, 0.336057
Copper	0.955008, 0.637427, 0.538163
Chromium	0.549585, 0.556114, 0.554256
Nickel	0.659777, 0.608679, 0.525649
Titanium	0.541931, 0.496791, 0.449419
Cobalt	0.662124, 0.654864, 0.633732
Platinum	0.672411, 0.637331, 0.585456

Figure 4.7: Specular chart for metal materials derived from research.

Figure 4.8: Initial metal materials generated from researched data.

4.3 Creating Additional Materials

Additional materials were needed to create variation in the material library. For most of these cases previous data was utilized to expand the original materials into

new materials. For example, different levels of wear for copper were generated to create unique varieties for that material:

- Lightly worn copper.
- Medium worn copper.
- Heavily worn copper.

The next 3 sections illustrate the development of these copper materials and are used as examples for how the additional materials were created for the library.

4.3.1 Lightly Worn Copper

Initial copper values were utilized as a starting point for this material. As copper ages, the color of the material becomes darker and more saturated until it entirely oxidizes and a different material coats the top of it, patina [25].

The following were applied to the initial copper material to illustrate a slightly aged appearance:

- Color was saturated and darkened.
- Color variation was added through texture mapping.
- Bump mapping was adjusted to include scratches and specular breakup.
- Roughness mapping was added to include specular variation.

The resulting copper material can be in figure 4.9.

Figure 4.9: Lightly worn copper shader results.

4.3.2 Medium Worn Copper

A key component for creating this copper material was to illustrate medium wear which meant the material has experienced a fair amount of usage. The results are illustrated in figure 4.10. This would imply that the material has micro-scratches, large scratches, and some tarnish which would affect the material in the following ways:

- Shader would have lower values and more variation in the glossiness map.
- The color would be darker and more saturated with more variation.
- Scratches and dents would be present in the bump map.

4.3.3 Heavily Worn Copper

The heavily worn copper was an extension of the medium worn copper. All the maps painted were heightened to reflect a more heavily used copper material. The results are illustrated in Figure 4.11.

Figure 4.10: Medium worn copper shader results.

Figure 4.11: Heavily worn copper shader results.

4.3.4 Populating Additional Materials

The same approach was taken to achieve the materials in figure 4.12:

4.4 Creating Masks for the Layered Materials

Texture masks were created to blend between the different materials to maximize the realism in the materials. A variety of approaches were utilized to generate these masks:

- Tileable painted texture maps.

Figure 4.12: Additional shaders created.

- Edging geometry aware masking with tileable painted texture maps multiplied on top.

- Crevices geometry aware masking with tileable painted texture maps multiplied on top.

A shader network was built per mask type to control overall amount, edge amount, and crevices amount.

4.4.1 Controlling Overall Amount

To control the overall amount of the texture that is utilized the following considerations were taken:

- Texture maps needed to be painted full range.
- Texture maps needed to be tileable.

Full range allowed for a remapping node to control how much black and white values are being remapped. A remap is a utility node that allows the user to take an input value and remap the black and white values to be different values [3]. Figure 4.13 shows the adjustment of this remap on one of the painted map inputs.

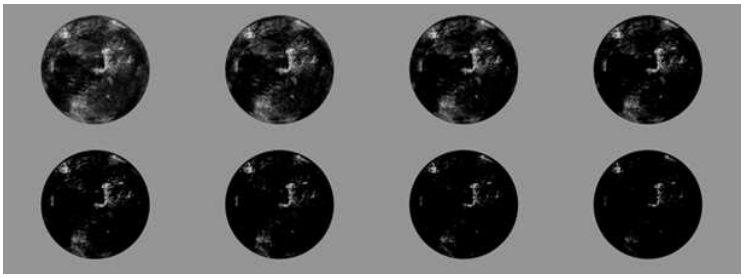


Figure 4.13: Effects of adjusting the remap nodes on the image input.

These maps were created using photo reference then converted to high contrast black and white images with grey values in between. This method provides a natural

blend between the materials. Hand painting was utilized to finalize the masks and make them tileable.

4.4.2 Controlling Edge Amount

Edge amount controls the amount of the texture that is utilized purely on the edges. This control is driven by the VRayCurvature which is a texture that samples the underlying mesh for curves. In order to compute this, a region around each shaded point gets sampled with additional rays, and all additional normals, are averaged to get a smooth normal at the original point. This normal is used to compute the color, which is shaded as a grayscale value. This means darker values are dents and lighter values are peaks.

To control the edge amount in this method, the sample spread is adjusted which controls the radius of the sampled region [16]. The figure 4.14 illustrates the control:

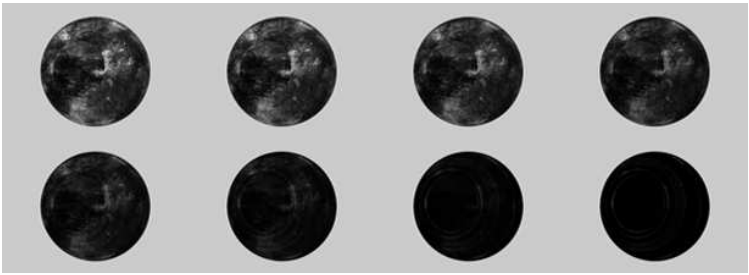


Figure 4.14: Effects of adjusting the curvature sample spread attribute with an image multiplied on top.

4.4.3 Controlling Crevices Amount

VRayDirt is a texture map that is typically used to simulate numerous effects like dirt accumulation around crevices. Ambient occlusion is used to calculate this effect, which simulates global illumination by faking darkness that the viewer perceives in

corners, mesh intersections, creases, and cracks. This occurs primarily where light is diffused by accumulated dirt and dust [15] .

A remap is connected to the VRayDirt node to control the crevices amount. The input min, max and output min, max are the parameters used to drive the amount of the crevices effect. The figure 4.15 illustrates the effect of these controls.

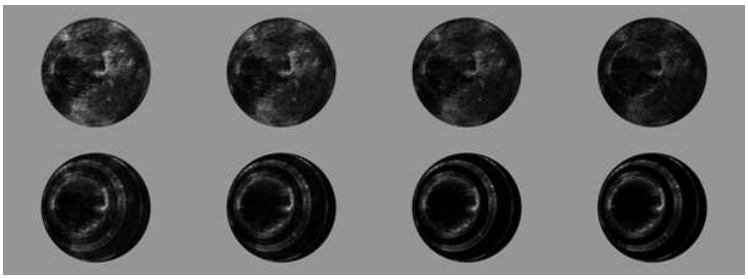


Figure 4.15: Effects of adjusting the V-Ray dirt node by adjusting a remap node with an image multiplied on top.

4.4.4 Creating Mask Shading Networks

To keep the shader networks flexible, a network for the individual masks were created combining the following methods:

- Amount (section 4.4.1) .
- Edges (section 4.4.2).
- Crevices (section 4.4.3).

Curvature and the dirt texture with a remap node are added together using a layered texture. The painted texture map is then multiplied on top of the combined curvature and dirt to add in a natural break up. Figure 4.16 shows a screenshot of the mask shading networking.

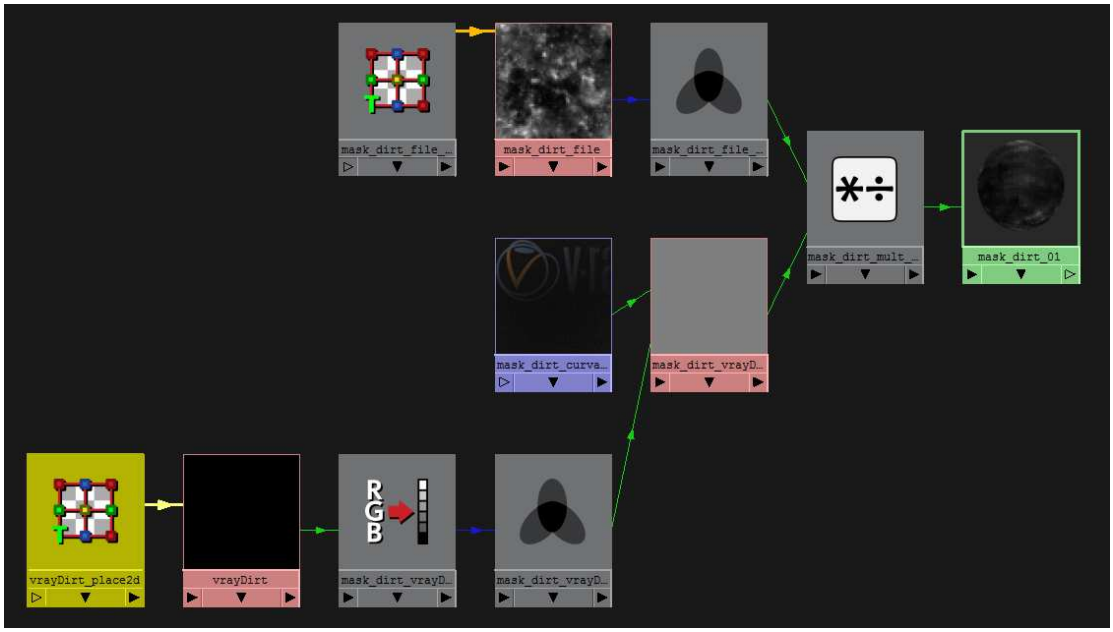


Figure 4.16: Screenshot in Maya's hypershade of mask shading network.

4.5 Categorizing Materials

To avoid overwhelming the user, materials need to be broken into categories to allow for easy browsing. The following are the categories that help divide the library in respective sections.

1) Architecture: Contains materials that are found on a building and used for construction. Examples include: Brick, concrete, stucco, and plaster.

2) Automotive: Contains materials specific to vehicles. Examples include: different types of car paint.

3) Cloth: Includes materials that are woven threading. Examples include: cotton, satin, silk, and denim.

4) Food: Includes materials that can be consumed. Examples: different types of

fruit, meats.

5) Glass: Includes materials that are made of glass. Examples: clear glass, thick glass, frosted glass.

6) Ground: Includes materials that are used for grounds. Examples: asphalt, moss, sand.

7) Grunge: Includes materials that are primarily used for layering to achieve wearing. Examples: dirt, oxidation, and rust.

8) Metal: Includes metallic materials. Examples include: copper, aluminum, steel, iron, gold.

9) Plastic: Includes materials that are made of plastic: shiny plastic, rough plastic, laminate.

10) Stone: Includes materials that are made of natural stone or rock materials. Examples include: clay, porcelain, rock.

11) Wood: Includes materials that are made of wood. Examples include: different types of wood shiny and natural, and painted wood.

To keep the organization consistent a similar categorizing method was used to organize material masks, resulting in the following:

1) Chips: Contains masks that are used for chipping materials. Exampling include: chipped paint, chipped wood.

2) Dirt: Contains masks that are used for dirty materials. Examples include: different levels of dirt patterns, dirt splotches.

3) Smudges: Contains masks that depict smudges. Examples include: smudges on shiny surfaces and finger prints

4) Scuffs: Contains masks that depict scuffs. Examples include: scuffs on floor, scuffs on doors, scuffs on wood.

5) Color Wear: Contains masks that depict different color break. Examples include different levels of noises.

6) Scratches: Contains masks that depict scratches. Examples include scratches on metal, plastic, wood, paint, etc.

4.6 GUI Design Phases

An artist friendly GUI, was designed through a series of iterations, to connect the materials and mask networks using Adobe Illustrator [1]. Illustrator allowed maximum flexibility while providing good conceptual design.

4.6.1 First Iteration and Feedback

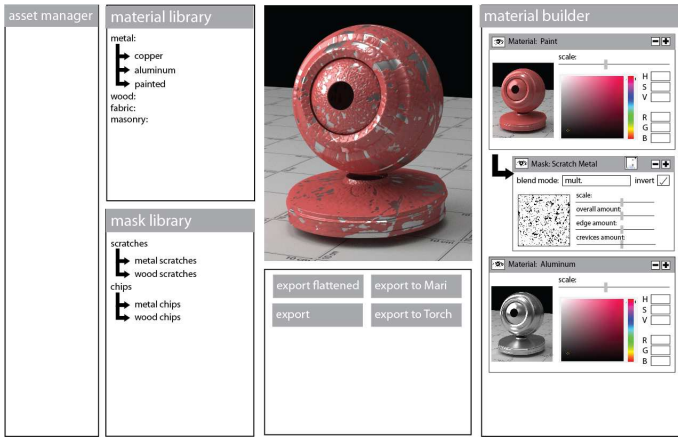


Figure 4.17: Initial pass of the interactive material library

This iteration defined what the material builder would look like. The main concern with this design was that it did not show us a preview of the base material or the masks. For an artist to make quick decisions, a full preview of the base material and masks are necessary. The asset manager was still in early development. This raised the question, to provide fast coverage, how was the artist going to apply the materials to the assets quickly. This component became the focus of the next iteration.

4.6.2 Second Iteration and Feedback

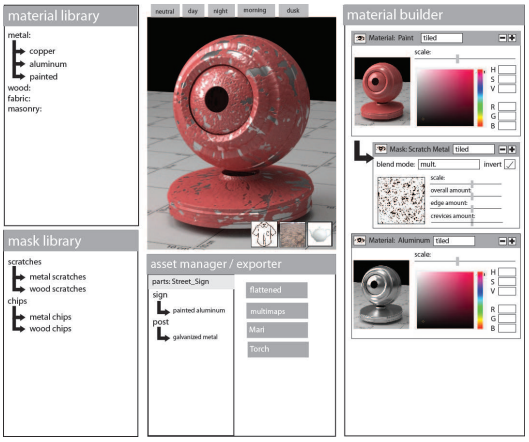


Figure 4.18: Second pass of the interactive material library

In the next iteration, the focus was on how the materials would be applied to the asset. A tree design is utilized to show the user the assets per the Maya session. Additional controls were provided to view the material under different lighting conditions.

4.6.3 Third Iteration and Feedback

This iteration includes a more refined material library, which includes previews of the materials. The display of the customized material was made smaller, to allow for more room while still maintaining full material evaluation size. This idea is inspired from Renderman's Slim [27]. The display now includes a description of the base material.

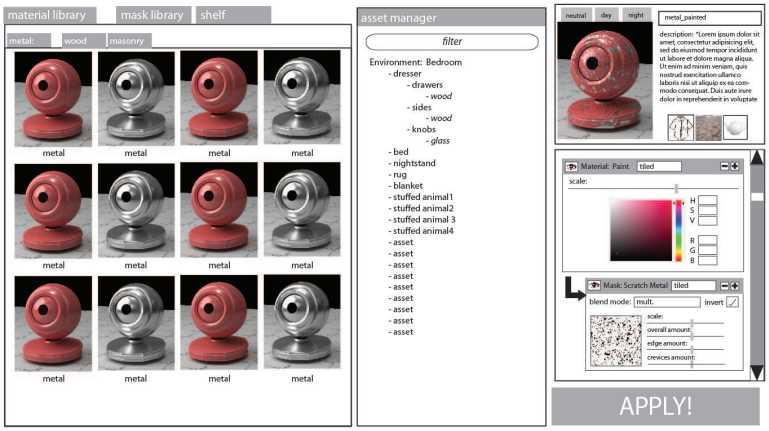


Figure 4.19: Third pass of the interactive material library

A search feature is now included for preset layered materials that the individual has created. These materials have designated tags associated with them that allow the artist to filter out the appropriate material stacks they want. This approach was inspired from Zappos [29]. Zappos interface and filtering methods are user friendly and give the user multiple ways of searching.

4.6.4 Final Iteration

For the final iteration, minor adjustments were made to consolidate the layer stacking. There were several characteristics gathered from this design phase which

included:

- The UI needed to display materials for fast browsing.
- The UI needed to display masks for fast browsing.
- A method for applying materials to the scene was necessary.
- The user would need a visually simple display to create customized materials.
- The user needed to see the material being customized.

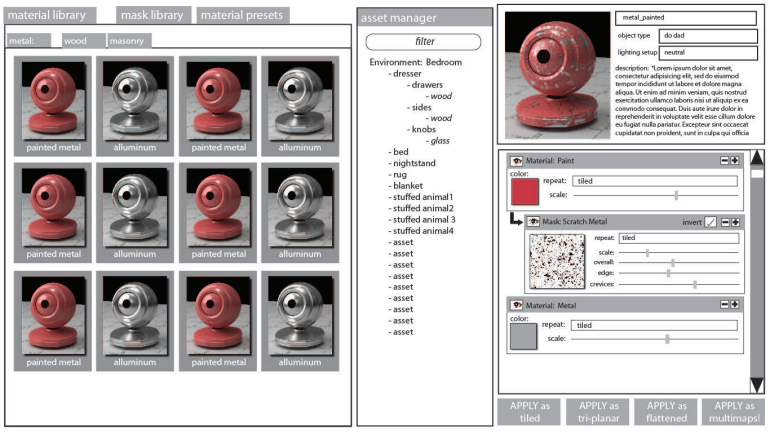


Figure 4.20: Final pass of the interactive material library

5. IMPLEMENTATION

PyQt was used for the UI implementation of this method [11], and utilized within Maya. The goals of the UI were discovered in the previous section:

- The UI needed to display materials and masks for fast browsing.
- A method for applying materials to the scene was necessary.
- The user would need a visually simple display to create customized materials.
- The user needed to see the material being customized.

5.1 Phase 1 - Initial Setup

Using PyQT, the overall layout was constructed from the designs developed in 4.7.4. The individual widgets to fit within this layout were then constructed. Thumbnails generated from the methodology section were utilized to represent the materials in the material library. A dictionary was created to contain the necessary information utilized in this tool:

- Name of the material.
- Image path location.
- Description of material.
- Category it is contained to.
- Mask or material identifier.
- Unique dictionary key that correlated to specific materials and masks.

Figure 5.1 shows the pseudocode for storing these parameters.

These thumbnails, with the material name from the dictionary, populate the left hand widget, and are differentiated by the material categories into separate tabs. The same principles applied for the mask portion. Figure 5.2 shows the code for populating the individual category tabs.

Dictionary keys:

dictKey	= stores the unique key that correlates to each material/mask
displayText	= stores the name of that material/mask
imageFileName	= stores file name of the thumbnail of the material/mask
description	= stores the description of the material/mask
listWidgetName	= stores the category name the material/mask correlates to
nodeType	= stores whether this material is a mask or material

Figure 5.1: Pseudocode for material and mask library dictionary.

Populating Category Tabs:

```
Loop for number of keys in dictionary:
        If key nodeType = “material” and key listWidgetName = material category tab name
                Add the material image to the corresponding category tab
        Elif key nodeType = “mask” and key listWidgetName = mask category tab name
                Add the mask image to the corresponding mask tab
```

Figure 5.2: Shows the population of the individual category tabs.

In the middle section, a display of the geometry names in the scene is created. This portion reads in the geometry names based off the geometry in the Maya scene, and populates a tree widget. Figure 5.3 shows how this was setup.

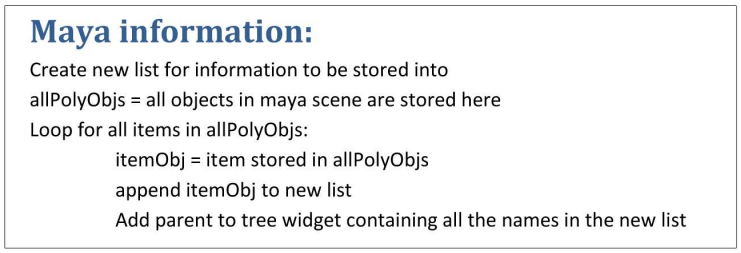


Figure 5.3: Shows the population of the asset manager with what is contained in the Maya scene.

5.2 Phase 2 - Interactivity Phase

A few additional steps needed to be incorporated to allow the material thumbnails and the geometry names in the tree to work together. The functionality for multiple material thumbnails to be dragged and dropped onto the geometry was incorporate, which illustrated a material being applied to the asset.

Materials are added to geometry names, by creating a child, based on mouse release after dragging and dropping. Additional functionality was created so the user can reorder the materials and masks as they want, within the tree widget.

To further improve the functionality of this tool for future implementation, the material should be added wherever the material is dropped into the layer stack, not always the top of the tree. Figure 5.4 shows this interactivity pseudocode.

5.3 Phase 3 - Interactivity Phase Layering Abilities

Individual widgets were then created to allow the user to customize their material and masking parameters. Material widgets were created to include the following placeholder parameters:

- Color change abilities.

Interactivity between Asset Manager and Material Thumbnails:

```
Define startDrag for Material/Mask thumbnails widget:

Define dropEvent for Asset Manager:
        If parentLocation = a parent of the tree widget:
                Add child to parent location containing the material name dragged over
        Else:
                Do nothing
```

Figure 5.4: Shows pseudocode for the interactivity between the thumbnails and the asset manager.

- Scale sliders for texture repeatability.

The masking widgets contained the following for placeholder parameters and were connected to the masking networks defined in section 4.4:

- Scale sliders for texture repeatability.
- Overall amount of the texture.
- Edge amount to effect the texture.
- Crevices amount to effect the texture.

The diagram 5.5 illustrates how the UI elements and the masking network parameters were connected.

A list widget is created to house the individual widgets for materials and masks defined by the user.

(1) Amount controls remap input and output values of remap node attached to texture user has chose

(2) Edge controls sample spread attribute of the curvature node

(3) Crevices controls the remap node attached to the VRayDirt node

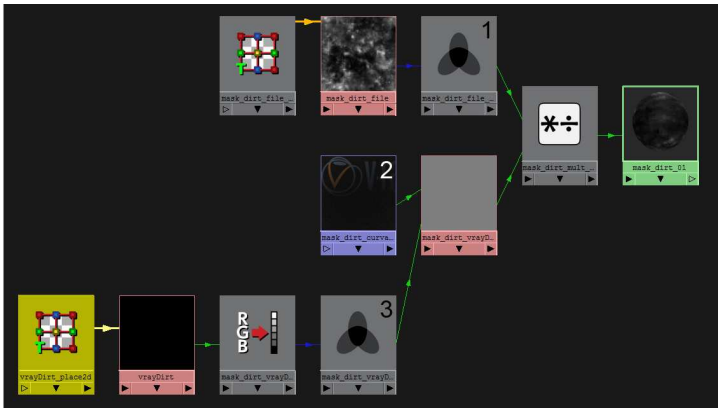


Figure 5.5: Shows a diagram of how the sliders from the UI and the masking network is connected.

Populating the Material Layer Stack:

Define mouseReleaseEvent for asset manager:
- If parentLocation = a parent of the tree widget:
  - Populate the layer stack with information gathered in the next step
- Else:
  - Do nothing

Define information needed to populate layer stack:
- Loop for the number of children on the selected parent in asset manager:
  - Loop for number of keys in dictionary:
    - If displayText = material name in selected parent in asset manager and nodeType = “material”
      - Add material widget to layer stack with designated material name
    - If displayText = mask name in selected parent in asset manager and nodeType = “mask”
      - Add mask widget to layer stack with designated mask name and mask thumbnail

Figure 5.6: Shows pseudocode for the layer stacking abilities.

5.4 Phase 4 - Material Display

The materials were composited together to illustrate what the material would look like roughly based off of what materials the user has selected to layer.

The material thumbnails were used for the designated materials selected from the asset manager. For the masking portion, the masks were rendered on the material ball which included an alpha and were used as a mask to blend between the materials selected from the user. Multiple materials and masks can be added based off

the input from the asset manager. To achieve this compositing effect, the Python module Python Imaging Library or PIL which adds image processing and graphics capabilities to the Python interpreter [28]. The display updates if the user chooses to rearrange the materials/masks in the asset manager. The following pseudocode in figure 5.7 and 5.8 illustrates the functionality described here.

Compositing the Materials in the Material Display:

```
In mouseReleaseEvent for asset manager:
        Define materialImage  list
        Define maskImage list
        Loop for number of keys in dictionary:
                If displayText = material name in selected parent in asset manager and nodeType = "material"
                        Add imageFileName to materialImage list
                If displayText = mask name in selected parent in asset manager and nodeType = "mask"
                        Add imageFileName to maskImage list
        Call compositeImage(materialImages, maskImages)
        Display_image_path = file path to display widget image
        display widget code(display_image_path)
```

Figure 5.7: Shows pseudocode adding images to display widget.

Define compositeImage:

```
Loop for items in materialImage:
        Add full file path header and ".png" to imageFileName
        Append to new materialNames list
Loop for items in maskImage:
        Add full file path header and ".png" to imageFileName
        Append to new maskNames list
Loop for (i) the length of materialNames – 1:
        If i = 0:
                Mask = Image.open of maskNames[length of maskNames – 1]
                Img1 = image.open of materialNames[length of materialNames – 1]
                Img2 = image.open of materialNames[length of materialNames – 2]
                Comp_img = Image.composite of img1 and img2 with mask
        Else:
                mask = Image.open of maskNames[length of maskNames – i – 1]
                img1 = comp_img
                img2 = Image.open of materialNames[length of materialNames – i – 2]
                comp_img = Image.composite of img1 and img2 with mask
Save comp_img to display_image_path using background.save
```

Figure 5.8: Shows pseudocode for compositing the images to create the displayed image.

6. PRACTICAL APPLICATION

Further extensions are necessary to make this prototype applicable in a large studio environment. For the application to work for DreamWorks for example, a number of artists were asked to evaluate and provide feedback on the prototype of the interactive material library. The feedback provided further insight into the functionality necessary to develop this prototype into a fully functioning application for the department, and included the following recommendations:

1. The department needed a way to assign different materials to various pieces of geometry parts of a specific asset.

2. Artists needed a visual representation of what materials were assigned to these various parts of geometry.

3. Artists need access to additional material information for further explanations and renders from different lighting scenarios to get a full understanding of how the material responded in different lighting conditions.

These key changes required a restructuring of the design of the UI to accommodate a more robust material and geometry assignment capabilities while also providing maximum functionality. The first pass of the UI yielded these results according to figure 6.1:

After our initial pass on restructuring the layout, another pass was taken for the layout of the UI. This new design organized the UI more efficiently by keeping the workflow from left to right. We also wanted to keep the interface as clean and simple as possible to accommodate a more efficient workflow. The new UI has not been

(a) DWA, GUI implementation

Figure 6.1: First pass for DreamWorks implementation

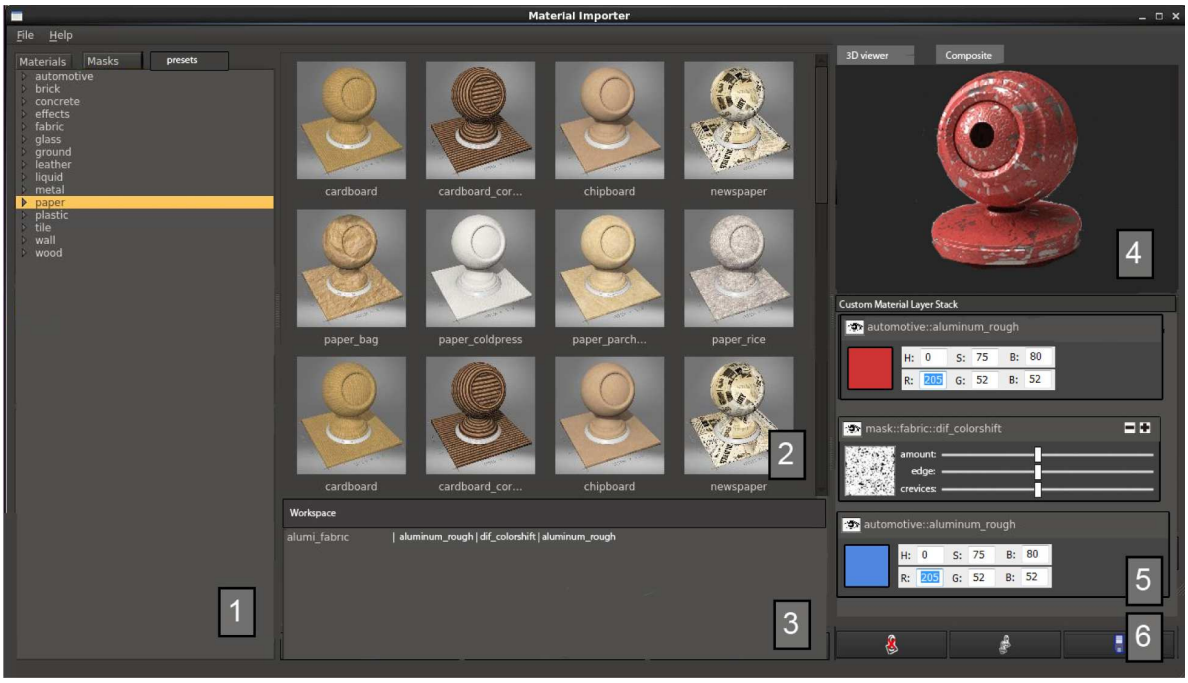
implemented completely on the studio level, however the draw overs can illustrate the future functionality of the application, refer to figure 6.2.

1. This is the mode for browsing materials, masks, and preset layered materials contained in the library.

2. The artist can choose a category of materials and the middle portion changes to allow the user to view the materials in that category. Additional information can be found by double clicking the icon in the center.

3. This is the workspace for artists to create their custom materials. The user can drag materials from the top into this workspace. This gives the user the

flexibility to have multiple materials in one session and eases the material assignment process by containing the custom materials in one section.



(a) DWA, GUI draw overs

Figure 6.2: Draw overs for future implementation mode 1

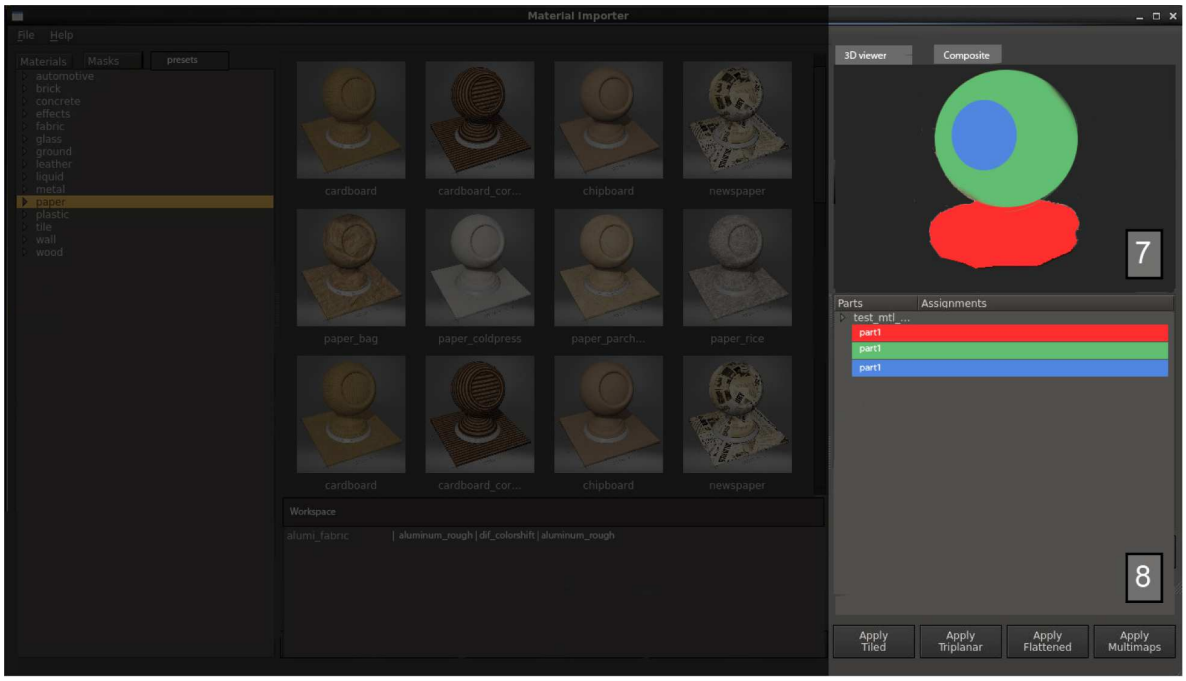
4. This is a 3D viewer which displays the custom layered material the artist is working on. As the artists adjusts various parameters, the display will update to show what the material will look like.

5. This section is where the artist can further customize their layered materials. Parameters like color of the overall material can be adjusted, along with scale and other parameter provided for that specific material.

6. This portion is where the artist can choose to export the material to the proprietary shader builder software.

7. Similar to the 3D viewer for displaying custom layered materials, the portion in figure 6.3 displays the geometry but depicts the material in the form of a random color.

8. This color corresponds to the list of geometry parts listed below it.



(a) DWA, GUI draw overs

Figure 6.3: Draw overs for future implementation mode 2

The artist can retain the functionality of layering materials together in a compositing manner, while having more control over assigning materials to various parts of the geometry. With the material assignment portion of the tool, the artist can drag and drop materials directly onto specific parts of the geometry, provided in a list format underneath a 3D view of the geometry itself. When the material is assigned

to a specific geometry part a random color is assigned to its listed counter in the 3D viewer. This gives the artist the ability to clearly relate materials to geometry parts.

Once the artist is satisfied with their customized material, they can export the materials from this UI into the proprietary shader builder program.

7. RESULTS

7.1 Evaluation of Materials

The evaluation of these materials are subjective, however we can state that the majority of the attributes used for each material were derived from acquired real world data. This should indicate that the materials are accurate respresentation of the material they are defined as. There were several cases when artistic eye was relied upon simply because data was not provided for those materials, however these specific materials were created to match specific looks. The maps used for some of these materials were generated to help support reference of what the materials look like while taking into account the physically correct values that defines them. These materials can be seen as accurate representation simply because we have an understanding of what materials should look like. The test layered materials created can be seen as successful because they match closely to real life reference of those worn materials and the user is able to create these variety of materials with minimal effort. Figure 7.1 shows a variety of some of the materials the user can create from this method.

7.2 GUI Breakdown

Primary goal behind the UI created is to generate materials quickly as if surfacing an entire environment rather quickly. There are several considerations that needed to be taken into account.

1) The UI needed to display materials for fast browsing

2) The UI needed to display masks for fast browsing

3) A method for applying materials to the scene was necessary.

4) The user would need a visual simple display to create customized materials.

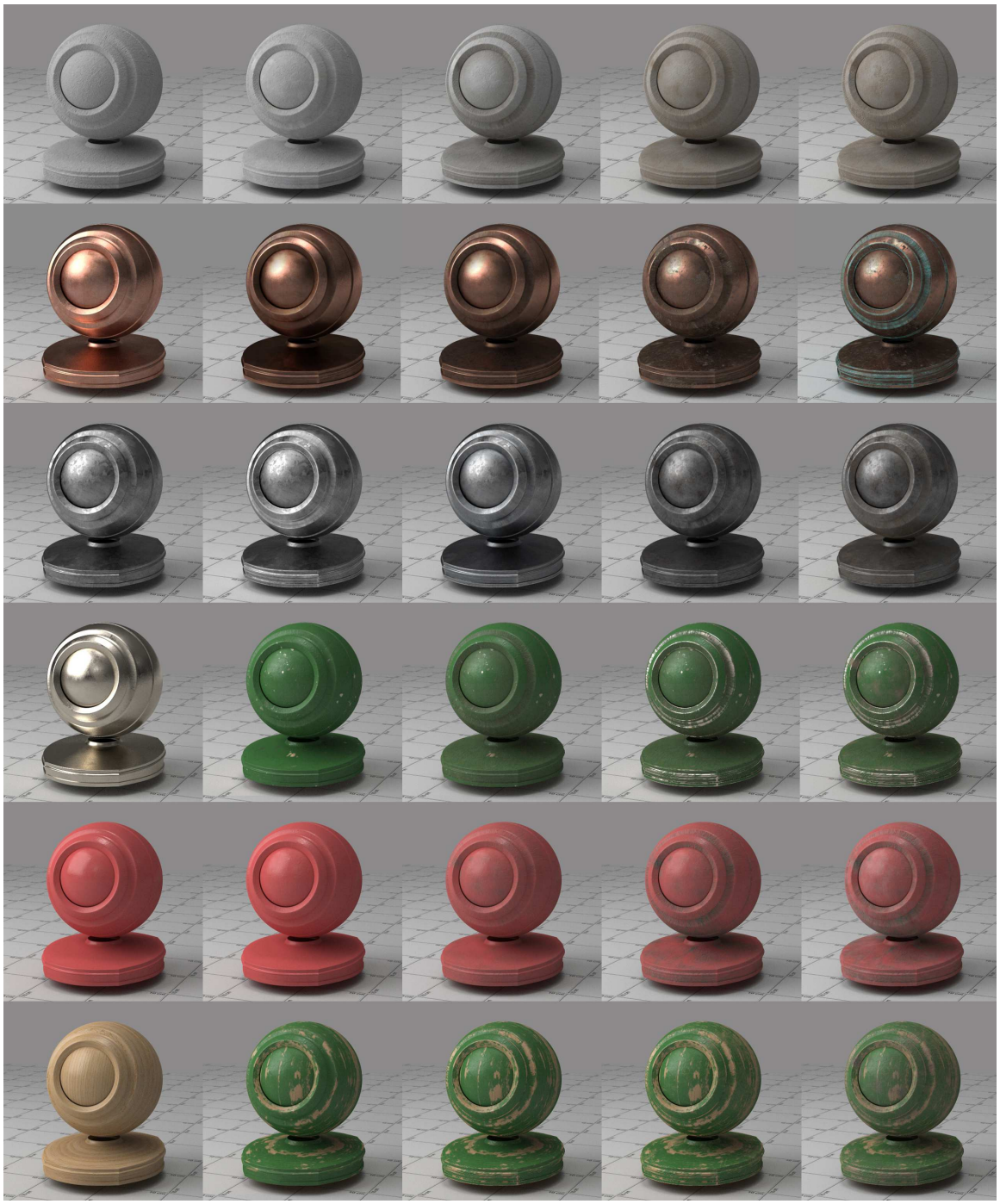


Figure 7.1: Resulting materials from method.

5) The user needed to see the material being built up so the viewer would know roughly what the customized material would look like. Before creating materials for the material library. I decided to setup the GUI with a test material: painted metal. This provides me with 2 materials to test the layering capabilities: metal and paint. I first began with creating this quick material in Maya and V-ray.

7.2.1 Material Library

The material library is the component which contains all the base material generated for the library. The user can toggle between different types of material types like metal, wood, grunge, etc. The material library works with all the other components for the GUI. Once the user has chosen a material, they drag the material into the asset manager, which populates the asset manager, the material builder, and the display widget. These materials are rendered images generated in Vray. The materials are demonstrated on a render object created in Maya. The render object is round to show the reflectance while also providing insight to what edges and crevices would look like with the material attached to it.

7.2.2 Mask Library

The mask library is similar to the material library. It contains images of all the masks the user can utilize to build the appropriate material they want. These masks can be dragged and dropped into the material builder to mask between 2 materials. The images themselves are masks generated from photoshop. The images themselves are tileable so they will not include seems on the asset. These images are also high resolution (at least 1024x1024) in order to provide fine detail for the artist .

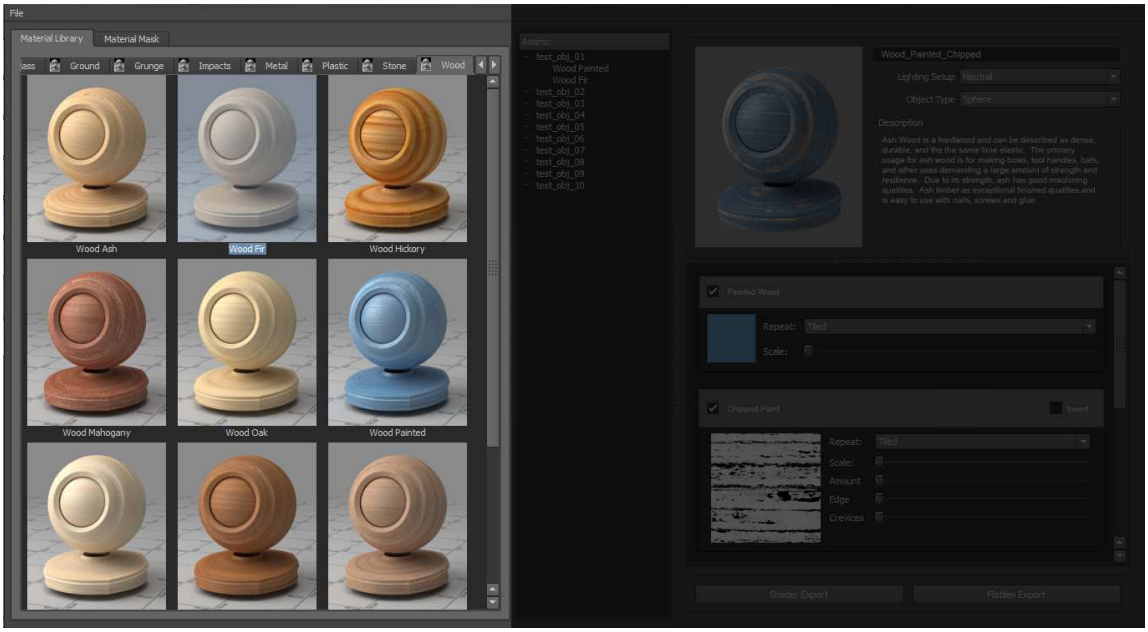


Figure 7.2: Material library and mask library portion of the UI

7.2.3 Asset Manager

The asset manager is a tool to display all the assets in the Maya scene once the interactive material library is loaded in. The user is given the option to filter out assets as they choose. The assets are displayed in a tree layout. Once the user drags the material/materials they want into the asset manager, a child is created under the appropriate asset they drug the material to. This then populates the material builder appropriately.

7.2.4 Material Builder

The material builder is what sets up your layered material with all the materials the user has drug into the asset manager to attach to the asset they want. For example if the user wanted to create painted metal. First the user would find the metal they wanted and drag that into the asset manager and the material builder

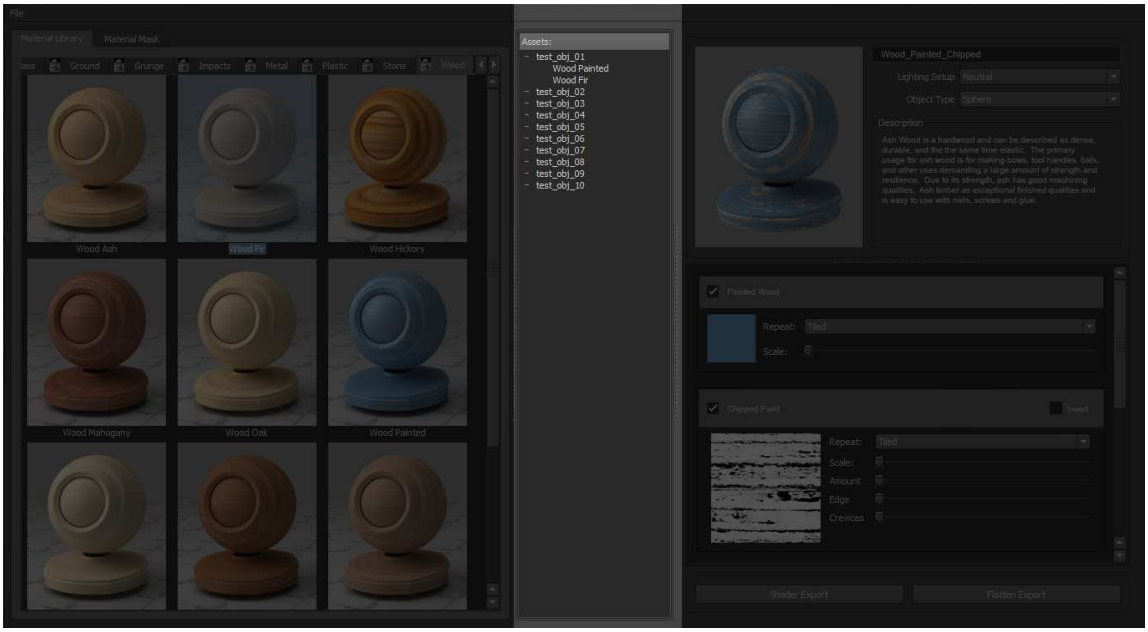


Figure 7.3: Asset manager portion of the UI

will populate the material builder. Next, the user would find the paint material they want and drag it into the asset manager. This will populate the material builder as well. The user will then go to the mask library and find a mask they want to use to mask between the metal and paint. The user will drag and drop this into the material builder in between the 2 materials. This will update the display widget with an image that depicts the artists changes. The user has several options to adjust the material and the mask. For the material, the artist can choose the color of the material from a color picker. They also have the option to adjust the scale of the textures for that material. The mask has a few more options than the material itself. The user can adjust the scale much like the material, but now they can choose to tile the image map or use a triplanar projection for the image maps. They also have the option to adjust the amount of the wear they want that is provided from the mask, the amount in the crevices and the amount that effect the edges. While the artist is

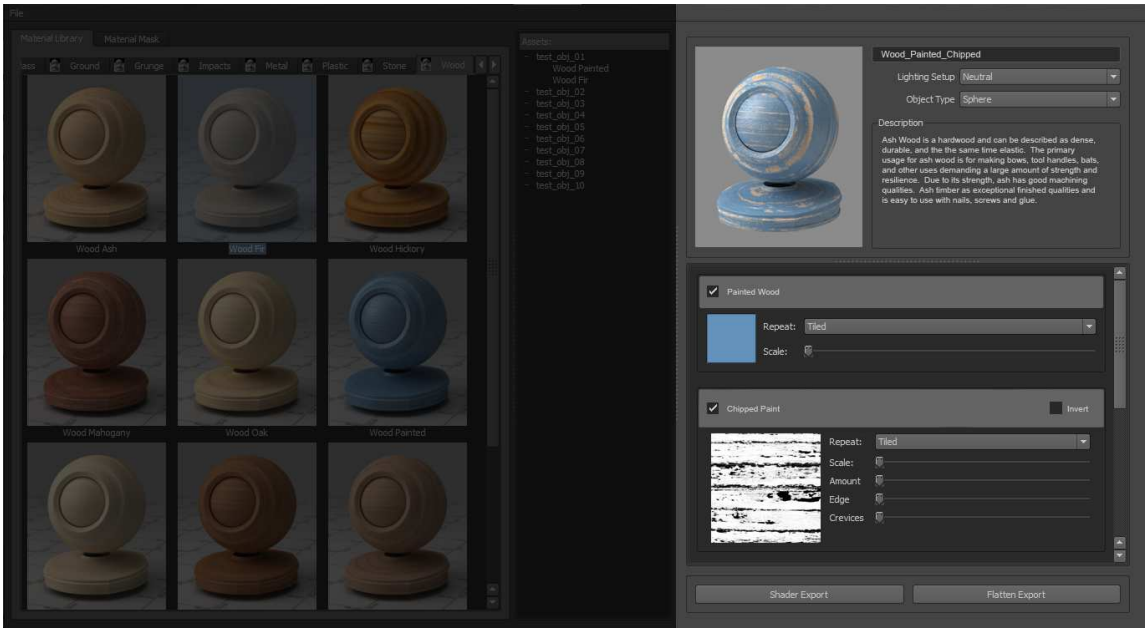


Figure 7.4: Material builder portion of the UI

making these adjustments the display widget will update with the appropriate image that depicts the artists changes.

7.2.5 Display Widget

This widget displays the composite image of the material the user is building. As the user builds and adjust the material in the material builder the displayed image will change based on the parameters the user is generating. A description is also included in this widget. The description provides detail information about the base material that user is utilizing.

7.3 Studio Implementation Summary

This thesis can be seen as a success because these concepts are being pushed forward in a production setting. The current phase of this implementation is still early and in the testing phases, with several artists testing the current implementation

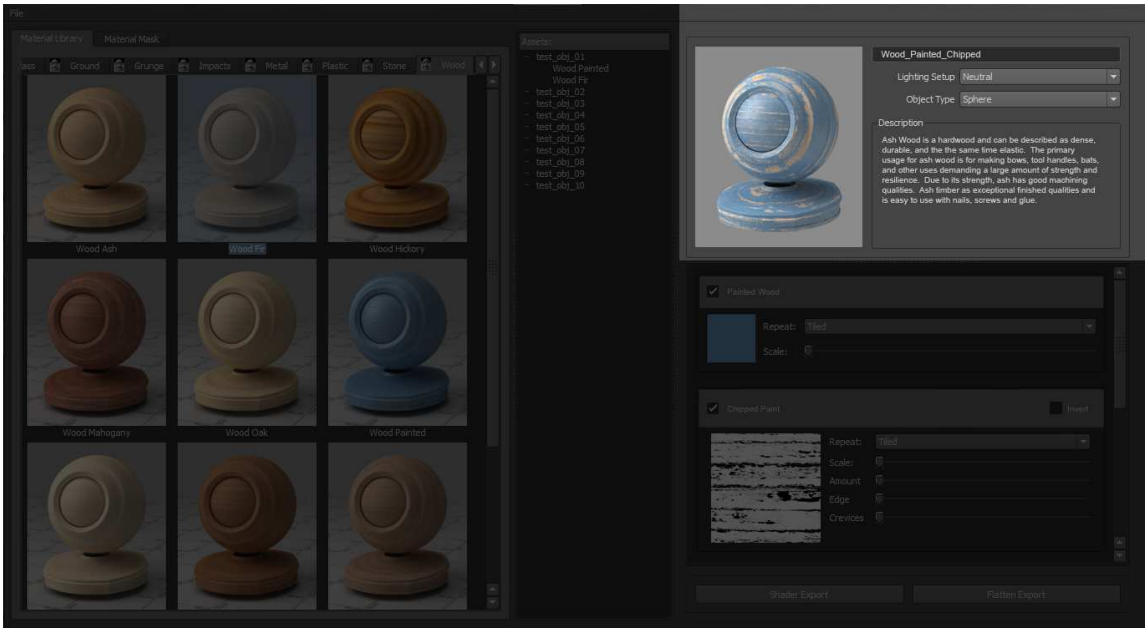


Figure 7.5: Display widget portion of the UI

and providing feedback on how to improve the process. The feedback for improving the workflow with this has not affected the overall design of this tool, but has more focused on fixing minor technical issues. A current production is pushing forward with full implementation of these concepts because they are wanting to unify workflow and material responses across the entire department. In addition to creating this tool, an entire workflow is being built around these central ideas provided in this method. This unification becomes extremely important especially when dealing with a department being split into several different locations worldwide.

8. EVALUATION

8.1 Evaluation Summary

As stated in the previous section, the process for implementing this method at the studio level is still early, however three studio employees were interviewed and asked to provide feedback on the method described in this thesis and how this affects the efficiency at the studio. The feedback provided here consists of how the current workflow is affected by this new process. All three individuals agreed the tool created for this method can be seen as a vast improvement and streamlines the current surfacing methods. The consensus was that the tool creates an efficient workflow by providing fast surfacing coverage of assets and even entire environments. Patrick noted that the materials used were utilized as "a first and sometimes final pass."

Since the materials are predefined, artists do not have to worry about depicting accurate shader attributes. Jeremy reveals that this method "streamlin[es] our process...and [gives] the ability to rapidly fill an entire environment with accurate, render efficient, [and] visually sophisticated materials." The individuals agree that this concept provides a strong foundation for the materials. The artist can spend their time designing out their materials and concentrate on painting necessary detail where necessary detail is appropriate.

All individuals agree we gain efficiency with setup time with the new method. The initial approach was to "[create] materials from scratch every time [a material is] needed, or [copy a material] from asset to asset using a very clunky, unintuitive, and archaic method. This has resulted in an enormous amount of waste not only within our department, but in our deliveries to downstream departments." - Jeremy notes. With the new method, "a user friendly tool for rapidly assigning" is has

helped “dramatically reduce re-inventing the wheel for surfacing.”

Both Christian and Patrick agree that this approach can be expanded to additional departments to help standardize other departments workflows.

For further reading on the individuals opinions, the individuals interviews are included in the sections below:

8.2 Technical Director and Shading Technical Director, Christian

Christian is a technical director and has been with the studio for nearly 3 years. His many responsibilities include supporting artists within the department and outside the department. He also develops tools for production pipeline with a focus within the surfacing department and the next generation of pipeline adoption.

“The material library is an incredibly valuable tool in streamlining surfacing workflows. By providing a global place where artists can store proven materials, and a user friendly tool for rapidly assigning these materials onto new assets, we’re able to dramatically reduce re-inventing the wheel for surfacers. Vetted materials from the library provide a strong foundation for an artist to quickly build upon without getting bogged down in technical details.

This has a lot of benefits, aside from just allowing surfacing artists to focus on new material looks and artistic decisions. The approvals process can now go faster, and initial material look decisions can be done before any hours are spent on an asset by going through the library with a film’s creative leadership. It’s now much easier to achieve a consistent look for a show; instead of relying on an informal system of loose show specific standards for various materials, an artist can instead quickly look up what their show’s approved version is of a material and apply it on an asset.

Pre-vizualization gets a massive jump in quality from this as well. Rather than having to wait for a surfacer to finish the materials on an asset, or substitute their

own approximations of how a material might look, pre-viz can take proven materials from the library and apply them to their work. Combined with temporary tiled textures or texture synthesis, this becomes a powerful way to get pre-viz renders much closer to what they might look in final.

Finally, the material library paves the way to improved asset re-usability in other areas. Other libraries can follow its example to have a standardized place for animations, models, effects, and other processes that might want to be used across shows."

8.3 Technical Director, Patrick

Patrick is a technical director and has been with the studio for 7 years. His many responsibilities include supporting artists within the department and outside the department. He is also responsible for maintaining and streamlining the surfacing pipeline during productions which includes initial setups and implementing solutions to assist artists with pipeline issues.

"The tool has already garnered users within the studio and the adoption of material library within our pipeline has facilitated the speed with which surfacing artists can work. We have long had the idea of a material library but the difficulty of getting materials from the library into assets left it unused in many cases. This UI and workflow modification has greatly changed the way in which surfacing artists interact with the materials in the library and eased their transition into using it as a first and sometimes final pass.

Surfacing has started using the tool as a way to get an initial set of materials onto an asset quickly when setting up an entire location or group of assets combined into the same environment. This has given surfacing artists faster feedback when working. They used this tool to make sure they are starting with the same materials

on alike surfaces in an environment as a whole, where before they would need to setup shaders by hand or copy and paste them from one asset to another. With faster setup using this tool a surfacer can now visualize all the assets together sooner and thus allowing them to present environments as a whole for director reviews.

Lighting artists see a benefit when surfacing is using this tool for some of the same reasons. Once a location is setup with an initial pass using responsive materials, lighting can move faster with their initial setup of lights in a location.

We are excited to branch use of the tool to modeling where they will be able to assign within Maya the materials that can be used further down the pipeline. This makes it so background assets may not need additional work. More likely, this frees up artist time in downstream departments to paint finer details or setup more complicated networks for unique or close-up assets.

Having predefined materials gives a show a more consistent look overall. Instead of requiring artists to keep track of what the defaults for a shader should be set, the materials come with those defaults, giving the artist a fewer number of parameters to modify in order to get the look they want. Materials do come with defaults assigned by shader writers, but often these fall quickly out of sync with changes to the render engine. The material library is easily updated concurrently with changes to the render engine and allows the surfacing department to keep up with those changes.

We have studios in several parts of the world and the material library is keeping all studios in line with what materials should be used on a given show. Since the tool has tagging, people working cross-site can filter down the materials to use just those for a given show regardless of where they are working.

There is a notion within the studio of running a series of tests on an asset before sending it on. These tests can cause the look of an asset to change, but materials from within the library are less likely to fail tests because of the vetting process those

materials go through. So in the end this prevents a look change during the standized fixes made before sending assets downstream."

8.4 Surfacing Department Head, Jeremy

Jeremy Engleman is a surfacing supervisor and department head and he has been with the studio for 15 years. Jeremy has been a supervisor for 3 years and his shows include Home and Croods 2. As supervisor, Jeremy guides artists technically and artistically while also providing solutions for the two. He is also responsible for identifying pipeline issues and leading the department with solutions for streamlining and supporting the department.

"I've had the opportunity to work with Megan on a tool that I believe will fundamentally change the way we do our film production work in the DreamWorks Surfacing Department. For a long time weve created materials from scratch every time they are needed, or copied them from asset to asset using a very clunky, unintuitive, and archaic method. This has resulted in an enormous amount of waste not only within our department, but in our deliveries to downstream departments.

Because Megan possesses a firm understanding of both the technical and artistic nature of materials and their properties, and the demands of an artist centric workflow, she has been able to design an entire process for working with materials. She has designed tools facilitating the cleanup and packaging of existing, approved materials, tools to browse, import, and customize materials in the catalog, and procedures for approving and maintaining those assets. She has worked with multiple departments in order to design a system which satisfies the needs of complex deliveries across the studio and encourages best practices between those departments.

All of this has been instrumental in streamlining our process, leveraging existing material assets, ensuring consistent error-free deliveries to the Lighting Department,

and having the ability to rapidly fill an entire environment with accurate, render efficient, visually sophisticated materials."

9. CONCLUSION

A new method for generating materials was created in this thesis, which allows the user to create a variety of materials quickly and efficiently while also maintaining flexibility within the tool itself, allowing for full customization. Using real world data, gathered through research, a wide range of realistic materials and masks that accurately depict age and weathering were generated. The aging and weathering interacted with the original materials realistically, but also allowed the artist to modify the age and weathering as needed.

A wide range of unique materials were generated to illustrate the capabilities and efficiency of this method. Additionally this method was integrated into a studio setting to facilitate artist workflow and create streamline efficiency among the departments at the studio level.

10. FUTURE WORK

There are several additional features that could help make surfacing more efficient to work with this workflow. These features would include speeding up the painting process, additional searching capabilities, exporting materials to the tool to be used, and live rendering. These additional features would help round out the process and make it more streamlined.

10.1 Painting

Currently there is no integration with a painting package. In order for the user to edit maps, they would have to open the textures used and would edit them from there. Ideally the tool would allow the user to either export out a live shader network with full network editing to be utilized in Maya as well as give the user the option to export the shader network to Mari and/or Photoshop. This method would help make the painting process by setting up your painting layers in a painting package. The user could directly paint within the painting package without having to worry about importing in the textures used. There are additional benefits to having setups in Mari. The user can maintain a nondestructive workflow, but also export out a baked down shader network. This would make the shader networks extremely more efficient, currently the user could experience inefficient render times with numerous amounts of material layers and masks. There are benefits to having both methods available to the user, so the flexibility to have both in the tool would help the user make their decision with what benefits best suit their needs.

10.2 Search Capabilities

Currently searching for materials is organized, but with a more expanded library searching for materials could become overwhelming to the user. Additional searching methods would help alleviate this potential problem. Tagging the materials with different tag types like material type, material name, and created by the user could potentially filter down what materials they need. Search fields would also help this problem. If the user knew exactly what they needed, they could just type in into a search field what they are looking for and that would filter the results for them. Search engines like Amazon and Zappos cater to making browsing for products as simple as possible, research these engines and how this could be integrated into the tool. Breadcrumbs would also help with filtering materials, the user can know exactly what filters they have and what they can remove at any time.

10.3 Real-Time Rendering

Currently the tool allows for the user to see how their material is being built up in the display widget. This method utilizes compositing techniques. While this makes the interaction relatively fast, it would be ideal to connect this tool to a live renderer or even Mari. This would give us a closer 1 to 1 result by providing the actual model we want to customize as opposed to guessing on a material sphere. This would also allow the user to change lighting setups a lot more simply and see how the material changes with camera changes. This is an essential part to fully understanding how a material reacts.

10.4 Exporting Materials to the Library

Currently in order to add additional materials to the library, a manual process of exporting materials, creating maps, and editing code is involved. An additional

tool to automate the exporting of materials to the library would help the user add additional materials to the library and make the entire material library process a lot more complete. Materials will constantly need to be added based off of art direction and users needs. Providing a simpler method will be necessary to make this process usable for those cases.

BIBLIOGRAPHY

[1] Adobe. Adobe illustrator cc. `http://www.adobe.com/products/illustrator.html`, 2015. [Online; accessed 05-April-2015].

[2] Allegorithmic. Substance painter: the new generation of texture painting. `http://www.allegorithmic.com/products/substance-painter`, 2013. [Online; accessed 09-November-2014].

[3] AutoDesk. Nodes remapcolor. `http://download.autodesk.com/us/maya/2008help/Nodes/remapColor.html`, 2008. [Online; accessed 05-April-2015].

[4] Yukihiro Bandoh, Guoping Qiu, Masahiro Okuda, Scott Daly, Til Aach, and Oscar C. AU. Recent advances in high dynamic range imaging technology. *Intelligent Modeling and Analysis Research Group*, pages 1–4, 2010.

[5] Brent Burley and Walt Disney Animation Studios. Physically-Based Shading at Disney, 2012.

[6] Yao-Xun Chang and Zen-Chung Shih. Physically-based patination for underground objects. *Computer Graphics Forum*, 19:109–117, December 2001.

[7] Yao-Xun Chang and Zen-Chung Shih. The synthesis of rust in seawater. *The Visual Computer*, 19:50–66, January 2003.

[8] Julie Dorsey and Pat Hanrahan. Modeling and rendering of metallic patinas. In *Proceedings of the 23rd annual conference on computer graphics and interactive techniques.*, pages 387–396, 1996.

[9] Julie Dorsey, Holly Rushmeier, and Francois Sillion. *Digital Modeling of Material Appearance*. 193-225, Burlington, Massachusetts, 2007.

[10] David J. Eck. Lighting and materials. `http://math.hws.edu/eck/cs424/notes2013/09_Light_and_Material.html`, 2013. [Online; accessed 05-April-2015].

[11] Source Forge. Pyqt. `http://pyqt.sourceforge.net/`, 2013. [Online; accessed 05-April-2015].

[12] The Foundry. Mari. `https://www.thefoundry.co.uk/products/mari/`, 2015. [Online; accessed 05-April-2015].

[13] Scott Games. Albedo cheat-sheet for physically-based rendering. `http://www.sc0ttgames.com/?p=464`, 2014. [Online; accessed 05-April-2015].

[14] Scott Games. Specular cheat-sheet for physically-based rendering. `http://www.sc0ttgames.com/?p=443`, 2014. [Online; accessed 05-April-2015].

[15] Chaos Group. Vraydirt map parameters. `http://help.chaosgroup.com/vray/help/150SP1/vraydirt_params.htm`, 2012. [Online; accessed 05-April-2015].

[16] Chaos Group. Vraycurvature. `http://docs.chaosgroup.com/display/VRAY3MAYA/VRayCurvature##`, 2014. [Online; accessed 05-April-2015].

[17] Chaos Group. Vraylightdome. `http://docs.chaosgroup.com/display/VRAY3MAYA/VRayLightDome#`, 2014. [Online; accessed 05-April-2015].

[18] Hoppe and Schrenk Marketing GbR. Chaos group v-ray materials. `http://www.vray-materials.de/all_materials.php`, 2011. [Online; accessed 09-November-2014].

[19] Sebastien Lagarde and DONTNOD Entertainment. Feeding a physically based shading model. https://seblagarde.wordpress.com/2011/08/17/feeding-a-physical-based-lighting-mode/, 2011. [Online; accessed 01-April-2015].

[20] Sebastien Lagarde and DONTNOD Entertainment. Dontnod physically based render chart for unreal engine 4. https://seblagarde.wordpress.com/2014/04/14/dontnod-physically-based-rendering-chart-for-unreal-engine-4/, 2014. [Online; accessed 01-April-2015].

[21] Sebastien Lagarde, Laurent Harduin, and DONTNOD Entertainment. The art and rendering of remember me. Game Developers Conference Europe 2013, 2013.

[22] The Foundry Visionmongers Limited. Udim workflow. http://modo.docs.thefoundry.co.uk/modo/801/help/pages/modotoolbox/UDIM_Workflow.html, 2001. [Online; accessed 05-April-2015].

[23] Jon McCormack. Interactive evolution of l-system grammars for computer graphics modelling. *Complex Systems: From Biology to Computation*, pages 118–130, 1993.

[24] David Neubelt, Matt Pettineo, Nathan Phail-Liff, and Ready at Dawn. Crafting a next-gen material pipeline for the order: 1886. Siggraph, 2013.

[25] Weathervanes of Maine. Information and facts about copper. http://www.weathervanesofmaine.com/landing/information-facts-about-copper/, 1997. [Online; accessed 05-April-2015].

[26] Eric Paquette, Pierre Poulin, and George Drettakis. Surface aging by impacts. *Proceedings of graphics interace*, pages 175–182, June 2001.

[27] Pixar. Slim - the shader tool. `http://renderman.pixar.com/view/slim-shader-tool`, 2015. [Online; accessed 05-April-2015].

[28] Pythonware. Python imaging library pil. `http:/www.pythonware.com/products/pil/`, 2009. [Online; accessed 05-April-2015].

[29] Zappos. Zappos. `http://about.zappos.com/`, 2014. [Online; accessed 05-April-2015].