SME Annual Meeting
Feb. 24 - 27, 2018, Denver, CO

Preprint 19-052

MINING ASSET DEVELOPMENT FOR VIRTUAL REALITY

T. J. Orr, CDC NIOSH, Pittsburgh, PA
J. L. Bellanca, CDC NIOSH, Pittsburgh, PA
J. Navoyski, CDC NIOSH, Pittsburgh, PA
B. Macdonald, CDC NIOSH, Pittsburgh, PA
W. J. Helfrich, CDC NIOSH, Pittsburgh, PA
B. Demich, CDC NIOSH, Pittsburgh, PA

DISCLAIMER

The findings and conclusions in this paper are those of the authors and do not necessarily represent the official position of the National Institute for Occupational Safety and Health, Centers for Disease Control and Prevention. Mention of company names or products does not constitute endorsement by NIOSH.

ABSTRACT

The pervasiveness of high-fidelity video games raises expectations of quality, detail, and lighting in virtual reality (VR) applications. However, high-quality models of mining equipment and environments are often not readily available, and developers may not be familiar with the latest techniques, such as physically based rendering (PBR). Because a consistent aesthetic is critical for maintaining immersion and controlling visual stimuli in research as well as in training, researchers at the National Institute for Occupational Safety and Health (NIOSH) created an asset development workflow. This paper details this workflow, including precision modeling and PBR techniques as well as a rigorous review process. It also describes lessons learned in the creation of photorealistic VR assets.

BACKGROUND

Virtual reality (VR) describes the simulated experience generated by an assemblage of hardware and software that are used to immerse the user. Over many years VR developers targeted the development of virtual environments (VE) toward various purposes: entertainment, the treatment of phobias [1], rehabilitation [2], and training [3]. A large part of the VE development includes digital assets, where digital assets refer to the non-programmed elements including 3D models, audio, and user interface. Because vision provides nearly 10Mbps of data [4] and accounts for the majority of all sensory input combined [5], a focused effort on asset appearance in VEs is warranted. Furthermore, research has shown that higher-resolution imagery increases search performance and training transfer, necessitating high-fidelity assets [6, 7]. In addition to being visually appealing, it is important that the assets and environment are also dimensionally accurate and consistent in order to elicit appropriate responses from the end users as they interact with and among these items [8, 9].

Researchers at the National Institute for Occupational Safety and Health (NIOSH) have developed VEs and digital assets under several research projects over the course of the last two decades [10, 11, 12, 13]. This work has provided researchers with a strong knowledge base for digital asset development. However, the assets themselves have had limited reuse value. Due to the extended timeframes and limited scope of these projects, the previously developed assets are not consistent in visual quality and dimensional accuracy. Moving forward, researchers wanted to minimize future development time and use of resources as well as adopt new asset development tools and game engine technology. Therefore, the aim of this paper is to document a new asset development workflow including geometry, materials, UV layout, and shader implementation combined with review steps in order to produce high-fidelity simulation assets that are accurate with consistent characteristics, using repeatable methodologies that are well documented.

INTRODUCTION

Workflow

A workflow is a formalization of the steps required to move a concept from ideation to realization. Each process step encapsulates the input and output required to flow along a directed path. Steps for asset development include the primary tasks, review, and documentation. Reviews help catch errors before additional effort is expended on downstream processes. Review and documentation also help provide metrics for maintaining consistency, particularly between various artists. Additionally, formalization of a workflow affords the opportunity to identify tasks that can be parallelized to improve efficiency. Minimally, an asset development workflow for a VR simulation must include the development of basic geometry, the development of materials, the creation of UV layouts, and the configuration of the rendering engine. A description of each of these steps and their importance to the ultimate fidelity of the simulation is described in the sections below.

Geometry

The geometry of a digital asset describes the underlying mesh of polygons and vertices that makes up the basic shape and dimensions of an object. The geometric representation of an object can be realized through a number of methods such as computer-aided design (CAD), laser scanning, photogrammetry, or box modeling. While starting out with detailed digitalizations of the equipment to be modeled might seem ideal, these formats each come with challenges that can require significant effort to format, simplify, and optimize for real-time rendering in VR. CAD models often do not include small details like welds or small fasteners that might seem trivial, but will be noticeably absent to system users. Additionally, because large mining equipment is often customized, as-built measurements of that machine may be required, depending on the requirements of the simulation. Even in the case of laser scans or photogrammetry, verification of model dimensions requires some manual measurements of the physical object. Furthermore, laser scans and photogrammetry methods produce holes or discontinuities when one part of the geometry obscures another. Lastly, all of these methods require some form of optimization because the total number of polygons on screen affects the rate at which objects can be rendered. In order to achieve real-time rendering at high framerates, polygons should be minimized. Building geometry from scratch—box modeling—affords the most flexibility in quantity, scale, and UV layout of mesh detail for a model of an object.

Regardless of the method of geometry creation, to further optimize rendering speeds, lower polygon models can be substituted when an object is farther away from the rendering camera. Level of detail (LOD) models can be generated using the mesh optimization tools with minimal artist manipulation. Mesh optimization software tools further refine a mesh by systematically reducing vertices while maintaining the critical features of the original model.

Copyright © 2019 by SME

Material

Materials are a collection of 2D texture bitmaps—textures—that control the way light interacts with a surface. Early computational rendering work used measurements of light intensity at various angles of incidence and reflectance to produce mathematical models that describe reflection and energy conservation termed physically-based rendering (PBR) [14, 15]. PBR algorithms require that the total light energy coming off a surface cannot be greater than the incident light and that light behaves with reciprocity, adding to scene realism by accounting for light bounce from object to object. Overall, the intention of PBR is to create synthetic images that are indistinguishable from actual ones.

Practically, PBR techniques provide consistent and predictable material behavior that is independent of the lighting conditions under the application of a single rendering engine. This also means that several artists can create content for a simulation, and by adhering to common color and reflectivity principles they can produce comparable assets. These principles are realized by standard textures applied to a material based on the prospective rendering engine. Some example textures include diffuse, normal, reflectivity, metallic, and roughness maps. Base properties for common materials are available for most of the popular game engines as libraries included in commercial software. Additionally, software tools can be integrated into some game engines like Unity and Unreal to further streamline workflows and provide non-artists the ability to tweak material characteristics.

UV Layout

The creation of a UV layout of for a digital asset describes the process of applying 2D materials to a 3D geometric mesh. The U and V refer to the x and y axes in the material coordinate space. UV denotes the plane typically used for mapping these materials. Materials are mapped to the geometry so that the desired properties are represented at the intended location. The process of making this transformation from 2D to 3D space is referred to as UV mapping. Mapping 2D materials onto complex 3D geometry directly is challenging. Therefore, polygons comprising the 3D geometry can be flattened out into 2D space in a process called unwrapping. The artist can then resize regions of the unwrapped mesh to control how much of the material is mapped to each mesh region in order to change the local texture resolution—texel density. Artists can also change the alignment of the mesh with a material to add edge padding in order to avoid mipmapping induced artifacts. The mesh can also be divided among different materials to improve texture resolution. However, artists should try to minimize the number of materials assigned to each mesh, because this can reduce rendering speed. There are various software packages that include tools designed to make UV mapping easier.

Shader

Previously, PBR methods had been used to render photo-realistic computer-generated imagery off-line for movies and animations [15]. However, advances in hardware graphics processing and ray tracing algorithms, now allow these sophisticated techniques to improve visual realism in real-time game engines like Unity and Unreal with the use of PBR shaders.

Shaders are software components embedded in a game engine's rendering pipeline that calculate the color of each pixel based on the material and lighting. They are the last component in a real-time rendering pipeline. Because PBR shaders vary in their mathematical modeling of light interpretation, they require different material parameters and textures to implement [16]. PBR shaders vary from engine to engine and different PBR shaders can even be used within an engine to support stylized visual effects (e.g. cartoon shading).

ASSET DEVELOPMENT

The following sections outline how NIOSH researchers have implemented an asset development workflow in order to develop more consistent, high-fidelity mining assets.

Workflow

The NIOSH asset development team workflow begins with a customer request for a simulation asset. The team and customer create a rough draft of the specifications document at the initial customer meeting. The specifications document includes key information about the requested model such as purpose, geometric detail, dimensional accuracy, animation, rigging, kinematic constraints, and material requirements. The customer and team will also agree upon what source material will be used for model development (e.g. CAD drawings, photographs, access to the equipment to be modeled) and who is responsible for providing this information. Source material acquisition can often be time intensive, but it is critical, as it is the backbone of development. Lastly, the team and customer must agree upon a delivery date based on the required features and level of fidelity.

After this initial meeting, the asset team refines the request to produce an asset concept. The concept may contain marked-up sketches, photographs, or prototype 3D models and descriptive text to document the customer expectations. This touch-point is important to formalize the acceptance criteria for this asset. Scope creep and mismatches in expectations have led to serious delays in past projects. One source for these shifting expectations can come when a secondary customer is identified who has additional constraints or use cases for the asset under development. Identifying all customers at the outset of each asset effort can minimize such impacts. Overall, effective, frequent communication is key, particularly when moving through the initial steps of the workflow.

Team communication is also vital for success as much of the asset development can be done in parallel. Figure 1 depicts the generic asset development workflow, showing how geometry and material development can be completed in parallel. Depending on the project, the workflow may be more or less parallelized. While efficiency is important, it is also critical that the team has good understanding as to how the components fit together and are interdependent. Early in the adoption of this workflow, team members tried to work ahead, rush through reviews, or skip reviews entirely. Errors discovered downstream led to greater inefficiencies. For example, if an error in the geometry was not identified until the final asset review, it could require additional changes to the materials as well as remapping of the UV layout.

Similarly, the development team also discovered that prototyping solutions all the way through the workflow to see the final effect in the engine shader was helpful. Because material properties differ based on the rendering engine, rendered objects in the development software do not look the same as in the game engine. Prototyping can make the final asset development proceed more efficiently with fewer errors overall. For example, the team created prototype instances with various options depicting welds using modeled geometry, with only material normal maps, and a combination of both (Figure 2). If the wrong method had been carried all the way through, the UV layout for the entire machine would have had to be redone, which can be especially time consuming with the amount of manual modifications made to the normal maps around the welds. More time for experimentation and testing as a formal step in the workflow is recommended for future asset development.

Documentation also plays a pivotal role for team communication about expectations, responsibilities, and task scheduling. A team wiki was used to provide a central location for this information and proved to be a valuable resource by providing a living document for reference as assets were added to the library. Providing detail about each step of the process was important. If changes to an asset are required, it is critical to be able to go back into the early steps of the workflow and understand the design choices that were made previously, possibly by a different artist. To assist with these cases, the documentation included information regarding software tools, key settings, and process walkthroughs. The detailed documentation helped maintain a more complete shared knowledge base; this can be especially critical for project continuity if team members rotate or leave the project.

Geometry

As work on the geometry of an asset begins, decisions about required accuracy are driven by intended purpose of the model in the simulation that should have been identified in the requirements. For

Copyright © 2019 by SME

generating engineering precision models, CAD files from the product manufacturer can save a significant amount of time and effort. Dimensioned, orthographic views (top, side, and front/rear) may also be available. As depicted in Figure 3, such drawings can be used for development or as verification of accuracy and scale. Heavy equipment brochures also often include tabular data on parameters, such as wheel base, tire type, and minimum or maximum values, for key features that are invaluable to the artist. Lastly, if access to the equipment is available, direct measurement of key dimensions is advisable. This is true even if CAD data is readily available, especially for high-cost equipment that may undergo significant customization either during manufacture or at some point during the service life of the equipment. If the goal is to simulate the equipment as it exists in the real world, an idealized version will not suffice.

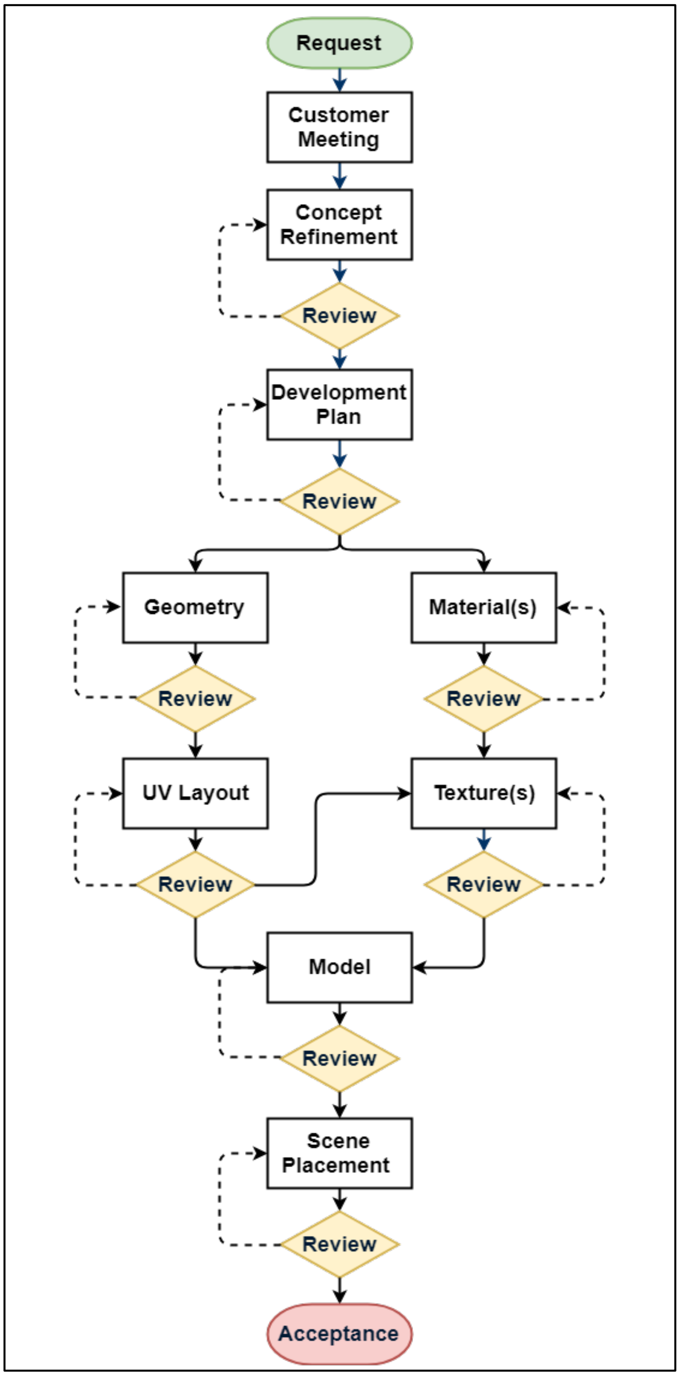


Figure 1. Flowchart depicting NIOSH's generic asset development workflow.

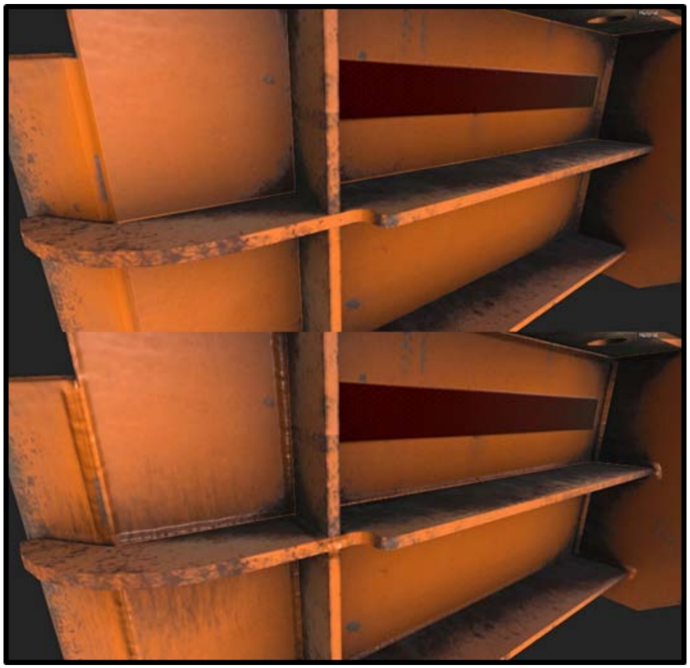


Figure 2. Detail view of heavy equipment showing seams between plates with and without welds.

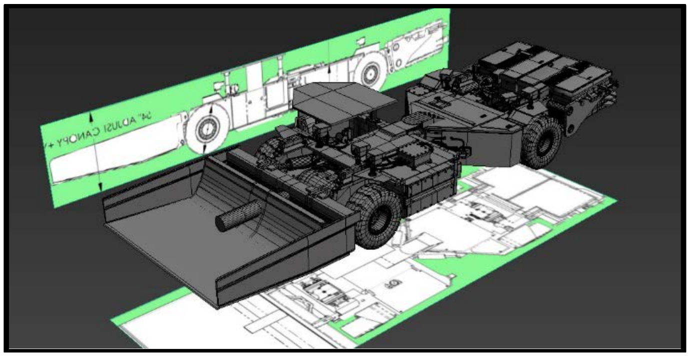


Figure 3. A view from the geometric development software, depicting the use of orthographic drawings of the side and top views that have been scaled and positioned around the model to be used to verify the accuracy of the dimensions.

Laser scanning and photogrammetry can be useful when a high level of precision is required, but CAD files or drawings are not available and direct measurement is difficult or risky. These methods can be particularly useful for as-built applications. For example, in order to correctly calculate a point of regard on a curved 360-degree screen, an exact mesh was required. The screen was fabric stretched over a metal structure resulting in an irregular shape, but it was unable to be touched, and therefore could not be physically measured. Figure 4 shows the progression from the actual screen, to a wireframe, and finally to the polygon mesh developed using photogrammetric methods.

For some of the assets, the team uses standard box-modeling techniques. Creating a model from base shapes, extrusions, and other vertex and edge manipulations allows for better control and organization of polygons. This also ensures a clean topology, meaning there are no unintended discontinuities or overlapping polygons that may have been created through import or mesh optimization. A good topology aids in the ease of editing when performing overall mesh optimization, LOD optimization, and UV layout tasks.

Regardless of the creation method, LOD models will need to be generated to be used in a VE. It is important to remember that features included in each level will depend on the use case and implementation. Communication with the customer upfront can prevent costly mistakes.

Copyright © 2019 by SME

For example, when developing LODs for tires on a machine, treads may or may not be included in a medium LOD model (Figure 5).

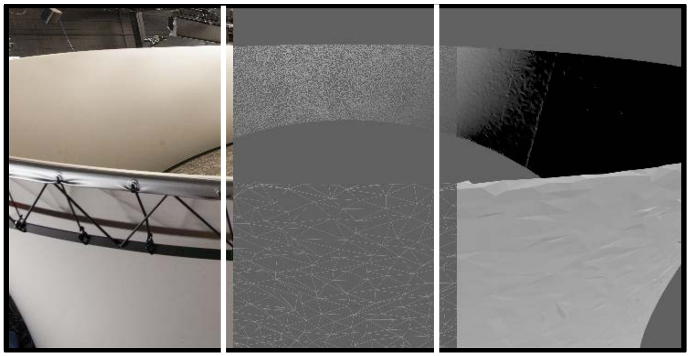


Figure 4. The progression of the creation of a photogrammetric model of a 360-degree VR theater screen showing, from left to right, the actual screen, wireframe, and polygon mesh.

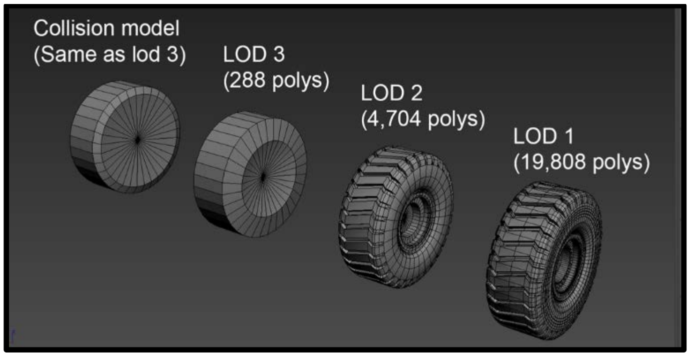


Figure 5. Level of detail and collision meshes for a mining vehicle tire.

When developing geometry, it is also critical to examine how it all fits together in 3D. One of the main assets developed by the team was a set of tileable mine environment meshes used to generate the underground mine. Tile edges were modeled to have coincident vertices so that the tile edges would appear seamless. But, as the meshes were separated in the development software—3DS Studio Max—the vertex normal values on either side of the split were automatically recalculated, causing a visible discontinuity at the boundary in some lighting conditions in the game engine. Because this issue was not discovered during the geometry review step, it required significant additional effort to identify and correct the issue, test the solution, and then rework the geometry, materials, and UV layouts of assets.

Materials

The team uses a variety of sources and tools to develop materials including Allegorithmic's Substance suite. The software license includes access to a vast library of pre-made materials that require little modification for application in simulating industrial materials. Additional materials are also readily available at no cost from several online material libraries. Realistic objects can be built up from blending several materials and adding additional non-procedural layer effects. For example, the general material developed for a scoop was created by layering rust and wear on a base material of painted metal as depicted in Figure 6.

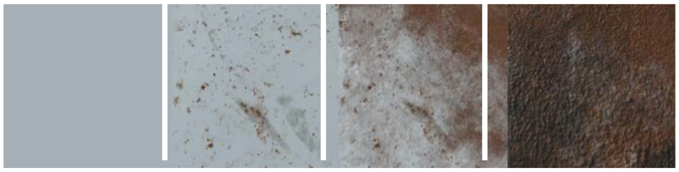


Figure 6. Four stages of rust and wear on a painted metal surface created by the artist by applying various material layers in Substance Painter software.

In addition to material libraries, new materials can be created from base photographs. Substance's B2M (bitmap to material) software helps the artist create tileable materials from photographs. Basic parameters can be changed to generate albedo, normal, metallic, roughness, and ambient occlusion texture maps, where diffuse, normal, and specular/metallic maps are the minimum for a basic PBR material. As an example, the team used the B2M tool to create a freshly cut coal material for the exposed coal in an underground mine VE. Starting from a photograph, it was necessary to adjust the color values on the albedo map so that the values fell within the recommended PBR range. In this case, the image needed to be brightened, as the minimum RGB color values for coal are 50, 50, 50, and the original photo pixel values were in the single digits. The other texture maps were then automatically generated from the edited image (Figure 7).

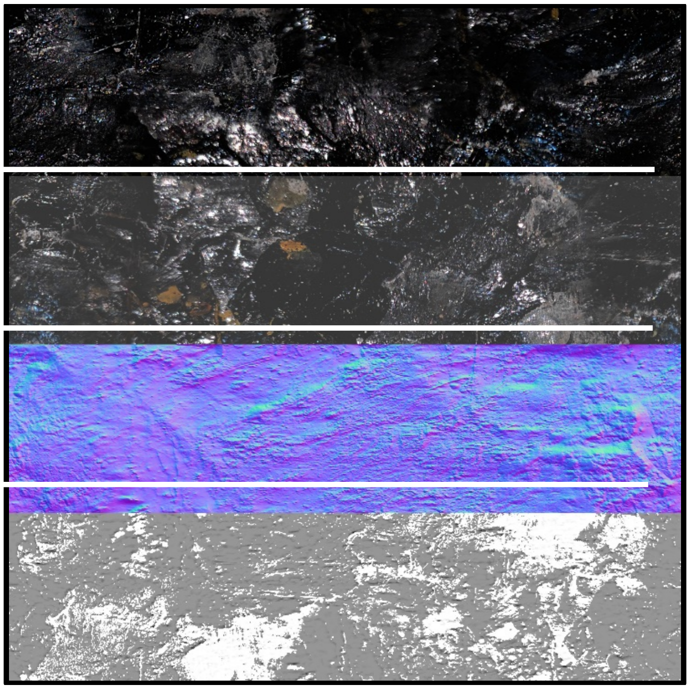


Figure 7. Material texture maps developed from a photograph of coal using Substance B2M software. From top to bottom: original albedo map, color flattened albedo map, normal map, and metallic map.

UV Layout

As discussed earlier, UV mapping should be done in order to achieve optimal texel density. Mapping should efficiently allocate the texture space to use more pixels in areas where detail is required, such as vehicle controls or gauges, and use less in areas where detail is not required, such as the canopy of the cab. However, care should be taken to not create a stark contrast in texel density between adjacent geometry, which could result in a distinct discontinuity. Figure 8 depicts an example of a UV mapping on a mantrip using a uniform checkerboard texture in order to better visualize the texel density of each surface.

Part of optimizing a UV layout includes accounting for the texture space required for padding around the edges to avoid mipmapping artifacts. These artifacts occur when background pixels of a texture bleed over the boundary as the image is rescaled for efficient rendering. As a general rule of thumb, one pixel of padding should be given for every 128 pixels of texture resolution. For an 8K texture, that means that 64 pixels of padding are needed. Figure 9 displays an albedo texture that was created for a section of an underground coal mine, where the striped areas of the images are the padding regions.

Mipmapping errors are also a good example of why prototyping can be important for complex or unfamiliar methods. In this case, the mine environment asset was being created to be viewed both as a first person player and in a zoomed-out overhead configuration. The

Copyright © 2019 by SME

overhead view was not included in the review. Therefore, the padding error was not identified until the final implementation. In this case, all the geometry, materials, and UV mapping had to be redone to fix the rendering artifact.

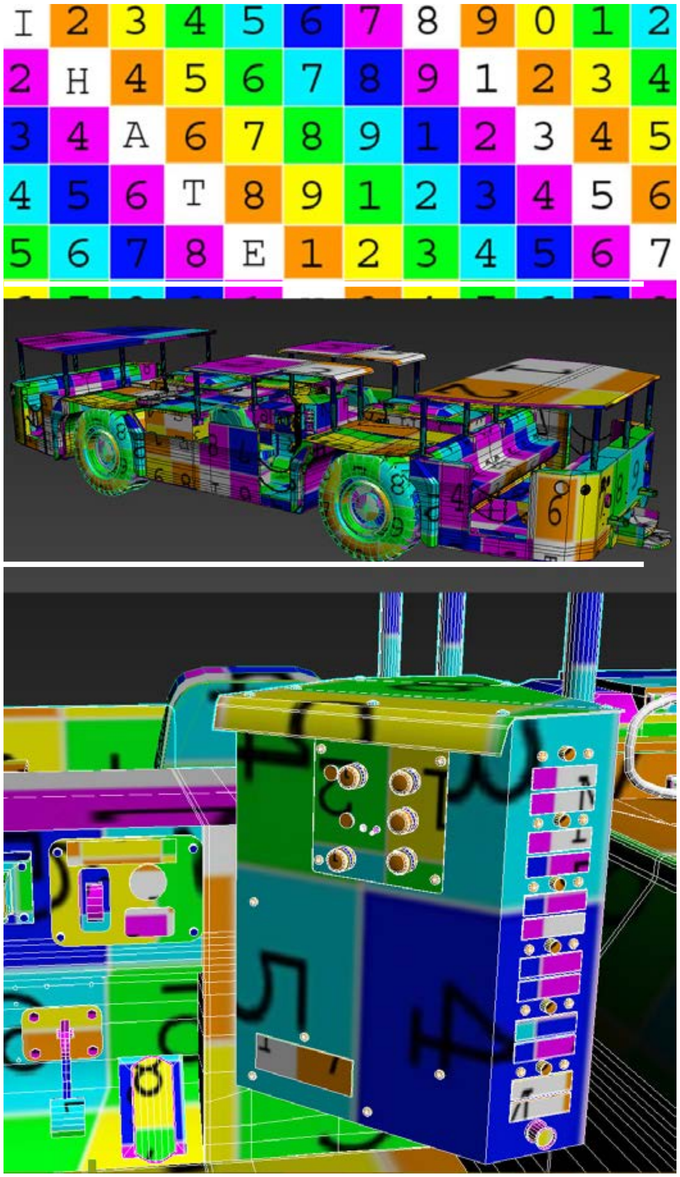


Figure 8. Screenshots from a UV mapping tool, where the test pattern pictured at the top is applied to a mantrip model such that the canopies use less pixels and the controls in the cab use more.

Lastly, the UV layout of a model should be created from the highest LOD. Though the geometry development generally needs to be completed before any unwrapping can be done, the UV layout should be completed before creating the additional lower LODs. By developing the UV layout first, this allows the lower LODs to inherit the UV layout of the higher LOD. Furthermore, it is important to inherit from the highest LOD because it includes the detailed geometry and smaller sub-components such as welds, bolts on tires, and cab controls on vehicles that would not otherwise be mapped.

Shader

Shaders and the lighting pipeline of a game engine are the last direct line of control for visualizing an asset in a VE. The NIOSH development team uses Unity's standard shader for most of the developed assets. The standard shader by default includes a metallic map to control areas of the material that represent exposed metal. With the metallic setup, the color and intensity of specular highlights (direct reflections of light) are generated in a more naturalist manner based on the underlying albedo texture and reflections from the environment. For efficiency, the metallic option uses the alpha channel of the metallic map to modify the overall smoothness of the material. Another configuration of this shader is "specular option." This configuration adds the functionality to directly control the color and intensity of specular highlights (e.g., the perceived glint on objects). While this configuration is still considered a PBR shader, in that it respects conservation of energy principles, because you are manually manipulating the specular highlights, it can result in unrealistic materials. Therefore, this option was not used by the NIOSH development team.

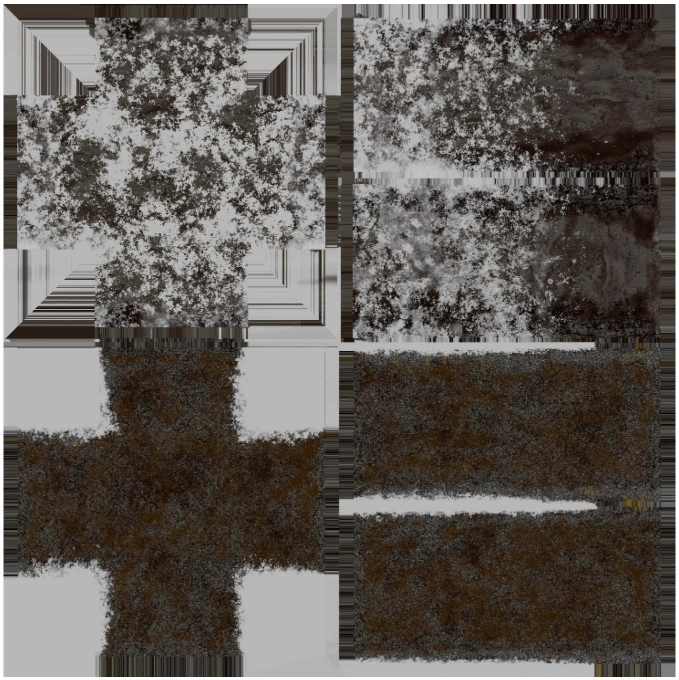


Figure 9. Albedo texture map for a tileable underground coal room-and-pillar intersection model depicting edge padding to eliminate artifacts introduced during mipmapping.

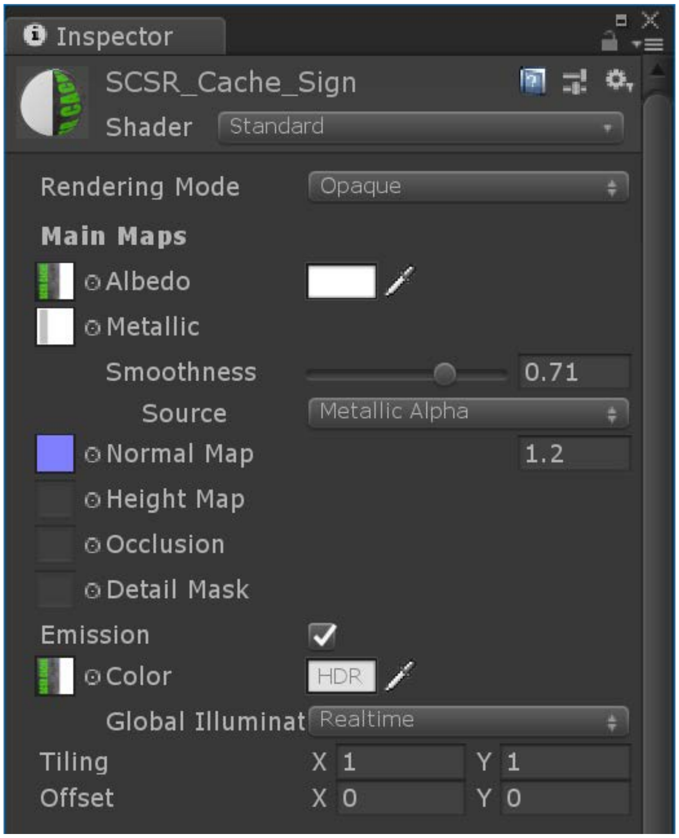
It is also possible to the same textures reused by different materials or have materials that use the same set of textures but have different parameter settings. Exemplifying some of these parameters, Figure 10 displays the user interface for Unity's standard shader in the "standard" configuration. In this case there is a multiplier on the normal texture map—currently set to 1.2—that can change the perceived depth of geometry that is modeled as a texture (e.g., wrinkles or folds).

Overall, shaders help the developer to fine-tune the appearance of assets in the VE. Because of this, it may be tempting to use shader parameters to compensate for poorly constructed materials (e.g., increasing the intensity for a normal map). However, such errors should be returned to the artist and corrected because this limits the flexibility and utility moving forward. For example, if a normal map parameter is decreased because the roughness does not correctly mimic scratches, the shader can no longer be used to modulate wear in the VE.

CONCLUSIONS

Virtual reality (VR) investment has increased rapidly in recent years, and the VR market could exceed 117 billion US dollars by 2020 [17]. The mining industry stands to benefit from this growth as virtual reality (VR) solutions (hardware, content, and development tools) become widespread. Asset development methods for VR must evolve as continual improvement of VR technologies drives the fidelity of these systems closer to reality (Figure 11). As simulations become more convincing, it is more important than ever to ensure the underlying validity of these visualizations.

Copyright © 2019 by SME



Inspector
SCSR_Cache_Sign
Shader: Standard
Rendering Mode: Opaque
Main Maps
Albedo
Metallic
Smoothness: 0.71
Source: Metallic Alpha
Normal Map: 1.2
Height Map
Occlusion
Detail Mask
Emission: ☑
Color: HDR
Global Illuminat: Realtime
Tiling: X 1 Y 1
Offset: X 0 Y 0

Figure 10. A screenshot of the Unity Inspector window, illustrating how individual texture maps are assigned to the material channels in the standard shader.

Figure 11. An example of the realism provided by physically based rendering as demonstrated by the rendering of a cab of a battery hauler.

Adherence to structured workflows can be helpful in achieving this goal, especially for part-time or non-technical developers as they continue to use PBR and other advanced techniques. Focus on communication, including documentation, ensures that customer needs are addressed throughout the process. Attention to detail with frequent review steps provides the ability for teams to self-correct before effort is wasted. Prototyping and experimentation are valuable for team members to try new techniques, learn new skills, and manage customer expectations. Overall, a well thought out and executed asset development plan can significantly improve the quality and consistency of virtual environment (VE) digital assets.

REFERENCES

1. Parsons TD, Rizzo AA (2008). “Affective outcomes of virtual reality exposure therapy for anxiety and specific phobias: A meta-analysis.” Journal of behavior therapy and experimental psychiatry 1:39(3): 250-61.
2. Parsons TD, Riva G, Parsons S, Mantovani F, Newbutt N, Lin L, Venturini E, Hall T (2017). “Virtual reality in pediatric psychology.” Pediatrics 1:140 (Supplement 2): S86-91.
3. Seymour NE, Gallagher AG, Roman SA, O’brien MK, Bansal VK, Andersen DK, Satava RM (2002). “Virtual reality training improves operating room performance: results of a randomized, double-blinded study.” Annals of surgery. 236(4): 458.
4. Koch K, McLean J, Segev R, Freed MA, Berry II MJ, Balasubramanian V, Sterling P (2006). “How much the eye tells the brain.” Current Biology, 16(14): 1428-1434.
5. Rosenblum LD (2011). “See what I'm saying: The extraordinary powers of our five senses.” WW Norton & Company.
6. Ni T, Bowman D, Chen J (2006). “Increased display size and resolution improve task performance in information-rich virtual environments.” In: Proceedings of Graphics Interface 139-146.
7. Lee C, Rincon GA, Meyer G, Höllerer T, Bowman DA (2013). “The effects of visual realism on search tasks in mixed reality simulation.” IEEE transactions on visualization and computer graphics 19(4): 547-56.
8. Sargent RG (2013). “Verification and validation of simulation models.” Journal of simulation 7(1): 12-24.
9. Schofield, D (2018). “Guidelines for using game technology as educational tools.” technology, 4: 31.
10. Orr, TJ, Filigenzi, MT, Ruff, TM (1999). ”Hazard recognition computer based simulation.” In: Proceedings of the Thirtieth Annual Institute on Mining Health, Safety and Research 21-28.
11. Navoyski J, Brnich Jr MJ, Baurle T (2015). “BG 4 benching training software for mine rescue teams.” Coal Age
12. NIOSH (2009). “Underground coal mine map reading training.” Retrieved Oct 27, 2018, from https://www.cdc.gov/niosh/mining/works/coversheet1825.html.
13. Orr TJ (2016). “NIOSH mine emergency escape simulation technology available for developers.” NIOSH Science Blog. Retrieved October 27, 2018, from https://blogs.cdc.gov/niosh-science-blog/2016/05/12/mine-escape-simulation/.
14. Greenberg DP, Torrance KE, Shirley P, Arvo J, Lafortune E, Ferwerda JA, Foo SC (1997). “A framework for realistic image synthesis.” In Proceedings of the 24th annual conference on Computer graphics and interactive techniques 477-494.
15. Pharr M, Jakob W, Humphreys G, (2016) “Physically Based Rendering: From Theory to Implementation.” Morgan Kaufmann.
16. Friston, S, Fan C, Doboš J, Scully T, Steed A (2017). “3DRepo4Unity: dynamic loading of version-controlled 3D assets into the unity game engine.” In: Proceedings of the 22nd International Conference on 3D Web Technology 15.
17. MarketWatch. (2018). “Virtual and augmented reality market is expected to exceed US$ 117 billion by 2022.” Retrieved Oct 27, 2018, from https://www.marketwatch.com/press-release/virtual-and-augmented-reality-market-is-expected-to-exceed-us-117-billion-by-2022-2018-07-26.

Copyright © 2019 by SME