Ale Neppius

3D GAME TEXTURING:
Comparative Analysis Between Hand-painted and PBR pipelines

Bachelor's thesis
Degree programme in Game Design

2022

South-Eastern Finland
University of Applied Sciences

Author (authors)	**Degree title**	**Time**
Ale Neppius	Bachelor of Culture and Arts	May 2022

Thesis title	
3D Game Texturing: Comparative Analysis Between Hand-painted and PBR pipelines	40 pages 8 pages of appendices

Commissioned by

South-Eastern Finland University of Applied Sciences

Supervisor

Brenda Jiménez

Abstract

Textures are 2D images projected on 3D models. Throughout the years many pipelines and workflows have been invented to create the appealing game art and further develop the technology of textures. Thus, game texturing has evolved into different pipelines representing different aesthetics and needs.

This thesis aimed to understand the similarities and differences between the hand-painted diffuse only texturing pipeline and the PBR pipeline. To conduct the study the theory of texturing was researched through a literature review and the results were further studied by conducting a case study.

First, this paper explained texturing, a brief history, and how optimization plays a big role in game textures. As well development blogs and interview articles from established studios were analyzed.

The theory was then tested by texturing both hand-painted and PBR pipelines on the same asset set to further see the difference in both workflows. Texturing was done utilizing programs called Substance Painter, Substance Designer and Photoshop.

Lastly, the results were analyzed based on the literature review and case study and conclusions were made as a comparative analysis.

Keywords

3D, game design, textures, PBR, non-PBR, hand-painted, texture maps, substance painted, substance designer

CONTENTS

GLOSSARY

2D – Two dimensional

3D – Three dimensional

Hand-painted – Texture detail painted by hand

PBR – Physically Based Rendering

Procedural texturing – Computer generated texturing

Node-based texturing – Textured by using nodes graphs

Polygon – Basic geometry shape of a 3D model

Mesh – 3D object

Asset – 3D object

Texture – 2D image projected on 3D mesh

UV – 2D image wrapped around a 3D mesh

Shader – a computer program that calculates rendering information

Baking – Saving information about a 3D mesh into a texture file

Mesh Maps – Substance Painter specific term for detail maps

Texture Maps – Contains the material attributes, light information, and texture

1 INTRODUCTION

This thesis aims to further enhance the author’s knowledge of different texturing pipelines, mainly concentrating on diffuse only hand-painted pipeline and procedurally generated PBR pipeline, as well as to have a wider understanding of the features different pipelines bring in the creative workflow. The purpose of this thesis is to compare hand-painted and PBR texturing methods with each other and to analyze the similarities and differences between these texturing workflows and how both could be utilized most professionally.

This thesis consists of a research and literature review from various interviews, articles, and talks regarding how to tackle the texturing process and determine the most efficient way to work. In addition, some games are analyzed art wise and research on how some studios built their textures is conducted. This information mainly came from interview articles from industry professionals and development blogs. The knowledge will be then tested by a case study of the author’s two texturing projects.

Most of the texturing will be conducted using Substance Painter and Substance Designer with little help using Photoshop. These assets were designed based on six different materials to better study the difference between hand-painted pipeline and procedural PBR pipeline.

Lastly, this paper explains the results with a comparative analysis between the hand-painted diffuse only texturing pipeline and the PBR pipeline. Conclusions are then made, and results are analyzed with hypotheses and general beliefs against the results from case studies.

This paper is aimed at students who want to better understand different texturing pipelines and how they differ from each other. The purpose of this paper is to summarize otherwise vast and misleading information.

2 GAME TEXTURE BASICS

Textures are 2D graphics laid onto 3D models to create an illusion of material and surface detail (Bogos 2008). They as well can determine the surface height and how the asset reacts to light. Textures have three main properties: material, light effect, and tertiary detail (Adib & Naghdi 2020). The material is the visible texture that gives the 3D object the general color and surface attributes. Light effects tell the renderer how the object reacts to light and shows how rough or metallic the asset is. Tertiary detail is there to give the illusion of fine details like bumps, scratches, and wrinkles on a lower polycount object. (Adib & Naghdi 2020) As seen in figure 1, the three different properties were tested on a simple Substance sample sphere.

Figure 1. Three main properties of a texture (Neppius 2022)

Nowadays, there are two ways to render texture maps: PBR and non-PBR. PBR stands for Physically Based Rendering, which is a modern, standardized, rendering model that simulates realistic light and interaction with the surfaces (Burda 2017). Non-PBR is essentially the old method of rendering textures. What makes the Physically Based Rendering system different from the former ones is more detailed reasoning about the behavior of light and surfaces (Russel n.d.).

There are also three current workflows for creating textures: hand-painted workflow, photo-sourced workflow, and procedurally generated node-based texture workflow. Hand-painting is a method of creating textures painted by hand. Usually, this texturing method has visible brush strokes and is associated with a specific more stylized style, but it can also look visually realistic. Photo-sourced textures are made from real-life photographs and edited to fit the game style and meshes. Before the node-based texturing pipeline photo-sourcing was the common standard for creating realistic textures. Lastly, there is procedural node-based texturing (Figure 2). Procedural texturing is the more technical and modern way of creating textures. These textures are computer generated using fix set of parameters and settings (Denham n.d.).

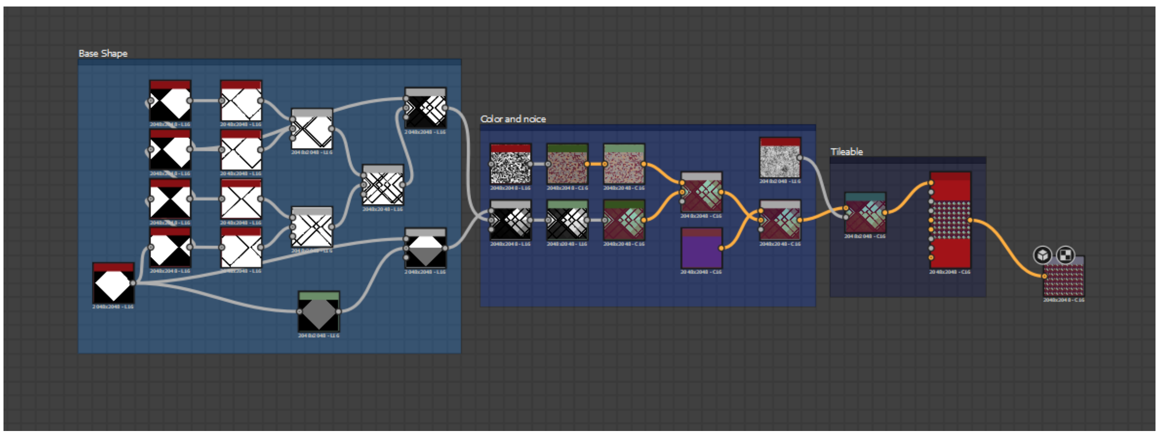


Figure 2. Substance Designer node tree graph (Neppius 2022)

2.1 Brief History of Texturing

In the early days of the game graphics textures had to be relatively simple. Due to the lack of computing power in consoles, the game graphics could not be as complicated as they are now. Texture artists had to come up with creative shortcuts to maintain the desired look. Some of the methods are used today even though game resolution and rendering have become much more powerful. For example, ambient occlusion maps are still used as a way of having more depth even though realistic PBR lighting is commonly used.

In his Twitter thread, Ryan Benno (2019) explained that many elements were pre-rendered to maintain better quality. Benno is currently the lead artist at game studio *Gardens* with many years of experience in game arts. His Twitter thread has been cited in many articles. He explains that games are inherently interactive and require real-time rendering. Though some static parts like the environment were pre-rendered (Benno 2019). Benno also explains that textures and lighting, during the 90s, were painted on models without proper real-time shadows.

During the second console generation in the early 2000s, only diffuse maps were used, but a lot of information was baked in the texture and even real-time lighting was able to be done. Later specular and normal maps came into the scene. First features in games like *Halo 2* (2004) and *Doom 3* (2004). With these texture maps, a lot more details could be stored in the textures and could react to light more realistically (Foundry 2022).

Nowadays PBR has become standard for the current generation of games. Even though computers are more powerful and real-time rendering has become more viable, game artists still use many old techniques to this day (Benno 2019). What started as a necessity nowadays is more of a stylistic choice.

2.2 General Workflow

One of the most important aspects of a good texturing workflow is speed and efficiency (Yang 2020). As the game production deadlines can get tight it is important to identify the level of needed detail on objects. Is the player going to drive fast past the texture, or will the asset be analysed closely? Thus, a non-destructible and well-optimized workflow is important to be able to make fast changes on the go without having to start everything over or make major fixes due to bad planning. Good pre-planning and building informative guidelines are important for an efficient workflow (Ahearn 2009).

The workflow differs a bit if it is for a game production or a personal game-ready asset. Big production, especially for environment art, tends to think of the big picture and how to create a lot of variations in a short amount of time while still having a unique artistic feel to it. Smaller projects, which usually are just one scene or a prop, concentrate more on a specific goal in mind and are done to showcase skill or to learn and test new pipelines. These projects are mostly made for portfolios (Figure 3).

Figure 3. Example of a personal project, the Toro & Spirits scene (Shedu 2020)

Whichever production is in question, the basic workflow starts from the research phase where the artist determines goals and gathers references for the project. Usually, in this phase, the pipelines, programs, and deadlines get determined as well. As the texture artist from *Deathloop* (2021) JB Ferder explained the workflow is good to keep clean and non-destructible for later adjustments as the project and its style develops (Ferder 2021).

Next artists will UV unwrap meshes, bake mesh maps, and prep the model for the actual texturing. The UV map is a 2D image wrapped around a 3D mesh (Figure 4). The U and V refer to the horizontal and vertical axes of the 2D space, as X, Y, and Z are already being used in the 3D space (Denham n.d.). Though with some processes such as tiled textures and foliage, the UV unwrap phase comes after the textures are done due to needing to adjust the UV layout according to the texture map.

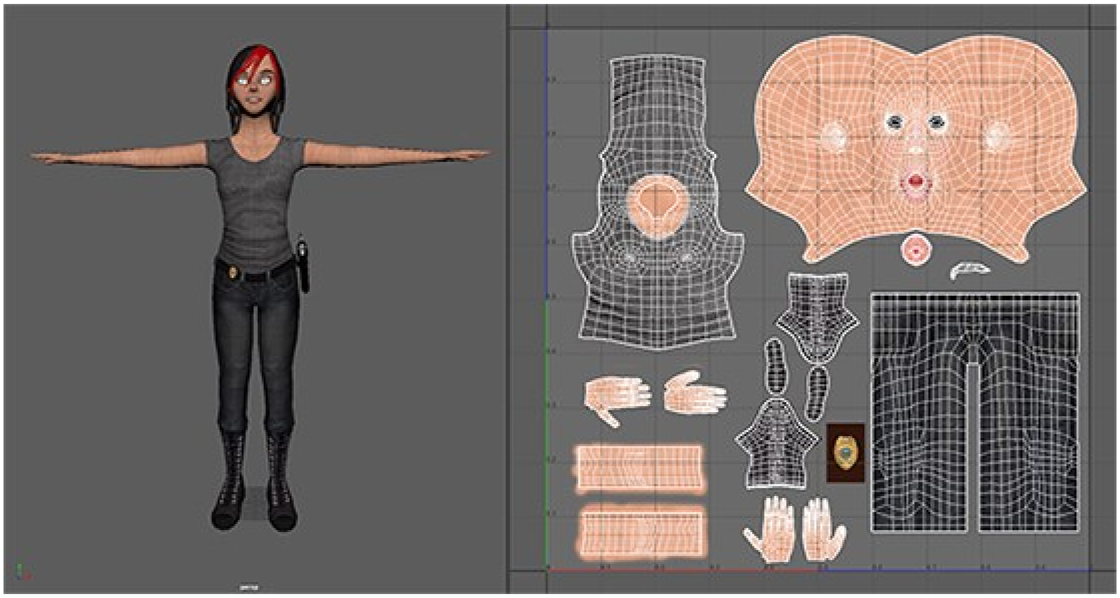


Figure 4. Example of UV layout (Autodesk 2022)

After the unwrapping and baking phase starts the actual texturing process. the first basic material libraries are built, and then final details are added. Materials and elements get analysed and it is decided how to implement the look of the material. Big teams build their material libraries in-house depending on each project. For an independent artist, it is advised to have a vast and organized texture library for efficient use (Yang 2020). For big studios having a huge library of generic textures before even having the style in place can become obsolete (Bosset 2017). At this phase artists mostly concentrate on base colors and getting the general feeling in place. The focus is on the overall material values, the colors, and the variation (Bosset 2017). The general rule of thumb is to start big and work up to small details. For example, Prey's environment artists utilized 80/20 division in their workflow where eighty percent of the texturing was done with Substance Designer and the last twenty percent of details with Photoshop (Bosset 2017).

The style of the game leads the way in working. For example, the game *Control* (2019) determined its style to represent brutalist architecture. To represent that style, they made many subtle variations to the same concrete material (Figure 5). This represents security and solidity in the face of the unknown (Vesterinen 2019). A similar pipeline was used in the game *Deathloop* (2021) where the team had a big emphasis on iteration. They made hundreds of enemy NPCs just by having a simple node-based texture base and made several variations from that (Ferder 2021). As the style of the game is a party of eternal life the characters painted themselves with the most extravagant colors and designs thus the team had to make a lot of elaborate iterations rather quickly (Figure 6).

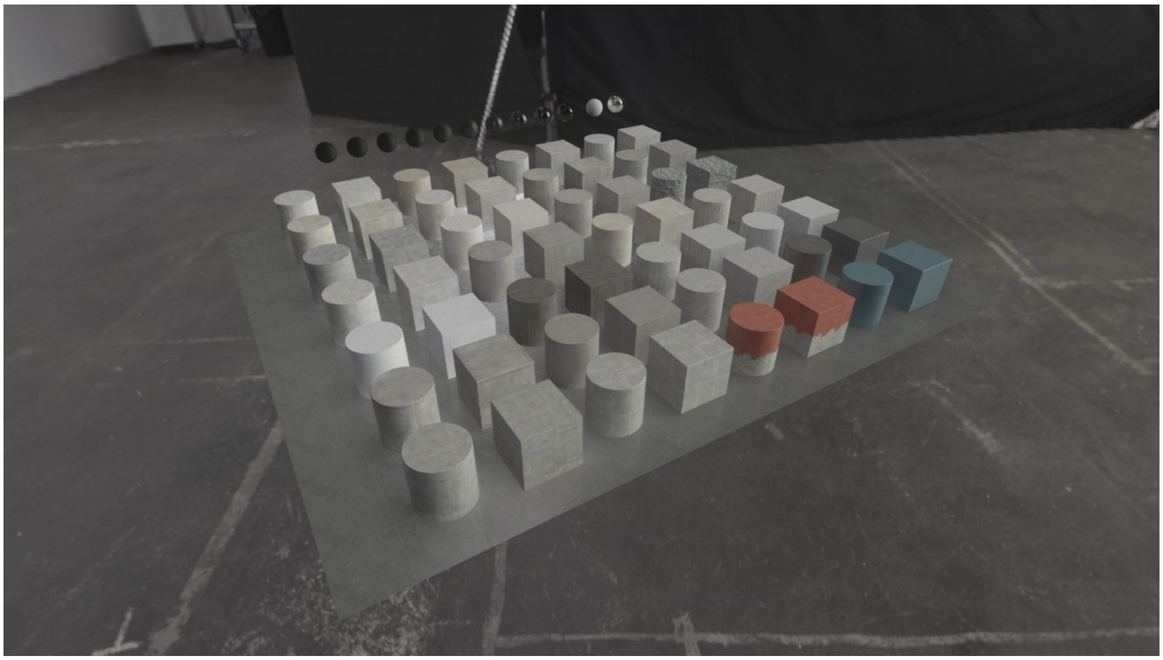


Figure 5. Variety of concrete textures from the game Control (Vesterinen 2019)

Figure 6. NPC texture variation from the game Deathloop (Arkane Studios 2021)

2.3 Optimization

Making a game run at its very best is called optimization (Ahearn 2009). As games render in real-time, especially during the texturing phase, artists must take optimization into account. The importance of optimization is what defines game textures from textures made for movies. Due to the strict polygon budget what cannot be conveyed from 3D models itself has to be emulated through textures (Pluralsight 2014).

One way to optimize game assets is to use texture maps such as transparency and normal maps. Transparency maps can give extra detail to assets such as wire fences or foliage which would otherwise be too high in polycount if modelled. The most common transparency maps are alpha maps and opacity maps. Transparency maps are black and white maps where black is transparent and white means solid. This way one can create very detailed assets with fewer polygons used. For example, to create a wire fence only one plane is needed with the alpha map when otherwise the mesh could have much more polygons in it (Pluralsight 2014).

Figure 7. An example of the alpha map (Unreal Forum 2018)

Another way to add detail without adding extra polygons is with normal maps. Normal maps are X, Y, and Z coordinates converted into RGB color maps. Normal maps are commonly used to project surface detail on a low-resolution model. Normal maps define how light bounces from the low poly object (Lemos 2020). Usually, it is done by sculpting high poly detail on a model, baking it into a normal map, and then projecting that detail onto a low poly version making it look more detailed than the model really is. Normal maps can as well be converted from images, done with node-based workflow, or even painted by hand, but the best result comes from high to low poly workflow (Polycount 2018).

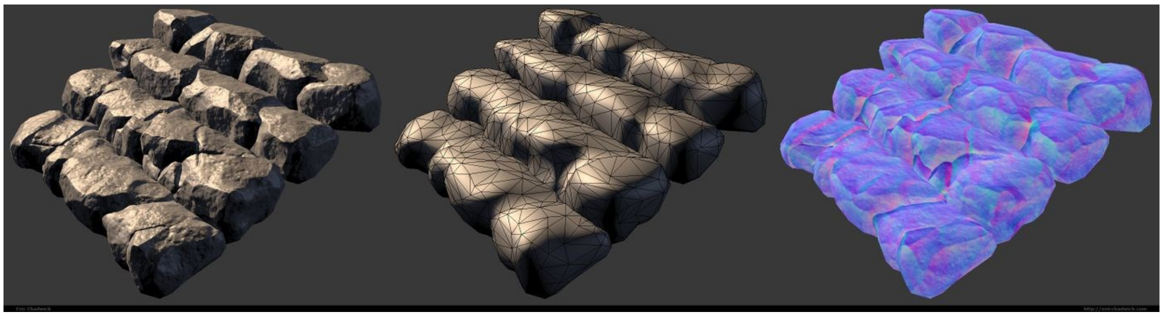


Figure 8. The normal map on a low poly model (Chadwick 2018)

One optimizing method is to use texture atlases (Figure 9). Especially common in environment art, texture atlases are used to optimize file space. A texture atlas is one texture file that contains many smaller textures packed together (Arm Developer 2019). This method is done to save space and lowers the number of draw calls. Atlases are as well beneficial for sorting since a much larger amount of assets are using the same file (Ivanov 2006). With the use of a texture atlas UV unwrapping need to be arranged depending on the atlas. Thus, building these can be very tedious for artists and require careful planning ahead (Ahearn 2009).

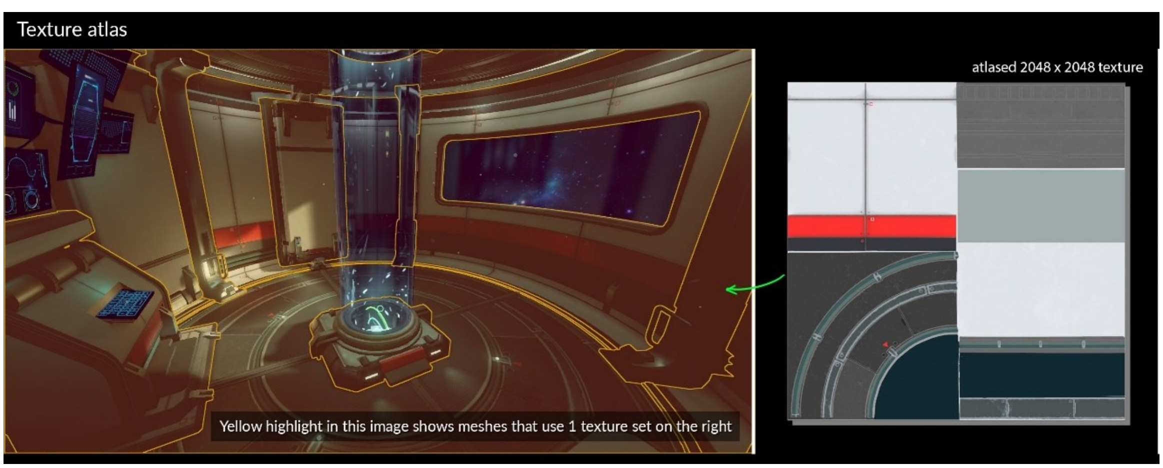


Figure 9. Texture atlas used in an environment setting (Arm Developer 2019)

Additionally, commonly used optimizing method in environments art is with tileable textures. The tileable texture is one texture plane that can be seamlessly placed next to each other (Pluralsight 2014). Instead of painting the whole game scene, different elements such as ground, grass, wall, and roof elements are made into square texture planes and placed along with the whole geometry. This method saves a lot of texture space and is easier to manage.

Another efficient way of increasing performance and better look with game environments is with mipmaps (Figure 10). Mipmaps are texture files scaled down in half showing a different level of detail known as LOD (Anuworakarn 2019). Mipmaps are used on assets that can be seen further away from the player. Using the mipmaps helps to reduce memory load and even makes the overall look better since the texture file is fitted to the resolution of the far-away object (Arm Developer 2019).

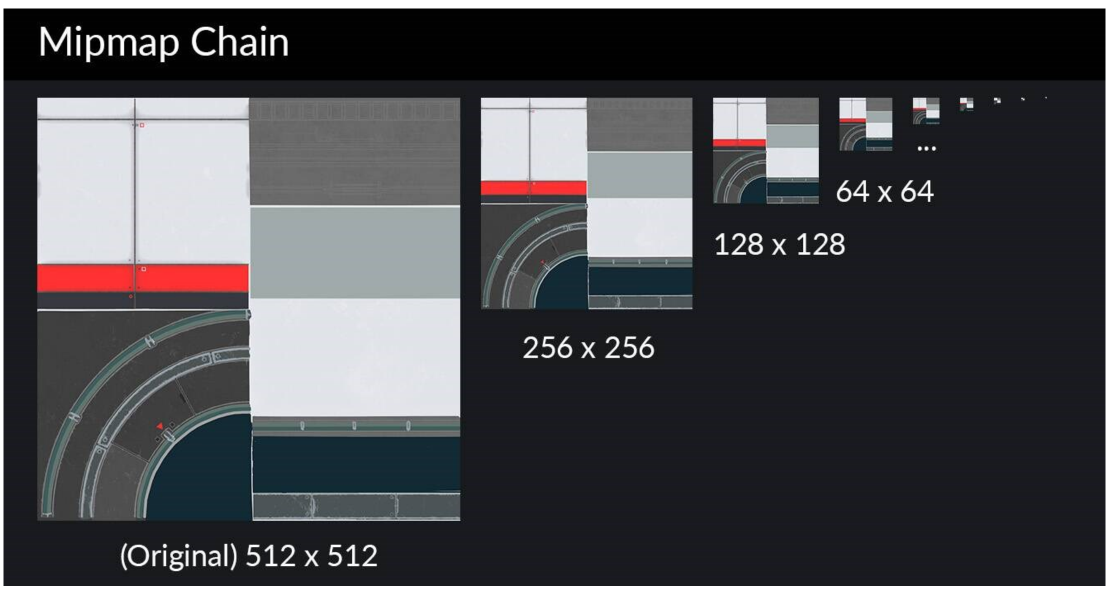


Figure 10. Example of a mipmap chain (Arm Developer 2019)

Generally, it is good to plan how much detail is needed. For example, mobile games do not need texture maps or high-resolution texture files. Especially with mobile game assets, the details, such as ambient occlusion and small highlights can be baked into the diffuse maps thus needing the fewer amount of texture maps (Arm Developer 2019). All assets do not need to be the same size either and even some texture maps within the single prop can have different sizes depending on needed detail. For example, the albedo map can be 1024 x 1024 pixels, but the roughness map is 512 x 512 pixels (Ahearn 2009). The general rule thumb is to make texture files as small as possible to have the needed result. A few other methods to save up memory are by packing several texture maps together into one map and by compressing texture files which reduces the data size.

A good example of optimizing the environment in the actual game can be seen in *Star Stable Online (2005-2022)* game. Carlsson explained in the Star Stable Entertainment development blog how the team optimized old textures in a modular town environment to have only 15 texture maps instead of 75 (Figure 11). For that, they used gradient maps. With gradient maps, the team could make greyscale tiled textures such as walls, roofs, doors, and so forth and color them with a set of different gradients. This way they only need to load in 15 texture maps and add color inside the game engine (Carlsson et al. 2022).

Figure 11. Texture optimization with gradient maps (Carlsson et al 2022)

Some of these optimizing methods, such as mipmaps and atlases, are mostly used by environment artists. Due to characters being rather organic, some optimization workflows are unnecessary. Even though nowadays games can render in higher resolution good optimization is important for a smooth run and good experience for players.

3 PHYSICALLY BASED RENDERING

Physically Based Rendering or Physically Based Shading is a realistic system to emulate light behavior on objects. As mentioned earlier PBR is a rendering method that emulates light. PBR is widely used and is a standard in many programs due to it being simpler and more intuitive (FlippedNormals 2019). PBR workflow makes texture artists more able to be creative with texture creation without needing to know all the technical aspects of rendering since the system is based on physically accurate formulas (McDermott 2018).

3.1 Light Behavior

When a light ray hits a surface it either gets reflected or refracted (McDermott 2018). The law of reflection states that when there is a perfectly smooth surface the light bounces off at the same angle as it hits the surface. Light can as well pass from one object to another and change direction.

Light can as well be absorbed and scattered inside the object. When light is absorbed, it changes form, usually converting to heat (Russell n.d.). Light can as well scatter inside the object and some of it bounces back out which is called diffusion (Figure 12). A good example of such an effect is when light radiates from the back of the ear (FlippedNormals 2019).

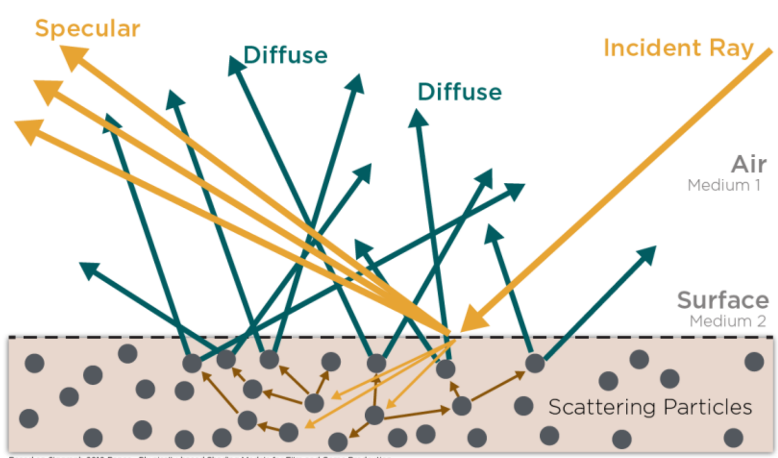


Figure 12. A light ray travelling from one medium to another scatters inside the object (Allegorithmic 2018)

PBR shader controls light physics such as Fresnel and energy conservation making it harder to break the physics behind the light behavior. This way the artist cannot break the law of physics which would result in poor-looking textures. Energy conservation is an important aspect of PBR (Russell n.d.) It simply means that the light that reflects off the surface is never brighter than it hit. The Fresnel effect is the percentage of light that the surface reflects at grazing angles (Wilson n.d.).

3.2 Texture Maps

Texture maps essentially tell how the asset reacts to light. Texture maps are the ones giving all the details and material attributes needed for the assets. PBR has two workflows for creating textures which are metallic/roughness workflow and specular/glossiness workflow (McDermott 2018). Both workflows work in the same fashion but have slightly different texture maps used (Figure 13). Transparency and normal maps were explained in the earlier chapter which are used in both workflows.

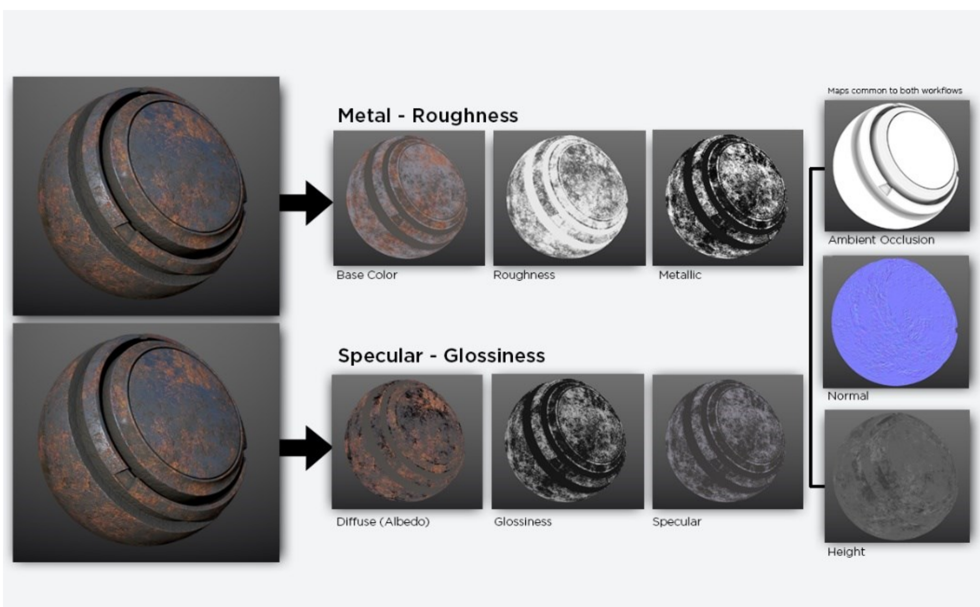


Figure 13. Metallic/roughness and specular/glossiness map comparison (McDermott 2018)

A base color map usually called albedo or diffuse map essentially is a color map. It does not have any shading information and works as a base for the asset. In metallic/roughness workflow base color map is called albedo and it represents color for non-metal and reflectance for metal. In specular/glossiness workflow the base color is called diffuse map and it does not tell reflectance for metal. The diffuse map only contains albedo color, and the raw metal is indicated as black. (McDermott 2018.)

In the metallic/roughness pipeline two maps that are used are metallic map and a roughness map. A metallic map simply says which part of the asset is metal and which is not. The map is black and white where black means non-metal and white means metal. There essentially is no gradience to it (FlippedNormals 2019). If there is to be any dirt or irregularities to the object it is conveyed through a roughness map. A roughness map is where all the details lie. In this map, black represents smooth, and white represents rough. A roughness map is the most creative map which tells the story of the asset (McDermott 2019).

In specular/glossiness workflow glossiness map is similar to roughness but the map is inverted. Whereas black represented smooth in roughness it means rough on the glossiness map. The glossiness map describes surface irregularities. A specular map tells if the material is metal. The map tells the reflectance of the material. Even though both workflows have very similar texture maps they represent different attributes and are not interchangeable (FlippedNormals 2019).

Lastly used in both workflows is the ambient occlusion map. Ambient occlusion tells how much ambient environment lighting is accessible to a surface point. It defines how close the object corners are to each other. The slight ambient occlusion type of effect is visible in real-life objects, though it is different. In hand-painted style, ambient occlusion is usually baked into the diffuse map, but it is usually not recommended to do. Adding it as a separate map allows the shader to use it in a more intelligent way (Wilson n.d.).

4 DEFINING STYLE & OBJECTS

Following the general texture artist's workflow, a mood board was built and collected into a program called PureRef (Figure 14). The texture elements were analyzed during this research phase, and material concepts were made. This way all the elements could be analyzed in the same file and see how they work together. As well the mood of the scene could be studied and give a good base for concepting.

To have an asset set that fits together and evokes environmental storytelling the theme was designed to be a coffee enthusiast's living room café based on the author’s interest in the subject and aesthetic. The chosen theme as well could have a wide range of different material elements implemented. These assets were designed based on six different materials to better study the difference between hand-painted pipeline with procedural PBR pipeline and to have more grounds for analysis. The materials decided for this project are wood, leather, metal, hard plastic, glass, and ceramic. All these materials are different in reflectivity to them. The materials have different attributes and elements, which requires a different kind of workflow making them more diverse and more extensive to analyze. For example, glass has some transparency to it which made it a good material to emulate and study.

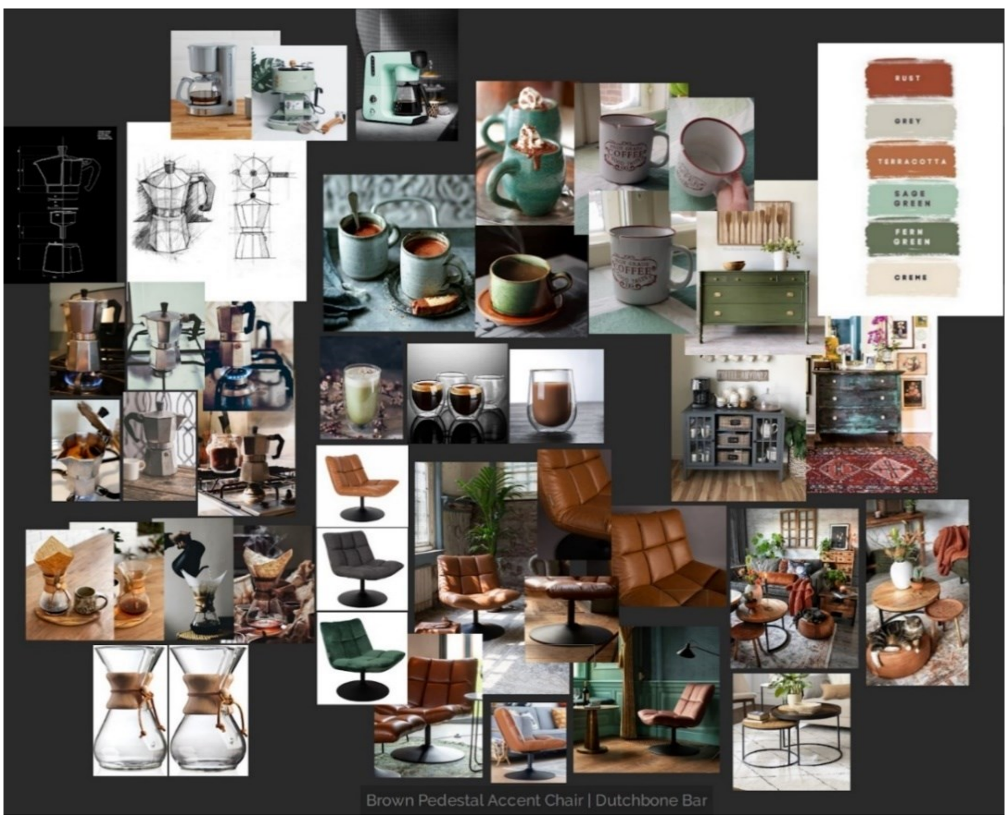


Figure 14. Moodboard reference of all the assets and elements for the scene (Neppius 2022)

The scene was designed to have a rustic feeling with muted complementary color palette. As the theme revolved around coffee a color palette called terracotta was found, which is used in interior design and seemed fitting for the project. To make it more compelling green and turquoise tones were added to the palette to evoke a close-to-earth feeling and work as an interesting contrast to the typical analogous warm brown palette usually associated with coffee. Overall, the wanted mood for the scene was cozy, rustic, and serene with more modern colors. The theme was aimed to evoke the feeling of early morning coffee.

To see if the reference materials work together a quick concept art was made along with the color palette (Figure 15). Since the objective was to study textures not much time was spent on concepting and a technique called photo bashing was utilized to speed up the process. Photo bashing is a method where photo elements are merged and painted together to look like finished concepts.

Figure 15. Concept of a scene with the color palette (Neppius 2022)

For this project, eleven objects were designed. The designed assets that were made were divided into three size sections, big, medium, and small. Big assets were a leather lounge chair, an old cabinet, and a coffee table set. Medium assets were a common coffee machine, moka pot, and a pour-over pot. The small assets were made of three different coffee mugs and a double-walled thermos glass. These painted concepts were made to study the material elements as well as to explore the style further (Figure 16). This way a better understanding of finer details could be achieved and helped to pre-plan and attain the wanted results. These concepts were especially helpful for hand-painted workflow and were used as a reference for said style. For PBR workflow the original reference board was mainly used due to realistic material attributes. The color palette could also be tested and modified to fit the designed aesthetic which made the texturing phase more efficient. Less time needed to be used in the hand-painting phase since the problem solving for the style and certain attributes was already done.

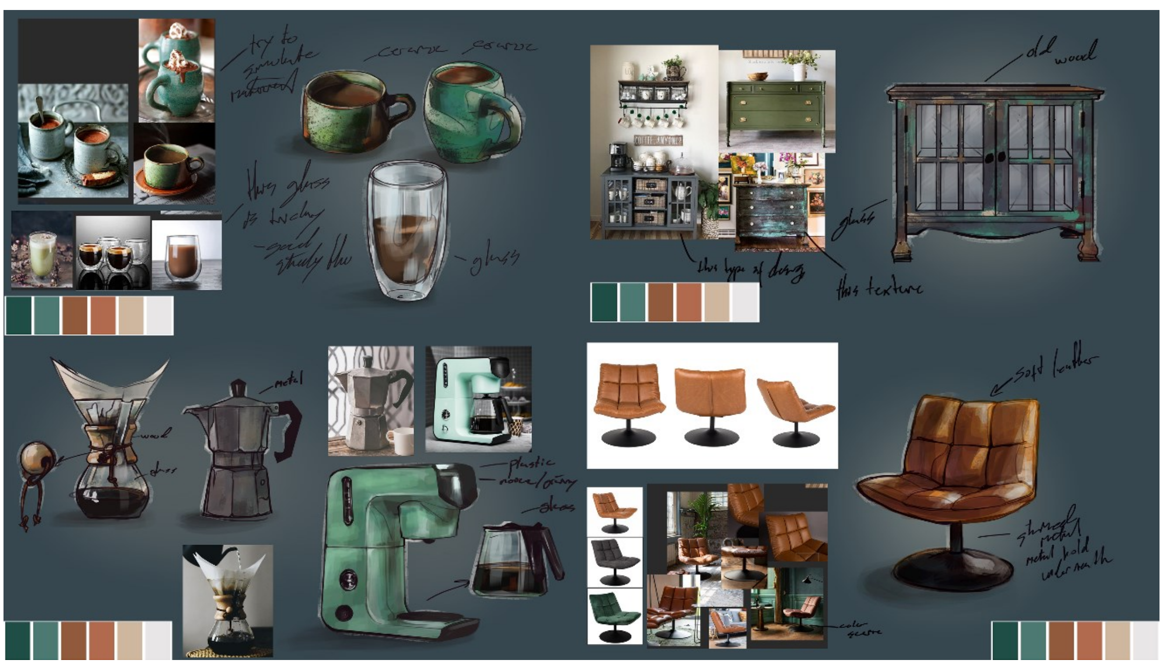


Figure 16. Concept studies of objects (Neppius 2022)

4.1 Hand-painted Diffuse only Style

For the further style studies, some inspiration was taken from games that use hand-painted texturing such as *Dishonored (2012)*, *Ruined King (2021)*, and *Star Stable Online (2022)* (Figure 17). These games have a distinct and iconic painted feel to them and were referenced mostly to represent how stylistic and diverse hand-painted style can be. Recent games have more complex details in them compared to the older ones as seen in the figure, *Ruined King* game prop (top left), and first *Dishonored* game (down left). *Ruined King* prop has finer detail than the Dishonored game.

Noticed in these games is the subtle rendering of light and shadows. Since game assets move and rotate in a 3D environment harsh lights break the immersion. Ambient occlusion maps are usually baked into the diffuse maps to have a sense of depth yet not be too distinct. Marie-Cécile Jacq (2022) explained in the Star Stable Entertainment development blog that since player characters have a variety of different skin colors it is important to find the fine line where the skin is not too dark during night lighting and too shiny during the daytime.

Figure 17. Examples of games that use hand-painted texturing style (Ruined King 2021; Dishonored 2012 & Star Stable Online 2022)

Additionally, Aurélier Predal’s concept for the Early Man movie was referenced along with some indie game artists Adam Wood, Arkady Nabirenkov, and Eric Coolmore (Figure 18). These worked as a style guide and example on how to tackle the hand-painting workflow. As seen below there is some prop concept art from Predal’s work for Aardman Animation studios which has a strong painterly effect and nice implementation of earthly colors. In the 3D props, the light information is subtle and is colored to have a softer and more delicate look to them.

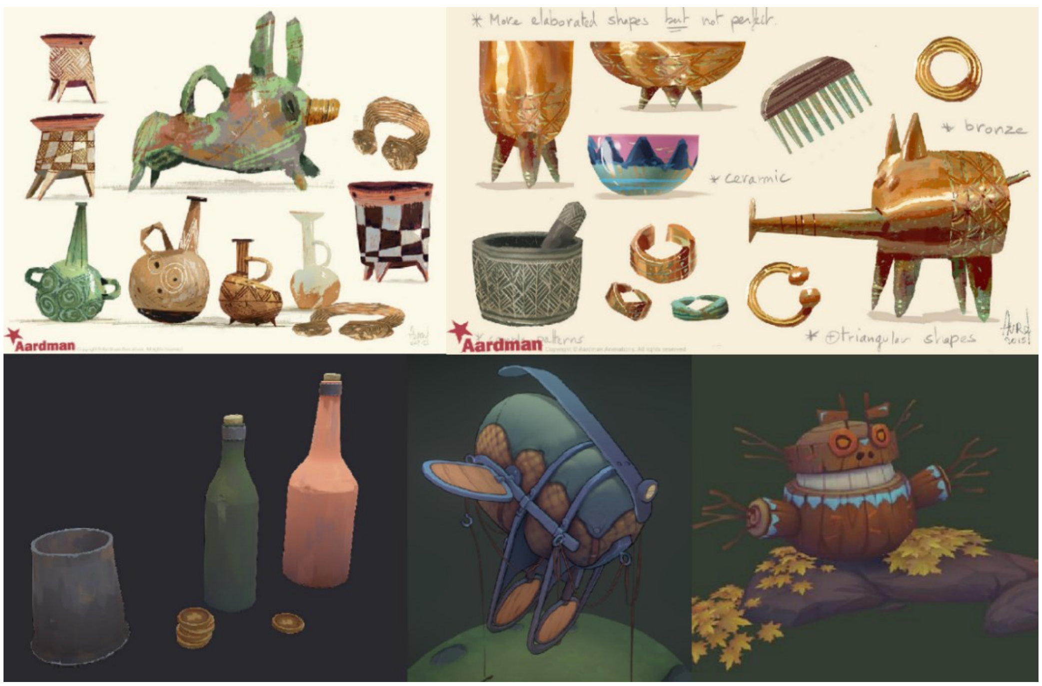


Figure 18. Prop concept art and 3D props in hand-painted style (Predal 2015; Wood 2022; Nabirenkov & Coolmore 2022)

The most notable elements noticed from these studies is the aspect of delicate light and shadow attributes. Hand-painted assets cannot have too distinct light information since the assets need to be able to rotate and be placed in different areas throughout the scene (Ahearn 2009). Though due to the lack of texture maps some material and light information still needs to be painted on the diffuse map to have some sort of material effect. As seen on many of these models’ slight ambient occlusion type of shadow information is there.

4.2 Node-based PBR Style

For the PBR workflow, *Deathloop (2021)* and *Overwatch (2016)* were analyzed for their PBR-oriented stylistic pipeline (Figure 19). Physically Based Rendering is quite common for realistic style but has as well become quite popular with a more stylized look especially after *Overwatch* became popular. PBR is based physically meaning that the system emulates realistic light, but it does not mean real (FlippedNormals 2019). After all, PBR shader was developed by *Disney* which is popular for its heavily stylized animation.

Nowadays the art style is widely known as stylized PBR, and many artists have started to create games with that pipeline. Stylized PBR still has the same attributes commonly found in a hand-painted style which is exaggerated silhouette and bright colors. Albedo maps can be also hand-drawn to have a more artistic feel and gradients are quite common in that style. Though most of the emphasis still comes from texture maps and realistic attributes. Especially rough base color and shiny metal is used for appealing contrast and emphasizing the reactivity with light information in the shader.

Figure 19. Example of stylized PBR art style (Overwatch 2016 & Deathloop 2021)

5 3D MODELING & UV UNWRAPPING

The models were done in a modelling program called 3ds Max. The modelling work started with the lounge chair. Originally the padding part was planned to be sculpted, but due to software limits, it was decided that the details were to be created with the texture maps only. Even though sculpting detail is an important part of texture creation for normal and AO maps, it was not needed for this project and the author felt like projecting the details was enough.

The common technique, that is the work from big shapes up to small details method, was used with the asset. First, the general proportions and shape were done by blocking out the object and with modifiers, the softer shape and curviness were attained. The lounge chair asset was as well split in half and mirrored to have a more symmetric result. The leg part was made from the cylinder and box basic shapes and attached for editing. Lastly, all the parts were merged (Figure 20).

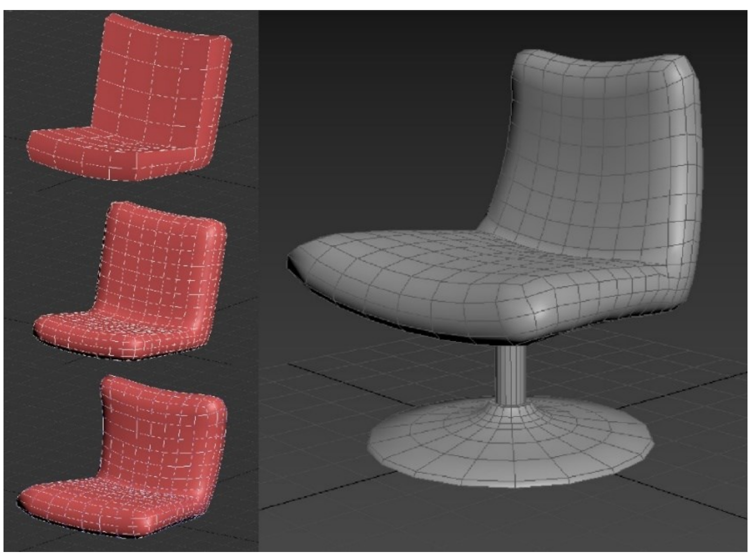


Figure 20. Lounge chair modelling process (Neppius 2022)

A similar pipeline was used with all the other assets. The general shape was built from basic shapes and smaller details were added by either editing them specifically with the box modelling method or by using modifiers for a more fluid result. Depending on the complexity and attributes of the asset different modifiers were used accordingly. The coffee mugs were done with a basic cylinder shape which was then intruded inside to have the mugs inside part. The handle was done with a spline that had a cylinder shape assigned to it. Since there were three different sizes to the mug the mug was just copied, and the size and shape were slightly modified (Figure 21).

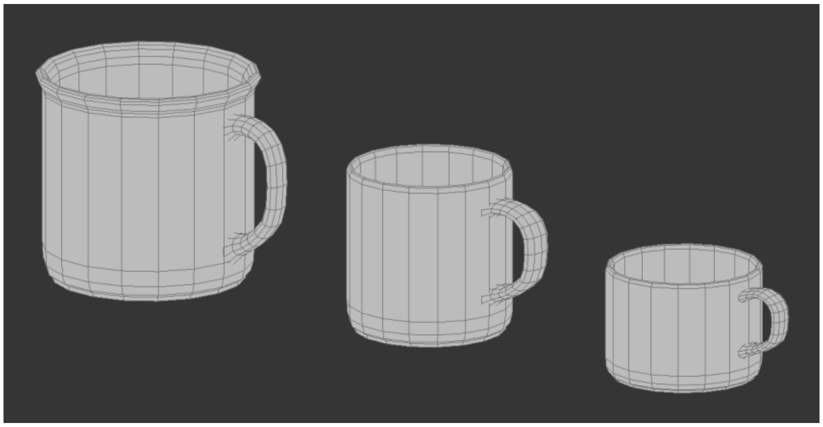


Figure 21. Different variations of mug assets (Neppius 2022)

Pour-over pot had quite time-consuming leather string which was done in a similar method as with the coffee mug's handle part but distorted more around the pot and to have a little knot on it. The glass assets were done as a simple cylinder shape and added a shell modifier to it. Shell modifier adds thickness to an otherwise one-sided plane. This was particularly important with slightly transparent models. Though the double-walled thermos glass was simpler than originally anticipated, it needed to be simplified even more later due to having the wrong kind of layering effect when the asset had more transparency to it (Figure 22).

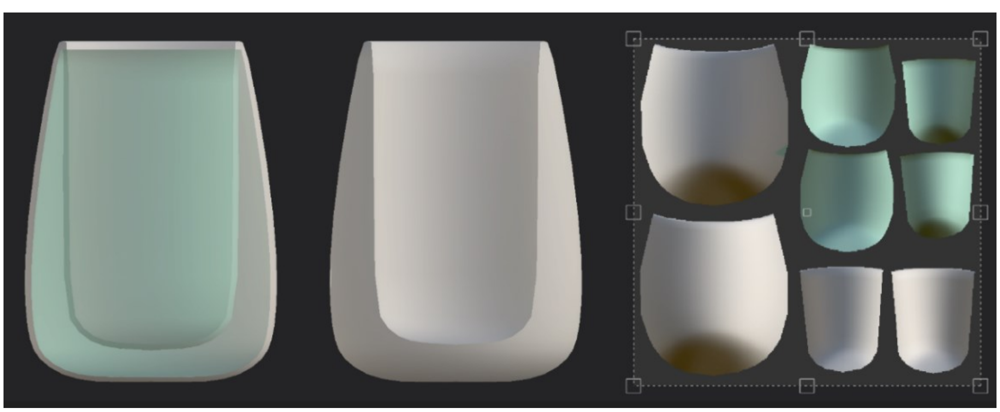


Figure 22. Double-walled thermo glass with and without shell modifier (Neppius 2022)

After all the modelling was done the assets were UV unwrapped. Simple unwrapping was done without any optimization techniques used due to needing more precise detail in the texturing phase. First, some assets were laid out on top of each other in the UV view but was later noticed that it screwed baking mesh maps and made using smart maps and generators impossible. Thus, there could not be any mirroring or overlapping UVs. The general rule is to have as few seams to the models as possible. Due to the models being rather static and trying to avoid too much angle distortion the models were split into multiple parts against the common rule.

Lastly, ID vertex colors were added. With ID mapping certain parts of assets are designed with different colors to indicate different parts of the model (Figure 23). This way picking different parts only needs to be done once and is a common method to speed up the process. This method is mostly used for small parts and when the model has much different material attributes to it.

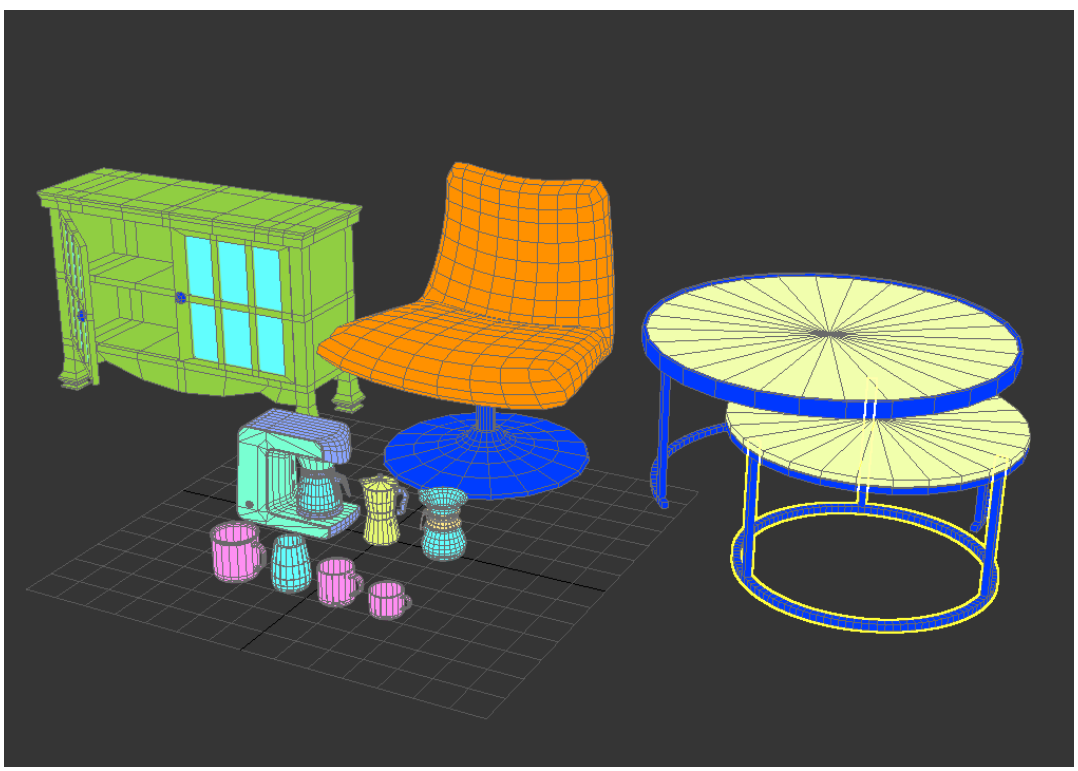


Figure 23. ID mapped assets (Neppius 2022)

6 TEXTURING

6.1 Preparing Models in Substance Painter

The texturing phase was started by preparing the models by baking needed mesh maps and setting the resolutions for the different sized models. At first, all the models were planned to be textured in the same Substance Painter file to keep better track of quality and consistency, but due to a high number of assets, it was too memory heavy to do for the program. This method also warped the position and world space normal mesh maps since the program counted all the assets together, which made texturing small assets difficult since big assets made a general brush size bigger. Due to those constraints, it was decided that assets would be textured separately.

During the mesh map baking phase, the importance of good UV was noticed. With the drawer asset, some UV elements were piled on top of each other, and the position map could not be effectively baked (Figure24). That made it impossible to use smart materials and generators.

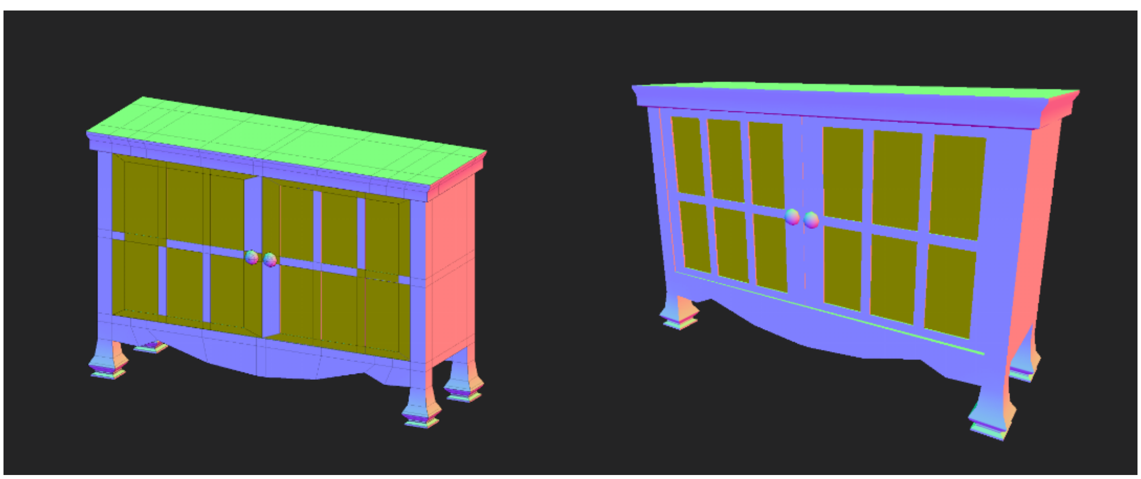


Figure 24. Position bad with overlapping UVs and fixed version (Neppius 2022)

To be able to create texture maps in Substance Painter one needs to bake mesh maps first. Mesh maps are Substance Painter-specific terms for detail maps that are generated from 3D mesh into a texture map. Mesh maps make it possible to use filters and smart materials (Adobe 2020). Some mesh maps that are used are ambient occlusion, ID, position, curvature, world space normal, and thickness (Figure 25).

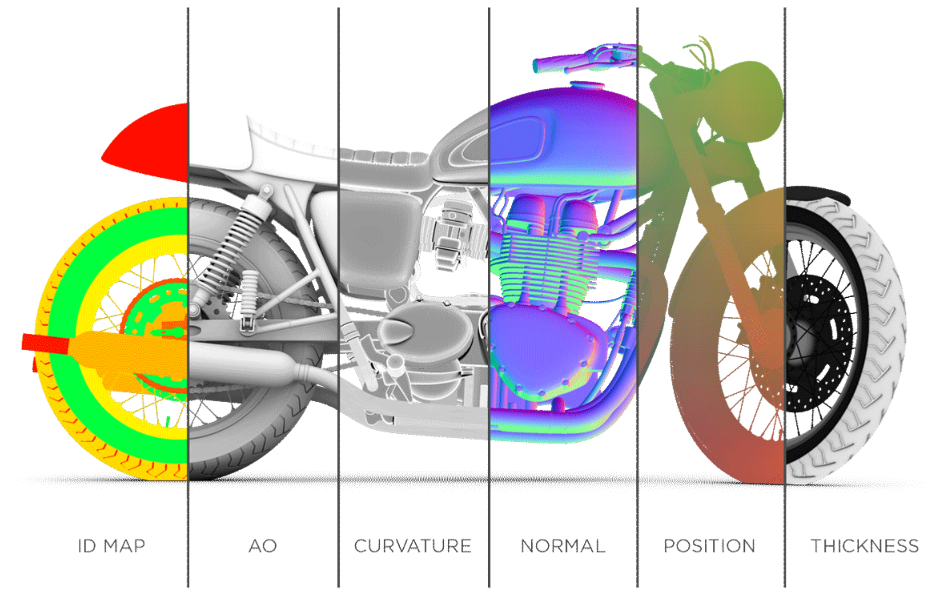


Figure 25. Different mesh maps (Adobe 2019)

6.2 Hand-painted Workflow

Utilizing the ID map, base colors were assigned to color-coded areas and made into specific folders. Position, ambient occlusion, and curvature maps were used for adding details such as colored gradient and ambient detail. A similar method of creating stylized albedo color was followed by 3D artists Robbie Walker (2021) and Lary Kummer (2019) on adding baked-in detail effects. Kummer (2019) advised using a strong top to bottom gradient and colored AO for a more vibrant separation of elements. A curvature map as well plays a big role in creating an appealing look. Layering up details with light information filters, basic gradients, AO, and curvature details already gives nice results (Walker 2021).

One way of adding generated details is with smart materials. This way the general use of filters and detail maps needed only to be done once. Once the filter effects, curvature details, ambient occlusion, and gradient are added the information is then converted into smart material which then is assigned to other assets. The smart mask technique was used to make a base for assets and as well used on other continuous material assets such as the coffee table's metal legs and ceramic mugs.

After the base materials and baked-in details were added started the actual painting phase. The 80/20 guideline was utilized where eighty percent of the hand-painting was done in Substance Painter and the last twenty percent was tweaked in Photoshop. The guideline was not completely followed depending on the asset in question. For example, a hundred percent of painting was done in Substance Painter for glass materials and with moka pot asset almost eighty percent of texturing was done fully in Photoshop.

Glass materials had opacity set to 0.2 to have some level of transparency on the material yet being slightly visible for painting. All the painting was done in Painter since the opacity map needed to be checked occasionally and fixed accordingly (Figure 26). The moka pot asset was mostly drawn in Photoshop due to it having large straight elements which would need separate gradient colors to it. Photoshop was more suitable for that due to the extensive lasso and gradient tool that Painter lacks. In Figure 27 one can see is how the details were built on the moka pot model.

Figure 26. Glass asset with an opacity map, albedo map, and both combined (Neppius 2022)

Figure 27. Building up details on moka pot asset (Neppius 2022)

To speed up some processes tileable textures for continuous materials such as ceramic and wood were made (Figure 28). This way these elements only needed to be done once and edited according to the asset. In Figure 28 there is an example of a tileable wood texture and how it was edited for different assets. The tileable materials were painted in Photoshop using the pattern preview view. This way one can actively check if the tiling is too visible or wrong and blend in details efficiently. The offset filter as well works as a good blending option for otherwise too visible seams in tiled textures.

Figure 28. Tiled wood texture and where it was used (Neppius 2022)

Most details came from the painting. As the style was hand-painted diffuse only pipeline all the work was done on base color view and in opacity view with glass materials. Since Substance Painter has quite a similar layering system and similar brushes to Photoshop most of the painting can be done straight on the model. With the ability to modify brushes with alpha masks one can create quite creative and visually appealing results.

Lastly, all the textures were exported to Photoshop to add final painted details and to edit so that the assets have a more consistent and painted look. Posterized filter and layer effects such as difference and exclusion were used to add similarity to the color tone and texture. After the final effects, the maps were yet again converted to the Substance Painter to blend out visible seams that occurred during the Photoshop editing process (Figure 29).

Figure 29. Example of hidden seams painted in Substance Painter (Neppius 2022)

In the end, the hand-painting style is just rendering detail into a diffuse map which is dependent on the artist’s personal style. Though some efficiency techniques could have been better utilized inside the Substance Painter such as smart masks, filters, and baked in detail. Kummer (2019) and Walker’s (2021) art breakdowns could have been better utilized and some sort of planning regarding the balance between baked-in details and actual painting could have been helpful and that way speed up the workflow. Some sort of a general guideline technical-wise would have been good to be planned as well other than style alone. All the finished assets are seen in Appendix 1.

6.3 PBR Workflow

Similarly, to the hand-painting workflow mesh maps were baked first. The metalness/roughness shader was used, and an opacity map was added to the workflow. For this workflow, a node-based texture creation pipeline was partly used which is becoming a norm in games such as *Deathloop (2021)* and *Control (2019)*. Node-based texturing is especially convenient for tileable textures due to Substance Designer's automatic tiling and non-destructible workflow. This workflow makes it possible to create a vast set of textures with less amount of work since one can just copy and edit texture graphs and make variations from those. The lounge chair's cushion, general leather, wood, and ceramics were done using Designer's node-based pipeline (Figure 30). Otherwise, Substance Painter's basic materials were enough to have the needed results.

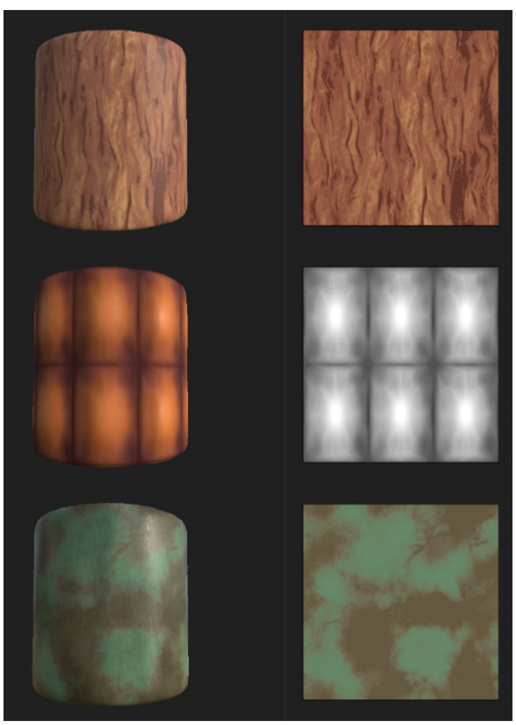


Figure 30. Substance Designer materials (Neppius 2022)

To test out the metalness/ roughness workflow in the PBR pipeline the texturing was started with the moka pot asset due to it being almost only metal. Used moka pots tend to have grease marks and visible roughness details which makes it a perfect example of said workflow. First, the models had assigned base material attributes such as metal, hard plastic for the handle, and copper for the fuse part by utilizing ID maps and organizing them to separate layers. A galvanized filter was added to have some stylized elements to otherwise simple metal. The smart material done for the hand-painting pipeline was as well used to further emphasize the stylized look.

Then started the detail adding phase where smart masks and filters were used to create wear and tear details. Since Substance Painter has quite an extensive smart mask library most of the effects could be conducted by utilizing and editing those. Color, roughness, and height detail were assigned as specific fill layers where said information is then constructed on specified areas with smart masks. To have more creative and personalized effects some painted details were done. In the PBR pipeline roughness map is the most creative map for artists (FlippedNormals 2019). Since albedo only shows the unlit base color and the metalness map informs if the material is metal the way to have more creative control is with a roughness map. With this knowledge, details were painted on the roughness layer. Lastly, some height details were added to the lid part to better represent the object's minor detailing. Seen in figure 31, is the buildup of material effects and surface detail. A similar pipeline was used with the other assets as well.

Figure 31. Process of moka pot assets in PBR workflow

With the lounge chair, the padding part that was done in Designer was converted to Painter and then projected on the seat (Figure 32). This method was proved to be far trickier than anticipated. Due to having to stretch the detail maps somehow destroyed height and normal information making the chair look flatter than would have been expected. Rendering information was slightly different in Substance Designer than in Painter which was further confusing at times. Due to the stretching detail maps could not show the needed information properly and needed to be fixed with the trial-and-error type of method and painting some albedo and roughness detail in specific paint layers. Some texture maps were as well exported to Photoshop to edit and hand-paint the maps to look better. With this asset sculpting high-poly detail and baking it into a normal map would have been a better option than having tileable cushion texture stretched on a specific area.

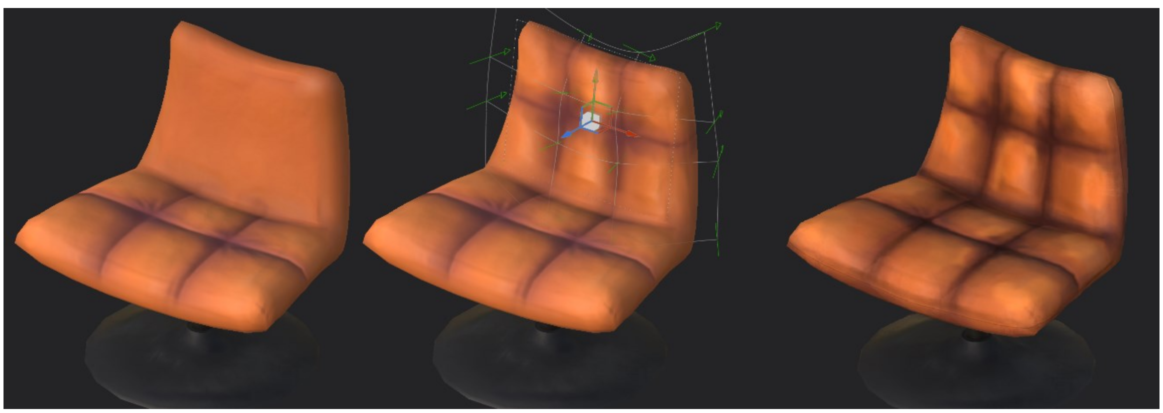


Figure 32. Projected cushion detail and the edited final version (Neppius 2022)

Another way to speed up texturing was with the use of custom alpha maps. This method was utilized with the big mug that had embedded text design on it and with some coffee stain details in cup and pot assets. A colored fill layer with height and roughness information was added. With a black mask, it was concentrated on the alpha map image and projected to the mug (Figure 33). This method made making detailed image creating very efficient due to only needing to draw the alpha mask once in Photoshop which is as well better equipped for such. The same mask was used on the hand-painted mug as well.

Figure 33. Custom alpha map and projected detail on mug asset (Neppius 2022)

Creating PBR textures was easier than first anticipated. Even though one works with realistic material attributes the workflow is slightly similar to hand-painting although more about adding generated details and changing them accordingly. Where the creativity is in the diffuse color map when working in a hand-painted pipeline, the PBR pipeline is most proficient in roughness map. One needs to think a bit differently since it is not as much about general material but the wear and tear of the object. This pipeline can be very efficient to work with since the shader does the heavy lifting of light and material information, and artists only need to think about the story behind the asset and one attribute of the whole asset.

Node-based texture creation was not as needed as first anticipated but was proven quite efficient for tiled texture creation. Substance Designer could have been better utilized to create custom filters and smart materials. This way the assets could have had a more consistent style. As mentioned earlier better guidelines for texture creation workflow and a more proficient style guide would have reduced the later problem solving regarding how to tackle the creation pipeline. An example image of all the finished assets is seen in Appendix 2.

7 COMPARATIVE ANALYSIS

Many texturing terms overlap (Ahearn 2009). This partly explains why the information can be scarce and sometimes even misleading. Neither of these texturing methods rules themselves out, but rather work in unison. Hand-painting and PBR workflow are usually associated with specific styles which leads to the common misconception that the pipelines cannot be blended. PBR workflow can be used alongside hand-painted workflow even though it usually gets mixed up as opposing styles. Though there are still some elements that are opposing to each other and change the meaning of terms such as the base color map and many terms of it.

Physically Based Rendering is a way of rendering light in a scene whereas hand-painting is a way of creating textures. To put the terms better in place it is important to determine what are the opposing elements. Since PBR is a way of rendering light then the opposite would be either not to have any physical lighting in the scene or an old approach to rendering light. An albedo map simply is a color map without any light information which means if one bakes or paints light information on the said map it is not considered an albedo map anymore. It is quite common to make ambient lighting in the hand-painted workflow to simulate depth and otherwise leave all light information out. In that case, the same color map is called a diffuse map to better fit the term. Another technique is to have in-game lighting as well, but since the AO shadow does not have any direct source of light it will not break the immersion. Henceforth, to speak in that sense the opposing elements would be the lack of painted light information versus painted light detail.

Hand-painting meshes were more time-consuming than creating them with filters and generators. Though it is more intuitive and easier from an artistic point of view than working around nodes and graphs. For more technical people node-based texturing is ideal. Node-based procedural texturing had a steep learning curve and can be overwhelming to start with, but due to it being non-destructible creating vast variations with little changes makes the pipeline very effective. PBR texturing is very simple at its core but can be overwhelming as well since the artist is working with realistic based attributes. In the end, PBR is what makes realistic texture creation more creative than older methods due to the use of texture maps. In hand-painting, one paints details on base color whereas in PBR the painting is done in a roughness map. PBR pipeline does not exclude the need to hand paint details since it is still used as a final detailing on albedo or done in the roughness channel. Otherwise, it is similar to hand-painting, but one needs to think differently about the texture itself.

Another way to think of these terms would be as an art style rather than a method of creating textures. PBR has generally been associated with a realistic style whereas hand-painting is more cartoony and exaggerated. A common misconception among artists is to ask the difference between hand-painted and PBR when they think about realistic versus stylized as an art style. To better see the difference between both styles, Appendix 3 demonstrates it.

As explained earlier good examples of combining both workflows are seen in games such as *Overwatch* (2016) and *Prey (2017)*. Both games use a PBR pipeline with hand-painted detailing for emphasizing stylized results. In that sense, style is not as strict and makes that kind of thinking obsolete; though to base this type of style thinking on the case study creating realistic PBR textures is much easier and more fluid than making it stylized. When trying to add stylized elements such as painting on albedo it can make assets look too busy and different maps can contradict the information. Stylized PBR can be more expensive to create than realistic PBR due to having to find a fine line between different pipelines without making the asset look bad.

Optimization is defining feature in game art. Through the case study, the importance of efficient workflow and optimizing texture creation processes came clear. Since the author needed to make a vast set of materials it drove them to think about how to cut corners so that one does not need to make the same progress twice. With just a single prop or scene that would not have been noticed. With the use of smart materials, alpha masks, and tileable textures some parts only needed to be done once and the same material could be utilized for other assets as well. Hand-painting workflow is less memory heavy due to only needing one or two texture maps whereas in PBR one needs at least three and up to eight maps. With fewer maps, it is as well easier to manage and not as overwhelming to organize.

In the end, it is not as much about how one creates textures, but more like how appealing and efficiently one makes them. There is no clear guideline on how these pipelines should be worked on, but knowing the features helps to distinguish the errors that might occur. Many senior artists emphasize efficiency more than technique.

8 CONCLUSION

Overall, the thesis answered the main questions, and the goals set were met. The literature review was successful and through research, the goal of enhancing the authors' knowledge was fulfilled. Based on the case studies the distinction between both workflows became clearer. Even though the original hypothesis of analyzing both workflows as an art style became obsolete at the start of the research the difference in pipelines was still noticed later while doing the case study.

The scope of the project phase was more time consuming than anticipated. Hand-painting and learning Substance Designer took more time than calculated, but the PBR texture creation in Substance Painter was surprisingly quick. The process of creating PBR textures was not as complex as anticipated and through research it became clear.

The high number of assets established more efficient "cutting corners" thinking to speed up the process. With a high number of assets, it made the author think more carefully about how to save time and not need to do things twice such as the same wood texture or ceramic.

Time management and planning could have been done better. More coherent and structured planning could have possibly saved time from unnecessary problem solving and backtracking.

Information is mostly based on industry veterans' experiences which can be subjective. The validity of information, since it’s mostly based on opinions, can be misleading depending on the source. Many terms overlap, or one word can mean many things which further confuses when researching the topic.

REFERENCES

Ahearn, L. 2009. 3D Game Textures Create Professional Game Art Using Photoshop. 2nd ed. Elsevier Inc.

Anuworakarn, B. 2019. Why you really should be using mipmapping in your graphics applications. Imagination Technologies Limited. Blog. Available at: https://blog.imaginationtech.com/why-you-really-should-be-using-mipmapping-in-your-graphics-applications/ [10 April 2022].

Arm Limited. Texture Best Practices. 2019. Article. Available at: https://developer.arm.com/documentation/102539/latest/ [Accessed 19 February 2022].

Benno, R. 2019. The history of 3D Video Game Texturing @BryanRenno. Tweet. 29 April 2019. Available at: https://twitter.com/BryanRenno/status/1122880458012934146 [Accessed 24 January 2022].

Bogost, I. 2008. Persuasive Games: Texture. Game Developer. Blog. Available at: https://www.gamedeveloper.com/design/persuasive-games-texture [Accessed 14 February 2022].

Bosset, P. 2017. Arkane Studios: Texturing Prey's Retro-futuristic Visual Style. Blog. Available at: https://store.substance3d.com/blog/arkane-studios-texturing-prey-s-retro-futuristic-visual-style [Accessed 23 February 2022].

Burton, A & Shedu, A. 2020. Toro & Spirits: Making a Hand-Painted Prop in Blender and 3D Coat. 80.lv. Blog. Available at: https://80.lv/articles/toro-spirits-making-a-hand-painted-prop-in-blender-and-3d-coat [Accessed 1 March 2022].

Burton, A & Walker, R. 2021. Robbie Walker, an Interior Artist at Rockstar, showed us how to create a majestic League of Legends-stylized gate using ZBrush, Substance Painter, 3ds Max, and Marmoset Toolbag. 80.lv. Blog. Available at: https://80.lv/articles/the-gate-creating-a-stylised-diorama-in-zbrush-3ds-max/ [Accessed 27 February 2022].

Bullock, M. Jacq, M & Serrano, I. 2022. Narrative Design, Servers & Player Character Update. Star Stable Entertainment. Blog. Available at: https://starstableentertainment.com/blog/star-stable-february-blog-narrative-design-servers-player-character-update/ [Accessed 25 February 2022].

Carlsson, O. Jacq, M & Skullman J. 2022. Marley's Farm, Tech & Character Update. Star Stable Entertainment. Blog. Available at: https://starstableentertainment.com/blog/star-stable-january-blog-marleys-farm-tech-character-update/ [Accessed 15 February 2022].

Ferder, J. 2021. DEATHLOOP's Award-winning Art Pipeline with Substance. Web site. Available at: https://substance3d.adobe.com/magazine/deathloops-award-winning-art-pipeline-with-substance/ [Accessed 23 February 2022].

Ivanov, I. 2006. Practical Texture Atlases. Game Developer. Article. Available at: https://www.gamedeveloper.com/programming/practical-texture-atlases [Accessed 9 April 2022].

Kummer, L & Tokarev, K. 2019. Hand-Painted Overwatch: Junkrat's RV Production. 80.lv. Blog. Available at: https://80.lv/articles/001agt-002mrs-hand-painted-overwatch-junkrats-rv-production/ [Accessed 27 February 2022].

Lemos, C. 2020. Tutorial: How Normal Maps Work & Baking Process. Article. Available at: https://80.lv/articles/tutorial-how-normal-maps-work-baking-process/ [Accessed 26 January 2022].

McDermott, W. 2018. The PBR Guide. 3rd ed. Allegorithmic. Book. Available at: https://substance3d.adobe.com/tutorials/courses/the-pbr-guide-part-1 [Accessed 24 April 2022].

Naghdi, A & Adib, P. n.d. 3D texturing in animation; step by step workflow + techniques & video. Web page. Available at: https://dreamfarmstudios.com/blog/getting-to-know-3d-texturing-in-animation-production/ [Accessed 19 February 2022].

Normal map. 2018. Polycount wiki. Article. Available at: http://wiki.polycount.com/wiki/Normal_map [Accessed 26 January 2022].

PBR Explained for 3D Artists - Physically Based Rendering. 2019. FlippedNormals. Youtube. Video clip. 17 May 2019. Available at: https://www.youtube.com/watch?v=a4dURVZEi3E [Accessed 24 April 2022].

Pluralsight. Tips and Techniques for Creating Seamless Textures. 2014. Article. Available at: https://www.pluralsight.com/blog/film-games/tips-techniques-creating-seamless-textures [Accessed 13 April 2022].

Pluralsight. Texturing for Games - Maintain a High Level of Detail without Extra Geometry. 2014. Article. Available at: https://www.pluralsight.com/blog/film-games/texturing-games-maintain-high-level-detail-without-extra-geometry [Accessed 19 February 2022].

Russel, J. n.d. Basic Theory of Physically-Based Rendering. Marmoset. Article. Available at: https://marmoset.co/posts/basic-theory-of-physically-based-rendering/ [Accessed 5 April 2022].

Tiling. 2014. Polycount wiki. Article. Available at: http://wiki.polycount.com/wiki/Tiling [Accessed 11 April 2022].

The Foundry Visionmongers Limited. Texturing in video games: evolution, application and outlook. 2022. Article. Available at: https://www.foundry.com/insights/film-tv/texturing-video-games [Accessed 5 April 2022].

Texture Maps Explained - Essential for All Texture Artists. 2019. FlippedNormals. Youtube. Video clip. 20 February 2019. Available at: https://www.youtube.com/watch?v=ZOHNRlrd1Ak [Accessed 24 April 2022].

Vesterinen, M. 2019. Control: How a Single Senior Artist Built a Brutalist Material Library. Web site. Available at: https://substance3d.adobe.com/magazine/control-how-one-senior-environment-artist-created-entire-brutalist-material-library/ [Accessed 23 February 2022].

Wilson, J. n.d. Physically-Based Rendering, And You Can Too! Marmoset. Article. Available at: https://marmoset.co/posts/physically-based-rendering-and-you-can-too/ [Accessed 24 April 2022].

Yang, C. 2020. Mari Tutorial: Photo-Realistic Texturing Full Process. Artruism Digital. YouTube. Video clip. 8 January 2020. Available at: https://www.youtube.com/watch?v=phqE5nPBCyc&list=PLeVbew7U6jzfv0yisV8kIFduYT1aLr4o4&index=11&t=5s [Accessed 21 January 2022].

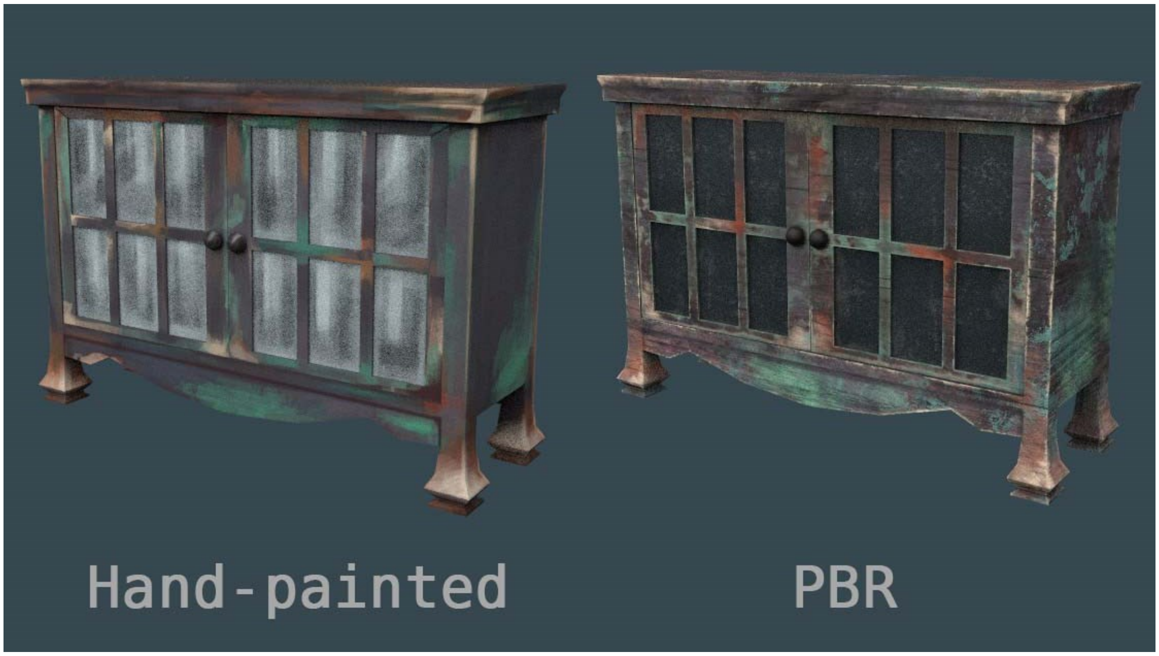
LIST OF FIGURES

APPENDICES

Appendix 1. Finished asset set in hand-painted style.

Appendix 2. Finished asset set in PBR style.

Appendix 3. Side by side comparison of assets.




Hand-painted
PBR

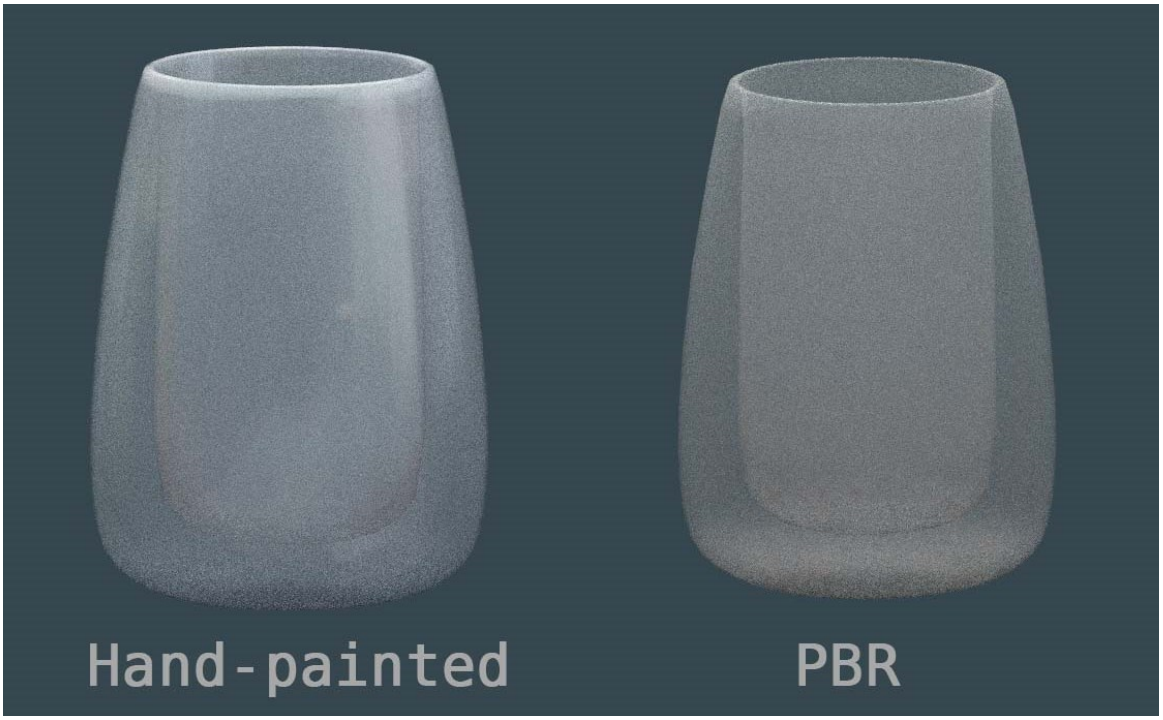

Hand-painted
PBR


Hand-painted
PBR


Hand-painted
PBR


Hand-painted
PBR


HIGH GRADE
COFFEE
GOOD TASTE
HIGH GRADE
COFFEE
GOOD TASTE
Hand-painted
PBR


Hand-painted
PBR


Hand-painted
PBR


Hand-painted
PBR