Advantages of Photogrammetry in creating Photorealistic 3D assets for Game Development using only Freeware: Indie's Delight

Vili Tuori

Bachelor's thesis
May 2022
Bachelor's Degree Programme in Business Information Technology

Jyväskylän ammattikorkeakoulu
University of Applied Sciences

jamk

Abstract

Tuori Vili

Advantages of Photogrammetry in creating Photorealistic 3D assets for Game Development using only Freeware: Indie's Delight

Jyväskylä: JAMK University of Applied Sciences, May 2022, 54 pages

Business, Degree Programme in Business Information Technology, Bachelor's Thesis

Permission for web publication: Yes

Language of publication: English

Abstract

3D modeling a complex model for Game Development purposes is a time-consuming task as there are many different phases that need to be completed before the model can be considered game ready. While there are many tools than can be used to reduce the amount of time needed to complete some of the phases, they are often costly and thus, not attractive choices for smaller developers. One of the tools that is both affordable and available for everyone is Photogrammetry.

The purpose of this study was to explore both the free methods of creating 3D models using Photogrammetry, and to compare the results to those created by using specialized hardware. The goal was to create a sort of a guide for the forementioned smaller developers (and for game developers in general) who might be interested in reducing the amount of work needed to create their 3D models and to present concrete examples of the end results of said methods. This was achieved by researching the different methods of using Photogrammetry to create game ready models, testing them by putting them in practice and putting the end results in direct comparison.

The conclusion was that the theoretical research using document analysis was a success when it came to discovering the viable Photogrammetry methods. However, the practical implementation was lacking due to time constraints and props available at the time when it came to completely free method while the method using specialized equipment was a complete success. Reliability of the results suffered somewhat due to circumstances, but another attempt with more time and equipment would no doubt create better results.

Keywords/tags (subjects)

Photogrammetry, photography, 3D-modeling, game development

Miscellaneous (Confidential information)

No confidential information was handled or gathered during the research

Tuori Vili

Fotogrammetrian hyödyt fotorealististen 3D mallien tuottamisessa pelinkehitystä varten käyttäen ainoastaan ilmaisia ohjelmia: Indien Ilahdutus

Jyväskylä: Jyväskylän ammattikorkeakoulu, toukokuu 2022, 54 sivua

Liiketalous, Tietojenkäsittelyn tutkinto-ohjelma, Opinnäytetyö

Verkkojulkaisulupa myönnetty: Kyllä

Julkaisun kieli: Englanti

Tiivistelmä

Monimutkaisen mallin 3D-mallintaminen pelinkehitystä varten on aikaa vievä tehtävä, koska on monia eri vaiheita, jotka on suoritettava ennen kuin mallia voidaan pitää pelivalmiina. Vaikka on olemassa monia työkaluja, joita voidaan käyttää joidenkin vaiheiden suorittamiseen tarvittavan ajan lyhentämiseen, ne ovat usein kalliita eivätkä siten houkuttelevia valintoja pienemmille kehittäjille. Yksi työkaluista, joka on sekä edullinen että kaikkien saatavilla, on fotogrammetria.

Tämän tutkimuksen tarkoituksena oli tutkia sekä ilmaisia menetelmiä luoda 3D malleja fotogrammetrian avulla, että myös verrata tuloksia erikoislaitteistoilla luotuihin. Tavoitteena oli luoda eräänlainen opas edellä mainituille pienemmille kehittäjille (ja pelinkehittäjille yleensäkin), jotka saattavat olla kiinnostuneita vähentämään 3D-malliensa luomiseen tarvittavaa työtä ja esitellä konkreettisia esimerkkejä mainittujen menetelmien lopputuloksista. Tämä saavutettiin tutkimalla erilaisia menetelmiä fotogrammetrian avulla luoda pelivalmiita malleja, testaamalla niitä käytännössä ja vertaamalla lopputuloksia suoraan.

Lopputulos oli, että dokumenttianalyysillä tehty teoreettinen tutkimus oli menestys, kun haluttiin löytää käyttökelpoisia fotogrammetrian menetelmiä. Käytännön toteutus jäi kuitenkin puutteelliseksi aikarajoitusten ja tuolloin saatavilla olevien rekvisiittausten vuoksi, kun kyseessä oli täysin ilmainen menetelmä, kun taas erikoislaitteita käyttävä menetelmä oli täydellinen menestys. Tulosten luotettavuus kärsi jonkin verran olosuhteista johtuen, mutta toinen yritys, jossa olisi enemmän aikaa ja laitteita, toisi epäilemättä parempia tuloksia.

Avainsanoja/tageja (aiheita)

Fotogrammetria, valokuvaus, 3D-mallinnus, pelikehitys

Muu tieto (Luottamuksellinen tieto)

Luottamuksellista tietoa ei kerätty tai käsitelty tutkimuksen aikana

Contents

Figures

Tables

Terms and Acronyms

Blender	Free 3D-modeling software
DSLR-Camera	A digital single-lens camera.
UV-Mapping	The 3D-modeling process of projecting a 2D image on 3D-model's surface (Deschatelets, 2016)
Meshroom	A free Photogrammetry software used to combine photographs into a textured 3D model. User interface is node based (https://alicevision.org/).
CUDA	Parallel computing platform that allows certain software to use the graphics processing unit (GPU) for general purpose processing (Abi-Chahla, 2008)
GPU	Graphics Processing Unit of the computer (Intel, n.d.)

1 Introduction

Game Industry has many different methods of creating 3D-models to be used in their games. Different workflows and tools, software and hardware. Some create models using the latest technology, sometimes spending thousands in their chosen currency to streamline the process and make the creation as efficient as possible. Other, smaller developers/Indie Game Development companies, use cheaper methods which are explored during this thesis.

The research topic was chosen according to the researchers own interest and to possibly act as a guide for anyone interested in Indie Game Development and in the possibility of using Photogrammetry as part of their 3D modeling workflow. The thesis will also be used for educational purposes in BIT Degree Programme as well as source material in Digi and Game Center.

2 Background of the Thesis

2.1 Objectives and Research questions

The goal of this research is to find what sort of differences show up in quality of the models when created using hardware/software available for everyone compared to using specialized hardware (such as DSLR-camera)?

The main objective of the practical part of the thesis is to create game development ready 3D-models using Photogrammetry combined with free software such as **Meshroom**, **Blender** and **Instant Meshes** and prove that they work in a game engine such as Unity and compare the differences in quality to. This is done to find out if Photogrammetry is something that Game Developers/Game Development companies should implement to be part of their 3D modeling workflow and what methods work the best.

2.2 Methodology

Thesis was done by using qualitative methods (Bhandari, 2019) and document analysis by analyzing articles, books, videos on the subject, etc. Quantitative methods through comparing performance statistics gathered during the practical research (Bhandari, 2021). The validity of the qualitative data was confirmed by checking multiple sources for information on the subjects and during the practical research by applying the information through direct testing. Practical research data validity/reliability was confirmed by running the same dataset with same setting through the software multiple times to prove it was repeatable if using the same hardware/software.

During the practical part of the thesis the creation of 3D assets using only free software and methods available to any indie developer was tested using methods learned during the research part. Differences in results between hardware used was also tested (laptop vs. Desktop PC, smartphone vs. Camera, etc.) during practical research by directly comparing the results both visually and through statistics (how much time did certain parts take, etc.).

The software and hardware were selected according to availability and ease of use. **Blender** is free and has literally hundreds of tutorials to its use across the internet making it a perfect 3D modeling software for beginners and more experienced users alike. **Meshroom** is also free and easy to use due to its simple node-based user interface. **Instant Meshes** automatic retopology software was chosen because retopology can be a time consuming and difficult process to get right and can be completely cut out by using automatic methods such as these. Instant Meshes was chosen as it is also free and simple to use with multiple tutorials in YouTube explaining how. With these software combined, even a beginner to 3D modeling and photogrammetry is capable of producing impressive results using them which makes them perfect if the goal is to test whether or not modelers wish to implement them into their own workflow.

As mentioned earlier, the goal is to research methods that are available for everyone and compare the results to those achieved using specialized hardware.

2.3 Information Gathering

Information was gathered by searching through different databases and Google search engine with terms such as: Photogrammetry, Photogrammetry in 3D modeling, Photogrammetric methods, 3D modeling, etc.

Theseus.fi was also searched using the aforementioned terms to see what sort of thesis there were already on the subject, alongside the JAMK web library.

3 What is 3D Modeling?

3.1 In General

3D-modeling refers to “computer aided design”, in which computer software is used to assist the modeling process of a three-dimensional object/character. It can be used for many different purposes such as Game Development, testing functionality of hardware parts to see if they fit together, to visualize different materials and object sizes and to create finished models of a product for marketing purposes. The main draw of 3D-modeling is the present things far more accurately compared to any 2D-creations (Petty, 2018). This overview of basic 3D-modeling workflow will help highlight the amount of work/time that using Photogrammetry as part of it will cut out.

3.2 Workflow

While many people use different workflows depending on how they learned to use the modeling software, the main process remains the same (Planning->Modeling->Unwrapping->Texturing->Optimization). Though optimization is something modelers should do thorough the project as it reduces the number of problems present at the end of the process.

3.2.1 Planning

The first phase of any efficient 3D modeling project is planning what you are going to create. Finding good reference pictures from the internet allows one to import them inside the modeling software itself to serve as a good idea of scale and shape of the model one is creating (SaanaFB, 2018).

Of course, when using reference pictures, you must keep in mind that some of them are protected by copyright and creating a model that is nearly an exact copy of the object/character depicted in the picture(s) may violate said copyright, especially if you are creating a model that is going to be used for commercial purposes. This is one of the reasons why most 3D modelers only use the general shape of specific characters or objects from the picture instead of fully copying them.

3.2.2 Modeling

The modeling phase itself is often the most time-consuming part of the project as it consists of creating the basic shape of the model and making sure the topology of the visible surfaces of the model remains as clean as possible to avoid any graphical errors (lighting/texturing, etc.). There are multiple types of 3D modeling that can be used to create the model in Blender (only limited by the amount of tools/addons available).

Some of the most common types of 3D modeling include (Selin, 2020):

- **Box modeling**

Box modeling is based on choosing a simple primitive (such as a cube, cylinder, or a sphere) and modifying it using loop cuts, extrusion, and other tools to create a base shape of your desired model.

Dividing the cube gives the modeler more vertices to work with allowing the creation of more complex shapes to work as the base of the final model. The exact workflow of this method is up to the modeler as different objects/characters require different approaches.

Box modeling is favored by beginners and veterans of 3D modeling as it allows the modeler to see the results of their efforts in a very short amount of time. And when combined with some reference pictures imported inside the software, one can very quickly create the base for just about any model by only using the basic tools of Blender.

After the base shape is finished it is time to add details. In box modeling this should be done layer by layer, adding the biggest details first followed by the smaller and more intricate ones. This detailing requires subdividing the mesh into smaller parts or addition of polygons using tools such as the knife tool and loop cuts.

The more subtle details and finishing touches are often done through sculpting as it allows addition of more organic details such as wear and tear marks on wooden/metallic/other objects and different small bumps and patterns created using brushes.

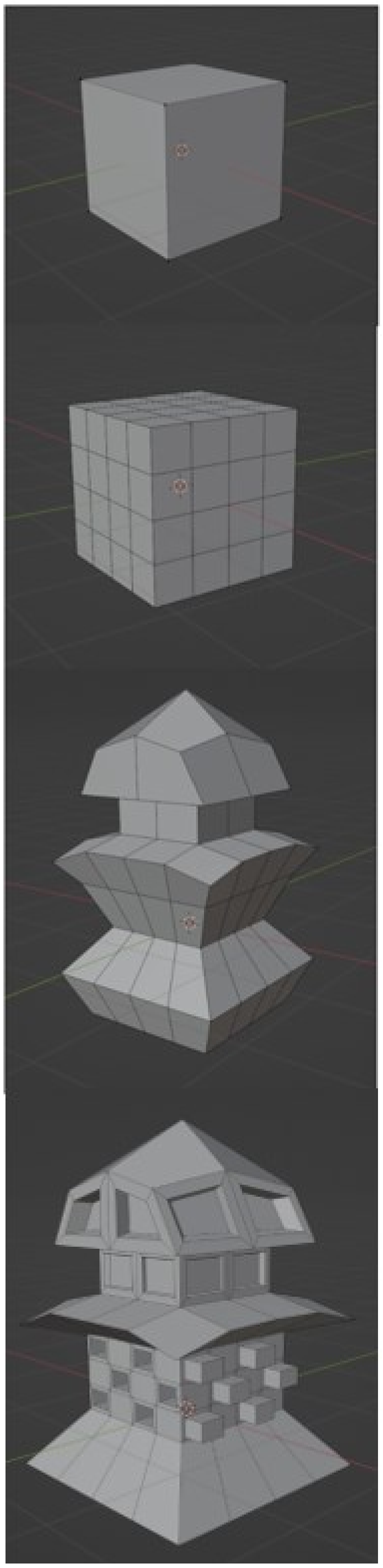

Figure 1 Very basic Box Modeling example

- **Digital 3D sculpting**

Instead of manipulating vertices, edges and faces based on selection, sculpting manipulates the mesh using different brushes. The brushes organically reshape the geometry according to the brush type and its settings. This technique is mostly used with in creating organic characters rather than objects, but it can still be used to add detail to objects that would be hard to achieved using box and polygon modeling.

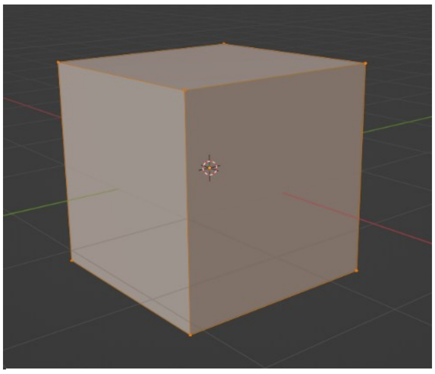

Figure 2 Sculpt starting point Cube

One thing about sculpting is that it requires a lot of geometry to be available from the start or usage of different modifiers/tools to create it. Modifiers such as multiresolution and Dynamic topology feature of Blender allow easy creation of more geometry from the low-poly base mesh.

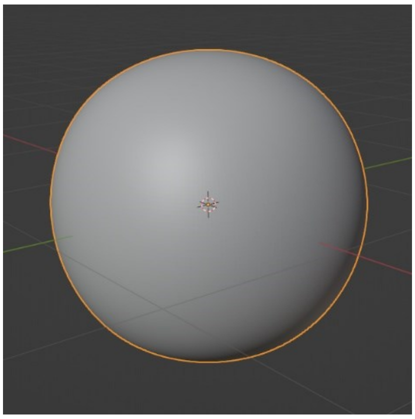

Figure 3 Same Cube after Multiresolution modifier

As with other modeling types, sculpting has multiple different workflows modeler can use to create the model they want, but the result often leaves the mesh itself in rather unworkable condition for game development purposes. The sculpted mesh is often high-poly and dense in vertices meaning it would be exceptionally hard to work with and any animation created with it is often far too demanding when it comes to performance to be useful. This requires an operation called retopology, which will be detailed later in the optimization section.

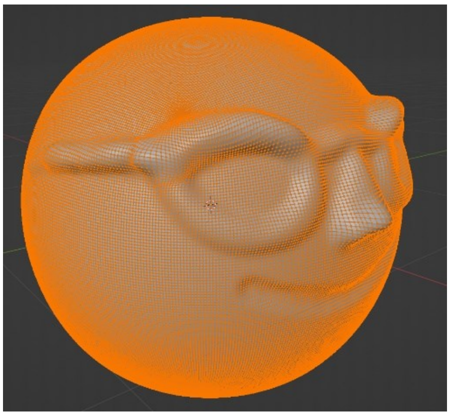

Figure 4 Dense high-poly mesh after sculpting (every orange dot is a vertex)

With the continuously developing technology, it is more common to first create a high-poly character model through sculpting rather than starting with box modeling a low-poly base mesh as the performance of hardware and software has increased to a point where this method has become more viable.

- **Kit bashing**

Kit bashing involves using a kit of ready objects that are then combined into more detailed object. This is often used when modeling more elaborate hard surface objects such as guns as they share many parts between different models, and it is faster to just combine different ready parts to form the object rather than modeling every single one from the beginning.

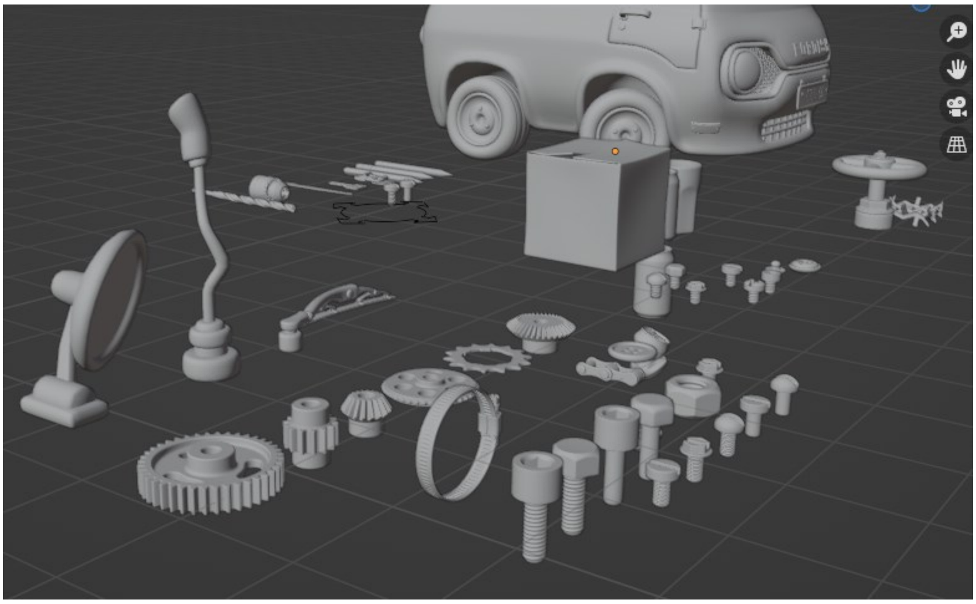

Figure 5 Example of free Kit Bash asset pack

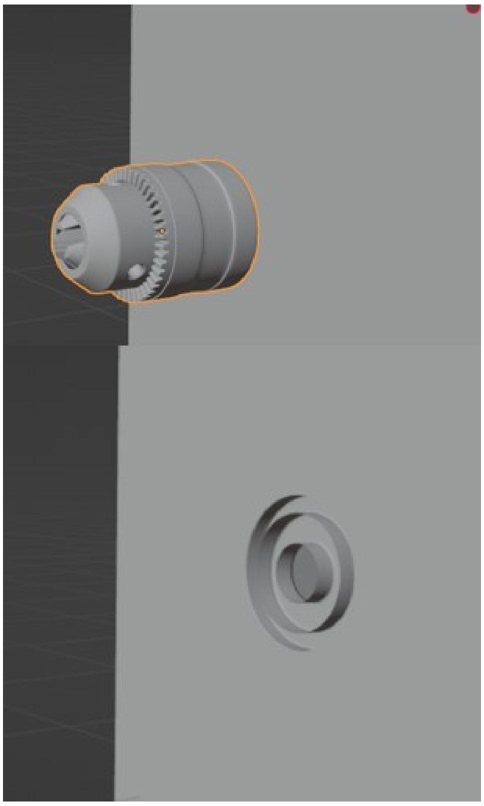

Figure 6 Cube with boolean cut

Kit bashing workflow can also be enhanced by using the Boolean modifier when combining parts to cut holes and create indents to objects to give them extra detail or to join them together.

Of course, this method is not restricted to simply making different objects as you can also find both free and commercial asset packs containing scenery assets such as modular buildings, nature assets like trees and bushes, etcetera. Using Kit bashing can save hours and hours of time when modeling a complex scene or an object.

3.2.3 UV-Mapping/Unwrapping

Figure 7 Trashcan model by Me

Concisely put, UV-Mapping refers to mapping of a 2D texture on a 3D object. The 3D model is unwrapped into 2D plane and 2D texture is applied to it according to the unwrap. There are multiple different methods of UV-unwrapping available in Blender both automatic and manual.

The automatic unwrapping is the quick method of creating unwraps but is not particularly accurate. It can be directed somewhat by changing the settings of the automatic process and by marking seams on the model itself, but the best results are gained by learning how to mark seams and unwrap manually.

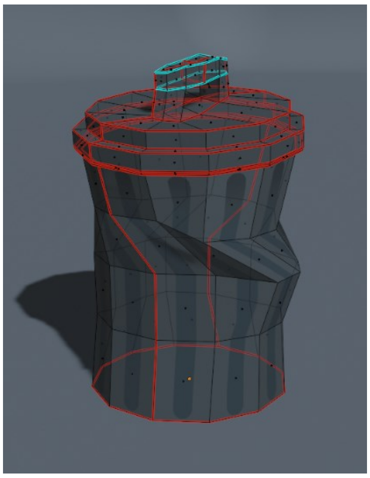

Figure 8 Trashcan model, seams in red

Resulting UV-map is then used as a tool to help spread out the texture over the model. It can also be exported outside Blender and used in other software.

3.2.4 Texturing

After the Unwrapping is done it is time to start applying the textures and modifying the unwrap when needed. Texturing itself can be accomplished inside Blender but it can also be assisted/made more efficient by using other software such as Adobe Photoshop, 3D-paint or other software specialized in 3D-texturing/texture painting. As a texture is essentially just a 2D image, it means that there are multiple different methods of creating one. The texturing method used is often dependent on the artists own preferences and whether the said preferred method/software can produce the desired result.

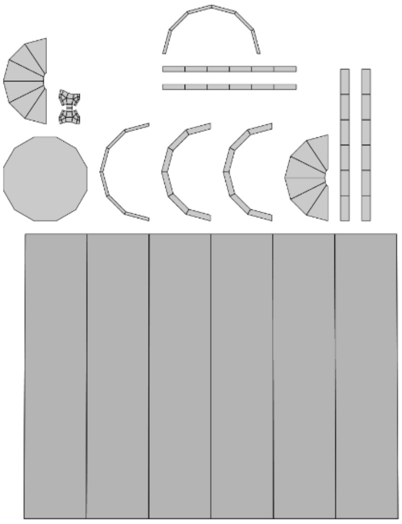

Figure 9 UV-map of a trashcan model

Using different Vector Graphics software to produce the texture is a quick way of doing things as it allows the artist to take advantage of the UV-map created during the unwrapping phase. The software can trace the lines of the UV map and create grids from the edges/faces that can be easily colored accordingly.

Texture painting is perhaps the fastest way to create a texture file inside Blender as it only requires an UV-map to be created before it can be used. Depending on the object/character and the UV map available, painting the texture can take even as little as few minutes to finish.

Blender allows one to paint the texture in 2D mode over the UV map (as shown in Figure 10) or the model can be painted in 3D mode directly using either Blender's own brushes (figure 11) or imported images that can be used as stencils (figure 12).

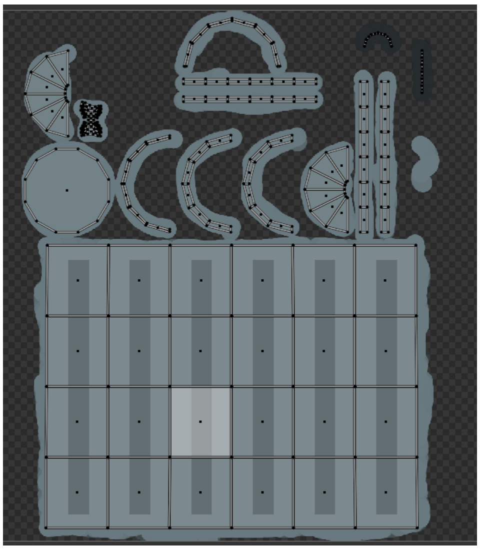

Figure 10 Trashcan model with UV map spread over texture made using Blender's Texture painting and Photoshop

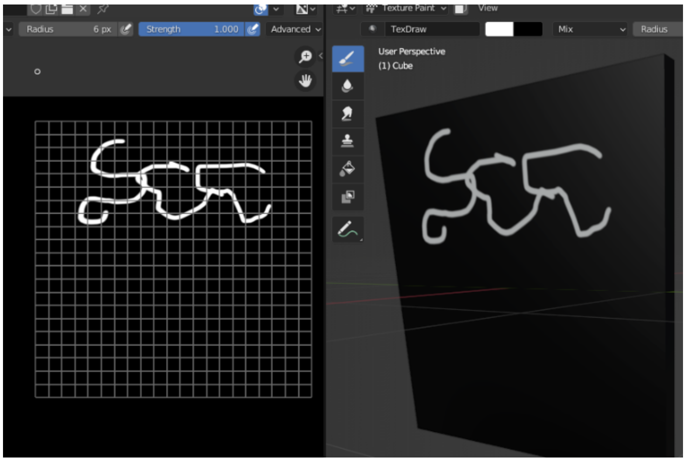

Figure 11 Basic texture painting using a brush

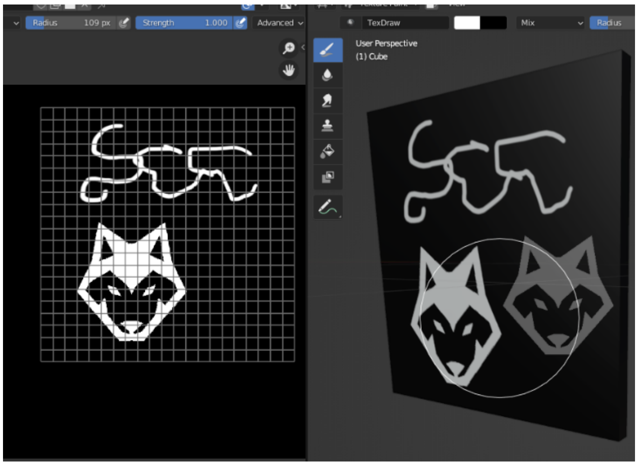

Figure 13 Stencil texture painting

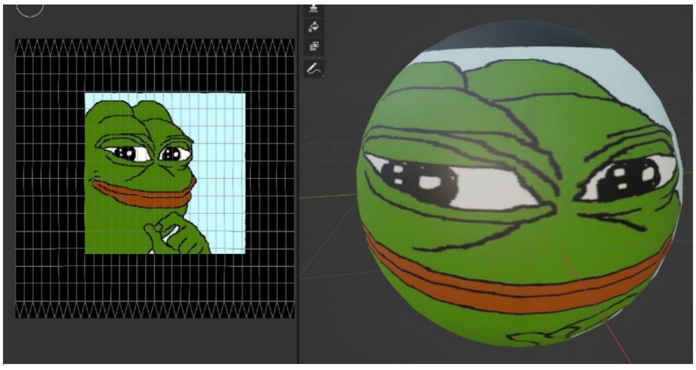

Figure 14 Stencil painting on top of the 2D texture

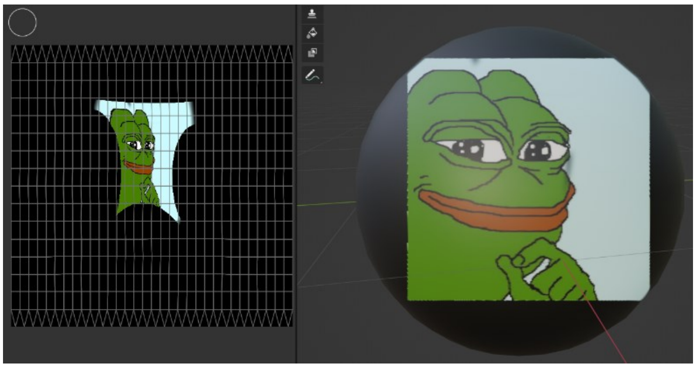

Figure 12 Stencil painting on top of the model

3.2.5 Optimization and polish

Optimization is something that should be done all the way through the modeling project, for an example, small bits during the modeling phase such as ensuring the edge- and face-loops are done correctly (especially when modeling a character). When creating a high-poly character/object it is a good idea to create a low-poly version for animation/game-engine purposes.

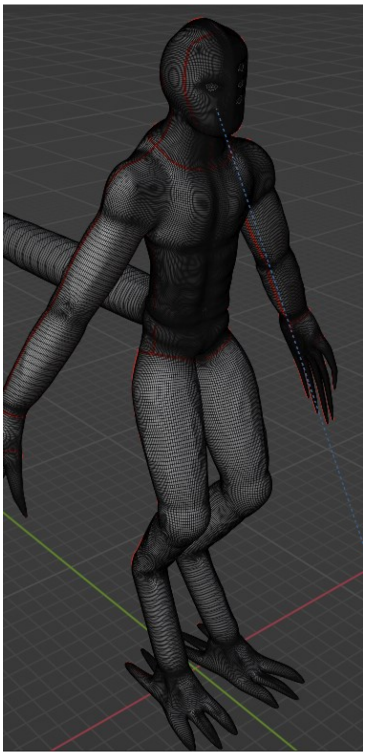

Figure 15 high-poly mesh monster mesh

Retopology is the process of converting high-resolution models into low-polygon meshes which are far more efficient for animation purposes. Essentially, either through automatic remesh algorithms or manually, the modeler creates a low-poly mesh on top of the high-poly one. The goal is to reduce the number of polygons as low as possible while still retaining most of the quality of the model. After the retopology has been finished and an acceptable low-poly version of the model has been created, there are still ways to make the model look as good as the high-poly one by baking the high-poly details to low-poly version. Texture baking in 3D modeling refers to “the process of transferring texture data from one model to another “(May, 2021, para 1).

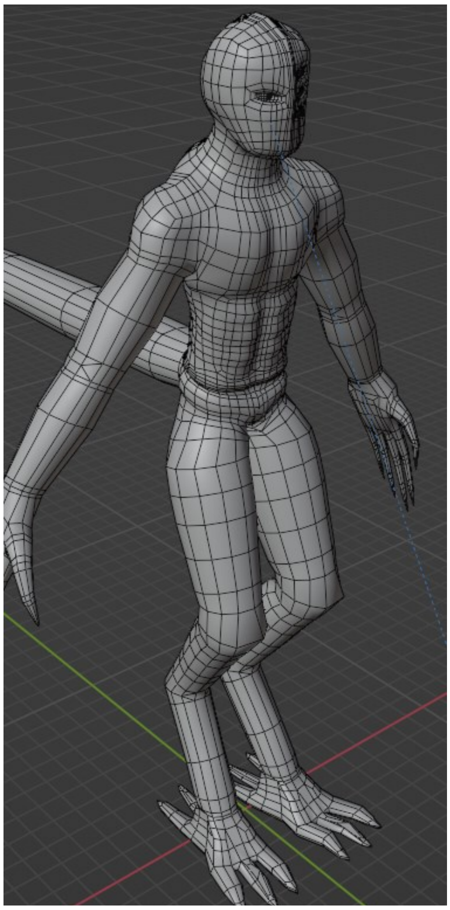

Figure 16 Retopo low-poly monster mesh

This allows the modeler to create Normal-, Bump-, Roughness- and Height(displacement) maps that can be used to create an illusion of detail on the low-poly model by simulating the surface’s details. Though displacement maps, as their name would suggest, can displace the mesh which they are applied to. And while they do give the greatest results when applied to a low-poly mesh, they are also significantly harder on the system and add lot more to render times compared to others making them an outlier.

To make the model as detailed as possible without making it too heavy to use inside a Game Engine (such as Unity) requires the modeler to combine these different maps and strike a balance between detail and performance. This becomes less of a concern over time as the hardware is constantly updated and the software made more efficient, but it is still something to be considered as the game worlds are also growing in scale and complexity.

- **Diffuse Maps**

Diffuse map is the most commonly used texture map as it defines the color and the patterns of the object. Essentially, mapping diffuse color is painting the image on the surface of the model (Trafagander, 2019). Good diffuse textures have no directional light painted on to them (different surfaces reflect light different ways) and any cracks, indents and holes should be darkened to help break up the image. Repeating patterns close together should also be avoided to make the model look organic/natural. Depending on modelers needs, diffuse maps can also be baked if needed.

Figure 17 Cube with Diffuse texture

- **Normal Maps**

A normal map is an RGB image that stores a direction at each pixel and said directions are called Normals. The red (X-axis), green (Y-axis), and blue (Z-axis) channels are used to control the direction of each pixel's normal which is why the maps look so different from other texture maps at first (Trafagander, 2019). These maps are used to simulate the subtle details of the surface such as scratches rather than bigger silhouette-altering details.

Figure 18 Cube with Diffuse and Normal map

- **Roughness Maps**

Roughness maps are used to simulate surface irregularities that would cause light diffusion, such as with reflections the rougher the surface of the object is the dimmer the reflections are. Depending on the software used, roughness may be replaced with smoothness, but they are essentially the same thing when it comes to texture maps, just inverted. These maps are used to give the object imperfections like stains or wear and tear (Trafagander, 2019).

Figure 19 Cube with Diffuse, Normal and Roughness maps

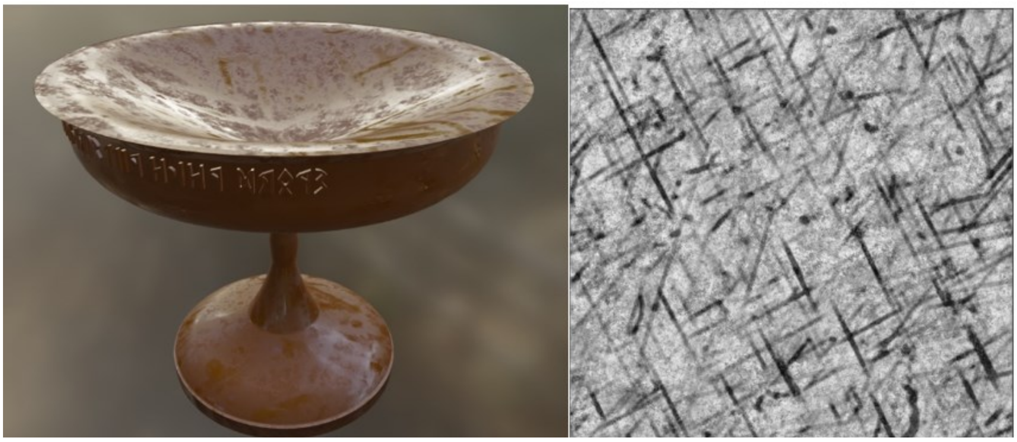

Figure 20 Example of Roughness map a on different model (effect more visible)

- Displacement Maps (Combined with Diffuse and Roughness map)

As mentioned before, displacement maps are the more powerful choice to creating details as they can deform geometry making them THE choice when it comes to creating photorealistic models and generating terrain (called height map for large-scale mesh deformation). The downside of this is the forementioned problem of being computationally intensive and thus not the best choice for animated characters/models in performance demanding things like video games (Griggs, 2019).

Figure 21 Cube with Diffuse, Roughness and Displacement maps

4 What is Photogrammetry?

4.1 History

Photogrammetry has a history reaching all the way to the 1480s to the time of Leonardo da Vinci was the first to work on the subject during his research on perspective/projective geometry followed by other scientists according to The Center for Photogrammetric Training (n.d.). Around 1849, a French scientist Aimé Laussedat was the first person to use terrestrial photographs to create the first topographic map compilation using process that was called iconometry which was essentially the precursor to Photogrammetry.

The first time the term “Photogrammetry” was used happened in an article published in 1867 by a civil engineer/architect Albrecht Meydenbauer (Albertz, 2001b, p. 21) who had been working on the photogrammetric methods in his free time for years and designing his own camera while doing so. The camera was the first wide-angle lens used for mapping and was used to create the topographic map of city of Freyburg in Germany alongside the structural drawing of St. Mary’s church.

As stereoscopy was becoming more widely used and the Wright brothers developed their airplane in 1903, photogrammetry was getting its next big development leading to creation of analog photogrammetry and the first aerial imagery taken by Wilbur Wright. This led to the first aerial mapping photography by Captain Cesare Tardivo of Bengasi, Italy (The Center for Photogrammetric Training, n.d., p. 9). Many innovations then followed in the field of photogrammetry over the years, in both the technology and the methods used, until the modern **Digital Photogrammetry** was developed.

4.2 Basics of Photogrammetry and Photographing Methods

According to simplest definition I could find: "Photogrammetry is the art and science and of extracting 3D information from photographs. The process involves taking overlapping photographs of an object, structure, or space, and converting them into 2D or 3D digital models" (Autodesk, 2021, para 1). To put it simply, the modeler takes pictures of the chosen target from all the angles that have details that they wish to capture and use a photogrammetry software to convert them into a single 3D model. This allows the modeler to add more texture and features to the model than is reasonable using the traditional 3D modeling methods, thus creating Photorealistic 3D models.

There are many ways of acquiring the required pictures and the method used can be determined by examining the target. Big targets such as villages, dig sites and other large structures often require the use of a flying drone as walking around and taking pictures becomes inefficient. Smaller targets, that can be captured from all angles by a person, are often photographed by using two different methods (Holmes, 2017):

- **Shooting while moving the camera around an object**

This is a viable method especially when taking the photos outside as you often have more room to move around in and most objects that are photographed this way are either a bit too big to film inside or things that are not

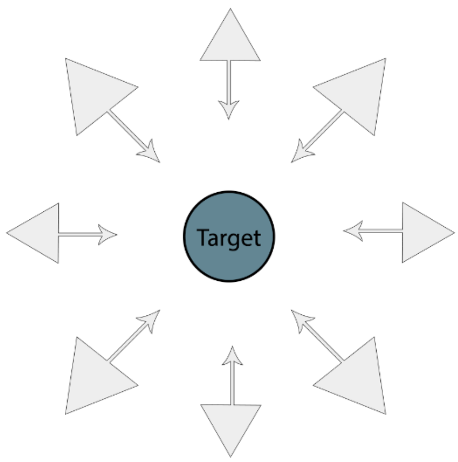

Figure 22 Moving the camera around the target

meant to be moved. Public statues, small landmarks, tree stumps, boulders and many other objects that can't/shouldn't be moved can be filmed this way. Though, while shooting outside, the results are dependent on the something as random as weather making it a risky choice.

One thing that must be considered while shooting outside is the lighting. When shooting around the object it must be lit all the way around and that can be difficult to achieve while taking photos outside. To get the best results, the photos should be taken at noon with overcast skies as the even layer of clouds will help to diffuse the light in a way that it is spread evenly around the object and that there are no random shadows ruining the shot (Holmes, 2017).

- **Rotating the object while camera remains stationary**

This method is mostly used when capturing smaller objects and when done right produces the best results. Unlike when shooting outside, this type of photogrammetry requires a prepared studio to shoot in. Said studio can either be created using professional equipment or using things you have laying around, but the DIY method needs a lot of planning to produce high-quality photos/models.

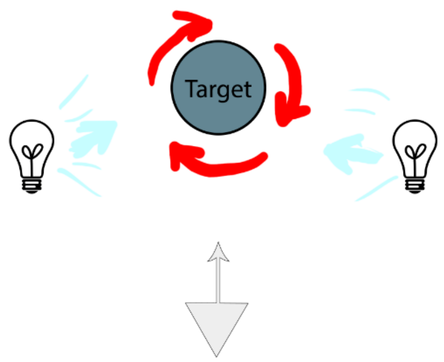

Figure 23 The camera remains stationary while the object rotates.

The background must be perfectly blank and big enough to allow the photographer to take pictures from high and low angles, which can be difficult with a slightly larger object. The model also must be lit up from the front and the top, while avoiding creating odd shadows/highlights on the object and background. This requires lights to be diffused with the help of other slightly transparent props (Holmes, 2017). The color tint of the light matters as well as it affects the end texture of the object and if the lights are slightly different tints (example. warm light on the left, cold on the right) it will confuse the photogrammetry software as it tries to guess which direction the photo was taken from (foreshadowing for the practical phase).

While the exact number of photos depends on the model used, in most cases it is recommended to take at least 25 to 250 or even more pictures of the target. The more pictures the software has to process, the better the quality of the model (most of the time). On the flip side, this also means it takes much longer for the less powerful computers to process them and generate the model.

4.3 Usage in General

As mentioned before, the earliest applications of photogrammetry were in mapping. In the modern era, photogrammetry has been used in various other fields as well such as entertainment, engineering, medicine, manufacturing, architecture, and forensic analysis. Despite there being other methods of creating 3D models of environment/objects such as laser scanning and the developing 3D scanning apps for phones, photogrammetry remains used as it is above all else, affordable, and accessible (PhotoModeler Technologies, 2020).

4.4 Photogrammetry in Game Development

Photogrammetry in game development is used to create photorealistic models both in shape and texture efficiently compared to trying to create them from nothing. It captures the exact shapes and formations of the target object's surface making the result more immersive experience for the players and less work overall for the developers.

5 Photogrammetry on smaller objects: The equipment and camera settings

5.1 Choosing the camera

When shooting photogrammetry, the choice of camera matters as it has a big impact on the quality of the pictures and the model. A DSLR-camera is a good choice for those who can get one, as it allows the manual adjustment of camera's aperture, focus, white balance, and ISO to fit the lighting environment. A smartphone with a decent camera will as well if you can't, but to get consistent and quantitative data, a DSLR-camera is required (Haines, 2019).

The lenses are also something that need to be considered as lenses lower than 24 mm in focal length start to cause distortion around the edges of the frame (Haines, 2019). Using prime lenses gets the best result as they have a fixed focal length (no zooming) which minimizes the noise and distortion in the pictures.

For smartphones, you shoot with what you have. If it is possible to borrow a phone with a better camera and more settings for manual mode, then do so.

5.2 Settings

The 5 major things to take into account while using a DSLR for photogrammetry are: Zoom/Focus, ISO range, white balance, aperture size (f/stop), and shutter speed. While the exact settings you need depend on the shooting environment, these are the most important things to keep an eye on (Haines, 2019):

- **Zoom/Focus**

Both zoom and focus must stay consistent thorough the photo shoot, which is why it is highly recommended to make sure that these do not change by setting both of them at the start to frame the target and taping the focus setting down using painter's tape. It may be necessary to adjust the focus while moving the camera to different heights, but it is important to not change anything else while doing so. Setting the focus can be somewhat difficult on smartphones as it requires one to have a rig to keep the phone in place (which most people don't have) as otherwise the focus will be adjusted automatically. This happens constantly which makes getting unblurred photos difficult.

- **ISO range**

The camera's ISO range affects its sensitivity to light and as photogrammetry requires the pictures to have as little noise as possible, it is important to set the ISO range low (lower the ISO, the less light sensitive the camera is, the less noise there are) to somewhere around 100-200 range. Some of the older smartphones can't change this (mine can't) but it seems to be dependent on the model.

- **White Balance**

Setting the white balance helps in rendering the white in the photo as it should be. Thus, it is important to adjust the white balance according to environment. If the photo shoot is happening outside, change the white balance accordingly. Same with the inside shoot. Smartphones can change white balance either automatically or manually.

- **Aperture (f/stop)**

Aperture controls the size of the opening of the lens on the camera which changes the amount of light it lets in. The higher the f/stop value, the smaller the opening is. This is important as it affects the depth of field with the higher values making the depth of field longer which helps getting as much of the target object in focus (Holmes, 2017). It is recommended to set it around f8 to f11.

- **Shutter speed**

The shutter speed affects the amount of time the shutter on the camera is open and is measured in fractions of seconds (Haines, 2019). Setting this can be tricky as slower shutter speed means that any movement can cause blurriness in the pictures as it means low amount of light due to small aperture and low ISO settings. Using a tripod with DSLR camera gets the best results as it eliminates most of the movement, but with smartphones trying to shoot photos without dedicated rig is difficult and adjusting shutter speed is not possible on some of them (my phone can't change shutter speed). Recommended to set it to slow (1/8) if stable camera stand such as a tripod is available.

6 Creation of 3D assets using photogrammetry

6.1 Practical research

The goal is to learn to create 3D-models using photogrammetry and compare the results of doing so with different methods. Using tools available for everyone and specialized tools allows direct comparison between the quality of the 3D-models created and the efficiency (through statistics) of both methods.

6.2 Workflow using tools available for everyone

6.2.1 Preparation phase

Preparation first required construction of a DIY photogrammetry studio. This took a while as finding right type of lights to illuminate the model evenly, objects to place the model on, turntable, tripod for the camera, light diffusers and other props turned out to be difficult. Luckily though, there was no need to buy anything new for the project as there were many substitutes available inside the garage and around the house.

The target was chosen based on availability (from the shelf in the woodshed) and only required small adjustments before being ready to be photographed.

On the hardware and software side, research was conducted on the recommend photogrammetry practices (as shown in the theory part), and on the workflow of importing the photos inside Meshroom, model to be polished/textured in Blender and lastly to Unity game engine (Serino, 2019).

Figure 24 The DIY Photogrammetry studio V1, another light was added above the model and the light on the left side replaced with similarly tinted light as on the right

6.2.2 Photographing phase

As we are not using specialized equipment, filming happened by placing the smartphone on top of the camera stand and steadying it by holding it. The object was turned on top of the rotating stand with picture taken approximately every 10 degrees turn. This was repeated until the model had been photographed from every angle which took 36 pictures (360 degrees). The photos were saved in JPEG format as the phone didn't allow saving in RAW format.

To ensure that the Meshroom Photogrammetry software had as much picture data to work with as possible, photos were also taken from and elevated angle with camera pointing down at the model from an approximately 45-degree angle. Another 36 pictures were taken to ensure the maximum coverage of the model.

Figure 25 The target set on top of a roll of toilet paper on top of an electric gramophone used as a turntable. DIY at its best

6.2.3 Combining the photos to form the model (Meshroom Laptop)

The pictures were imported inside the Meshroom to begin creating the model which at first presented some problems as multiple errors and crashes happened before being resolved. The lacking processing capabilities of the laptop caused the program to crash if too many pictures were imported at once. The lacking hardware later one became one of the main problems of using the "free" method.

Another problem emerged when starting the process as the integrated GPU (Graphics Processing Unit) of the laptop was not capable of using all the features of the Meshroom and time had to be spent troubleshooting the issue. Instead of being able to use default method of processing the images (which used **CUDA)**, the process had to be done using a less accurate "draft meshing" method ("Meshroom manual", n.d.).

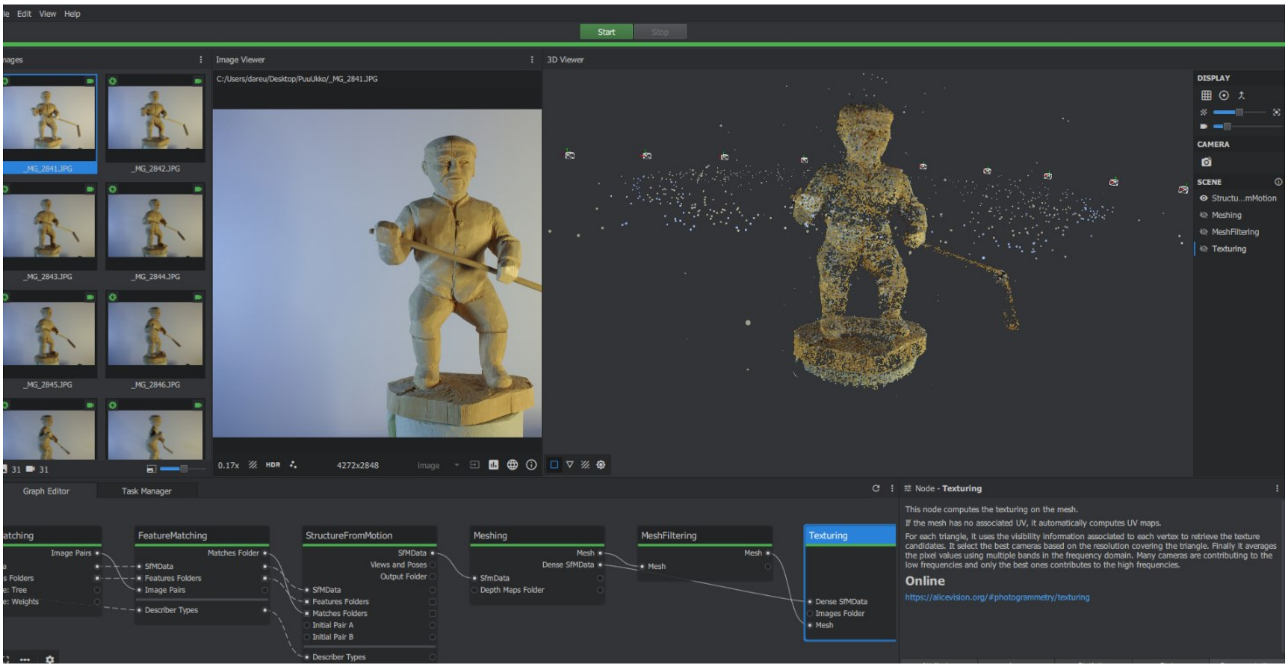

Figure 26 Testing if Meshroom recognizes the target and the camera directions with less pictures

This method allows makes it possible to do a reconstruction without using the Depthmap node of the process, but the resulting mesh is lower quality. In an attempt to make things work, the Feature extraction node settings were set to high, and three the unusable nodes removed (Figure 28).

As can be seen from the picture on the right, the first try at draft meshing didn't create an end result that would be acceptable for game development purposes. The software itself rejected 5 of the pictures as their quality was not good enough which was part of the cause for the missing pieces of mesh.

The whole process took 21 minutes from start to finish.

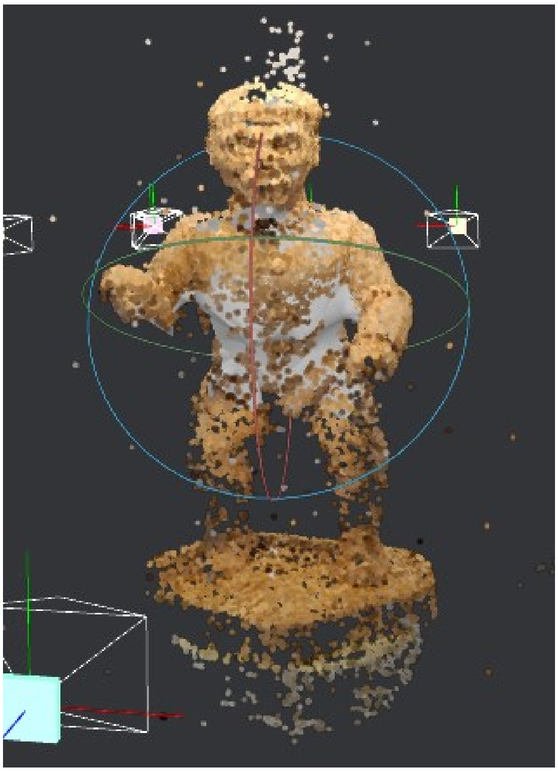

Figure 27 First attempt at Draft meshing, didn't even make it to Blender

6.2.4 The second try at Draft Meshing, now with more knowledge gained

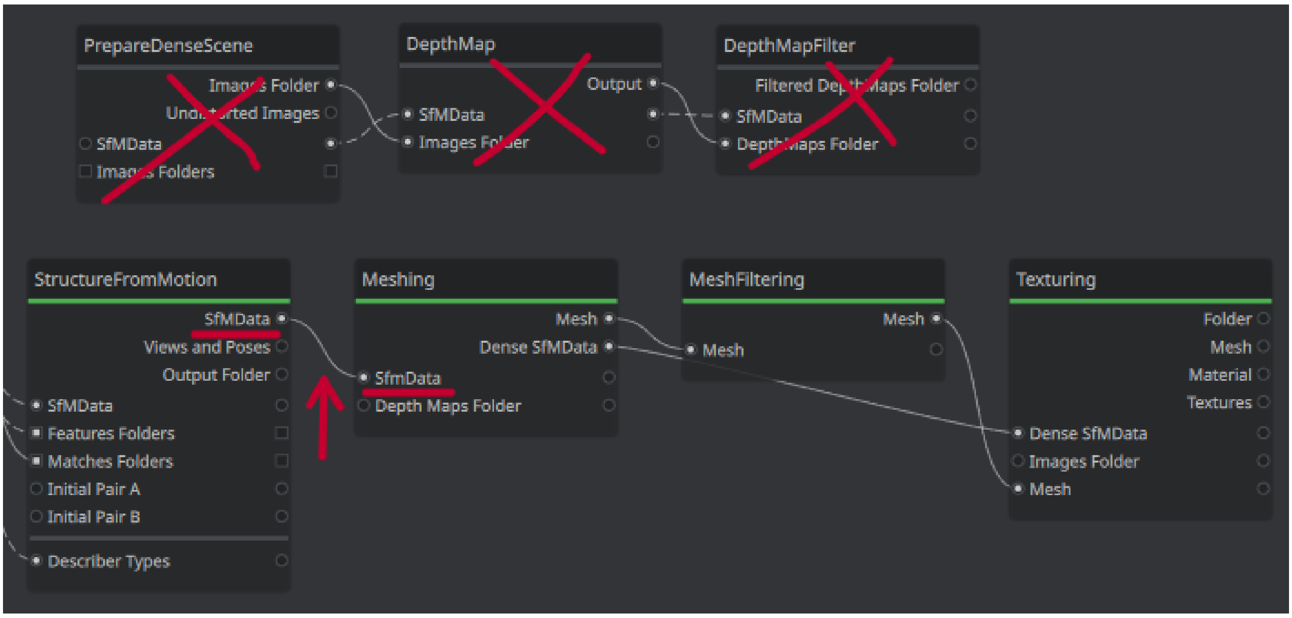

Figure 28 Meshroom Draft meshing: remove crossed out nodes and connect StructureFromMotiont to Meshing node

Figure 29 One of the Pictures taken with improved lighting

Instead of trashing the first try, I decided to leave it in as it shows the development of the methods and workflow that happens during the practical research. One of the major issues during the first try was the differently tinted lights which confused the software as it tried to understand the why the surface color tint stayed the same despite the target turning.

After setting up the nodes as needed and moving the importing the pictures (5 at a time to avoid crashing) the process begun again. Using JPEG format instead of RAW (as it is unavailable). The correctly tinted lighting and steadier shots were implemented to increase the quality of the pictures. The feature extraction node's setting preset remained in high preset as a short test using Ultra caused the laptop to almost freeze completely.

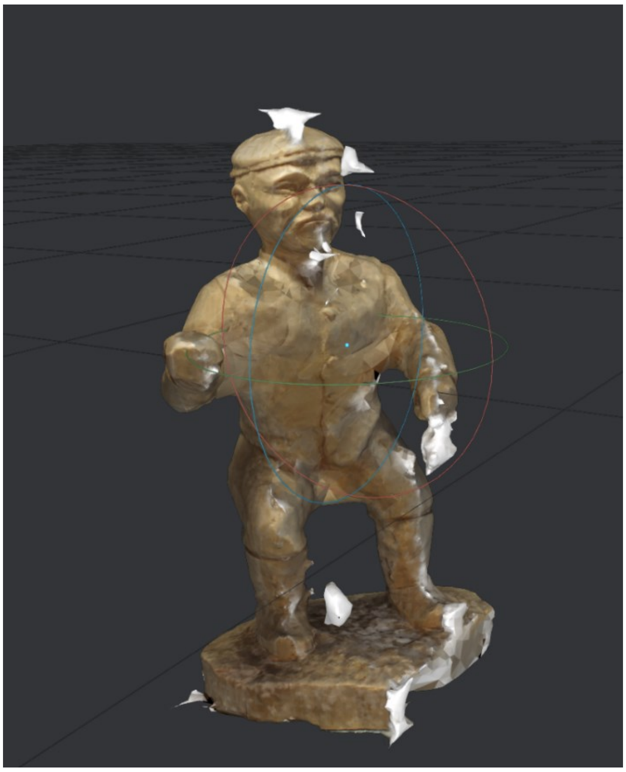

Figure 30 The second attmetmt

The second try went better, but as can be seen from the picture it still didn't produce a workable result. The floating pieces of mesh and some of the texture missing leave this attempt a failure as well. While it would be possible to transfer the model to Blender to work on it and paint the missing pieces of texture on the model, the purpose of the process is to create a model that can be run through rest phases without adding extra work.

Time taken: 19 minutes

6.2.5 One final try at Draft Meshing, this time outside

As the current setup was refusing to work with smartphone and the time was running out (the researcher had to move back home the next day) it was decided to make a final attempt at Draft Meshing by taking the pictures outside.

Figure 31 Outside pictures

An overcast sky served to diffuse the light to create perfect conditions for an outside shoot. The photos were taken while moving around the target and steadying the phone on top of a tripod. While the results of the shoot can't be directly compared with the results of photos taken inside using the DSLR they still serve as an example of what can be created using Draft Meshing method when it works.

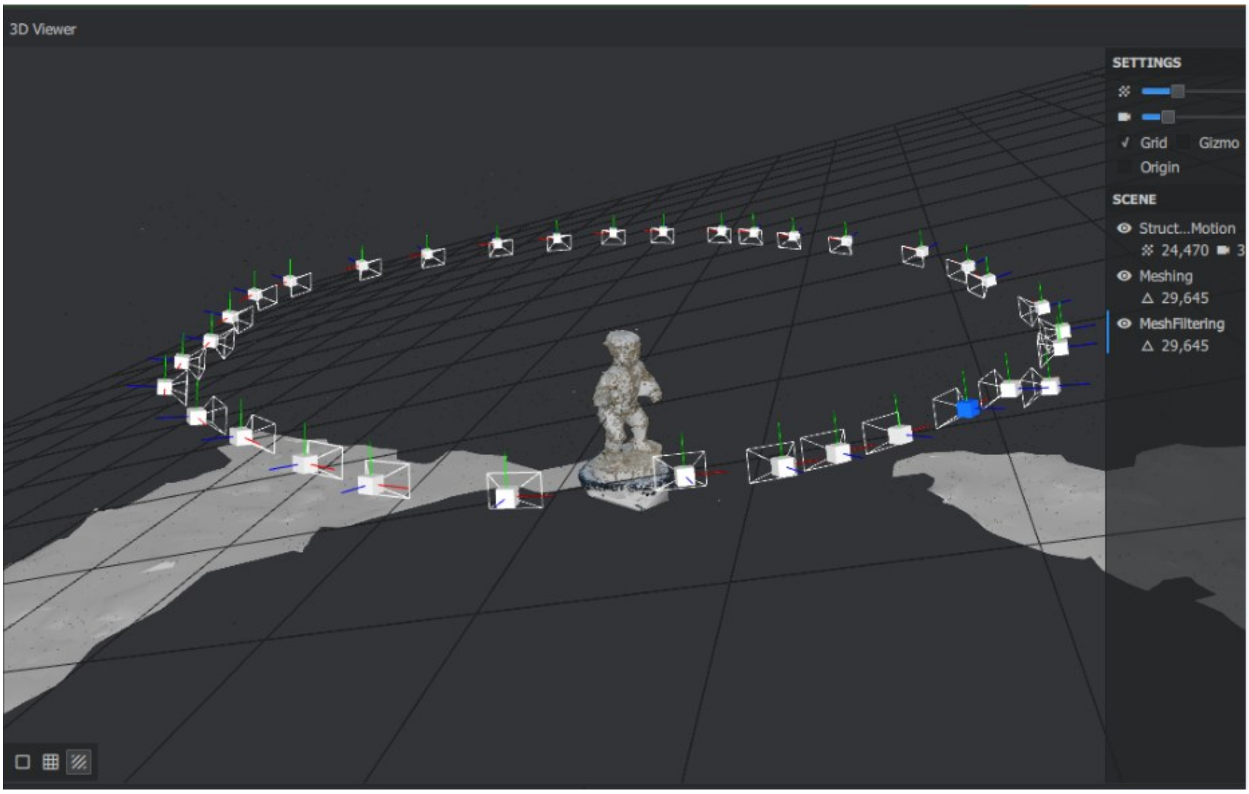

Figure 32 Running the photos taken outside through Meshroom, only the horizontal ones at first

Figure 33 End result of Draft meshing outside photos (high-poly)

The result of the outside photos was acceptable as it produced a model without missing textures or random pieces of mesh either floating around it or missing from the main body.

Time taken: 17 minutes.

6.2.6 Optimizing model for Game Development purposes

After Meshroom had generated a working model, it was time to generate a low-poly version of it by using **Instant Meshes,** an automatic retopology software as the model generated by Meshroom is too heavy for game development purposes as seen from the picture on the right (every black dot is a vertex, 178 938 vertices exactly). The generated model is located within the Meshroom cache folder chosen for the project (project folder\Texturing\folder) inside one of the folders in the Texturing directory.

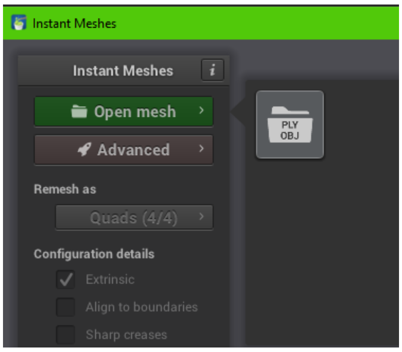

Figure 34 Importing the .obj to Instant meshes

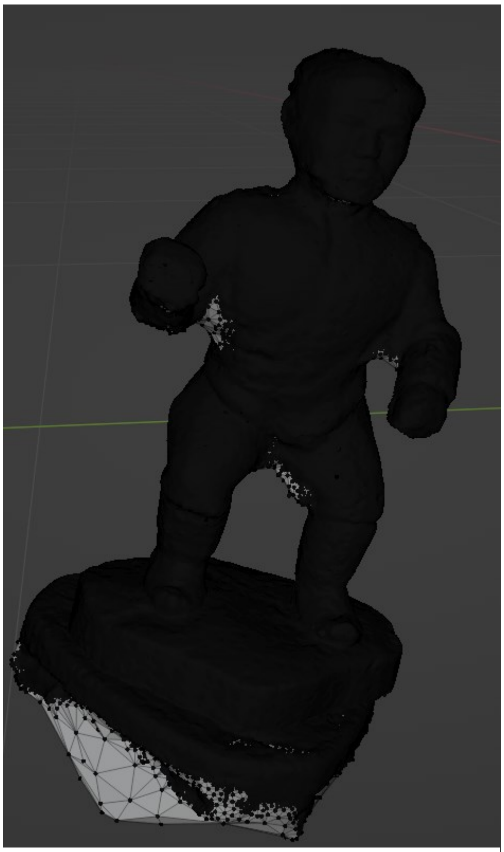

Figure 35 The high-poly model inside Blender, Vertex example

Instant Meshes works with .obj files which can also be found within the Texturing folder.

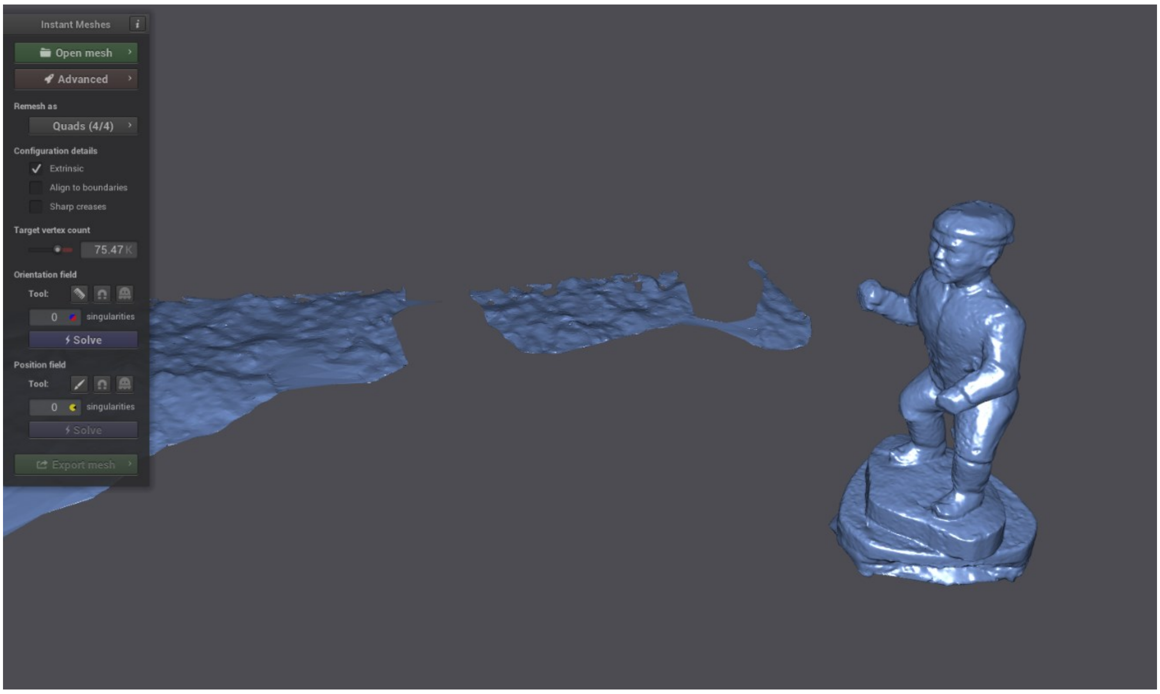

Figure 36 What the model looks like inside Instant Meshes

The software has an easy-to-use interface that allows the user to reduce the amount or vertices by changing the “Target vertex count”. In this case, the target vertex count was set around 75 thousand as many of the vertices could be found in the surrounding terrain that was picked up by Meshroom. With this model, using default settings was perfectly fine and the process could be started by pressing both of the “Solve” buttons.

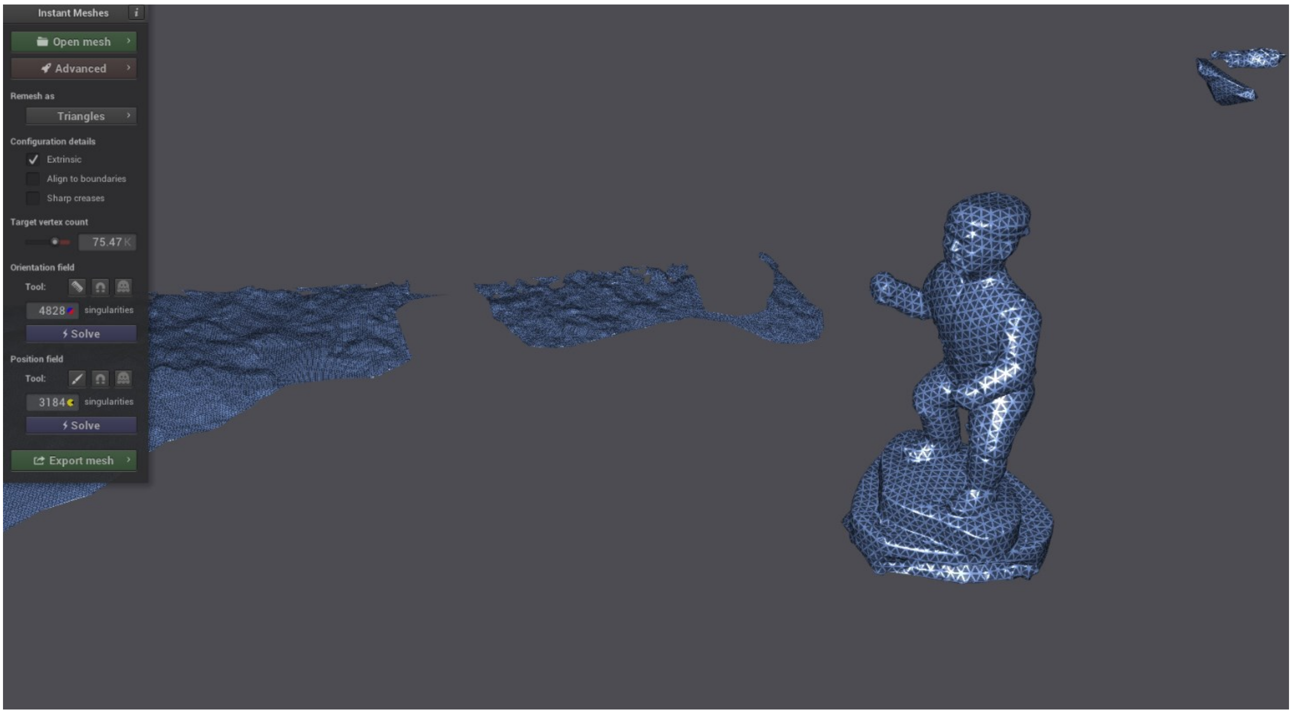

Figure 37 Instant Meshes dividing the mesh into triangles while attempting to preserve as much detail as possible

The resulting low-poly mesh can then be imported into Blender to be worked on.

Time taken: 15 minutes

Figure 38 Instant Meshes low-poly version ready to be exported to Blender

Figure 39 The Imported high-poly + low-poly models overlaid inside Blender

When importing both the high- and low-poly meshes inside Blender it is recommended to avoid moving, rotating, or otherwise manipulating them even though they are always upside down for some reason. If left alone, they will sit perfectly on top of each other when imported which makes the following actions much simpler.

The first thing to do after importing both models is to choose the low-poly mesh and clear sharp from all the edges inside edit mode as models generated by Instant Meshes have all the edges marked sharp (Figures 40 and 41). This allows the use of Smooth Shading giving a better representation of what the model looks like.

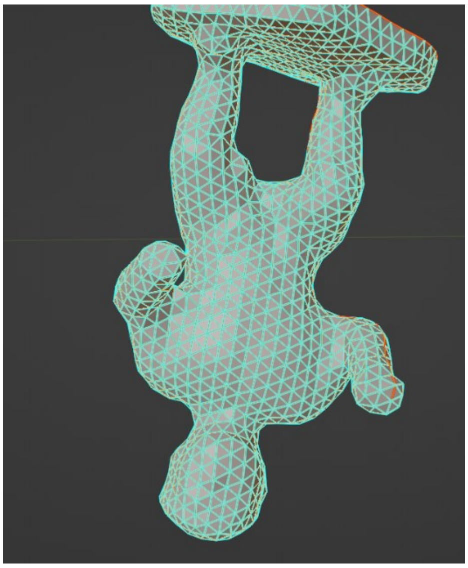

Figure 40 Edit mode with edge select on (blue edges mean sharp)

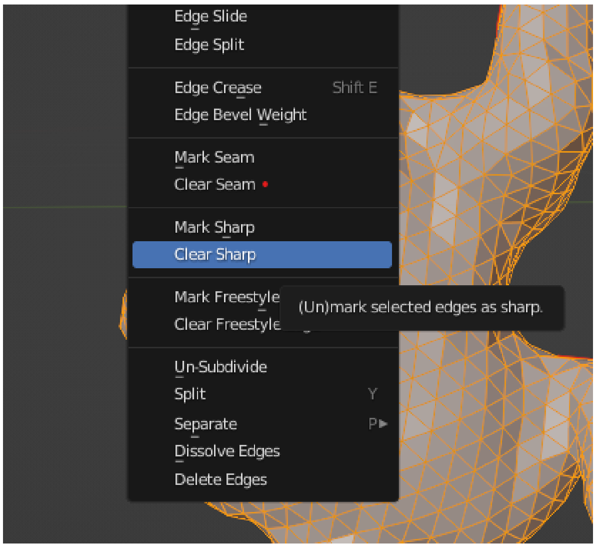

Figure 41 Clearing the sharp by right-clicking and choosing the option "Clear Sharp"

After that, it was as good time as any to clean up the model by deleting unnecessary vertices/mesh from the low-poly model and leaving only the parts that are going to be used.

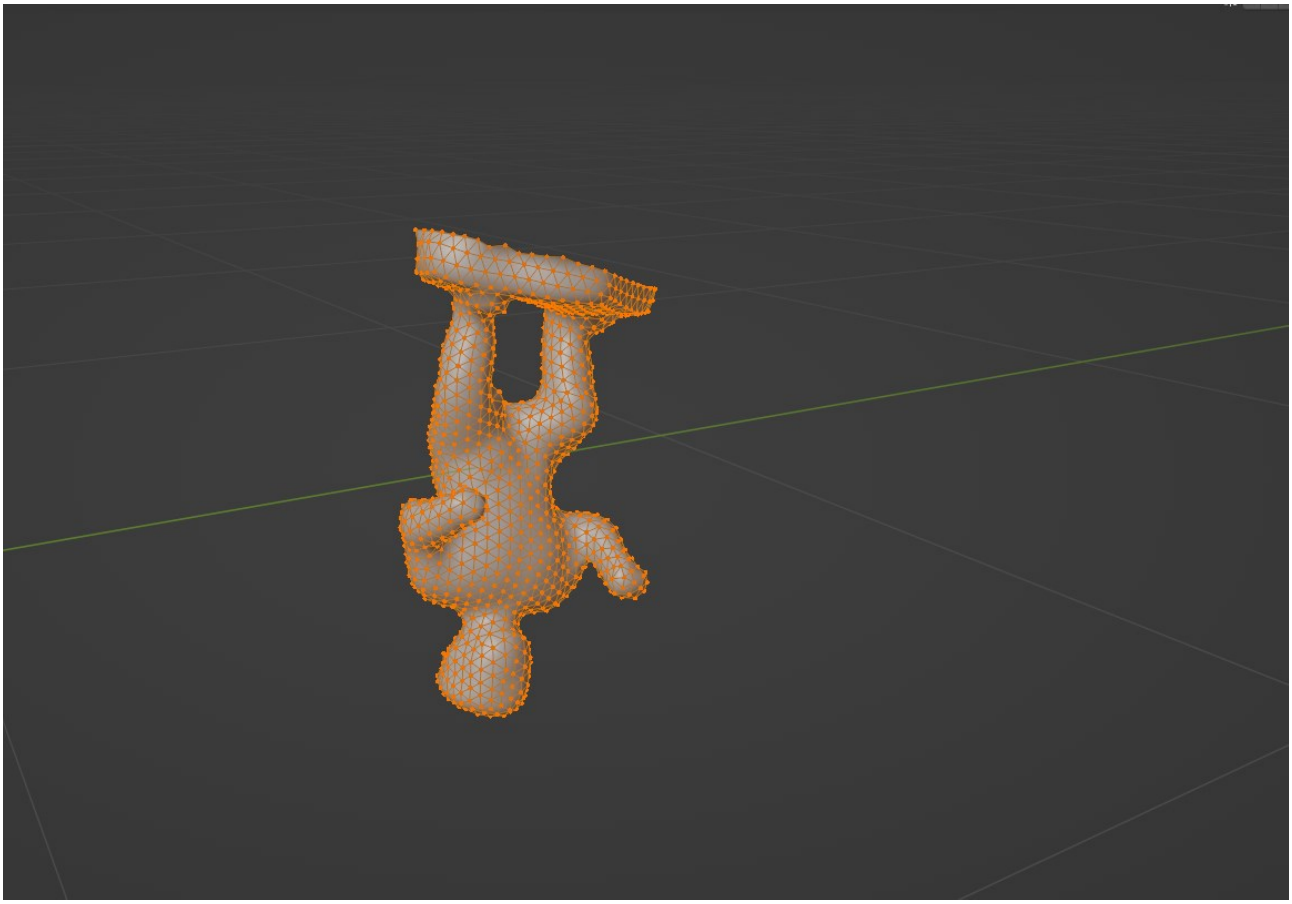

Figure 42 All the floating pieces of mesh have been deleted and the bottom of the model cleaned

With the mesh clean, it was time to start thinking about texturing, or more precisely, UV-mapping.

It is a good idea to try and hide the seams in the harsh edges but as the model lacks hard edges the seams are instead marked on the parts that are rarer to be seen by the player. In this case the seams were placed around the back and at the underside of the arms.

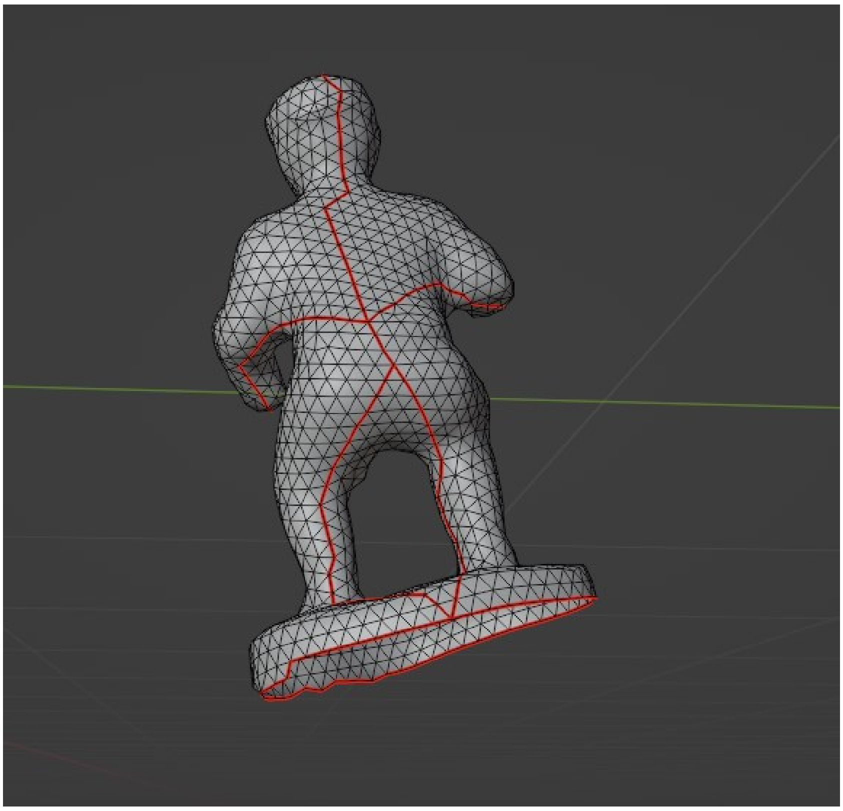

Figure 43 Edges marked with red are seams

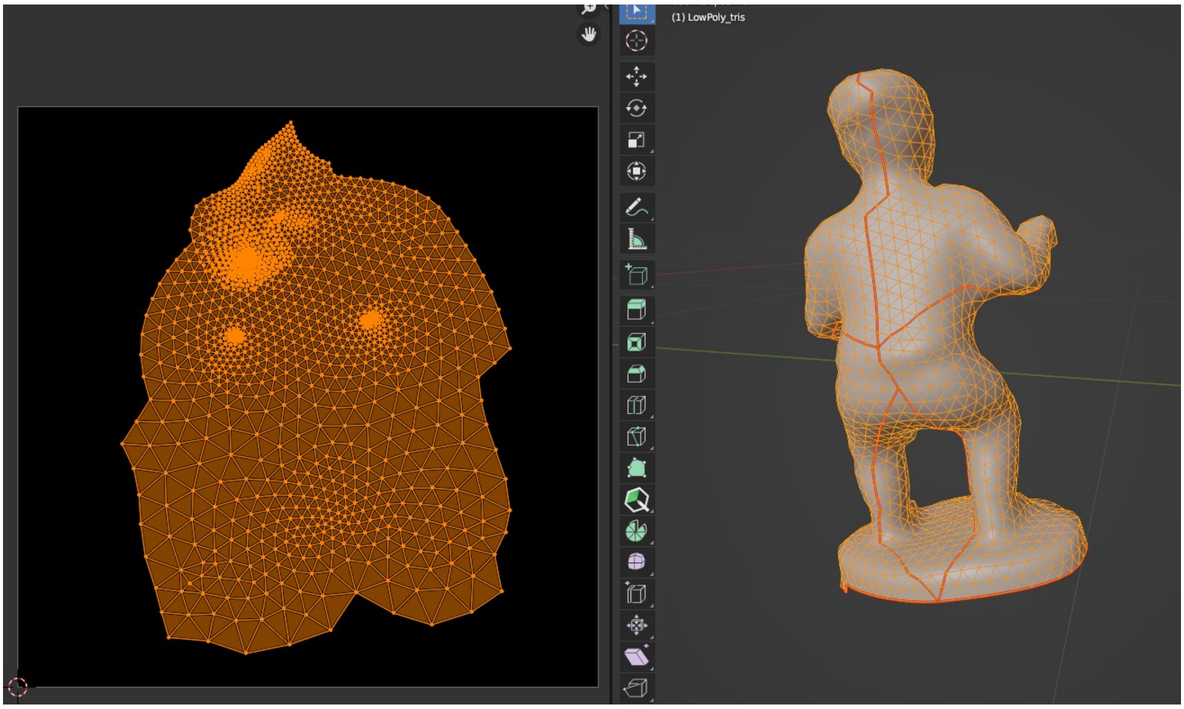

Figure 44 Low-poly UV-unwrap with a new 4096x4096 texture

The next step of the process was to bake the diffuse and normal textures and for that using a Cage to bake them was the best option as there are a lot of mesh poking through the low-poly version when overlaid with the high-poly mesh (Figure 39). Cage is a ballooned-out version of the low-poly mesh that is created by duplicating the low-poly mesh (cage needs to have the exact number of vertices as the low-poly) and scaled in edit mode (Alt + S with all the faces of the cage selected) by the Normals to be just big enough that there is no longer mesh clipping through anywhere (Blender Documentation Team, n.d.). Note. Remember to uncheck auto-merge vertices before scaling the cage.

When the cage was ready, it was time to bake the Diffuse texture and the Normal map.

Figure 45 Cage created from the low-poly mesh

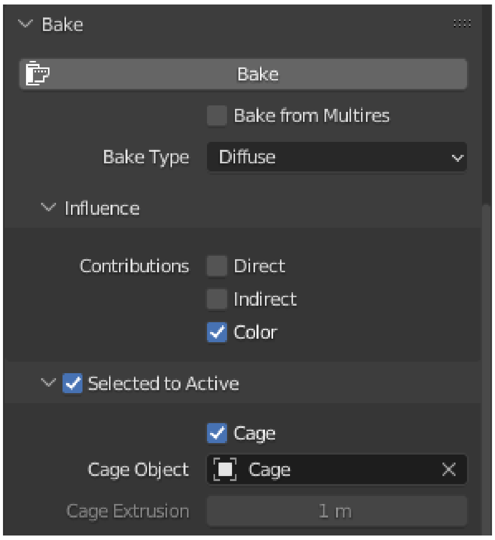

Figure 46 Cage settings inside Render Properties

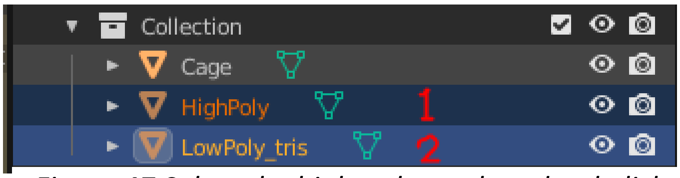

Figure 47 Select the high-poly mesh and ctrl-click the low-poly

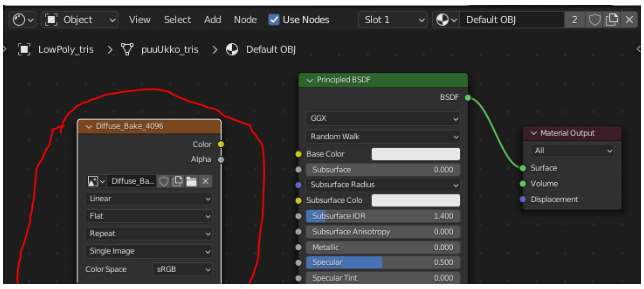

Figure 48 Make sure that the Diffuse bake is selected in the Shader editor and press Bake in the render tab

Figure 49 The Baked diffuse Texture

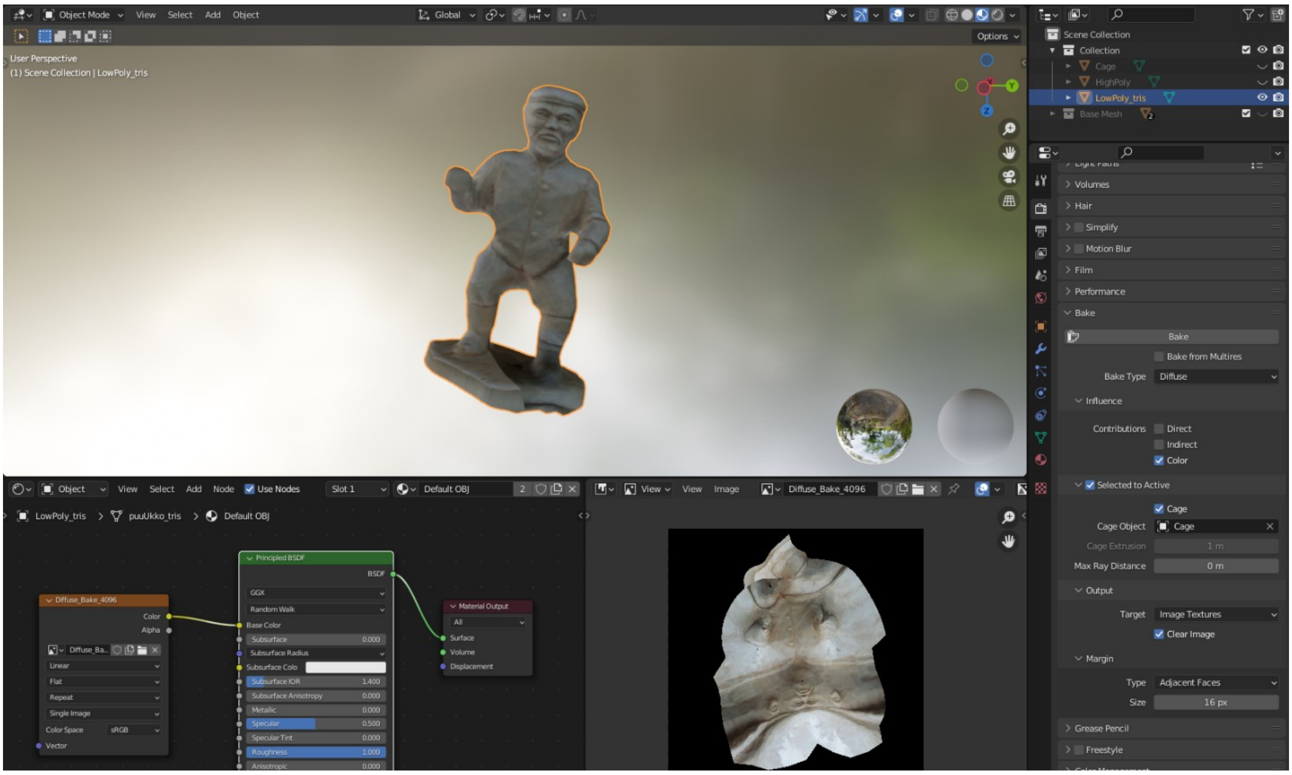

Figure 50 The low-poly model with the baked diffuse texture

After the Diffuse texture had been baked and connected to the low-poly model, it was time to add the small details by baking the Normals. This was essentially the repeat of Diffuse bake with the only difference being the “Bake type” set to “Normal”.

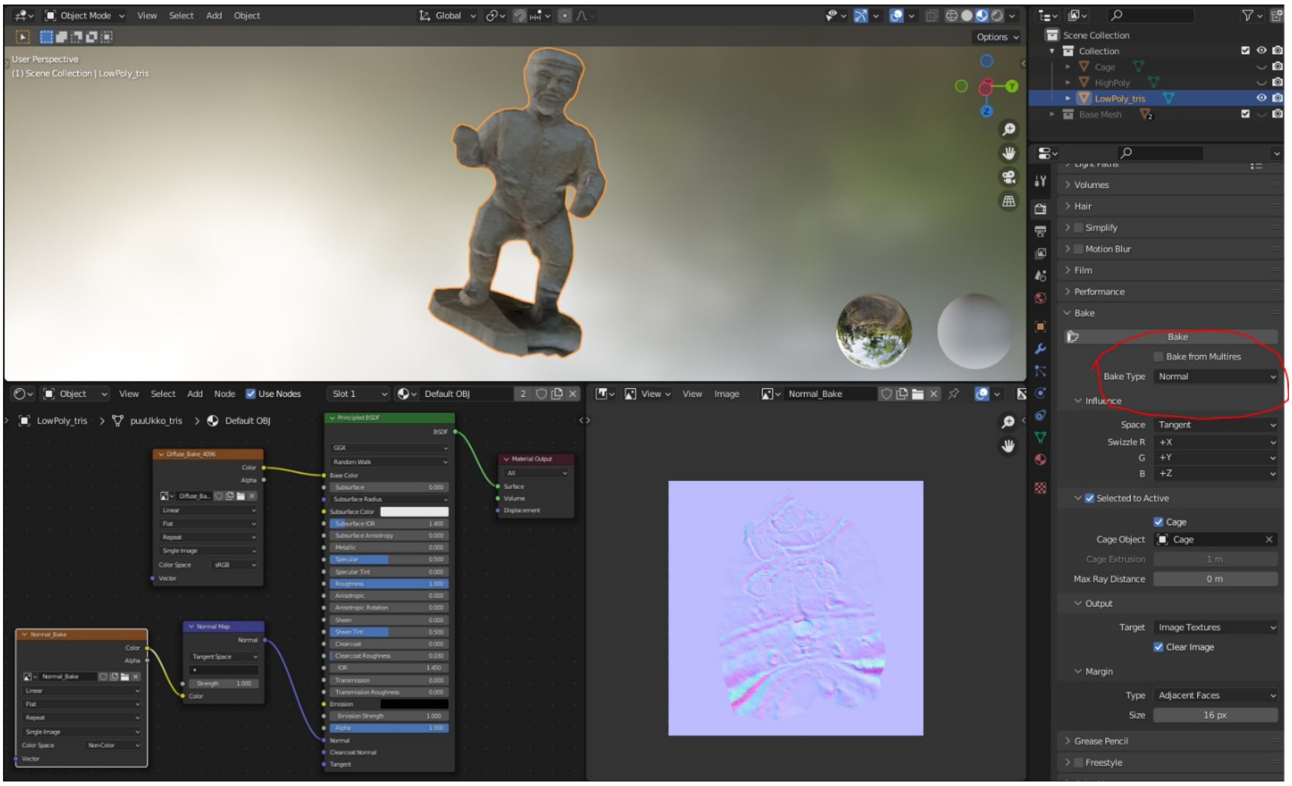

Figure 51 Normal bake done

Figure 52 Side by side comparison of models without (left) and with (right) Normal map

With both of the textures baked the model was ready to be exported to Unity.

Time taken: 23 minutes

6.3 Workflow using more specialized tools

6.3.1 Preparation phase

Preparation for the DIY photogrammetry studio was already explained in the earlier part and the preparation process is essentially the same.

Figure 53 DSLR-Camera used (Canon EOS 450D)

6.3.2 Photographing phase

The photographing phase was done by using the same methods as with the previous smartphone session. The only differences were the usage of a DSLR-camera and the far more stable platform as the camera could be fastened into place on top of the tripod ensuring the increased quality and sharpness of the taken pictures.

The model was rotated approximately 10 degrees between the pictures with 72 photos total.

With the DSLR camera being capable of saving the pictures in RAW (CR2 in this case) format as well as JPEG, the process was tested with both to compare processing times and the effect on quality.

The camera settings were set in manual mode according to the recommended settings seen in the theory section.

- Zoom/Focus: Set to frame the model correctly and taped in place
- ISO range: set to 100
- White balance: set to indoors
- Aperture (f/stop): set to 8
- Shutter speed: set to 1/8

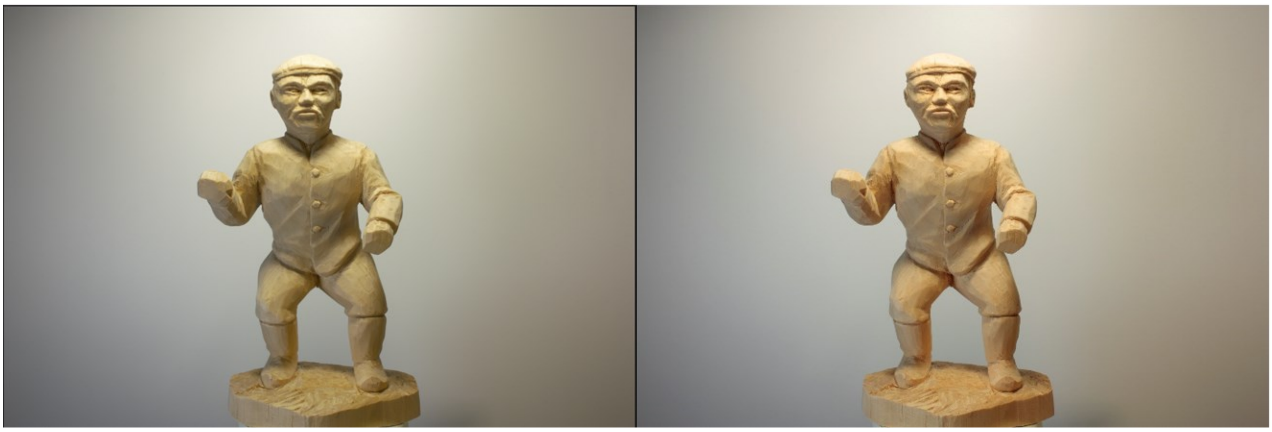

Figure 54 Comparing the CR2 (on the left) and JPEG photo (on the right)

6.3.3 Combining the photos to form the model (Meshroom Desktop PC)

This time, with access to a Dektop PC with an actual GPU capable of using CUDA, Meshroom could be used as it is supposed to be used with all the nodes present in the graph editor. The settings were left to default except for setting the FeatureExtraction node to High again to match the lap-top settings.

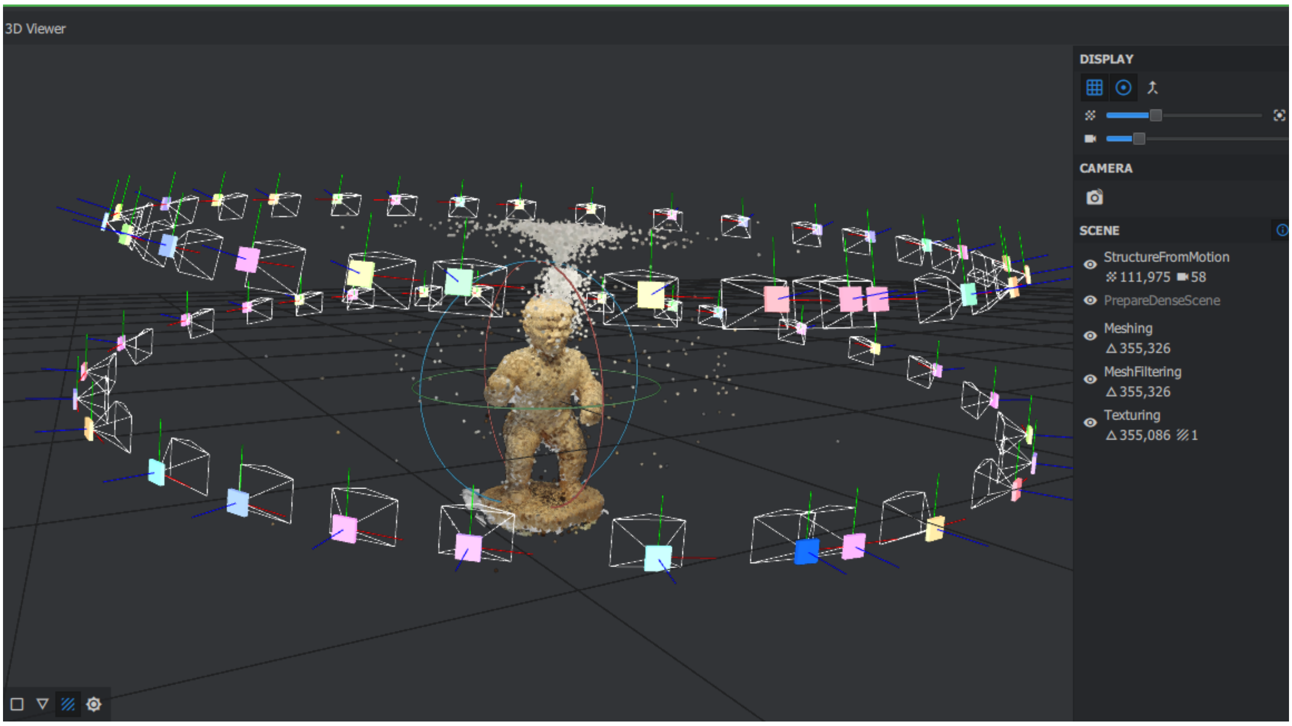

Figure 55 Model inside Meshroom after the process had finished

At the end after the Texturing node had finished computing, the model seemed acceptable even with the strange white point-cloud hovering over it. With the introduction of three additional nodes to compute, the time used increased, but it resulted a more detailed model because of it.

Time taken: 1 hours 51 minutes

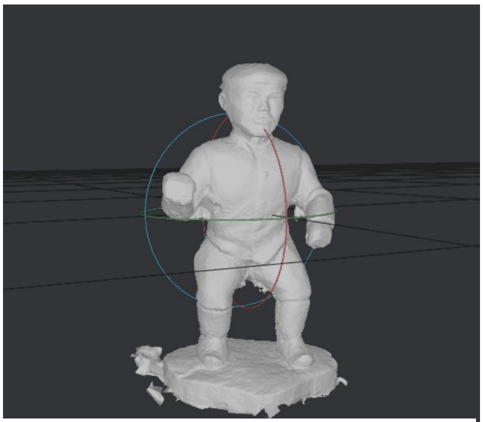

Figure 56 The model inside Meshroom in Mesh view

6.3.4 Optimizing model for Game Development purposes

Retopology was done using Instant Meshes again to similar results as with the draft meshing model. Target vertex count was set around 5450 to reduce it as much as possible while retaining the shape.

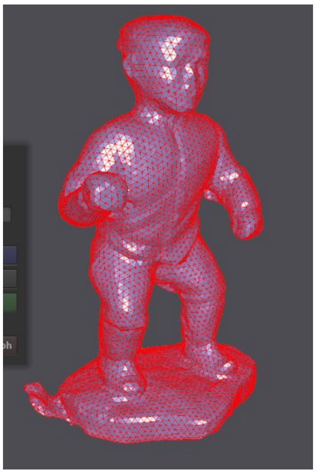

Figure 57 Model inside Instant meshes

The process still stays the same as with the Draft Mesh version:

- Import the low-poly
- Clear Sharp edges
- Clean up unnecessary parts of the low-poly
- Mark the seams and UV-unwrap (marking similar seams to keep the results somewhat consistent)
- Create a blank 4k texture

Figure 58 High-poly inside Blender

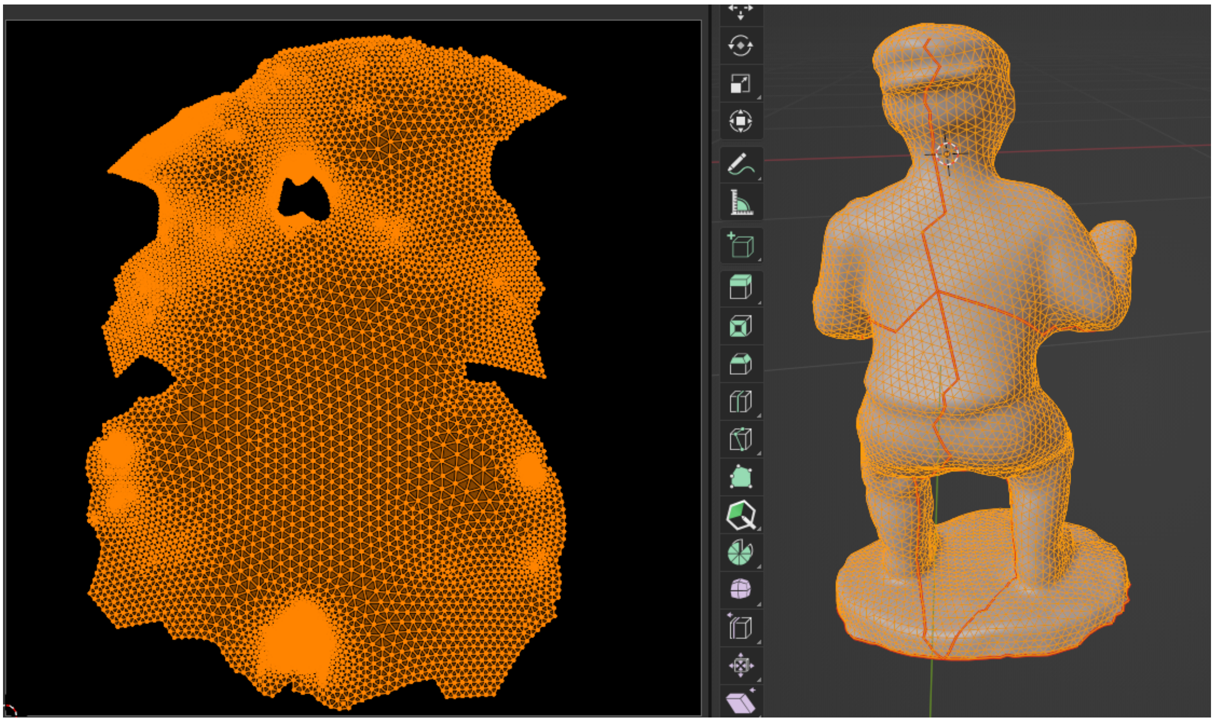

Figure 59 UV-Unwrap of the low-poly model

- Create the Cage by duplicating the low-poly and scale along the Normals
- Set Cage settings inside the Render Properties and all the preparation for Diffuse and Normal Bake.

Baking the high-poly details to low-poly model

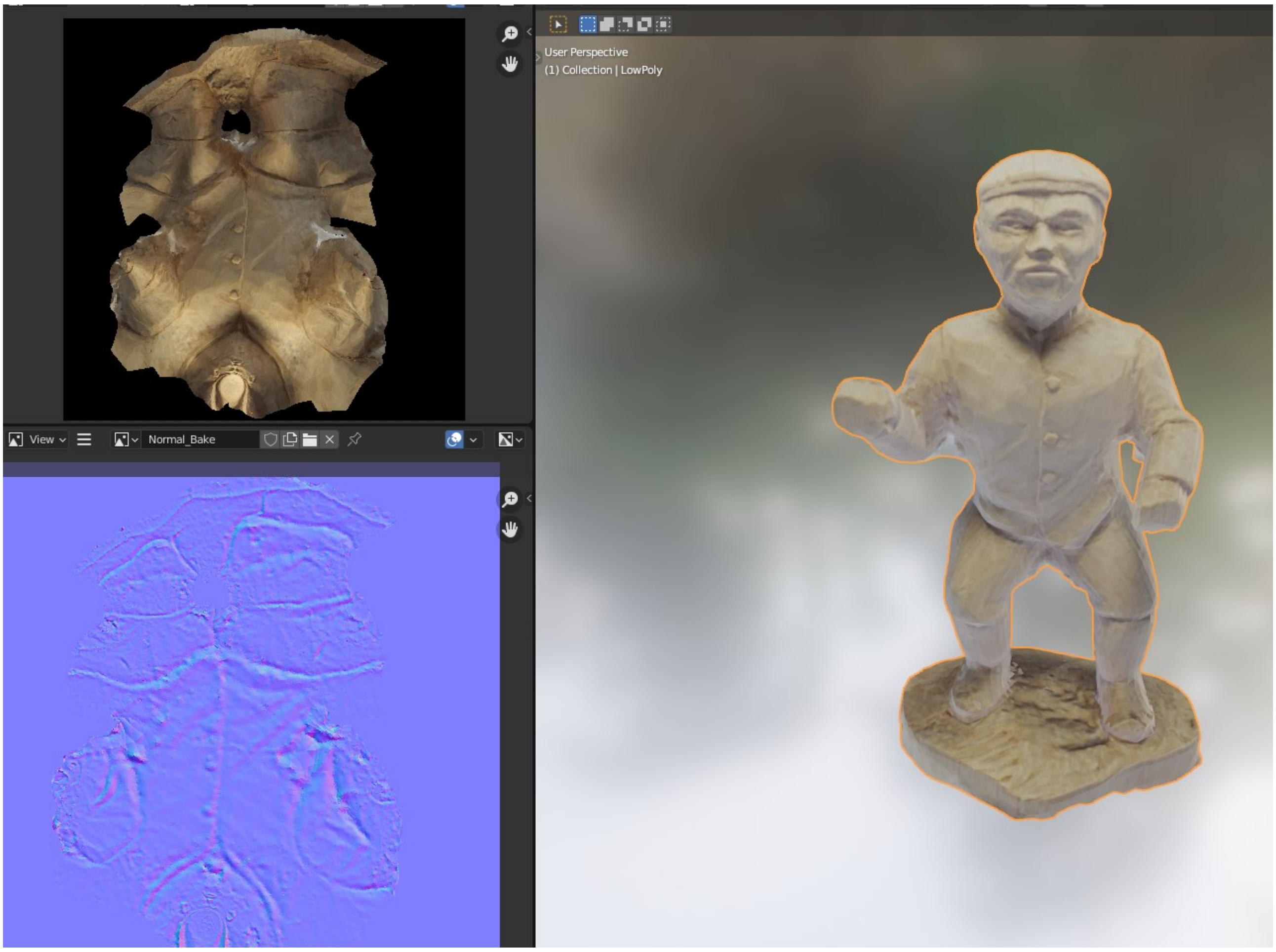

Figure 60 The DSLR-camera version of the low-poly model with Diffuse and Normal textures

There were no problems during the process as the Draft Meshing version was enough practice to succeed in a single try.

Time taken: 15 minutes

7 Making models work in Unity (Game Ready)

7.1 Importing

Importing the 3D models into Unity doesn't require much, especially when they are static models instead of animated ones. The only things that need to be considered are the import settings. Making sure that material and mesh settings are correct will save a bit of work when adding models to the level.

7.2 Shaders/Shader Graph

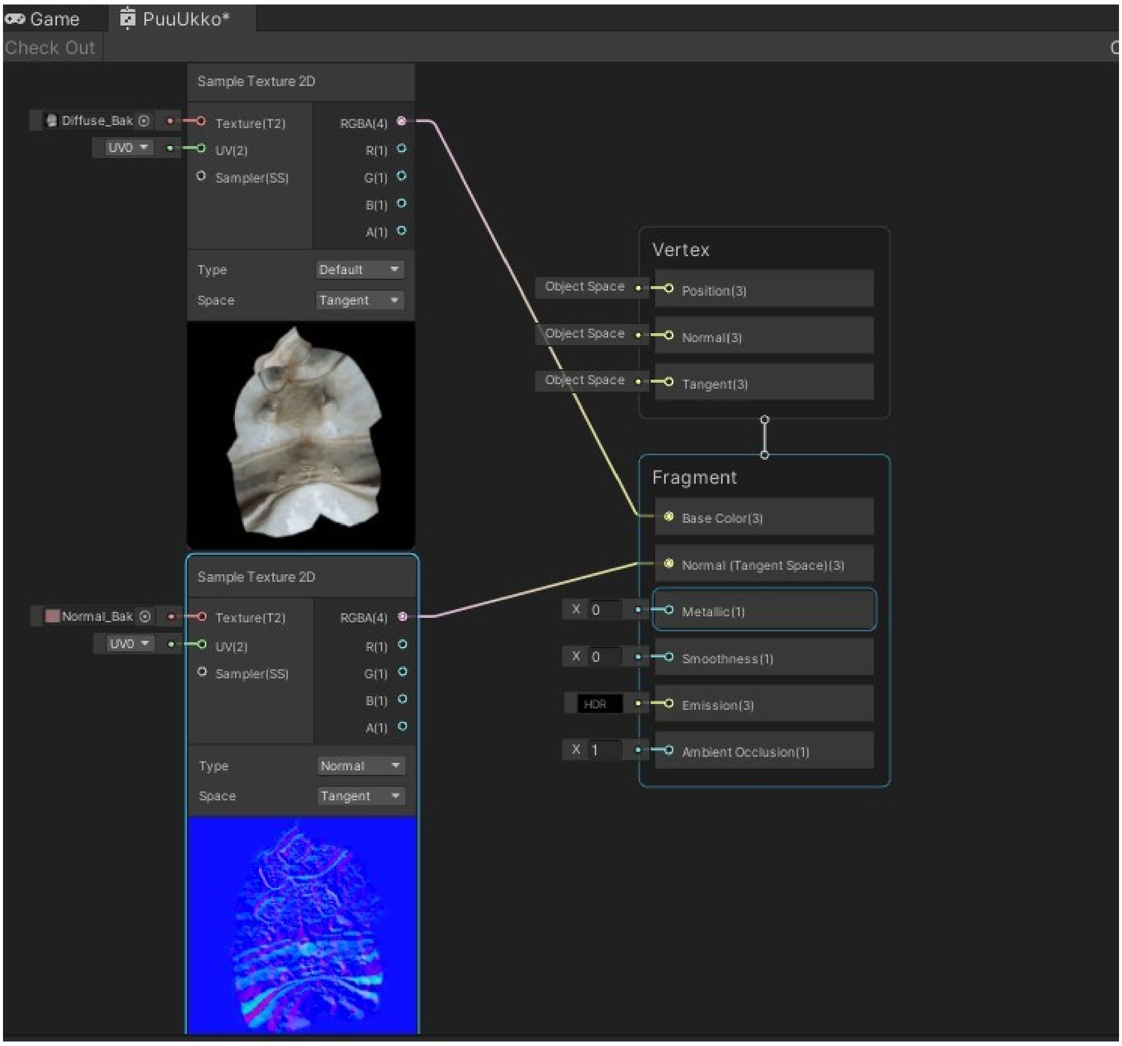

Figure 61 Shader graph layout in Unity to make the Textures work. Same for both models with only the used Sample Texture changing

When adding the textures to models in Unity, the shader graph ended up being the preferred method over others as the graphic interface was very easy understand compared to lines of shader code. Connecting the Diffuse to Base color and Normal to Normal worked just fine. The exact layout of the graph editor (or the shader code) is dependent on the model and on the types of baked texture maps used.

8 Results and Comparison

8.1 "Free" Method

Figure 62 Model created with Draft meshing, imported inside Unity

The result of the Draft meshing process was a 3D model that while usable as a static part of scenery in a game, had some issues such as the very apparent shadow caused by a passing darker cloud while photographing leaving the base of the statue strangely colored and with an odd seam. The result didn't come as a surprise as the limitations of the method were already pointed out during the theory section (p 21, para 1).

The process itself was very demanding with all the crashing and troubleshooting issues that cropped up. Whether it was troubles with taking photographs using smartphone or with the software it could certainly be called a learning experience. And by using the lessons learned from it, the DSLR-photographing went a lot smoother.

8.2 Specialized Equipment

Figure 63 Model created by using the Default settings of Meshroom, imported inside Unity

The result of being able to use DSLR-camera and the Meshroom software to its fullest potential caused the quality of the model jump far beyond the Draft meshing one. With the camera steadily attached to the tripod to eliminate any camera shake and with full control over the settings the pictures came out crisp and sharp and with deeper colors. The work itself went far smoother as well as it was a matter of rotating the model and pushing the button rather than trying to desperately make sure that the phone didn't shake at all (p. 24, para 1).

Meshroom didn't cause any problems either by rejecting pictures or by crashing because of the odd settings and the lack of performance of laptop attempts. To put it shortly, everything worked.

8.3 Direct Comparison

Figure 64 Draft Meshing-model (left) and the DSLR-model (right) inside Unity, side by side comparison

When it came to visual quality, the DSLR-model was clearly superior both with the Diffuse texture and the smaller details of Normal map.

Without a dedicated rig and using a DIY studio, the pictures taken with the smartphone were simply not good enough quality to create a working model which led to an attempt to take the pictures outside during a decent weather for photogrammetry which produced the model that was used in the end. This does lead to the fact that it cannot be directly compared with the DSLR-camera produced model as the circumstances were so different during the shoots. However, the result should still be indicative of how much easier using DSLR-camera to take the pictures was and the how much higher the general difference in quality was.

Table 1 Meshroom Processing time comparison (nodes)

The method used->	Draft Meshing (laptop, smartphone)	Default (PC, DSLR-camera)
FeatureExtraction	3 minutes	22 mins
ImageMatching	1 minutes	1 min
FeatureMatching	4 minutes	22 min
StructureFromMotion	4 minutes	21 min
PrepareDenseScene	Not used	1 min
DepthMap	Not used	31 min
DepthMapFilter	Not used	7 min
Meshing	1 minutes	4 min
MeshFiltering	1 minutes	1 min
Texturing	3 minutes	1 min
Total time	Time: 17 minutes	Time: 1 hour 51 minutes

The times were recorded into the Meshroom log files and displayed the start and stop times of each node. While the Draft meshing was significantly faster method, the results of using CUDA en-able GPU to process the images created far better results. If needed, the Draft Meshing could serve as a way to test the taken pictures and see whether or not it’s worth it to run them through the longer process and thus act as a time saving measures.

9 Ethicality

Ethicality of the research was not an issue as the thesis had no client/organization and all the material used and displayed was not confidential. None of the data contained any sort of identifiable material and no copyright violation was possible as all the images used were captured or created by the researcher.

Any plagiarizing was avoided by paraphrasing the sources and citing them correctly with information added to the reference list (JAMK University of Applied Sciences, 2018).

10 Conclusion and Final Thoughts

The goal of the thesis was to explore the differences between using methods available for everyone and specialized methods of creating 3D models using photogrammetry and whether or not it would be beneficial for the Game Developers/Companies to adopt it as part of their modeler’s 3D modeling workflow.

The thesis turned out to be a true learning experience with all the research into the different aspects of 3D modeling and methods of photogrammetry as many things went wrong during the process and a lot of extra research was conducted to find ways to fix said problems.

The photographing and the preparation for it required more time than expected which left some of the results lacking in quality. Better lighting and some modifications to the DIY studio where most of the photographing was done would have made a big difference as Holmes explained (2017, para 19), but the lack of time and available props led to the results seen in the thesis. Still, the result of the DSLR-camera produced model was something I am quite proud of and serves as an example of what photogrammetry is capable of even when the methods used are not the most optimal ones. And when the objective of the thesis is considered, “testing methods available for everyone”, even the smartphone version taken inside/outside would count as a success as attempts were after all done using equipment available at that moment. Practical research was 75% successful then, as true direct comparison of quality would have required the smartphone attempts taken in the DIY studio to succeed.

The research itself was something that revealed that photogrammetry has a lot of potential when used for game development purposes and it is a tool that I am planning on using myself during my own projects as it cuts down the time needed to model/texture a real-world object by hours or days depending on the complexity of the model. One of the things that also came up regarding complexity of the 3D models is the increasing capability of different game engines such as **Unreal Engine 5** (through the **Nanite Virtualized Geometry**) to handle large amounts of geometry (Bailey, 2021, para 3). Such developments make creating low-poly versions largely pointless and allow for more efficiency as models generated by Meshroom and other photogrammetry software can then be directly added to the game engine. Further research on the most optimal methods and practices is something that I would like to see myself.

References

Abi-Chahla, F. (2008, June 18). *Nvidia's CUDA: The End of the CPU?* Tom's Hardware. Accessed May 2, 2022. Retrieved from https://www.tomshardware.com/reviews/nvidia-cuda-gpu,1954.html

Albertz, J. (2001a). *ALBRECHT MEYDENBAUER - PIONEER OF PHOTOGRAMMETRIC DOCUMENTATION OF THE CULTURAL HERITAGE*. Accessed March 29, 2022. Retrieved from https://www.semanticscholar.org/paper/ALBRECHT-MEYDENBAUER-PIONEER-OF-PHOTOGRAMMETRIC-OF-Albertz/65a85156676884a8d266108fc87b2a12018bb199

Albertz, J. (2001b). *ALBRECHT MEYDENBAUER - PIONEER OF PHOTOGRAMMETRIC DOCUMENTATION OF THE CULTURAL HERITAGE*. Accessed March 29, 2022. Retrieved from http://www.theulegium.de/fileadmin/user_upload/Texte/Meydenb.pdf

Autodesk. (2021, December 10) *Photogrammetry Software | Photos to 3D Scans | Autodesk.* Accessed February 25, 2022. Retrieved from https://www.autodesk.com/solutions/photogrammetry-software

Bailey, R. (2021, October 29). *What is Nanite in Unreal Engine 5?* On-Set Facilities. Accessed May 4, 2022. Retrieved from https://onsetfacilities.com/what-is-nanite-in-unreal-engine-5/

Bhandari, P. (2019, June 8). *An introduction to qualitative research.* Scribbr. Accessed May 2, 2022. Retrieved from https://www.scribbr.com/methodology/qualitative-research/

Bhandari, P. (2021, December 8). *An introduction to quantitative research.* Scribbr. Accessed May 2, 2022. Retrieved from https://www.scribbr.com/methodology/quantitative-research/

Blender Documentation Team. (n.d.). *Render Baking [Manual page]*. Accessed April 30, 2022. Retrieved from https://docs.blender.org/manual/en/latest/render/cycles/baking.html

Deschatelets, B. (2016, April 25). *Understanding UV Mapping and Textures*. Spiria. Accessed April 26, 2022. Retrieved from https://www.spiria.com/en/blog/desktop-software/understanding-uv-mapping-and-textures/

Griggs, M. (2019, January 11). *A beginner's guide to displacement and bump maps.* Creative Bloq. Accessed on April 25, 2022. Retrieved from https://www.creativebloq.com/features/a-beginners-guide-to-displacement-and-bump-maps

Haines, J. (2019, September 24). *Photogrammetry Workflow using a DSLR Camera*. Scholars' Lab. Accessed on April 30, 2022. Retrieved from https://scholarslab.lib.virginia.edu/blog/documentation-photogrammetry/

Holmes, M. (2017, April 26). *Shooting for Photogrammetry.* Instructables workshop. Accessed January 13, 2022. Retrieved from https://www.instructables.com/Shooting-for-Photogrammetry/

Intel. (n.d.). *What Is a GPU? Graphics Processing Units Defined*. Accessed May 2, 2022. Retrieved from https://www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html

JAMK University of Applied Sciences. (2018, December 11). *Ethical Principles for JAMK University of Applied Sciences*. Accessed May 4, 2022. Retrieved from https://www.jamk.fi/en/media/34826

May, S. (2021, April 7). *What Is Texture Baking And Why Is It Important*. | *VNTANA*. VNTANA | 3d CMS & AR eCommerce. Accessed on April 16, 2022. Retrieved from https://www.vntana.com/blog/what-is-texture-baking-and-why-is-it-important/

Meshroom Contributors. (n.d.). *Draft Meshing [Manual page]*. Accessed April 30, 2022. Retrieved from https://meshroom-manual.readthedocs.io/en/bibtex1/faq/draft-meshing/draft-meshing.html

PhotoModeler Technologies. (2020, January 14). *Photogrammetry vs 3D Scanning*. Accessed April 28, 2022. Retrieved from https://www.photomodeler.com/photogrammetry-vs-3d-scanning/

SaanaFB. (2018). *Asset Workflow: Concepting – Frozenbyte Wiki.* Frozenbyte Wiki. Accessed on January 3, 2022. Retrieved from https://wiki.frozenbyte.com/index.php/3D_Asset_Workflow:_Concepting

Serino, T. (2019, August 23). *3D Photo Scanning Tutorial | Blender 2.8 | Meshroom | Instant Meshes* [Video]. YouTube. https://www.youtube.com/watch?v=kGBhM3-W7ic

The Center for Photogrammetric Training. (n.d.). *History of Photogrammetry.* Accessed on February 22, 2022. Retrieved from: https://ibis.geog.ubc.ca/courses/geob373/lectures/Handouts/History_of_Photogrammetry.pdf

Trafagander, C. [Get Learnt w/ Chunck]. (2019, November 25). *What is a Diffuse Map* [Video]. YouTube. https://www.youtube.com/watch?v=IkXMr20xcZU

Trafagander, C. [Get Learnt w/ Chunck]. (2019, December 11). *What is a Normal Map* [Video]. YouTube. https://www.youtube.com/watch?v=jK11WjGuJaI

Trafagander, C. [Get Learnt w/ Chunck]. (2019, November 18). *What is a Roughness Map* [Video]. YouTube. https://www.youtube.com/watch?v=e9KF64hMn98