MASTER THESIS

Setting Standards for 2D-Normal Map Generators

Author:

Amine El Bouhattaoui

Supervisor:

Laure Dhooge

Linda Effinger

This thesis is submitted in fulfilment of the requirements
for the degree of Master of Science in Game Technology

In the programme

Professional Master Game Technology
Academy for AI, Games and Media
Breda University of Applied Sciences

Date

June 21, 2024

Declaration of Authorship

I, Amine El Bouhattaoui, declare that this thesis titled, “Setting Standards for 2D-Normal Map Generators” and the work presented are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date: *June 21, 2024*

Empty page

Working with and researching 2D-Dynamic lighting has only made me more certain of the dreams I wish to accomplish. I must always keep learning so I can keep looking back to my past with pride.

Amine El Bouhattaoui

Empty page

Breda University of Applied Sciences

Abstract

Academy for AI, Games and Media

Master of Game Technology

Setting Standards for 2D-Normal Map Generators

By Amine El Bouhattaoui

Several tools on the market aim to automate normal map generation for 2D-dynamically lit games, but they have yet to gain widespread industry adoption.

Multiple methods exist for building such normal map generators, with no consensus on the best algorithmic or deep learning-based approach. These tools are developed based on stakeholder needs through requirements engineering, thus fitting into established workflows. However, previous research has often used subjective methods to evaluate 2D-normal map tools, leaving developers unclear on the ideal tool features and methods.

We employ an exploratory Q-Sort methodology to assess artists' needs in 2D-normal mapping, combined with expert opinions, to create guidelines and technical criteria for these tools.

Our findings provide ordered requirements for developers and highlight artists' need for quality generation and a properly enabled trial-and-error workflow. They additionally expose the lack of objective evaluation metrics and public learning resources for 2D-normal maps. Lastly, the data exposes a controversy within the field between the trio of separate tools and engine- or artist- tool integration that could warrant further research.

Empty page

Acknowledgements

I would like to thank my primary project supervisor Laure Dhooge, for always being available to give me advice and always being the positive voice whenever I felt it was all falling apart. Your care and positive attitude made me look forward each week to share the progress I'd made on my thesis and my life every single meeting. As this chapter closes you've truly left a positive mark on my life.

I would also like to thank my second reader Linda Effinger, for always listening to me ramble about little things, and for helping me out whenever I felt like I had some huge crisis. Your time and effort were truly appreciated.

I would also like to thank everyone who helped me with guidance or participation on the project, special thanks here go to Thomas Feichtmeir for not just helping me out in the early stages of the project. But also, for letting me use his art during my experiment. My respect for you has only increased as we have interacted.

Lastly, I'd like to thank everyone who has helped throughout this program, my family, my friends and my wonderful colleagues in this fantastic field. Your support throughout this project helped me a lot in making it through the year.

Empty page

Contents

Empty page

List of Figures and Tables

Empty page

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
RGB	Red Green Blue
API	Application Programming Interface
GAN	Generative Adversarial Network
CNN	Convolutional Neural Network
UMUX	Usability Metric for User Experience
GAI	Generative Artificial Intelligence
SMART	Specific, Measurable, Attainable, Realistic, Testable

Empty page

Dedicated to Mika, my drive and the purpose of my life.

Empty page

Chapter 1: Introduction

Within the past two decades, there have been many advances within the field of game visuals. The gaming industry moved from predominantly 2D games, utilizing pixel art, to 3D visuals. Nevertheless, there is still a demand for 2D games as they can provide different visual and gameplay experiences to 3D games. Examples include 2019's Cuphead, with over 6 million record sales (Minotti, M. 2020), and 2020's Celeste with over a million (IGN, 2020).

One avenue that developers have used to make 2D games more appealing is dynamic lighting. Dynamic lighting makes the world react to changes in lighting during gameplay by utilizing shadow casting, reflective surfaces, and realistic lighting. Research by El-Nasr, Zupko, and Miron, (2005) suggests that utilizing dynamic lighting, the developers can make a more believable visual experience, providing greater immersion to players. This makes dynamic lighting a powerful addition not just for narrative games, but also for pixel art games. Here it can help them stand out on platforms such as Steam (Valve, 2003) as outlined by Chris Zukowski on his IndieGameBusiness talk about the 3 things that impact how well your game sells (IndieGameBusiness, 2022).

Examples of games that utilize dynamic lighting include the recently released Sea of Stars (Sabotage Studio, 2023) and the critically acclaimed Ori and the Will of the Wisps (Moon Studios, 2020). Both releases received praise for their immersive visuals and environments.

While dynamic lighting is commonplace within the 3D games market, 2D-dynamic lighting is a much more recent development. While there are exceptions like Octopath Traveler (Square Enix, 2018) where 3D models are used on top of the 2D sprites, most 2D games do not utilize 3D models. To utilize dynamic lighting, normal maps are needed to help simulate the depth needed for realistic lighting. They store depth data for an object that can be used to simulate bumps and additional geometry. Since 2D-art does not have any depth associated with it for lighting purposes, developers must simulate all depth with normal maps. However, unlike 3D-modelling software, there are no automatic generation options for 2D-normal maps in 2D art software. Instead, game artists must draw their normal maps by hand. This can not only exponentially drive up the costs of development as it is a large amount of additional work that artists need to put into every single asset.

While bigger well-funded studios can utilize more expensive 3D-based solutions for their 2D games, smaller indie studios will often refrain from creating games involving dynamic lighting. Experienced 2D-art director Thomas Feichtmeir explained in a private interview that these studios cannot afford the extra development delay and costs added to the artistic budget by dynamic lighting (Feichtmeir, 2023).

Automation could be the solution here, as normal map generation could eliminate the need for artists to spend time drawing each normal map by hand. The creation of normal maps is an uncommon skill and automating this process may eliminate the need for this skill. While this is a valid ethical concern, the development pipeline still needs such skills for the purpose of quality control. Additionally, this frees up artists' time to spend on other visual parts of the development pipeline, such as making more or higher scale images without needing to hand paint their normal maps.

There are already several tools on the market that attempt to solve this problem, such as 2014's Sprite Lamp available on Steam and lesser known but more recently updated tools, such as SpriteDLight (Faas, 2014) and Laighter (Azagaya, 2020). However, such tools have yet to see major industry adaptation. Aside from Laighter, these tools are closed source Blackbox tools, so their process and code are not accessible to developers who would want to modify these tools to serve their development pipeline. Additionally, there is little information publicly available regarding developer needs in this field. Many of the available tools are designed based on unknown means, rather than grounded in publicly available research.

What is lacking is a public resource that provides a proper set of software requirements and evaluation criteria for developers. Software engineering has long known tried and true methods to design and evaluate software. This paper wishes to apply these methods to this currently unstructured field. Additionally, due to the number of different approaches currently used, it would be useful to create an evaluation method to compare the output of such a tool, to further unify the field towards one clear methodology grounded in research.

This paper wishes to accomplish this by answering the following research questions:

- **Can requirements engineering techniques such as those outlined by David Lightbown in 2015, be used to create a basic set of software requirements and**

technical evaluation criteria for the design of 2D-normal map generation software.

The goal is to unite development on such tools towards a single set of provably correct criteria such that further research and development can reference these criteria to develop a more applicable tool for the industry. The creation of such tools could lead to an increase in successful 2D-game releases featuring 2D-dynamic lighting. Especially from smaller studios. Lastly, this paper hopes to expose some information about the methods used in the current Blackbox tools, to streamline further research.

Chapter 2: Literature Review

2.1 Normal maps

While normal maps are vital to the usage of dynamic lighting in 2D-games, this is a recent development. Normal maps have already been commonly used in 3D game development for a far longer time, with 1998's Jurassic Park: Trespasser being the first commercial game to use primitive normal mapping methods (BanZeken, 2022). Thus, it is necessary to first investigate the underlying technology and its intended purpose before diving into normal maps for 2D assets.

2.2 Normal maps in 3D

Normal mapping is one of multiple texture mapping techniques used to simulate different variations of surface detail on a 3D object. Every mesh is exported with surface normal data, which consists of vector information at each vertex of the mesh. This vector points to a direction, which is used during lighting calculations to visualize a smooth surface. This technique does not change anything about the polygons, only how light renders the object. For this paper, normal maps are considered tangent space normal maps as these are the current industry standard (Unity Technologies, 2018).

A normal map consists of a single RGB image file. Within this coloured image, each RGB channel encodes 3D-vector displacement data for the mesh. When rendering, this normal map causes a change in how the surface normals visualize interactions with light. This causes bumps to be simulated on the surface. While normal maps are generated, they encode the direction vector values -1 to 1 to a colour channel ranging from 0 to 255 in a process known as baking. This means that a (0,0,1) is encoded to an RGB encoding of (128, 128, 255), which causes the blueish colour of a normal map.

In 3D game development normal maps are used as a tool to balance quality and performance. The usage of normal maps is important, since large 3D models can cause decreases in a game’s frame rate as they take much processing power to render (Autodesk, 2007). Baker (2022) explains that while Unreal Engine 5’s Nanite system gets close to eliminating this issue, high-polygon models still take up far too much random-access memory and hard drive space to be commonly usable. Normal mapping is a compromise between model quality and performance, you display more visible detail while using less geometry (Lampel, 2022). Developers utilize low-polygon 3D-models and use normal maps to simulate additional details. High-polygon models are used to bake these normal maps, resulting in low-poly models with an elevated level of visual detail (Unity Technologies, 2018).

In his article on normal map generation for Unity, technical artist Ben Golus (2021) goes in-depth on the current methods utilized in the industry. xNormal (Orgaz, 2007) is recommended as an industry-standard tool for baking normal maps. Most 3D modelling software such as Houdini (SideFX, 1996), Substance Painter (Adobe, 2023), and 3ds Max (Autodesk, 1996) include this functionality. Golus (2021) also offers several warnings to those unfamiliar with creating normal maps. The two most popular vector graphics application programming interfaces (APIs), OpenGL and Direct3D utilize inverted Y orientations. Depending on which API is used, normal maps will have to be inverted before being applied. Additionally, while many pieces of software used to utilize their specific way of calculating tangent space, MikkTSpace (Mikkelsen, 2020) has since become the industry standard method used to calculate tangent space. MikkTSpace is an implementation standard that makes the generation of tangent space resistant to the movement between different software applications. Lastly, it is important to remember that while some engines only support pixel- or vertex-baked normal maps, most industry-standard software can bake normal maps using either method.

2.3 Algorithmic normal map generation in 2D

2D game developers are not as lucky with their normal map generation options. As the industry standard tools used for 2D-art such as Photoshop (Adobe, 1990) for high resolution and Aseprite (Igara Studio S.A, 2014) for pixel-art do not feature any methods to bake normal maps. The comprehensive paper on normal map generation for 2D sprites is the paper by Moreira, Coutinho, and Chaimowicz (2022). This paper on the analysis and compilation of normal map techniques for pixel art goes in-depth on multiple techniques used to generate normal maps for pixel art.

Several relevant techniques outlined in this paper involve creating heightmaps for the purpose of normal map generation. Heightmap creation works as follows: textures are recoloured such that anything protruding outwards is coloured darker than a base colour of 50 percent grey. Anything protruding inwards is coloured lighter than this base colour. The brightness of a pixel shows its depth on the heightmap. There are several known methods used to generate normal maps based on these heightmaps, Moreira, Coutinho, and Chaimowicz (2022) uses the Sobel-Feldman operator (Sobel, 2014). This edge detection algorithm puts a heavy emphasis on the edges of an image when creating a normal map. The Sobel filter takes a greyscale image and performs a matrix convolution with Sobel's matrices resulting in a horizontal and vertical map. These two maps are then merged with a third blue map controlling the intensity of the gradients resulting in a normal map.

A post by nbickford (2022) on the Nvidia developer forum describes some heightmap generation methods used in Nvidia's Texture Tools Exporter. They describe the calculation of the normal at each pixel using a normalization function. Sobel filters, 2-sample approximation, or other derivative kernels are used afterwards to perform the transformation from a heightmap to a normal map. Software Engineer Nika Tsankashvili (2018) additionally describes Prewitt's operator as a potential alternative to a Sobel filter, putting less emphasis on pixels closer to the centre of the sprite mask. According to Tsankashvili this results in slight improvements at the cost of computation power.

While Moreira, Coutinho, and Chaimowicz (2022) argues that the output from a heightmap-generated normal map lacks quality compared to the hand-painted normal map. Given that the heightmap was hand-painted for this study, it is unverifiable whether the output could have been caused by a flawed hand-drawn heightmap, or the techniques utilized.

Since the manual painting of normal maps is not sustainable for development, substituting this for the manual painting of heightmaps is not a valuable alternative. Moreira, Coutinho, and Chaimowicz (2022) prepared for this and studied an even more algorithmic approach that they refer to as bevelling. This method by Matt M. (2011) is used as an attempt to emulate the Sprite Illuminator (codeandweb, 2015) program and it works as follows: The algorithm first generates a heightmap by merging the distance transformations of two binary masks. These two masks encode the external and internal shapes of the sprites and are automatically generated. Afterwards, the results are passed through an Euclidean distance transformation. This returns the distance to the closest background pixel for each pixel of a black-and-white binary image. This output is then merged and passed through a Gaussian filter for smoothing. The algorithm exposes several parameters that can be used to finetune the heightmaps. These heightmaps are then used instead of handpainted heightmaps to generate normal maps.

However, there are more ways to generate heightmaps that are not based on bevelling. The open-source tool Laigter (Azagaya, 2020) instead estimates a heightmap based on the RGB values of the input image. Lighter colours are at a higher height than lower colours. SpriteDLight (Faas, 2014) is another tool currently in production that utilizes a different depth estimation algorithm.

While most of the tools listed so far rely on the creation of heightmaps, Sprite Lamp (Snake Hill Games, 2014) instead takes a different approach. It uses an approach described by Moreira, Coutinho, and Chaimowicz (2022) based on that of Clark (2007) known as the merge of four illumination angles. It works by manually painting greyscale versions of your input image with the correct lighting from each of the four major directions. Afterward, opposite angles are merged, with one of each inverted. Each of the resulting images then takes either the red or green channel, with the blue channel being used as the background. Once again it takes a high amount of artistic work.

Not all 2D normal map generation methods are based on purely algorithmic approaches. While there are currently no tools on the market that generate normal maps based on deep-learning methods, Su, Du, Yang, Zhou and Fu (2018) have published their Python-based implementation online for public usage. Their paper first described a method to generate normal maps utilizing a Generative Adversarial Network (GAN) framework using sketch inputs. Su, Du, Yang, Zhou and Fu (2018) verified that the generated normal maps had fewer errors than comparable methods. Shortly after Hudon, Lutz, Pagés and Smolic (2018) also established a method utilizing a convolutional neural network to generate normal maps from only character outlines as the input. Hudon, Grogan, Pagés, and Smolić. (2019) expand upon their previous work by introducing multiple modifications to the convolutional neural network (CNN), resulting in higher-quality normal maps. He, Xie, Zhang, Yang and Miyata (2021) would later utilize a similar method in their paper. They utilized a GAN framework using sketch inputs. However, they additionally adopted a U-Net structure discriminator for more accurate results. After utilizing this new technique, He, Xie, Zhang, Yang and Miyata (2021) verified their work to be an improvement on the work done by their predecessor.

2.3.2 Evaluation of 2D-normal maps

In the previous chapter a number of deep learning-based 2D-normal map generation methods were discussed. Most of these papers used a similar method first established by Su, Du, Yang, Zhou and Fu (2018) to verify the correctness of their generated normal maps. This was done by restricting the dataset to images and drawings based on 3D models, this way they could obtain a verifiably correct ground truth normal map based on the existing 3D model. These were then compared to the generated normal maps.

Several variables were calculated based on the difference between the ground truth and the generated normals. An error map was also generated and used to visualize the positions where the generated normal map made mistakes.

Moreira, Coutinho, and Chaimowicz (2022) evaluated a number of algorithmic and a deep learning-based approach in their paper and concluded that none of them can properly replace the quality of hand drawn normal maps. However, the authors have done this based on observation, rather than any objective evaluation model. Along with the fact that many pieces of software utilize different approaches, this makes it unclear which of these approaches is the most accurate or the most desirable.

As outlined above, there are several tools on the market which utilize some of the purely algorithmic approaches as described by Moreira, Coutinho, and Chaimowicz (2022). However, none of their development is verifiably based on any publicly verifiable user testing or criteria and there is no public evidence of commercially successful games utilizing any of these tools. And while there are papers on machine learning-based approaches, these have not yet resulted in any market tools. Thus, it becomes relevant to investigate the design and evaluation of such tools, to see what is needed to result in properly adoptable developer tools.

2.4 Design and evaluation of game development tools

Developer tools are made based on the needs or requirements of stakeholders. The process known as requirements engineering, was first used by Thayer and Bailin (1997). Software Engineering by Sommerville (2016) describes requirements as a description of the services that a system or piece of software should provide, as well as the constraints that it adds to the operation. These requirements should reflect the needs and purpose of the stakeholder.

Software Requirements are split into user requirements and system requirements. In the context of developer tools, user requirements are effectively the needs of the developers translated into tooling functionality. System requirements are the operational constraints of the piece of software. Sommerville (2016) describes the steps involved in requirements engineering as follows: elicitation and analysis, specification, and validation.

Elicitation and analysis involve the discovery of requirements based on interaction with stakeholders. This can be done through interviewing or through observation, where you specifically observe the status quo during development.
Sommerville (2016) describes the specification as the standardization of such requirements to an established standard. While requirements have long been written in natural language, the usage of such language invites undesired ambiguity into requirements. Thus, it also suggests a standard format for writing down requirements which includes the now standard usage of unified modelling language (UML) and use case diagrams.

Lastly, Sommerville (2016) establishes the verification step by verifying the software against the requirements. This involves several important checks; validity checks verify that the requirements reflect the needs of system users. Consistency checks check for conflicts within the requirements. Completeness checks verify requirement completeness. Realism checks verify the possibility of implementation and verifiability checks whether the previous four checks are doable. While the steps of analysis, specification, and validation are listed in order, Sommerville (2016) states that this process is a continuously iterative process, with the activities being interleaved throughout development. Additionally, requirements tend to change throughout development, thus it is worth continuously verifying whether your requirements still hold.

A GDC talk by Lightbown (2021) on tools development by game industry legends breaks down the design of game tooling into three questions: What is the problem users want to solve, what can you learn from other tools and how do people use your tools in conjunction with their other tools? When discussing the first question, he emphasizes the need for tools to be developed in context. He states how it is necessary to verify the needs of developers while developing tooling, which is equivalent to elicitation and analysis as described by Sommerville (2016).

When discussing the other two questions, Lightbown (2021) recommends using older tools as a reference for the development of your tools. He describes how it is important to follow established design guidelines and conventions to stay compatible with current tools and conventions. He also discusses the importance of context, user interaction, and compatibility with the overall tool portfolio. These last two points match up well with the conclusions by Kasurinen, Strandén and Smolander (2013). Their paper concluded that tools need to fit well into current development pipelines, to be widely adopted by the industry as compatibility with the current pipeline is one of the most relevant criteria in tool adoption.

In his book: Designing the User Experience of Game Development Tools, Lightbown (2015) goes even further in-depth on the first step of requirements engineering. He describes that while interviews can be an effective way to elicit user feedback, there are some risks involved. Users will often say yes to feature suggestions even if they will not end up using them. This is dangerous, as more features will exponentially increase complexity. Instead, he suggests a less structured method of doing interviews based on observation. This involves asking questions based on the interaction between the developers and tooling with a specific goal in mind. He then proposes using task flows to help document these interactions. Task flows are flow charts that show the connections between users and specific tasks from the user's perspective. They can be used to help understand the behaviour of the user.

Requirements engineering is a proven process, in a study by Hofmann and Lehner (2001) requirements engineering was proven to be a relevant factor in the successful adaptation of software. Thus, applying requirements engineering to these 2D-normal map generation tools could result in better and more adoptable tools for the market.

Chapter 3: Research Methodology

The goal of the research is to create a set of requirements for normal mapping tools, it makes sense to first dissect the goal of the research. When trying to discover what makes a good 2D-normal map generator, it is relevant to first know what makes a good 2D normal map. Therefore, it is relevant to split this methodology into two halves trying to answer two smaller questions:

Q1: According to experienced 2D-normal map artists, what is the current quality standard for a good normal map?

The expected outcome here is that there is no objective metric. Since 2D-normal maps lack any objective geometry found in 3D objects it is likely based on personal artistic taste. However, mention of the trade-off between speed and quality is expected.

To answer this question, information from experts will be needed. The expert insights will additionally be used to provide additional context to the results derived from the second sub question, which is centred around the actual tool.

Q2: What relevant features do current artists value within 2D-normal map generation tools?

Due to the nature of this question, it was necessary that some knowledge of 2D-normal map generation tools would be preferred. However, during testing, it became clear that a trial run with a single normal map generation tool offered the necessary context needed for an experienced 2D artist to offer valuable data.

We expect to find that artists will primarily prioritize features that are relevant in the software that is already industry standard. This is mainly based on Lightbown's (2021) statements on tool development. Tools should be developed in such a way that artists will want to use them, thus it makes sense to incorporate features that are industry standard in other artist tools. Integration with their current tools is also expected to be valued highly for that reason.

The main artifacts that were eventually developed are two interview plans. The results of the Q1 interview would be used to create a set of technical requirements, after which the results of the Q2 interview would be used to create the non-technical requirements as well as introduce order to the requirements set. These requirements end up forming a list of guidelines that developers can use to develop their normal mapping tools.

3.1 The 2D-Normal Map Evaluation Interview

To answer the first question, it was decided to use interviews to research the quality evaluation of 2D-normal maps. The reason for doing an interview, instead of a survey or focus group was the following. While a focus group could be a great way to discuss such a subjective topic, in the context of this interview, it would simply be too difficult to connect enough people in the international community in one meeting. An interview was then chosen over a survey due to the subjectivity of the topic. It was expected that questions would be revised greatly based on the direction of the interview. The questions are left very open-ended for that reason, such that there is plenty of space to ask follow-up questions and in order not to overly constrain the interview.

3.1.1 The 2D-Normal Map Evaluation Interview Participant Selection

Due to the current standard in the development of painting normal maps by hand, the ideal candidates for this interview are 2D-normal map artists used to drawing 2D-normal maps. Participants for this interview were mainly approached through social media or email. Prior research was done to ensure that the potential participant was properly experienced with the topic before being approached.

3.1.2 Tools Utilized for 2D-Normal Map Evaluation Interview

While the tests for this interview were done in person, Microsoft Teams Classic was used to perform the online interviews in the final data gathering. The choice for Microsoft Teams came from a desire to minimize the number of tools needed. Microsoft Teams Classic allowed for an acceptable level of transcription and recording quality needed for this study. While Microsoft Teams was used for the interview, most communication was done through Discord. Notes during the interview were taken within Microsoft Word, which was done on a secondary monitor. A primary monitor and a webcam were utilized fully for the Microsoft Teams meeting.

3.1.3 2D-Normal Map Evaluation Interview Questions

During the interview, a few introductory questions were asked first. The participants were asked to confirm their name, current occupation, and position, years of experience in the field, years of experience with the topic, and level of familiarity with the topic. These questions mainly serve to evaluate their level of experience with the topic. They're then informed about the concept of stimulated recall (Lankoski, P., Björk, S, 2015, p. 117), which means they should express their thoughts out loud during the interview. This is followed by a basic explanation of the topic, informing them about the goal of the thesis and what information is being gathered. Afterward, they were asked to confirm their consent through a consent form (See Appendix C) sent to them before the interview, which informed them of the legal parts of the interview.

At the start of the interview, a few questions are asked specifically about normal maps in 2D and how they would evaluate the quality of a good 2D-normal map.

- Have you ever created a 2D-normal map by hand?
  - If yes, what do you look for when creating a 2D-normal map? Can you walk me through your process?
- When would you consider a normal map to be usable for a project? When are you finished?
- Do you think there is any objectivity present in the creation of normal maps, or is it all up to artistic taste?
- What is in your eyes, the main goal that the usage of 2D-normal maps intends to achieve?
- Within the development pipeline, where do you lie on the balance between speed and quality of a 2D-normal map?

The second set of questions is based on normal map automation and the interviewee's experience with those. It is meant to evaluate the current state of 2D-normal map generation from the perspective of experts and to provide insights about their quality.

- Do you have any experience using any normal map generation tools on the market, could you walk me through your experience with them?
- How would you evaluate the normal maps generated by such tools?
  - If you think the output of such tools is not up to standard, what about these normal maps is lacking? By what metric would you judge them?
- To what degree do you think automation will catch up to the manual creation of normal maps in the future?

        - Why do you think it has yet to become a major part of 2D-game development?

Lastly, they are asked the following closing questions:

- How do you feel about the current state of 2D-dynamic lighting in games? Do you feel it will become more prevalent in the future? Do you feel that any other fields could provide major value?
- Do you have anything you'd like to add to this discussion?

3.2 The Tooling Quality Interview

The choice of an interview was quite straightforward for the second question. Due to the subjective nature of the topic, it was not desirable to perform research through surveys as follow-up questions would be important for this research. As described by Sommerville (2016), during the elicitation and analysis step for requirements it is also relevant to observe the participant during the usage of a tool. As some participants might lack experience with 2D-normal map tools, it became relevant to let them experience a tool before asking further questions. Because of this, a demo was included in the experiment, to help provide context to each participant. This additionally integrates some of Lightbown's (2015) methods to avoid the risks mentioned in interviews (Chapter 2: Literature Review).

3.2.1 The Tooling Quality Interview Participants

To ensure a significantly large target group for this interview, any digital artist with a moderate 3-year or longer experience in visual arts on a student or professional level was included in this study. Originally the goal was to focus on artists with experience working with 2D lighting and while that would be the preferred qualification for this interview. It was decided that relaxation of the requirements was appropriate, as the desired data was solely focused on artist tools. These participants were mainly found within the Dutch game development community.

3.2.2 Tools Utilized for Tooling Quality Interview

While doing research online, it became clear that SpriteDLight (Faas, 2014) was often cited as people's tool of preference. SpriteDLight (Faas, 2014) is also desirable as it contains a large number of features to play around with for a demo. However, SpriteDLight has a few things going against it that prevented me from using it for this research. Firstly, SpriteDLight (Faas, 2014) is a closed source currently private tool, therefore utilizing it risks compromising the reproducibility of the research if the tool were to become unavailable in the future. Moreover, due to budget constraints, it was unrealistic to attain SpriteDLight for a significant number of participants due to the associated costs.

Therefore, when looking at other tools available, Laigter (Azagaya, 2020) turned out to be the most desirable tool to utilize for this research. Not only is the tool free and publically available, making it widely accessible for any researcher to utilize for as many people as necessary. It is also open-source and available in any version, which means reproduction with the exact version of the software will be possible in the future. For this study, the newest version of Laigter (Azagaya, 2020) will be used for the demo, Laigter 1.11.0 (Azagaya, 2023).

For the live demo, it was necessary to obtain a sprite of proven quality to give to the participants for normal map generation. The sprite used during the testing stages of the interview was a 64x64 character sprite with moderate detail. Therefore, it was vital to locate a similar sprite of proven quality. Credit goes to Thomas Feichtmeir for granting this study permission to use the following sprite depicted in Figure 1.1 within the confines of this study.

Figure 1.1 ***2D Knight by Thomas Feichtmeir (2019)****: The 2D-sprite utilized during the experiment.*

For the Q-sort (3.2.5), a Miro board will be utilized. It will allow the participants to enter, drag, and drop categories. Miro (Free Plan) allowed participants to join without an account, which made it a convenient pick for this research. The template for the Miro board is shown under chapter 3.2.4 in figure 1.2.

All software and hardware used for the interviews mentioned in Chapter 3.1.2 are also used for this group of interviews in the same way.

3.2.3 Interview Questions for Tooling Quality Interview:

At the start of the interview, the participants are told about the concept of stimulated recall (Lankoski, P., Björk, S, 2015, p. 117), as explained in 3.1.3. They are requested to utilize this technique throughout the entire interview. They are then told about the topic of research and what they should expect from the interview, after which they get space to ask some questions. Before the interview starts, they're asked to confirm their consent through a consent form sent to them before the interview, which informs them of the legal parts of the interview.

A few introductory questions were asked at the start of each interview:

- What is your name?
    - Serves as a basic introductory question.
- How much experience do you have as a visual artist?
    - Used as a basic metric to evaluate level of expertise.
- What tools are you most familiar with using while doing digital art?
    - Used to evaluate the industry standard tools used by the participants.
- Have you ever worked on a 2D Dynamic game before, if so, have you used any normal map generation tools before?
    - Used to evaluate the participant's experience with the topic.

Before the experiment, the participants were all sent the version of Laigter (Azagaya, 2020) to be used for the experiment and asked to install it. After this set of questions, the research moves on to this demo phase. The participants are asked to open Laigter and told how to input a sprite to use within the program and the basic relevant features. After which the participant is given 10 minutes to generate a normal map that they are satisfied with. They're told to use the preview window in Laigter to evaluate their work. After the demo, they are asked a few questions about their experience.

- What did you like about the tool? Did it have any capabilities that made it easy to use?
- Did any parts of the tool stick out as particularly frustrating?

- Are there any capabilities that are noticeably absent within the tool? Anything you may be used to, that is now clearly missing.

The interviewer won't home in on specific elements of the tool unless first mentioned by the interviewee. However, based on the observed behaviour during the experiment, the interviewer might ask questions about the thoughts or choices of the user while using the tool. This is to prevent discussion of superficial elements not relevant to the interviewee. Afterward, the participant is free to ask any questions before moving on to the next part of the interview.

3.2.4 Q-Sort Methodology and Possible Methods of Generating a Q-Set

A Q-sort methodology was chosen for this next part of the research. This research's main purpose is to establish a set of ordered requirements, therefore it is relevant to figure out the importance of various features found within 2D-normal mapping tools. A Q-Methodology such as the one by Millar, Mashon and Kidd (2022) is a good fit for that reason. A Q-Methodology entails the usage of a set of topics called a Q-set, which the participants are then asked to sort based on the level of importance they associate with it. This is most useful when combined with stimulated recall, as you are then able to evaluate their full reasoning. By utilizing a Q-Methodology, the dataset will already have some order that can be used while making the set of requirements.

The first methodology considered utilizes a pre-defined Q-set for the Q-sorting interview. This Q-set is often ideally based on prior research, due to the lack of research available on this topic, the Q-set was established based on prior research and experience with several 2D-normal mapping tools and their features. Based on this analysis, 16 topics were developed which were then split into 4 categories, Quality of the output, Testability of the work, User Interface and artist tool features, and User Experience. As they did not end up being utilized, these categories and topics can be found in Appendix A.

Another approach that was tested for this research was an exploratory Q-sort approach. Based on the approach utilized by Dhondt (2023), the exploratory Q-sort approach mainly changes the way the Q-set is created. Instead of utilizing a pre-established Q-set, the exploratory approach puts even more of a focus on categories that the participants find important, by letting the participants pick the topics included in the Q-sort. You tell the participants to name features for 2D-normal map tools that they find important and use those to generate the Q-set. It is however important to include an explanation for each item in the set, to avoid confusion in the final dataset.

After proper testing, it became clear that the pre-established Q-Set contained more abstract design approaches. Performing the exploratory Q-sort ended up with more actual features. Requirements engineering works best when the wishes of the client are as concrete as possible, therefore it was decided to go with a different approach for the final experiment that mixed the two approaches. The exploratory Q-sort approach would be used; however, the categories “Accuracy of the normal map” and “Automation” (See Appendix A) would be given to the participant at the start of the research to add to the Q-sort. This way, the research results will contain the features artists find most important, while still considering these two core design philosophies. Though artists are given the option to drop either option if they truly do not consider them important.

Throughout the experiment, the participants are given a free choice of the number of items to include in the Q-set. This was done to ensure that they could list all items they deemed relevant. During the development of the Q-set, they are free to change or remove any items, as long as they explain why they are committing to that change. Afterward during the Q-sort, only the 23 most valuable items can be utilized. This number can be reduced to 16 or 9 depending on the number of items given by the participant. If the participant lists more than 23 items, they are asked to consider the items used in the Q-Sort to be the 23 most important items present. They do however not remove the items from the set.

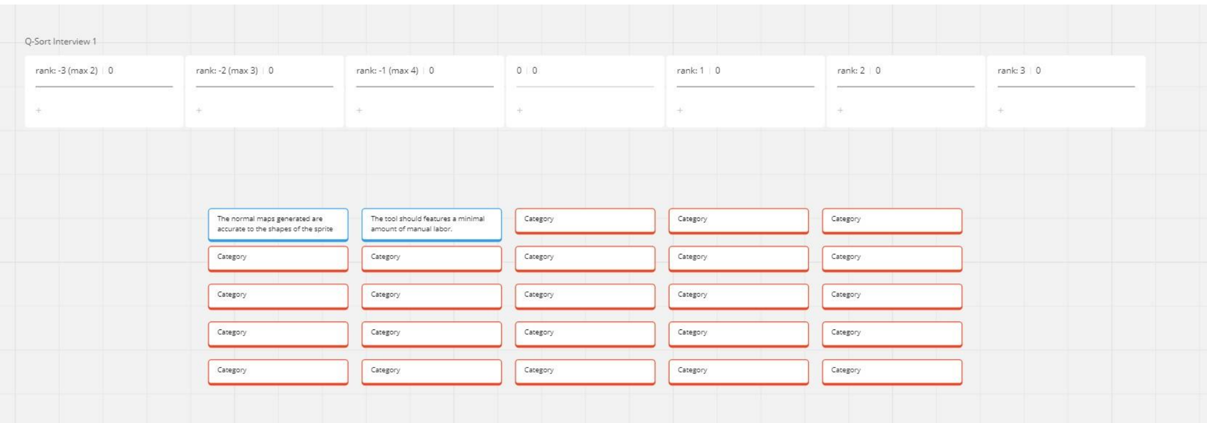

*Figure 1.2 **Miro Example**: The Miro Board utilized for the final interview.*

3.3 Ethical Considerations

Originally, the interview contained more questions regarding age and gender. However, after careful deliberation, it was decided that these questions were irrelevant parts of the dataset and were therefore removed.

There are potential concerns in the 2D-Normal Map Evaluation Interview regarding NDA. Mainly, regarding the questions concerning their workflow. During the interview, it will therefore be indicated that it is not the intention to break NDA and therefore it is optional to deny answering such questions. Additionally, any participants of the questionnaire would be anonymized within the research dataset. The same applies to the participants of the tooling interview. Before the start of the interview participants were also asked to sign a consent form, stating their consent to the interview, as well as their right to remove any part of the transcript before the thesis gets published.

It is worth considering that most of the participants for the research and interviews were recruited from artist groups in Europe and the United States. By taking recommendations primarily from these groups, it becomes likely that the results and following software requirements created will correspond mainly to a Western workflow. This cultural bias could limit the applicability of the results to non-Western software development.

Furthermore, while it is open-source software, the development team of Laigter (Azagaya, 2020) was made aware of this research using their software. For the image used during the tooling interview, the creator was asked for permission before their imagery was utilized.

Lastly, the recordings of all participants, as well as any of their information obtained for this research were stored following The General Data Protection Regulation. However, it was indicated to all participants that due to the nature of a master's thesis, the data used within this study would be accessible to examiners for the master's program.

3.4 Limitations of Research

The main limitation of the research is the small sample size of industry professionals, making it difficult to find proper correlations. Further research should focus on increasing the sample size, as well as increasing the number of experts on 2D-normal mapping included in this research. If possible, a focus group might allow for more useful research results than interviews concerning the first interview. Therefore, it is recommended that future research is performed with a focus group rather than individual interviews with experts.

During this interview, a specific focus was put on 2D-normal mapping tools focused on pixel art. This was mainly done due to the focus a lot of the current tools put on pixel art. Further research could potentially be run utilizing different tools, or by restricting the target group of the second interview to artists already experienced with 2D-normal mapping tools.

Additionally, language barriers currently make it difficult to access groups outside of the English-speaking world for interviews. Further research should also focus on other sides of the world, as a difference in workflow could lead to significantly different results.

While ethics concerns do not currently restrict the research, there are potential ethics concerns that might apply to further research. Due to this research not utilizing AI tools, AI ethics are not currently in play. However, due to the developments at the time of this research, future tools will likely utilize the AI-image generation technique. This will require added ethical considerations before research can be done.

Chapter 4: Data and Results

4.1 Initial Test run Results and revisions.

Just like the final version, the initial test run consisted of 2 different interviews, both of which were run twice in preparation. Changes were made to the final interview plan listed in this thesis based on the results and insights gained from these trial runs.

4.1.1 Test Run 2D-Normal Map Evaluation Interview

After defining the basic methodology for the 2D-normal map evaluation interview, test runs were performed to verify the quality of the experiment before the proper data gathering. The main insight gathered from these test interviews is the following.

While the questions themselves can derive usable answers, this only applies when the participant is experienced with the topic. Trying to run this interview with participants who are only experienced with conventional normal maps results in relatively weak answers. Therefore, it is worth restricting the target group of this thesis, to preserve the quality of the results.

During the interviews, it became clear that the participants sometimes did not know what information I needed. Originally this resulted in a more comprehensive introduction, but after more deliberation, it was decided to instead reform the questions to be less vague. Furthermore, it is necessary to ask more follow-up questions, especially when someone voices a strong yet vague opinion. While it was not done during the original test, it became clear from reading the transcripts that there was room for further questioning.

After running the tests, a few changes were made to allow for deeper and more insightful answers. Firstly, another question was added to differentiate between the quality evaluation metric and the point of success. This was done to clearly define a difference between the utmost point of quality, and the point of satisfaction. A question was also added regarding the quality of normal map evaluation tool output. This was done out of a need to understand the criticism that these tools receive.
Each of these tests was attempted in the same room as the participants. This did not work out as well as the test done online, as it caused messier recordings and did not

allow for the efficient note-taking that an online testing environment would permit. Therefore, it was decided to run the final experiment fully online.

4.1.2 Test Interview Tooling Quality Interview

To test the tooling quality interview, two interviews were done. To finalize the Q-sort methodology used, one interview would be done with the established Q-set, while the other would be done with the exploratory Q-sort methodology.

Both test interviews showed some benefits for each approach. The original Q-sort methodology took significantly less time than the exploratory Q-sort, taking only an hour to complete. In comparison, the test run with the exploratory Q-sort took closer to one and a half hours to complete one run. However, the exploratory Q-Sort ended up deriving significantly more useful results, as was explained in 3.2.4.

Miro was chosen as the tool for the Q-sort. During the testing process notepad was used for the Q-sort, but this made things more confusing for the participant. Hence the choice of Miro, a clear tool with visual sorting capabilities.

The original interview put more of a focus on utilizing the Usability Metric for User Experience (Finstad, 2010) to evaluate the participant's experience with the tool. This was found to be irrelevant data, as the tool's usage was meant to facilitate the Q-Sort. While the UMUX could help indicate the quality of the specific tool, this is out of the scope of the overall research question. Instead, the questions are now used as guiding questions to help the participant select categories for the Q-sort.

Lastly, after feedback was given during the tests, it was decided to expand the explanation of the research and topic to include a more extensive explanation of the topic. This way the participant can be more properly informed (Appendix B).

4.1.3 Data processing for testing interviews

After performing the test interviews, data processing was done on one of the transcripts to verify the amount of time needed for the process. In total, tagging and analysing the important points of a single interview took a total of 3 hours when including the time needed to perform basic cleaning. From the interview analysis, the following becomes clear from both interviews.

To analyse the normal map quality interview, a thematic analysis will be performed on the obtained interview transcripts. After comparing different participants and their artist profiles a hypothesis can be written on what makes a good normal map. This hypothesis will then be used during the process of writing the requirements to describe the qualities needed to derive a good normal map.

The artist tool interview will also go through a thematic analysis, which will be used to compare the answers and arguments given to the QSort itself, to derive the most important qualities of a normal mapping tool. Additionally, this research will look into topics spoken about positively and negatively across multiple interviews. Then based on the Q-sorts and the additional weight given to certain topics, a list of weighted requirements will be written, categorizing the most important requirements for a good normal mapping tool.

After the interview processing has been finished, requirements will be written according to Specific, Measurable, Attainable, Realistic and Testable (SMART) specifications, established by Mannion and Keepence (1995). These requirements will then be weighted according to their importance, indicating their value to any future developers.

4.2 Final Test Results:

After proper adjustment of the methods the final tests were held. These consisted of two sets of interviews. Three expert interviews, one of which was eventually done through text, in order to establish a quality standard for 2D-normal maps, and seven artist interviews answering the second research question and establishing important qualities for artist tools.

4.2.1 Data Processing Methodology Artist Interview:

It was decided to analyse the artist interviews first, as the expert interview can be used to explain patterns and themes found within the artist interview. The artist interview output three different relevant files, there’s the transcript of the entire session, the video files which show the movement of items during the Q-Sort, and a final .png file of the Q-Sort.

The analysis of the artist interviews was done using the following steps. Firstly, on a spreadsheet every single item from the Q-sort was noted down. The final ranking per participant was noted down before calculating each item's frequency and total rank. Items with a very high or very low total rank were then highlighted, as they are particularly interesting. The standard deviation per item was also calculated, in order to get an indication of each item's divisiveness between different participants. If a participant explicitly chose and justified why they did not include a certain item from the Q-Sort, it was also indicated in this spreadsheet.

In the second spreadsheet, the movement of each item is recorded per person. Additionally, it is noted how many people moved a specific item, to establish any items that were deemed especially difficult to rank.

The final spreadsheet was created for the purpose of a thematic analysis. In this document each item was sorted into one of six themes. These themes were created based by grouping each of the features into different goals as follows:

Automation: Each of these features is meant to minimize the time spent performing manual work within the tool and any other time-saving features.

- Automation
- Software is present within art program.
- Autosaving feature
- Game engine integration.
- Lighting presets

User Interface: Each of these features is related to how the user directly interacts with the tool, and the visuals and user interface of the tool.

- Intuitive UI
- Navigator Settings
- Simplistic UI
- Tabs/Multiple Edits at Once
- Customizability
- Visual Appeal of Tool
- Editing Accuracy

Normal Map Quality: Each of these features serves to enhance the quality of the final normal map, be it through generation or manual work.

- Normal Map Generation Quality
- Paintbrush for Manual Edits
- Material Settings
- Parameter Editing and Saving
- Exclusion Painter
- Generation of Different Map Types

Enables Experimentation: Each of these features enables a trial-and-error workflow, that quickly lets you test and experiment with the normal map.

- Accurate Live Previews
- Undo/Redo Tool
- Lighting Simulation Quality
- Experimentation
- Performance
- Live-Updates for Assets

Traditional Artist Features: Each of these features is commonly found within industry standard artist tools.

- Animation Support
- Layers
- Keyboard Shortcuts
- Copy/Paste/Delete
- (non)-Pixel-Art-Viable
- Colour Settings
- Button Controls
- Artist Brushes
- More Complex Backgrounds
- Undo/Redo Tool

Ease of Learning: These philosophies or features impact how easy it is to pick up and use the tool without much prior knowledge.

- High Quality Documentation
- Ease of Use

Based on these established themes a thematic analysis was done on each of the seven transcripts. No specialized tools were used for the thematic analysis, instead important phrases from each of the interviews were noted down, sorted and split into these themes. The results of which were then used to establish patterns or support findings within the Q-Sort.

4.2.2 Results Artist Interview:

In total, seven artists were interviewed in this round of tests, they will be identified by P(n) with n going from 1 to 7. To start the interview, each of them were asked a few questions to help form an artist profile. Of them, P1, P5 and P6 had little to no experience with dynamic lighting or normal maps, be it in 2D or 3D. P2 and P4 have worked with normal maps and dynamic lighting before, but only in 3D and the remaining 2 participants P3 and P7 had high amounts of experience with normal maps, both having drawn normal maps in the past. Most of the participants in this round were students, each having around 4 years of academic experience in visual arts, though P4 and P6 also had experience working as a freelancer and P6 also mentioned two and a half years of game industry experience.

When asked about their specializations, P1 and P3 mentioned an explicit focus on 2D-illustration, the others all draw a mix of 2D- and 3D-art. P3, P5 and P6 also noted experience with pixel-art and P5 talked about a specific specialization in animation. Lastly when asked about industry standard tools, the most common tools mentioned were Adobe Photoshop and Clip Studio paint for 2D-art, with Aseprite also being mentioned for pixel-art. In regard to 3D-Art software, Blender, Maya, Zbrush and Substance 3D all came up as commonly utilized software.

When analysing the Q-Set the first thing worth noting is the categories, based on each Q-Set and the definitions that the participants gave for each item descriptions could be written for each item. For the most part, participants agreed on the definitions of the items. However, there were a few items that were interpreted differently. The speed of a tool was a contentious definition, as often it was unclear whether the intention was to speak of a lightweight tool, a tool with a fast-running time or both. Eventually during the data analysis process, it was decided to combine the items that factored into the performance of the tool into the performance category. Generally, features and philosophies were combined as soon as achieving one logically achieves the other. Copy/paste and delete were combined for this reason. And so were parameter editing and parameter saving. In Appendix D, the definitions for each category used in the Q-sort can be found.

When looking at the results of the Q-sort in figure 4, several categories immediately stick out. Normal map generation quality immediately stands out as the highest-ranking item. Being ranked highest by P3, P4 and P7, and never being ranked any lower than a rank of 1. There is however some disagreement on whether normal map quality is ultimately the most important item. Three participants ranked it highest and thus consider it the most important thing. With one of them stating that the upper limit has to be as high as possible even if it takes effort to get there. According to these participants, there is no point to using the tool when this feature is not as good as can be. Participants often came to this realisation in the middle of their Q-Sort, which lead to two participants moving the item to the top halfway through, this can be seen in Figure 3.

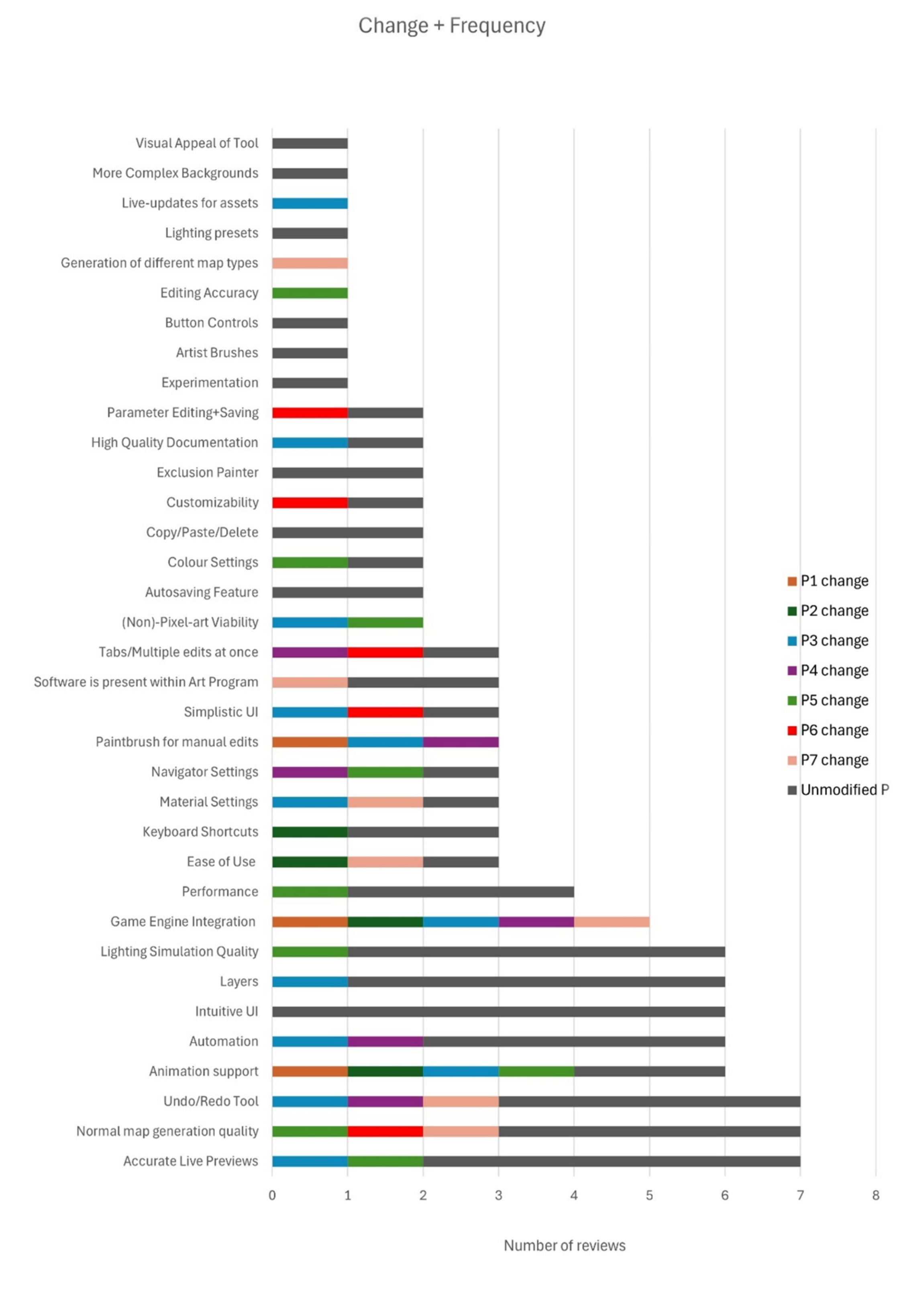

Figure 2.1 ***Frequency Graph****: A bar chart visualizing the number of participants that moved every item combined with the total frequence of every item in the Q-Sort. Bright colours indicate a change by a participant. Items are listed in increasing frequency.*

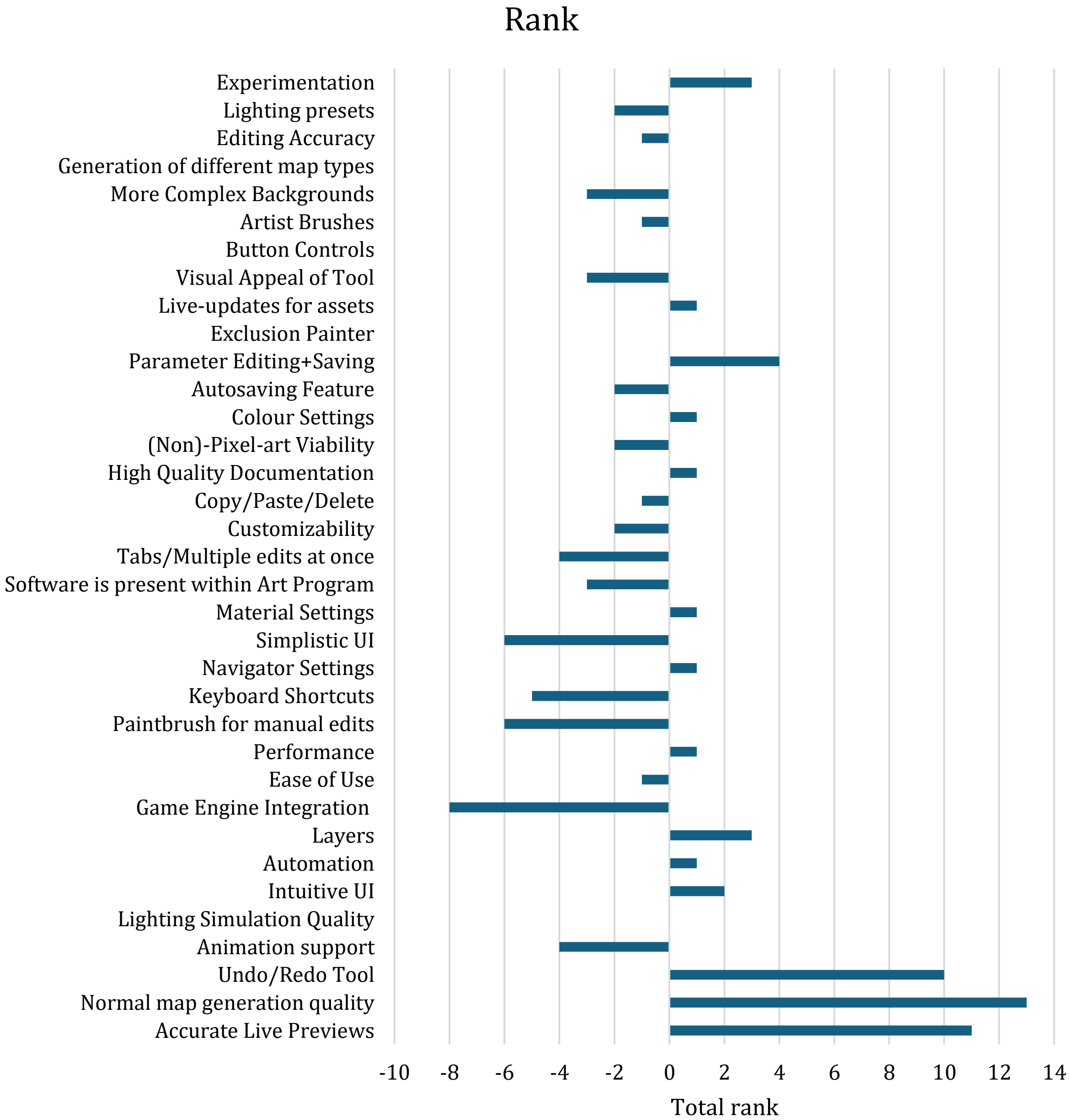

Figure 2.2 ***Total Rank Graph****: A bar chart visualizing the total rank for every item listed in the Q-Sort. Items are listed in the frequency they appear in from most to least frequent.*

However, the other four participants disagree with this notion. One of them stating that while the normal map quality is immensely important, there are categories present that are so vital to the tool, that the tool stops being worth using when they are no longer present. P5 described it as follows: "There's no point to existing if features such as undo/redo and layers are not there, you can only sacrifice so much quality of life".

P6 chose to prioritize parameters over the upper bound of normal map quality, arguing that there is no such thing as one good way to generate a normal map. They argued that parameters and manual corrections could make up for a sacrifice in generation quality as many normal maps aren't the same to begin with.

Two other categories that ranked similarly highly to normal map generations were the categories Undo/Redo and Live Previews. Undo/Redo was mentioned as an absolute must have multiple times, with P7 mentioning that it becomes even more important if your tool utilizes parameters. P5 additionally stated that it is highly unlikely for a tool to receive any following without undo/redo present as experimentation is so important to artists. Most participants agreed, stating it to be a core part of the workflow. Live previews were similarly ranked as important due to the artist's goal of experimentation. It was mentioned by multiple participants as a necessary thing for this software to be functional, with P2 even ranking it highest, stating that you're basically working blind without this functionality. Undo/Redo did see quite a bit of movement during the Q-Sort, with three participants moving it around, be it up or down. In two cases this happened in contention with other high-ranking categories that were all seen as must haves. Though in one case it was moved up from a -1 rank upon reflection from the participant.

Keyboard shortcuts and simplistic UI were two elements that were ranked significantly lower than a lot of other elements. Despite this however, many participants shared their support for having a more simplified UI that prevented visual clutter. The main reason many moved both of these category's low was as most of them stated: "These categories are really nice to have but aren't an absolute requirement for a good tool".

Perhaps one of the most divisive topics seemed to be the question between game engine and artist tool integration. Though artist tool integration ranked significantly higher than its game engine counterpart, it is worth noting that artist tool integration also suffered from more explicit rejections than any other category. P3 argued in favour of having the tool integrated with the engine, stating that it's bothersome not being able to tell how good the sprite will look in the final game. As the engine lighting will look different from the simulated lighting. P7 speaks very positively about having it integrated into their art software, stating that it could save a great amount of time. As you can benefit from all the artist tooling being already integrated. Multiple participants stated to support neither however, as the tool could be more focused as a standalone piece of software. Proponents of a separated tool stated that artists are already used to switching between tools, therefore it is not a problem to add another (lightweight) tool to their portfolio so long as it is compatible with their toolkit.

Though animation ranks average, it does see quite a bit of movement during the Q-sort. These cases primarily occur based on an explicit bias being utilized or thrown out. One participant stated that they ranked it high, despite knowing a lot of people would not use it. But another participant instead increased its rank. They explained that although they wouldn't use it, it could be important to artists making in-game assets.

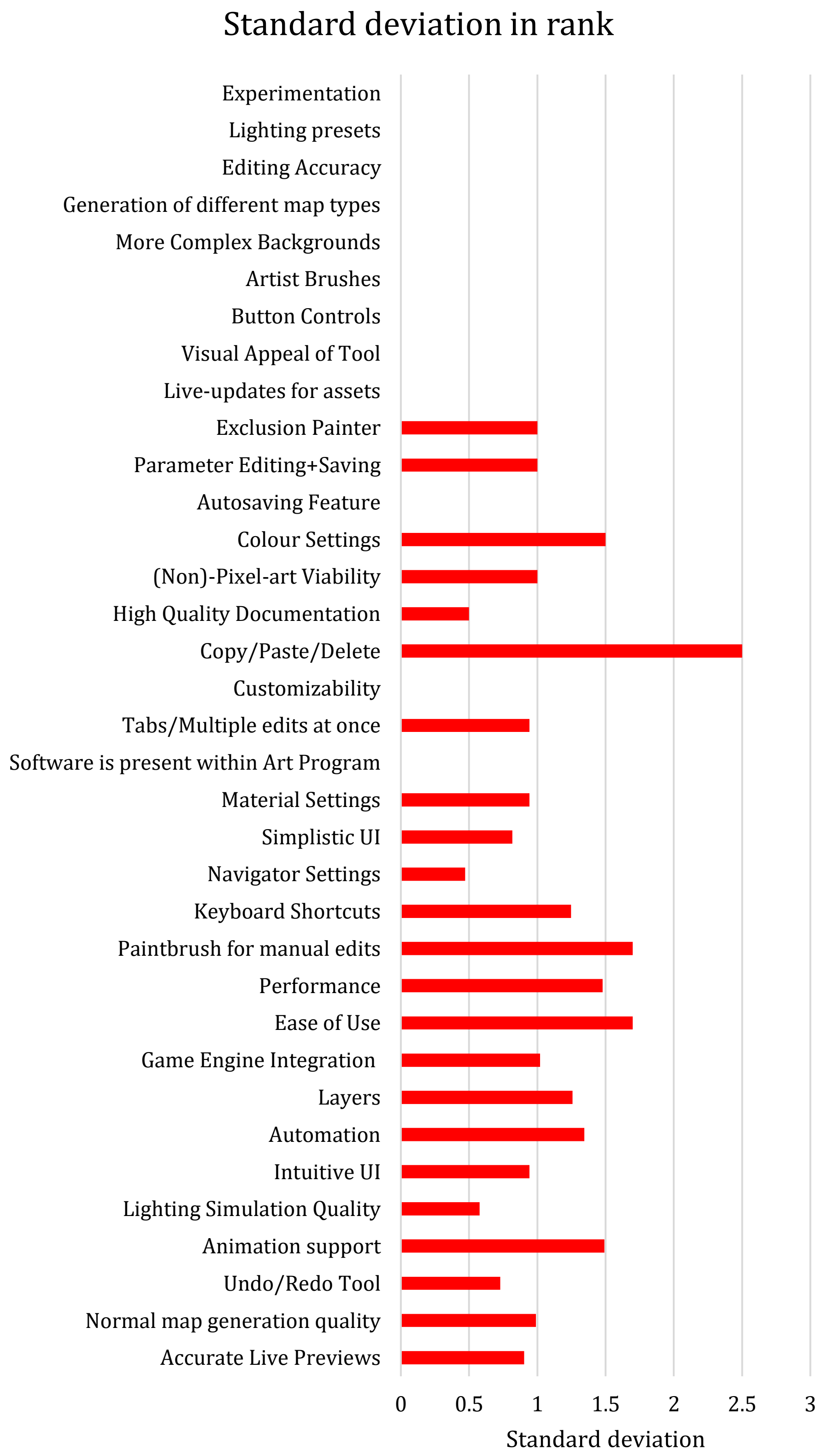

Figure 2.3 ***Standard Deviation Graph****: A bar chart visualizing the total Standard deviation for every item listed in the Q-Sort. Items are listed in increasing frequency of appearance.*

Though no patterns can be explicitly established based on this data. It is worth noting that due to the usage of an exploratory Q-Sort. Not every category was used the same number of times. As seen in figure 4 it is worth noting however is that the three highest ranking categories all appeared in every single Q-Sort. There are also some more niche categories that were only mentioned once or twice. One worth noting is (non)-pixel-art viability, which despite only being mentioned twice was moved in both Q-Sorts, indicating some difficulty in placement. One participant mentioned the following when moving it down: "While I care deeply about pixel-art being supported, a lot of pixel-art games also look fine without this technology".

The standard deviation of each category as visualized in figure 5 is another metric worth looking at. As it can be a good indicator of specifically divisive topics amongst the different participants. Perhaps the most controversial item in the entire Q-Set is the paintbrush for manual modification. Being moved around multiple times by every single person that named it, it is also one of the lowest ranking items. When discussing this item, participants were quite divided. Some of them argued that the sprite should not be touched in a program like this, and that manual edits should only be done to the normal map. While P6 and P7 would even rather edit the normal map in their conventional art program, worried about parameters or a lack of quality brushes respectively. P4 however was quite positive about the idea of a paintbrush, believing it to be a useful way to save time while making small manual corrections. Lastly, P2 argued that a good normal map generator shouldn't need manual corrections at all, as they are not used to doing this within 3D.

While the copy/paste/delete feature looks like a pretty contentious item, not many participants deemed it important to rank onto the Q-Sort. Additionally, it saw no changes in either of the Q-sorts and participants had very little to say about it.

Colour settings was actually quite contentiously ranked. P2 argued it to be one of the core tools needed in art software. While P5 saw it as another feature that would enhance the tool but wasn't a make or break for this type of software.

Performance was another pretty controversial topic, as some participants recommended simply acquiring better hardware to keep up with the state of new software. While other participants like P5 argued that accessibility is very important for artist tools, since other heavier tools like engines already weigh down your system.

Ease of use was also quite a debated topic. It was noted to have a high standard deviation between the three participants that considered it. One participant named is as quite important, but the explanation clarified that the ability to experiment and mess around with the tool was more important than the learning curve. Some rejected the idea of having an easy-to-use tool, such as P2 and P3, who both suggested documentation as a better solution. P3 also mentioned that the tool being too easy to use could be a risk, as it could keep them from doing the proper amount of research needed to become knowledgeable about the topic. Which could be an issue when trying to learn this skill for a professional project. P4 argued against this however, stating that learning by using is a better way of learning a tool than by reading tutorials or documentation.

Lastly it is worth looking at the overall themes, and how participants looked at those. From the visuals it becomes clear that items ranked under normal map quality and experimentation ranked significantly higher than the other categories. With items listed under automation and user interface commonly ending up at a lower rank.

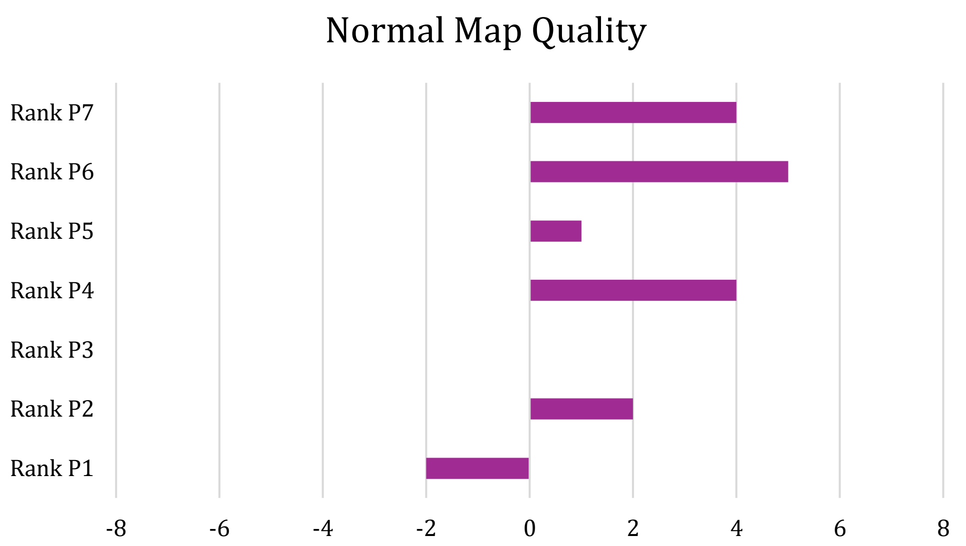

Figure 2.4 ***Normal Map Quality Theme Graph****: A bar chart visualizing the total rank of normal map quality for every participant.*

When discussing normal map quality, the same argument is brought up by nearly every single participant. It's the core of the tool, therefore, features that help you achieve the desired quality for your project are attractive. Parameters, exclusion, and masking were all mentioned as features that highly helped in increasing the quality of the final map.

Use cases mentioned included masking off outlines on pixel-art, excluding shadows on sprites, or splitting off certain darker parts from the normal map to be generated separately. Often these features were mentioned to add more manual work, however they were generally considered to be acceptable, given that the quality was seen as the most important metric.

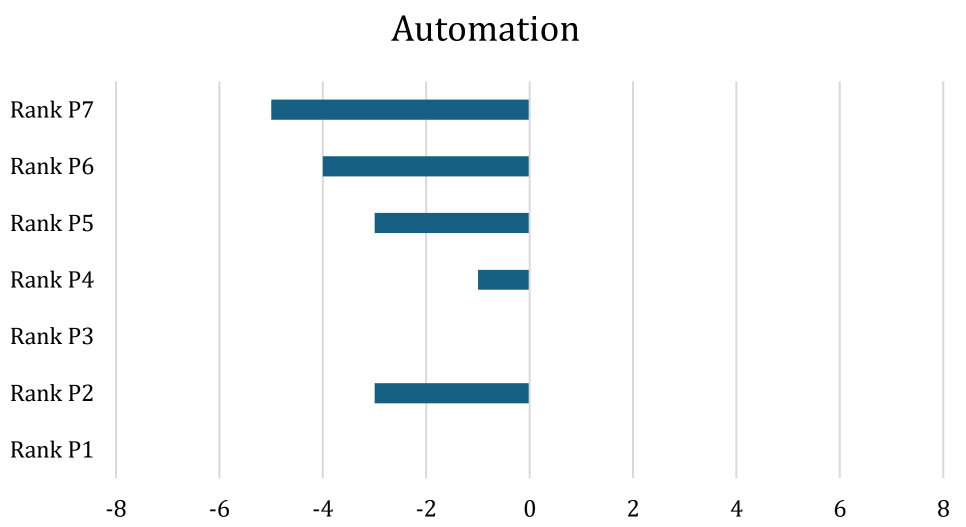

Figure 2.5 ***Automation Theme Graph****: A bar chart visualizing the total rank of automation for every participant.*

However, despite automation being ranked quite low, multiple artists still mentioned that the tool needs to be able to save you time compared to manually drawing the maps. Otherwise, it stops being useful. As P4 stated you can't be required to do so much manual labour that you might as well draw the entire normal map by hand. The Sprite Lamp software was named as an example of automation done wrong, where you have to do so much manual work that you lose the benefits of automation.

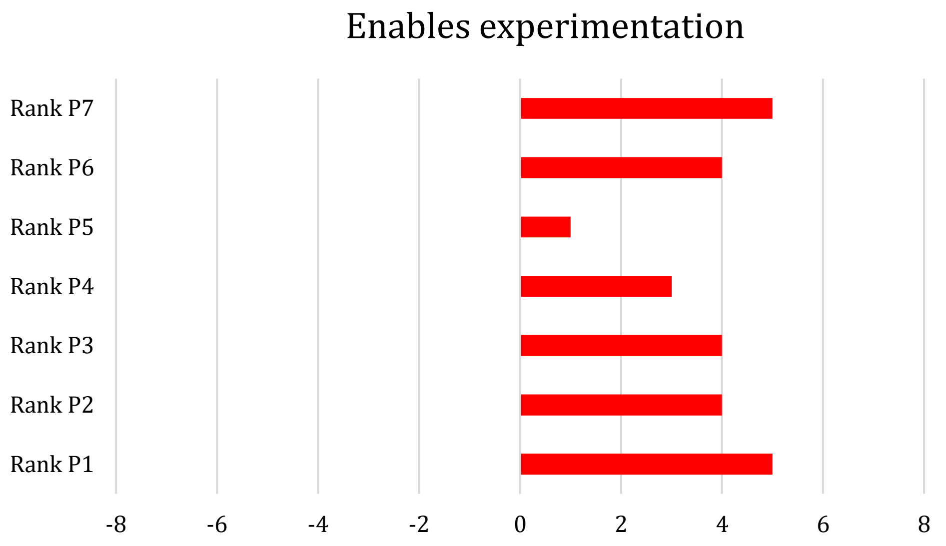

Figure 2.6 ***Experimentation Theme Graph****: A bar chart visualizing the total rank of experimentation for every participant.*

Experimentation has two very high ranked items. However, it is notable that none of the items listed under this category have a combined rank below zero. Comments about these features mostly go in depth about the degree to which they either help speed up the workflow or enable a useful trial-and-error workflow that artists like to work with.

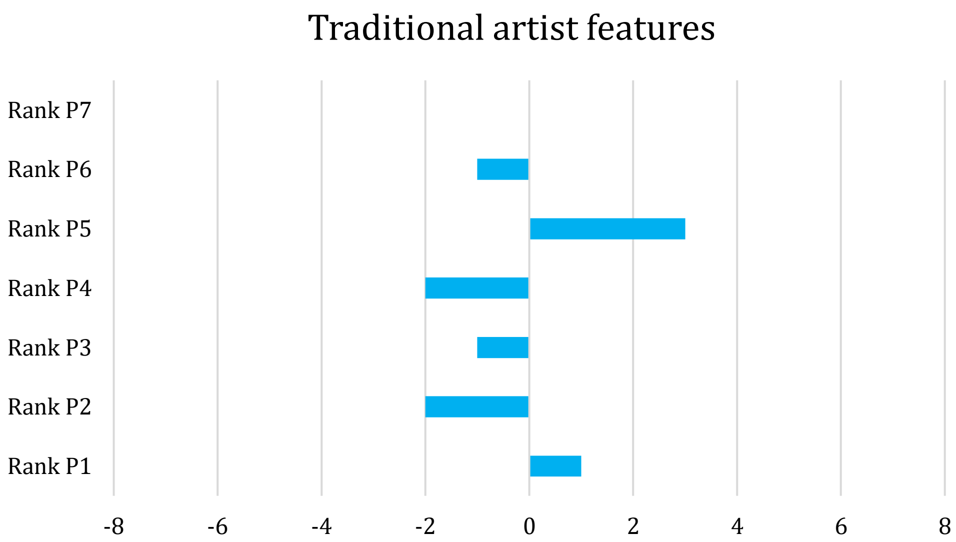

Figure 2.7 ***Traditional Artist Features Theme Graph****: A bar chart visualizing the total rank of traditional artist features for every participant.*

When discussing traditional artist features, there seemed to be a general consensus that layers were very important. Multiple participants argued that layers are necessary as they allow you to view the normal maps in the greater context by allowing you to setup entire scenes. Additionally, it is mentioned that layering multiple types of normal maps can have some utility, as it is also done in some 3D-software. Lastly, a few of these items suffered from simply being lower priorities, resulting in a lower rank.

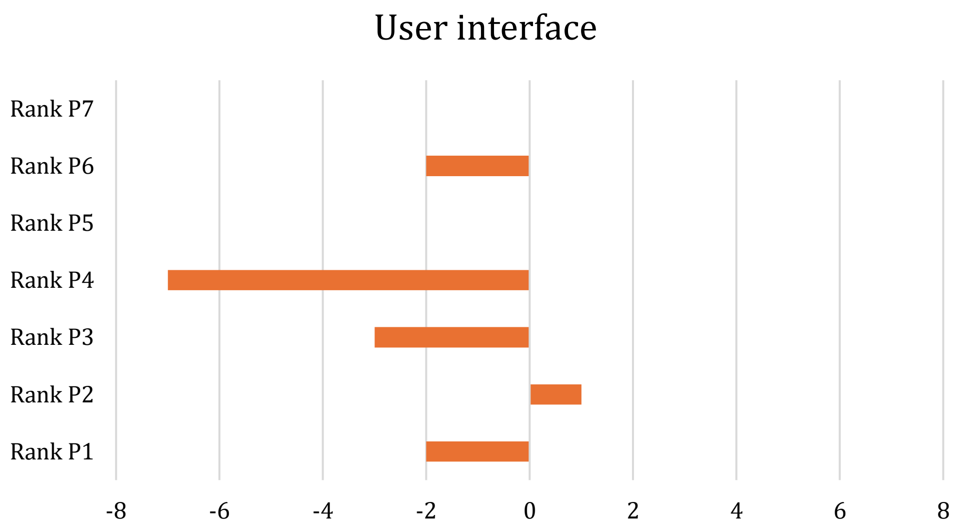

Figure 2.8 ***User Interface Theme Graph****: A bar chart visualizing the total rank of User Interface for every participant.*

Similarly, a lot of the items included in the User Interface category were classified in similar ways. Generally being seen as a lower priority than items belonging to different themes. However, participants did note during the demo that they enjoyed parts of the software that felt similar, such as being able to make tiny changes, or an intuitive scroller. Additionally, though many chose not to rank it, having proper visuals rather than programmer art was often appreciated.

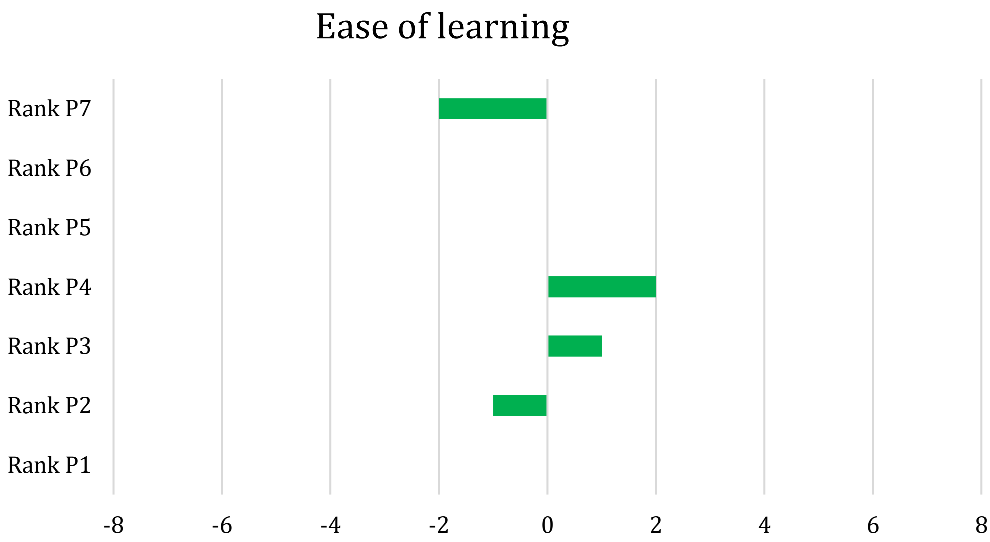

Figure 2.9 ***Ease of Learning Theme Graph****: A bar chart visualizing the total rank of Ease of Learning for every participant.*

As mentioned earlier. Ease of learning had a divide between participants wanting proper documentation and participants wanting an easier to use tool. Hovering guides were mentioned as a potential option for making the software easier to use, and so were tutorials. Though built-in tutorials were often looked down upon, with participants preferring external tutorials online.

4.2.3 Data Processing Methodology Expert Interview:

The analysis for the expert interview was done last, this way the focus could be put on the themes and patterns found most relevant within the artist interviews. The output from the expert interview only included the transcript of the interview session, or in one case the transcript of the text-based interview.

Due to the absence of a Q-sort in this interview, the primary source of data was a thematic analysis done on the transcripts of the interviews. Here, the focus was primarily put on the two primary themes of the interview: **Automation** and **Normal Map Quality.** However, during the analysis **Ease of Use** and **Experimentation** was also found as a relevant theme and therefore included in the thematic analysis.

4.2.4 Results Expert Interview:

In total, three experts were interviewed in this round of tests, each of which will be identified by E(n) with n going from 1 to 3. Of the interviewees, E2 and E3 were both artists having drawn normal maps for 2D-games in the past. However, while E2 mostly mentioned having limited experience with some lighting in 2D-games. E3 has a very high level of experience, being a very hands-on solo developer on a project that requires a high amount of 2D-normal map knowledge and application. E1 is not primarily an artist, instead they are a developer of various 2D-normal map plugins for the Aseprite pixel-art software. This requires knowledge of various 2D-normal mapping algorithms.

When discussing potential objective quality metrics for 2D-normal maps, there was unanimous agreement. All three of the participants agreed that there was little to no objective quality metric involved, and that most of it relied on the artist's judgement and standards. E1 mentions that it is all down to the artist's preference, as some artists want smooth and realistic angles, while others prefer working in a different way. They argue it all depends on the intended use case. Although they did also state that there are some absolutes, for instance a pixel should never turn fully black due to the lighting on the normal map. E2 and E3 agree with this sentiment, stating normal map generation to primarily be a trial-and error workflow where quality depends on the style. For realism you'll need more proper rendering, but when working with a stylized artstyle they argue you just need to be convincing. This goes back to the theme of experimentation, where all individuals argued that this trial-and-error process can go on forever. Therefore, it is vital that it is easy to quickly review and make minor changes.

P1 mentioned some interesting insights on how they develop these normal-mapping tools. They stated the paper by Moreira, Coutinho, and Chaimowicz (2022) to be a major inspiration, as they used its analysis of the Sobel Operator as the basis of their development. Based on their experience with these tools they argue that sometimes when you're involved with a 2D-Normal mapping project it can be good to create your sprites with the knowledge that you'll be generating maps for them. This way you can prevent certain issues before they even come up. For instance, by making your sprites slightly larger, or layering them in ways such that important edges are separated.

P3 does not use separated tools for their development, instead all their work is done using various scripts built for the Aseprite tool. They mention various benefits that they derive from using scripts within their artist tools instead of a separate tool. One of such is the ability to create different layers and use them to create a pipeline of multiple steps. This way they can generate edge normals first, which the tools can do very accurately. Before generating the rest of the normal map and layering them over each other. This process results in a more accurate normal map, with little added work. P3 states that they overall do very little extra work, with most of the manual work being some directionality added to the normal map by using a normals brush already included out of the box in Aseprite. According to P3, most of the changes being made are not to the normal map, but rather to the lighting.

On the theme of automation, there is a similar level of agreement between participants. All three argued the speed of the pipeline to be more important than quality overall. According to P1, most of the work can already be automated. With only some adjustments to the angles and the addition of some wear, tear and cracks being noted as manual work. While P3 agrees, stating that the tools can already do 80% of the work very well. They also mention that the tools can have an issue with darker colours and shapes, which can lead to more adjustments. P2 disagrees with this positive sentiment. In the interview they discussed how the current tools on the market require a lot more manual work than makes them worth. However, P2 does state that in a more professional production pipeline a lot of this manual work could be negligible. P1 and P2 both make another observation however, which is that not all normal maps are born equal. When designing a main character, both interviewees argue that it is worth putting a lot of extra time into. While a lot less time can be put into background elements that are rarely revisited. Lastly, P3 also quickly discussed the idea of generative AI in normal-map generation. They primarily voiced concerns about lack of consistency, especially in tile-maps. They argued that while the current tools generate consistent normal maps, normal maps generated by generative AI would include an undesirable level of randomness.

On the topic of Ease of use, P3 mentioned it being an overall barrier of entry to working with 2D-normal maps. According to their experience, resources are scarce and often difficult to find. They also mentioned there being a severe lack of experts in this field, making it difficult for newcomers to get into the field. On the other hand, since there’s so little experts, they also tend to have their own established workflow. P3 argues this makes it difficult for new tools to break into the field, as most developers won’t want to switch over to an unknown tool when they have a working tool for their project. P1 paints a similar picture, stating that while anyone can use these tools. It requires an expert to truly make these normal maps shine. They strike a comparison to utilizing an AI-language model to generate code without any coding experience. You'll have a finished product without the ability to modify or debug it properly. Without the necessary knowledge, you will never be able to make a normal map at the desirable level of quality. P2 adds to this in their interview, stating this overhead in learning and work to be a primary reason why companies refrain from utilizing this technology.

Chapter 5: Discussion of Data and Results

5.1 Data and Hypothesis

The results thus far indicate the high upper bound on normal map generation quality to be the single most important element for a 2D-normal map generation tool. Worth noting is that there is a clear divide between artists that value this as the most important element and artists that value other elements higher. This could potentially be explained by the fact the three artists that most highly rated this element all have prior experience with 2D-normal maps that some of the other artists did not possess. However, one could also argue that this element was always going to be ranked highly as it was one of the only 2 elements that was pre-included in the set of rankable elements. This does not seem to be the case, as the element of automation, despite being pre-included in every set, did not rank comparably high. Thus, these results do suggest that a high upper bound on the normal map generation quality is a highly desirable quality for the tools to possess.

The data does however include animation far lower than expected, this is surprising as animation is a staple feature in many of the tools already on the market as well as a very important feature in game development. A likely explanation can be found within the background of the artists interviewed. Many of the interviewees specialize in 2D-concept art or 3D-art rather than 2D-Animation. This is supported by the fact that the one 2D-animator in the dataset did rank animation especially high. Thus, animation is worth considering an outlier, as a more experienced testing group would have likely ranked it higher considering the primary purpose of 2D-Normal mapping tools for asset creation.

The data from the tooling quality interviews suggests a clear preference for normal map quality over speed and automation. This is a clear trend within the data, as none of the participants ranked the automation element over normal map quality. A potential reason for this is the target group of the study. Many of the participants in the study are current art students, not yet working within the industry. Therefore, a high amount of emphasis is placed on high effort assignments, rather than projects where time is a very high-pressure factor. This theory is supported by the fact that the experts look far more favourably at speed and automation. This is because they look more practically and project-oriented towards this topic, which results in a more nuanced look at the debate where the type of art needing a normal map warrants a different response. During the artist interview, the demo was done with a single character, this means the response might have been different had there been a demo with multiple different elements. Within the scope of this research, this was not done mainly to prevent overwhelming the participants, as well as to keep the interview time as short as possible. Therefore, while the artists and experts are split on the debate between the importance of normal map quality versus automation, it is believable to assume that the answer to this question depends on a piece-by-piece basis.

Some of the experts' statements put into question the conclusion delivered by Moreira, Coutinho, and Chaimowicz (2022). This paper proposes a number of generation methods, but none of them are capable of generating normal maps on a level of quality identical to those of handdrawn normal maps. However, according to the experts some degree of editing is always necessary. This means it is likely more worthwhile to compare different methods of 2D-normal map generation to each other, rather than comparing them to the handdrawn normal map. As it is more worth considering which 2D-normal map generation method brings one closest to the ideal.

The data shows the debate between artist tool integration, game engine integration and a separate tool to be highly contentious. Within the dataset, artist tool integration received more support than game engine integration, though neither ranked particularly high. Artists with more experience in the field generally ranked artist tool integration more highly. However, it is worth considering that artist tool integration also received considerably more rejections than its game engine counterpart. Despite this, it is likely that artist tool integration leads to more beneficial results than game engine integration would. This is mainly since the inclusion of such tooling in industry standard tooling already brings with it many available artist features, as well as the fact that it makes such tools more easily adaptable for artists. This is supported not just by the experts' claims, but also by the claims made by Lightbown (2021) in his GDC talk as referenced earlier in the literature review. When a tool is created as a plugin for an already industry standard tool, it becomes far more easily adaptable to the artist's current toolset, which might likely already include the industry standard tool. There are additional downsides to your artist's sudden dependence on in-engine tooling. A significant amount of the industry currently uses custom game engines, which makes it undesirable for such tooling to be engine-specific, rather than included in industry-standard artist tools.

When looking at the topic of ease of use, the data shows another clear divide between artists with a preference for documentation and artists that prefer proper tutorialization. When comparing this to the expert opinion it becomes clear that it might not be a simple debate between two options. Experts emphasized the lack of easily available information on this topic resulting in a severe lack of expertise. This leads to the idea that both might be vital to the success of such a tool, as current knowledge on the topic simply is not up to the desirable level needed to easily utilize such tools. This also became clear from the demo within the experiment, as many participants struggled utilizing the tools. One might propose this to be an issue of relevancy, where the topic lacks experts as it is currently undesirable. However, it is worth noting that the lack of projects utilizing 2D-Normal mapping is at least partially a result of the costs associated with it as stated by Feichtmeir (2023) many of which one of the experts attributed to overhead associated with learning and inexperience. Thus, it becomes clear that some combination of proper tutorialization and documentation will be necessary for a properly usable 2D-Normal Map generation tool.

Given the research done by Lightbown (2021), it is interesting to see many of the traditional artist features ending up with a relatively lower rank. I believe this can be explained as a side-effect of the methodology. Many of such artist features are small singular features rather than large overarching themes such as automation, normal map quality and ease of use, additionally such features are also competing with each other. Therefore, it is makes sense that many of these end up with a lower priority on the ranking. However, it is worth noting that lower priority does not mean unimportant. And with 10 different features, there are many artist features that were stated to be important by the artists, with many different artists naming features they were personally fond of. Hence, the result of this study makes sense when considering the claims made by Kasurinen, Strandén and Smolander(2013), specifically the fact that artists are more likely to want to use tools with capabilities they are used to from other artist tools.

5.2 The Research Question

At the beginning one research question was formulated:

Q: Can requirements engineering techniques such as those outlined by David Lightbown in 2015, be used to create a basic set of software requirements and technical evaluation criteria for the design of 2D-normal map generation software.

To answer this question properly we then split it into two sub-questions:

Q1: According to experienced 2D-normal map artists, what is the current quality standard for a good normal map?

Q2: What relevant features do current artists value within 2D-normal map generation tools?

5.2.1 Question 1

With the data gathered so far, we can answer the first question. Though not fully reliable due to the small number of experts, the group of experts in this study fully agreed that there is no objective metric by which we can evaluate the current quality standards of 2D-Normal maps. All experts interviewed within the scope of this research agreed that this seemed to be fully based on artistic taste combined with an experimental trial-and-error workflow. This matches the expected hypothesis, though future studies with a larger number of experts could provide more conclusive claims. These were outside of the scope of this research, due to the challenge of finding experts on this topic. The lack of an objective metric of quality would mean developers of future software will have to rely on rigorous testing and artist feedback to measure the quality of their tooling. These results may seem contradictory to the methodology used in the research by Su, Du, Yang, Zhou and Fu (2018), as that team did use an objective evaluation metric. However, it is worth noting that this methodology worked since the 2D-illustrations were based on existing 3D-models. When you introduce more stylized 2D it becomes more likely that this ground truth normal map is no longer the most desirable normal map for every artist. Additionally, for something such as pixel-art, having a true-to-3D normal map is undesirable to begin with. Therefore, the answer to this question is that based on the data there is likely no objective quality metric, the quality of a good normal map can only be evaluated by an experienced artists with knowledge of the technology and context of the asset.

5.2.2 Question 2

Based on the data there are a few patterns that immediately come to light. It is clear that features that positively influence the quality of the final generated normal map, as well as those that enhance the trial-and-error artist process are valued most highly. Automation is valued, but not particularly highly compared to other, more vital features such as undo/Redo, live previews and parameter editing and layers. Since the results of

Q1 do not result the idea of an objective metric of normal map quality, this once again supports the idea of directly involving artists in the development process of such tools. The expected hypothesis was that traditional artist features would be rated very highly; this is not entirely supported by the results. Though traditional artist features were generally seen as important in the data, they were generally not rated as the most important. One might question the validity of the results given the small sample size, but the choice for a Q-Sort was done primarily since a Q-Sort can yield valuable results with a small sample size such as this. Based on trends within the data one can thus pose the following recommendation: When developing 2D-Normal mapping tools, try to focus primarily on features that support the trial-and-error art process as well as those that enhance the quality of the final normal map. However, it is vital to make sure the tool requires less work than would be needed for a handdrawn normal map. It is recommended to test the tools with established 2D-Normal map artists, as they will be able to provide more concrete feedback on the potential improvements made compared to their established pipeline.

5.2.3 Overall Research Question

Based on the current data it is not possible to create a set of technical evaluation criteria for 2D-Normal map generation tools. However, it was possible to create a set of software requirements based on the results of the artist and expert interviews. This set of requirements can be found in the appendix (TO BE DONE) and can be used by future developers as a basis for the creation of their 2D-normal map generation tools.

5.3 Reflecting on the Methodology

There are a number of ideas that should be indicated on how the method can be critically reviewed and improved for the future. Firstly, elements that were explicitly rejected or stated and thrown out were not reflected within the results of the Q-Sort. Even if they were stated as important by other participants. This potentially resulted in these elements having a higher score even if other participants explicitly did not value them. However, outside of the debate between artist tool and game-engine integration this did not cause significant questions or outliers within the data as it mainly impacted elements around the middle of the rankings, rather than elements that explicitly stood out. Nonetheless, it might be a good idea to experiment with a way to properly include a rejection into the methodology.

Additionally, for future research the participants should be more specifically selected among potential future users of 2D-normal mapping tools. This way, bias for unnecessary features can be avoided. Due to time constraints this was not in the scope of this research, resulting in a few feature suggestions that may not be immediately relevant to the field.

The exploratory Q-Sort caused a small number of issues, due to differing terminology the researcher had to occasionally merge categories that were identical. However occasionally, some of the participants listed categories that were partially identical, or combined multiple separate categories, which made the data more difficult to analyse. Therefore, it might be worthwhile separating the exploratory aspect from the actual Q-sort methodology, this way categories can be appropriately named and unapplicable categories can be reviewed before the sorting is done. In this research it was unfortunately not possible due to time constraints and the small sample size. Additionally, the difference in knowledge the participants had on the topic did have an impact on the results, as some of the participants had significantly more difficulty coming up with enough categories for the Q-sort. This led to the researcher needing to supply randomized elements from finished datasets in a process where the participant could accept or reject the elements, which is undesirable as it introduces bias to the data. Because of this, no conclusions were made based on the frequency of elements. This can potentially be fixed in future research by either doing the exploratory part beforehand or by removing the minimum requirement on the Q-set elements.

While grouping each of the elements into themes, only one researcher was involved. This is undesirable as it can invite extra biases into the process. Therefore, during the research, the themes were only evaluated against each other, and individual elements were not evaluated based on their associated theme. In the case of a theme such as undo/redo which fit into multiple themes, it was put into each applicable theme to eliminate as much bias as possible.

The expert interviews were limited by a very small sample size; however, it can be argued that the rarity of experts within this fields means a small sample size is already quite valuable in answering such questions. While this means that no conclusive claims can be made based on this data, it is nonetheless incredibly valuable insight to be used while analysing the rest of the data and when combined with existing literature.

Lastly, while the interviews were a successful in gathering the data needed from the experts. It is worth reiterating that the potential value of doing a focus group instead could be very high. There are many potential discussions one can have in a focus group that are not possible in 1-on-1 interviews and while it was out of the scope of this research, it is worth considering if this research were to be repeated.

Chapter 6: Conclusion and Future Directions

While the data currently gathered can offer some guiding insights into potential answers and directions, it is important to look at other limiting factors raised during the research. Therefore, this study suggests that future research should focus more on methods to establish technical guidelines for 2D-Normal maps. As well as investigating the benefits of potential engine or artist tool integration of this functionality.

6.1 Research Conclusion

The primary goal of the research was to create a set of software requirements and technical evaluation criteria to guide the creation of 2D-Normal map generation tools by using requirements engineering techniques.

The first research question focused on establishing technical evaluation criteria of generated normal maps themselves. Though there was some variation in the personal approach of each of the three experts interviewed, they all agreed that currently there is no established set of evaluation criteria for 2D-Normal maps. Additionally, based on the expert statements we can raise the theory that the lack of public resources combined with the required expertise necessary to create usable 2D-normal maps therefore results in a lower output from smaller studios and developers. This would suggest opportunities for public research and a need for developing additional educational resources within the field.

The second research question focused on creating a set of software requirements for the purpose of guiding developers towards the features that artists value most. By means of an exploratory Q-Sort, the categories have been rewritten into software requirements and added to Appendix E. These requirements are ordered in importance based on the results of the Q-Sort. This document could be used by future tools developers as a first step of the process, to create more appealing tools for aspiring 2D-Normal map game developers and artists.

6.2 Future Research and Development

The results of this research have raised a number of additional fields of interest. Most of these questions are a direct result of the research results, though some are born from research limitations.

One of the most controversial elements that arose from the artist tool Q-Sort was the choice between artist tool or engine integration of 2D-Normal map generation tools. Future research could focus on outlining different benefits and opportunities present within current artist and engine tools for the implementation of such technologies. Though this idea is supported by Lightbown's (2021) statements on tools, various statements by participants support keeping the tool separate. Therefore, future research with a more focused target group could confirm the most attractive approach.

While the artist tool interviews suggested normal map quality to be the most important factor, data from the expert interviews suggest that there is currently no proper way to properly evaluate 2D-Normal maps. There are two main ways future research could tackle this problem. The critical approach would be to verify these findings with a larger sample size of experts from outside the scope of this research, this would include developers outside of Europe and the United States, as well as experts from the industry that weren't accessible for this study. However, future research could also focus on the creation of such an evaluation model. Be that through creating a set of artistic guidelines or by utilizing emerging technologies such as generative artificial intelligence. Though the expert data warns against the risks of utilizing GAI, as the randomness involved is a major risk factor. Overall, this research suggests Q-sorting to be an effective tool to add to the requirements engineering portfolio, especially for projects under stress of time and limited scope.

Lastly it is worth reiterating that for the sake of the advancement of the field, it is vital that future research and development into the field happen transparently and with proper documentation. This way the technology could become more widely usable within the gaming industry.

Appendix A: Scrapped Q-Set used for Test Interview 1

Quality of the output:

- The algorithm should output high-quality normal maps for shaded and unshaded art.
    - *SpriteDLight (Faas, 2014), one of the currently relevant tools lists features on their website that are said to help create normal maps for both unshaded and shaded maps. By including this feature, I am testing whether the quality drop for unshaded maps matters to artists. The paper by Moreira, Coutinho, and Chaimowicz (2022) also experiments with unshaded art, which makes it a relevant point to consider.*
- The algorithm should be capable of outputting a diverse amount of different normal maps, such that it can confirm exactly to the type of game I am developing.
    - *There are multiple ways to shade a 2D game. This point mainly raises the question of how relevant it is for the tool to be able to generate different types of normal maps. Tools such as SpriteDLight (Faas, 2014) and Laigter (Azagaya, 2020) all contain multiple sliders used to influence the final normal map, so it is relevant to test to what degree this customizability is important to artists.*
- Accuracy: How important is it for the normal map to be as accurate as the shades/shapes of your sprite
    - *This is the basis of quality, Moreira, Coutinho, and Chaimowicz (2022) establish in their paper that the main purpose of 2D-normal maps is to establish clear shape language with dynamic lighting. This topic means verifying its importance.*
- Consistency: How important is it for the tool to be capable of creating normal maps with consistently similar depth values?
    - *This is mainly relevant due to the recent influx of AI development, such as those outlined by, He, Xie, Zhang, Yang and Miyata (2021). AI-based methods often have a challenge, in reproducing similar results, therefore it is relevant to test the importance of this issue.*

Testability:

- Iteration: The ability to quickly make changes and create another normal map (live previews)
    - *This mainly concerns the speed at which the tool generates its various normal maps. Many developments to normal-map generation tools are done to increase the speed and performance of the algorithm.*
- Simulation: The ability to display lighting within the tool in many different ways.
    - *This feature is the bread and butter of testing. It is asking how important it is to be capable of testing the normal map under many*

different types of lighting. How much do artists value the ability to test within the tool?

- Correction: The ability to make corrections and revisions within the tool, to go back and forward.
    - *A very common artist tool feature that was added to further verify the importance of common artist tool features.*
- Ability to make manual edits to the normal map after creation.
    - *An artist tool feature present in SpriteDLight (Faas, 2014), due to my earlier stated hypothesis, it is interesting to verify the importance of current artist tool features such as this one. On top of that, due to the quality issues stated by Moreira, Coutinho, and Chaimowicz (2022), it is relevant to evaluate whether this is desirable.*

User Interface and Artist tool features:
User interface and artist tool features were grouped for one main reason, the user interface topics are often based on current artist tools as conversations with artists have highlighted the similarity of UI in industry-standard artist tools. Therefore, it makes sense to group them.

- Clarity: It's clear what every element of the tool does and what the most important variables are
    - *A basic category that serves to evaluate how important it is for the tool to have a clear and easy-to-understand user interface.*
- Advanced Artist tool features: I need this tool to feature elements such as Layering, and animation included in the previews.
    - *Very common artist tool features, it was added to further verify the importance of common artist tool features.*
- Modification: I want to be able to modify the image within the tool
    - *Another feature present mainly in SpriteDLight (Faas, 2014), that brings the tool closer to an actual artist tool.*
- Minimalism: I want the UI to take up as little space as possible
    - *This topic is meant to be a natural opposite to clarity, and serves the same purpose, to what degree artists value a clear UI.*

User Experience:

- Integration: I want this tool to be integrated well within my other game development tools.
    - *This topic is once again based on Lightbown's (2021) recommendation for tools to be well integrated into the current development tools used by your target audience.*

- Presets: I want preset indicators for settings that make for good normal maps.
    - *A category that reduces the amount of customization and prior knowledge needed and a feature already present in several artist tools.*

- Ability to include manual pre-work: I want to be able to input a height map alongside my normal map.

    - *The ability to include a height map is based on Moreira, Coutinho, and Chaimowicz (2022) multiple height map-based methods, some of which have also been adapted into the different tools available.*

- Automation: The tool should feature as little manual work as possible

    - *This topic is included based on the purpose of the research. To automate the process and reduce the amount of manual work, to what degree do artists want manual work to be reduced?*

Appendix B: Interview Plans

Interview Plan: 2D-Normal Map Evaluation Interview

Start Of Interview

- *Initialize recording + Webcam (Confirm that it works)*
- *Introduce yourself*
- *Confirmatory Questions*
    - *Name*
    - *Current occupation*
    - *Position*
    - *Years of Experience in the field*
    - *Experience with the topic*
    - *Level of Experience with the topic*
- *Explain Stimulated Recall*
- *Thesis Explanation*
    - *Explain relevance of topic: "Why are 2D-normal maps relevant"*
    - *Explain what I'm trying to do with requirements engineering*
    - *Explain why I need the participant's help*
- *Explain and discuss consent form*

2D-Normal Map Questions

- *Have you ever created a 2D-normal map by hand?*
    - *If yes, what do you look for when creating a 2D-normal map? Can you walk me through your process?*
- *When would you consider a normal map to be usable for a project? When are you finished?*
- *Do you think there is any objectivity present in the creation of normal maps, or is it all up to artistic taste?*

- *What is in your eyes, the main goal that the usage of 2D-normal maps intends to achieve?*

- *Within the development pipeline, where do you lie on the balance between speed and quality of a 2D-normal map?*

2D-Normal Map Tooling Questions

- *Do you have any experience using any normal map generation tools on the market, could you walk me through your experience with them?*
- *How would you evaluate the normal maps generated by such tools?*
    - *If you think the output of such tools is not up to standard, what about these normal maps is lacking? By what metric would you judge them?*
- *To what degree do you think automation will catch up to the manual creation of normal maps in the future?*
    - *Why do you think it has yet to become a major part of 2D-game development?*

Closing Interview Questions

- *How do you feel about the current state of 2D-dynamic lighting in games? Do you feel it will become more prevalent in the future? Do you feel that any other fields could provide major value?*
- *Do you have anything you'd like to add to this discussion?*

End of Interview

- *Thank the interviewee for their time*
- *Let them know they can contact you any time for amendments*

Interview Finished

Interview Plan: 2D-Normal Map Tooling Quality Interview

Start Of Interview

- *Initialize recording + Webcam (Confirm that it works)*
- *Introduce yourself*
- *Explain Stimulated Recall*
- *Thesis Explanation*
    - *Explain relevance of topic: "Why are 2D-normal maps relevant"*
    - *Explain what I'm trying to do with requirements engineering*
    - *Explain why I need the participant's help*
- *Explain and discuss consent form*

Artist Profiling

- *What is your name?*
    - *Serves as a basic introductory question.*
- *How much experience do you have as a visual artist?*
    - *Used as a basic metric to evaluate level of expertise.*
- *What tools are you most familiar with using while doing digital art?*
    - *Used to evaluate the industry standard tools used by the participants.*
- *Have you ever worked on a 2D Dynamic game before, if so, have you used any normal map generation tools before?*
    - *Used to evaluate the participant's experience with the topic.*

Demo

- *Remind the participant about the necessary software*
- *Send them the sprite through teams and tell them how to import.*
- *Explain the basic features*
    - *The modification menu*
    - *The relevant features*
    - *Explain you won't be able to offer any help*
- ***Participant is given 10 minutes for the demo***

Demo Reflection/Q-Set Creation

- *Explain the concept of a Q-Sort and open up Miro, send the link to the participant as well.*
- *Ask the reflective questions*

- *What did you like about the tool? Did it have any capabilities that made it easy to use?*
- *Did any parts of the tool stick out as particularly frustrating?*
- *Are there any capabilities that are noticeably absent within the tool? Anything you may be used to, that is now clearly missing.*

- *Afterwards, tell them to create a Q-Set with a total of* ***20 Elements.***

Q-sort

- *Remember to ask questions whenever the following occurs*
    - *An item is placed*
    - *An item is moved*
    - *An item is swapped*

End of Interview

- *Thank the interviewee for their time*
- *Let them know they can contact you any time for amendments*

Interview Finished

Appendix C: CONSENT FORM 2D-Normal Mapping Interview BUas

Welcome and thank you for accepting the invitation for this interview.

This interview is part of a larger study attempting to create a framework for 2D-normal map generation tools, such that future developers working on such tools know what makes a good normal map to artists, and what features artists need in such a tool. Within this interview, you will be presented with several questions about your workflow and opinions on the topic. Your responses are anonymous, in the sense that no names will be attached to the answers to any question. This study should take 1 to 1.5 hours to complete, your participation in this research is voluntary. At any point, you have the right to withdraw from this study by contacting 236468@buas.nl. If you have any questions about the data or the interview you can always contact the same email address.

That being said, please confirm the following.
I ________________________________ (your full name) provide consent to the following researcher:

Amine El Bouhattaoui, a Student in Master's Game Technology at Breda University of Applied Sciences (BUas) to use my data within the above-mentioned project.

- I have been informed about the goal of this project and I have had the opportunity to ask the research team any questions which may arise about the research and my participation.
- I confirm that I am 18 years of age or over.
- I understand that personal data will be collected from me, and this information will be held confidentially such that only Amine El Bouhattaoui and the responsible supervisor will have access to this data. The information will be anonymized before any publication takes place.
- I understand that my participation in this research is voluntary, I am free to refuse to participate and free to withdraw from the research at any time. My refusal to participate or withdrawal of consent will not affect my position and my relationship with the institutions involved.
- If I have any inquiries about the research, I can contact Amine El Bouhattaoui (236468@buas.nl), the responsible researcher, or if any concerns or complaints regarding the way the research is or has been conducted, I am free to contact Dr. Mata Haggis-Burridge (haggis.m@buas.nl, ethics board member at BUas.

By signing below, I am indicating my consent to:

- Provide access to the data gathered during the interview, including any statements said or opinions shared.
- Having the collected data be used for the sake of the study.
- Being contacted if further clarification is necessary to investigate statements present in the collected data.

I understand that the data collected from my participation will be used for the purpose of this research and will be securely stored, and I consent for it to be used in that manner.

Signature

.... Date Place

Appendix D: Q-Set Definitions

Accurate live previews
The ability to have live previews directly in the app that accurately reflect the quality of the normal map on the lighting

Normal map generation quality
The upper bound on the quality of the final generated normal maps after parameter modification.

Undo/redo tool.
An Undo/Redo tool as found within most artist tools.

Animation support
Supports animation capabilities for animating sprites and images.

Automation
The tool should feature a minimal amount of manual work.

Intuitive UI
The UI is easy to understand/follow without any guides.

Layers
Layers as found in most artist tools.

Lighting simulation quality
The quality and amount of different light sources you can test with

Performance
The tool runs fast and does not take a lot of time on generation.

Ease of use
Established as the ability to quickly pick up and learn the program.

Keyboard shortcuts
The ability to use your keyboard to quickly achieve different capabilities of the tool.
Material settings
The ability to distinguish certain parts of the sprite as different material, such that the generator takes it into account when making a normal map.

Navigator Settings:
The ability to perform rotations, zooms and drag and drops such as in most artist tools.

Paintbrush for manual edits
A brush to fix errors on the sprite or the normal map, to do manual paintovers with

Simplistic UI
UI takes up little space on the screen.

Software present within art program
The capabilities of this tool are instead integrated directly into the art software used to create the artefact.

Tabs/multiple edits at once
Allows you to generate normal maps for multiple sprites at once.

(Non)-Pixel art viability
The tool is viable for multiple types of art, be it pixel-art or larger traditional 2D art.

Autosave feature
The tool automatically saves back-ups.

Color settings.
The ability to select your colors in many ways.

Copy/Paste/Delete
The ability to copy, paste and delete elements where necessary.

Customizability
Custom settings within the tool that can be saved to the user's device.

Exclusion painter
The ability to exclude elements of the sprite to be excluded from the normal map.

High quality documentation
Documentation that clearly explains the capabilities of the tool.

Parameter editing/saving.
The ability to edit and save various parameters for normal maps.

Experimentation
The ability to quickly go in and out of the program to make changes and reflect.

Artist brushes
Artist brushes to perform manual changes with as found in common artist tools.

Button controls.
The ability to utilize buttons rather than the mouse for various features.

Editing accuracy
The ability to easily make small changes to settings and parameters through buttons.

Generation of different map types
The ability to generate other maps such as bump or height maps.

Lighting presets
The ability to save certain lighting presets to reuse for other normal maps.

Live updates for assets
The assets update in your game-engine as you change them within the tool, this should happen quickly.

More complex backgrounds
The ability to include gradients and other more complex backgrounds in the preview.

Visual appeal of tool
The artistic appeal of the tool and its user interface.

Appendix E: Requirements for 2D-Normal Map Generation Tools.

In this document, software requirements for 2D-Normal map generation tools will be shared. These requirements are written based on the data on 2D-Normal map generation tools found within the thesis titled "Setting Standards for 2D-Normal Map Generators" by Amine El Bouhattaoui (2024). SMART specifications, as established by Mannion and Keepence (1995) will be used to ensure that the requirements are specific, measurable, attainable, realistic and testable in their specification. However as not all of the categories in Amine El Bouhattaoui (2024) are specific and measurable, some liberties will be taken in the natural language specification of these requirements.

This document will solely cover user requirements, as the tool's system requirements depends on a number of factors outside of this document's control. Additionally, the requirements will be numbered in order of importance as seen in the data from the research. Multiple requirements on the same rank are however not ordered on importance.

It is worth noting that in the context of this document, each of these requirements are seen as relevant and desired features. However, some of these requirements might be mutually exclusive to one another, in this case that will be indicated.

User Requirements: 2D-Normal Map Generation Tool

UR1 Normal Map Generation Quality:

- The tool must be capable of generating 2D-normal maps of similar or superior quality to the user's current work so after minimal manual editing, they satisfy the stakeholder's established desire of quality for in-game assets.

UR2 Accurate Live Previews:

- The tool must contain live previews that accurately show the changes made on the normal map during generation or editing.

UR3 Undo/Redo Tool:

- The tool must contain an Undo and a Redo tool that allow the user to undo and redo specific changes made to the normal map with the click of a button or key.

UR4 Parameter editing/Saving:

- The tool must be capable of letting the user edit the various parameters involved in the generation of the normal map, these parameters must also be capable of being saved within the tool, such that they can be re-used for other normal maps.

UR5.1 Layers:
- The tool must contain layers that allow the user to input various sprites and generate separate normal maps for them, while still visualizing them onto the same canvas.

UR5.2 Experimentation:
- The tool must contain a testing procedure that allows the user to visualize the lit sprite before making more edits to the normal map in minimal time.

UR6 Intuitive UI:
- The tool must contain a user interface that users are able to utilize without needing documentation or tutorialization.

UR7.1 Automation:
- The tool must be able to generate normal maps that satisfy **UR1** while not requiring any additional work before generation as well as minimal manual editing afterward.

UR7.2 Performance:
- The tool must be able to operate without significant slowdown on the user's system.
  - Note: The specifics of which devices to support are left to the developer.

UR7.3 Navigator Settings:
- The tool must allow the user to perform rotations, zooms and drag and drops on the canvas while utilizing the mouse.

UR7.4 Material Settings:
- The tool must contain the capability of marking parts of a 2D image as a specific material and utilize this data during generation of the normal map.

UR7.5 Documentation:
- Every capability of the tool, as well as its usage must be well-documented in a file shared alongside the tool.

UR7.6 Colour Settings:
- The tool must contain multiple ways to select colours within the tool, this includes atleast a colour wheel and spectrum, Hex code and RGB selection and a colour picker.

UR7.7 Live-Updates:
- The tool must contain the capability to immediately update the normal map in-engine after it is exported from the tool.

UR8.1 Lighting Simulation:
- The tool must contain the ability to simulate various lights in order to preview the effect of the normal map under lighting.

UR8.2 Exclusion painter:
- The tool must include an exclusion painter that allows for the user to exclude parts of the image from the normal map by process of painting an exclusion layer.

UR8.3 Button Controls:

- The tool must contain buttons in its UI rather than exclusively utilizing sliders.

UR8.4 Generation of different maps:

- The tool must be capable of generating specular, occlusion and height maps.

UR8.5 Ease of Use:

- The tool must contain basic tutorialization such that the user only requires only basic knowledge of 2D-Normal maps from the user in order to utilize its full set of features.

UR9.1 Copy/Paste/Delete:

- The tool must contain the capability for the user to copy, paste and delete elements on the canvas.

UR9.2 Artist Brushes:

- The tool must contain the capability for the user to draw onto the normal map.

UR9.3 Editing Accuracy:

- The tool's various parameters must contain buttons that allow the user to make specific single digit changes.

UR10.1 Customizability:

- The user is able to modify and save each of the various settings within the tool.

UR10.2 Artstyle Variability:

- The tool can viably be used on both pixel-art as well as 2D-artwork of higher resolution.

UR10.3 Autosaving Feature:

- The tool must create back-ups of various works in progress.

Mutually exclusive

UR11.1: Artist Tool Integration:

- The tool must be integrated within an industry standard art program
    - Note: Which program is left up to the developer.

UR15: Game Engine Integration:

- The tool must be integrated within an industry standard game engine
    - Note: Which engine is left up to the developer.

UR11.2: Visual Appeal of Tool:

- The tool must contain appealing visuals for the target audience.

UR11.3 Complex Backgrounds:

- The tool must be capable of changing the background of image previews as outlined in **UR2** and **UR 8.1.** This includes various backgrounds, gradients and colours.

UR12.1 Animation:

- The tool contains the capability to load an animation sheet and to perform each of the features on the normal map on each of the animation frames simultaneously.

UR12.2 Tabs:

- The tool allows for separate tabs, such that multiple normal maps can be generated on separate canvases.

UR13.1 Keyboard Shortcuts:

- The tool allows for keyboard shortcuts to be utilized to access various features or buttons with the use of solely the keyboard.

UR14.1 Paintbrush for manual edits.

- The tool must contain a paintbrush to draw onto the provided 2D-Art input sprite.

UR14.2 Simplistic UI

- The tool must contain a simple UI that takes up minimal space on screen. Additional menus must be hidden under dropdown menus or hotkeys.

Bibliography

Akenine-Mller, T., Haines, E., & Hoffman, N. (2018). *Real-Time Rendering, Fourth Edition* (4th ed.). A. K. Peters, Ltd.

BanZeken. (2022, April 10). *The Bump Mapping of Jurassic Park: Trespasser - Per-Pixel Lighting in 1998*. FragZone.

Ben Golus. (2020, February 3). *Generating Perfect Normal Maps for Unity (and Other Programs)*. Bgolus.Medium.Com. https://bgolus.medium.com/generating-perfect-normal-maps-for-unity-f929e673fc57

Blake Reynolds. (2023, May 13). *A Pixel Artist Renounces Pixel Art*. Dinosaur-Games.Com.

Zukowski, C. (2022, March 11). *The 3 things that actually impact how well your game sells*. IndieGameBusiness®. https://www.youtube.com/watch?v=dHzVio6L0zU&t=1549s&ab_channel=IndieGameBusiness%C2%AE

David Lightbown. (2015). *Designing the User Experience of Game Development Tools DAVID LIGHTBOWN*.

Dhondt, M. (2023). *Analysing the value of project management literature in the game industry*. https://doi.org/10.13140/RG.2.2.27246.18240

El-Nasr, M. S., Zupko, J., & Miron, K. (2005). Intelligent lighting for a better gaming experience. *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, 1140–1141. https://doi.org/10.1145/1056808.1056852

Finstad, K. (2010). The Usability Metric for User Experience. *Interacting with Computers*, *22*(5), 323–327. https://doi.org/https://doi.org/10.1016/j.intcom.2010.04.004

Gandeaga, G., Iliash, D., Careaga, C., & Aksoy, Y. (2022, July 27). DynaPix: Normal Map Pixelization for Dynamic Lighting. *Proceedings - SIGGRAPH 2022 Posters*. https://doi.org/10.1145/3532719.3543238

He, Y., Xie, H., Zhang, C., Yang, X., & Miyata, K. (2021). *Sketch-based Normal Map Generation with Geometric Sampling*. https://doi.org/10.1117/12.2590760

Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software*, *18*(4), 58–66. https://doi.org/10.1109/MS.2001.936219

Hudon, M., Lutz, S., Pagés, R., & Smolic, A. (2019, December 17). Augmenting hand-drawn art with global illumination effects through surface inflation. *Proceedings - CVMP 2019: 16th ACM SIGGRAPH European Conference on Visual Media Production*. https://doi.org/10.1145/3359998.3369400

James OHare. (2014, August 14). *Normal Maps and How Not to Do Them*. Opengl-Tutorial.Org.

Jonathan Lampel. (2017, June 15). *Normal vs. Displacement Mapping & Why Games Use Normals*. Cgcookie.Com.

Kasurinen, J., Strandén, J.-P., & Smolander, K. (2013). What do game developers expect from development and design tools? *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, 36–41.

Kevin Baker. (2022). *Do game artists still have to bake details on their 3D models to save space and reduce polygons even using the Unreal Engine 5? What is the current limit of polygons for game design?* Quora.Com. https://qr.ae/pKnzbH

LIFTING THE LID ON VIDEO GAMES ALL FORMATS THE MAKERS OF THE MESSENGER RETURN WITH AN ACTION RPG EPIC. (n.d.).

Lankoski, Petri (2018). Game Research Methods: An Overview. Carnegie Mellon University. Journal contribution. https://doi.org/10.1184/R1/6686765.v1

Mannion, M., & Keepence, B. (1995). *SMART Requirements*.

Millar, J. D., Mason, H., & Kidd, L. (2022). What is Q methodology? *Evidence Based Nursing*, *25*(3), 77. https://doi.org/10.1136/ebnurs-2022-103568

Matis Hudon, Mairéad Grogan, Rafael Pagés, & Aljoša Smolić. (2018). *Deep Normal Estimation for Automatic Shading of Hand-Drawn Characters*. https://v-sense.scss.tcd.ie/

Mike Minotti. (2020, July 28). *Cuphead launches on PS4 | VentureBeat*. Venturebeat.Com. https://venturebeat.com/games/cuphead-launches-on-ps4/

Moreira, R. D., Coutinho, F., & Chaimowicz, L. (2022). *Analysis and Compilation of Normal Map Generation Techniques for Pixel Art*. https://doi.org/10.1109/SBGAMES56371.2022.9961116

nbickford. (2022, February 17). *Algorithm behind converting a diffuse texture to a normal texture*. Forums.Developer.Nvidia.Com. https://forums.developer.nvidia.com/t/algorithm-behind-converting-a-diffuse-texture-to-a-normal-texture/203220

Nika Tsankashvili. (2018, January 20). *Comparing Edge Detection Methods*. Medium.Com. https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e

Sobel, I. (2014). *An Isotropic 3x3 Image Gradient Operator*. https://www.researchgate.net/publication/239398674

Sommerville, I. (2016). *Software engineering*.

Su, W., Du, D., Yang, X., Zhou, S., & Fu, H. (2018). Interactive Sketch-Based Normal Map Generation with Deep Neural Networks. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, *1*(1). https://doi.org/10.1145/3203186

Thayer, R. H., & Bailin, S. C. (1997). *Software requirements engineering* (M. Dorfman, Ed.; 2nd ed.). IEEE Computer Society Press, Los Alamitos [etc.], 1997.

Tom Marks. (2020, April 2). *Inside EXOK games: The Brand New Studio that's already sold a million copies*. IGN. https://www.ign.com/articles/inside-exok-games-the-new-old-studio-from-the-team-behind-celeste

Unity Technologies. (2018). *Normal map (Bump mapping)*. Docs.Unity3d.Com. https://docs.unity3d.com/2018.1/Documentation/Manual/StandardShaderMaterialParameterNormalMap.html#:~:text=A%20normal%20map%20is%20an,(or%20smooth%20interpolated)%20polygon.