Tampere University of Applied Sciences

# Fur Simulation

## Creating and Animating a Character with Fur in Blender

Anu Hautakangas

BACHELOR'S THESIS
May 2024

Degree Programme in Media and Arts
Interactive Media

**ABSTRACT**

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Media and Arts
Interactive Media

HAUTAKANGAS, ANU:
Fur Simulation
Creating and Animating a Character with Fur in Blender

Bachelor's thesis 62 pages, appendices 1 page
May 2024

---

Hair and fur simulation has been a rapidly evolving technology in 3D computer animation in the past few decades. As 3D animation has become accessible with more powerful hardware and free software, it is worthwhile to assess if the tools hold up to industry standards. The purpose of this thesis was to research if Blender is a viable software in creating fur simulations from an artist's perspective. What tools there are in Blender, and what kind of artist-friendly workflows can be used when simulating fur were the focus of the thesis.

The theoretical section examined the history, evolution and current methods of hair and fur simulations, and what makes simulations so difficult to realise. To investigate the viability of Blender as a software for hair and fur simulations, a workflow evaluation in the form of a practical project was made.

The results suggest that Blender can be a viable software for hair and fur simulations, but it is heavily reliant on available hardware. Additionally, Blender Hair Nodes tool was still under development at the time of writing this thesis and was essentially missing the possibility to simulate hair. This led to using a workflow, which was a combination of two different systems.

---

Key words: fur simulation, hair simulation, blender

**CONTENTS**

**GLOSSARY**

| | |
|---|---|
| Cortex | inner structure of hair fibre, keratin filled cells surrounding the medulla |
| Cuticle | inner structure of hair fibre, scale-like formation over the cortex |
| GPU | Graphics processing unit |
| Guard hairs | Long and coarse hairs in animal fur |
| LOD | Level of Detail |
| Medulla | inner structure of hair fibre, inner core of hair |
| NST | Neural Style Transfer |
| OS | Operating system |
| Steam | Digital distribution platform for games and software |
| Texel | 3-dimensional texture map |
| Undercoat | Dense and short hair in animal fur |
| VFX | Visual effects |
| Vibrissae | Whiskers, hair type in animal fur |

# 1 INTRODUCTION

Hair and fur simulation, which is part of a larger subject matter of computer graphics, has been around for decades in film, animation, and games. Simulation is a complex process of using mathematical functions to produce movement or other phenomena based on real-world physics. Its main purpose is first and foremost to automate time consuming work and achieve realism. As hardware capabilities and its accessibility has increased over the years, so too has the attainable realism and efficiency of simulations. With this exponential growth, it is no wonder that recently, the use of Artificial Intelligence is also becoming more prominent in the media industry, as arguably an ultimate automation of the creative process. The impact of which is yet to be seen in its entirety.

Does then a software like Blender, from an artist’s perspective, seem like a worthwhile investment? For hobbyists, students and others with limited budgets, Blender has been an approachable choice when it comes to creating 3D models, animations, and visual effects. In addition, this software has become increasingly popular with industry specialists. Thus, it is beneficial to assess software and its functionality if it is a viable option to use when creating specifically fur simulations, which is the main topic of this thesis.

The concepts and processes behind hair and fur simulations were examined to have a better understanding why this subject is so challenging. The main objective of this thesis is to find an artist-friendly workflow for creating fur simulations in Blender by using the built-in systems of Hair Nodes and Hair Particle Physics. This was achieved as a practical project by creating a 3D model from start to finish, explaining the process and choosing a suitable workflow for simulating fur. This chosen workflow was then evaluated based on how approachable it is for an artist, and whether it is an efficient method when creating fur simulations.

To limit the contents of this thesis, using Blender is the main focus, with some additional software and addons for Blender being mentioned during the practical project creation, but their functions are not explained more than is necessary.

Testing or other practical use of other simulation software is also not explored in detail.

# 2 SIMULATION IN ANIMATION

## 2.1 What is simulation?

Simulations are used in various media, such as films, animation, or video games. The purpose of simulations in the animation pipeline is to automate a process that would be too time-consuming to animate by hand alone. For hair simulations specifically, the process of simulation automates movement that is described in some of the key animation principles, such as follow-through, overlapping and secondary actions. This adds to realism to the movement of the simulated object, as the movement itself is based on real-world physics, commonly known as Newtonian physics. (Roda 2022, ch 6.)

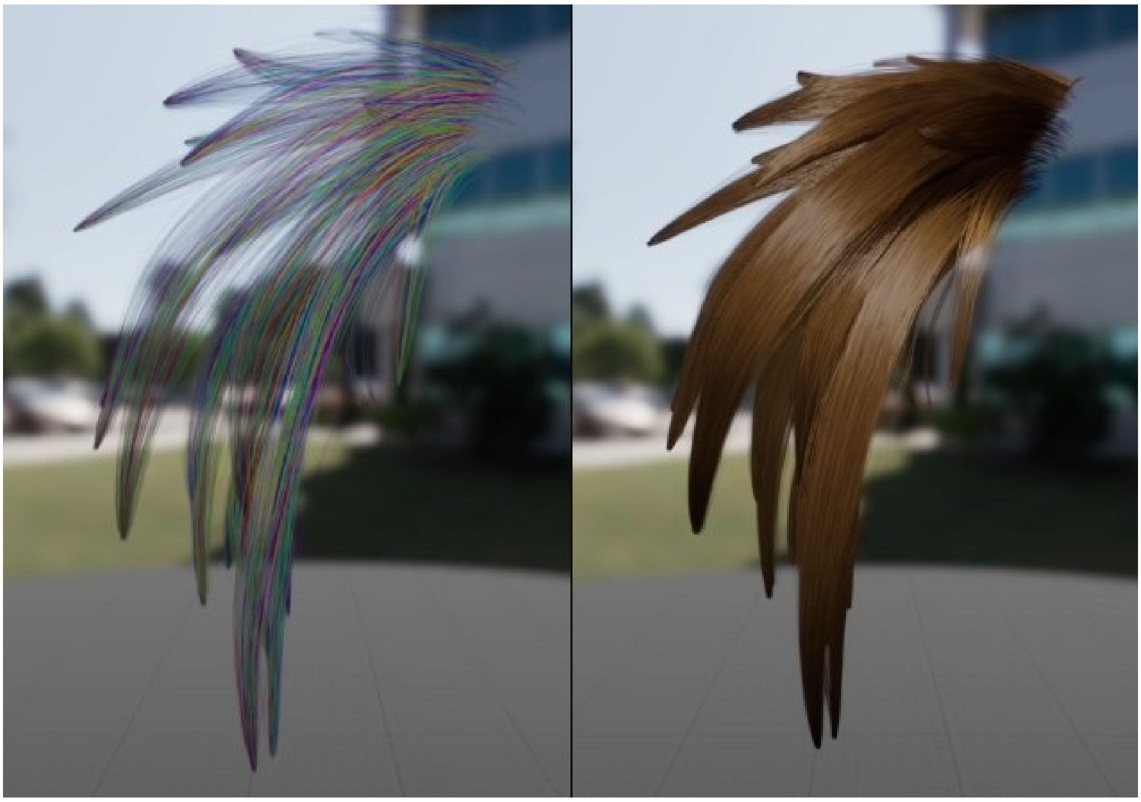
Based on these Newtonian physics, simulations can be categorised into two types, soft body, and rigid body simulations. Objects that have rigid body physics are non-deforming, hence the name “rigid”. Rigid body simulations are usually computationally on the cheaper side, and easier to implement. Non-organic objects are commonly treated as rigid bodies, such as vehicles or buildings, or rocks and other environmental objects. Soft bodies on the other hand can deform, some examples of which would be hair or cloth. (Roda 2022, ch 6.) The way simulations are handled, depends on what the end usage of them will be.

Simulations can be handled either real-time or offline. Real-time simulations are common for video games, and offline simulations can be used for films or other such media that do not need to be interacted with in real-time. Real-time simulations often focus more on performance, rather than realism. Offline simulations on the other hand can have added realism and scale, but with a cost to computation. (Roda 2022, ch 6.) For complex simulations, such as hair, offline simulations are generally preferred. Though due to increased hardware capabilities, real-time simulation of hair is becoming more achievable.

Real-time simulation of hair has the massive upside for animators to see how the hair acts and looks when character animation is ready, and hair is generated on the model. Thus, there is the possibility to fix or edit things immediately and not

wait for the technical artists to work on the simulation and lighting first, but rather that the animators and technical artists can work in parallel. (Unreal Engine 2020.) If the objective is to use hair simulations in an interactive setting, such as a game, it needs to be adjusted to increase efficiency and smooth interactivity.

In order for real-time hair and fur simulations to work in a game setting, some performance variables are usually used. For example, Unreal Engine uses a method where only guide hairs have simulated physics which are depicted as various coloured strands, and the interpolated hairs that follow these guides are then rendered as actual hair strands as seen in Picture 1. Another system that enables the real time simulation of hair to work more smoothly is the level of detail, or LOD system. In this system strand density can be reduced, or the engine can change the rendering of hair to use hair cards instead of strands. Hair cards are a common method of rendering hair in video games. (Unreal Engine 2021.)

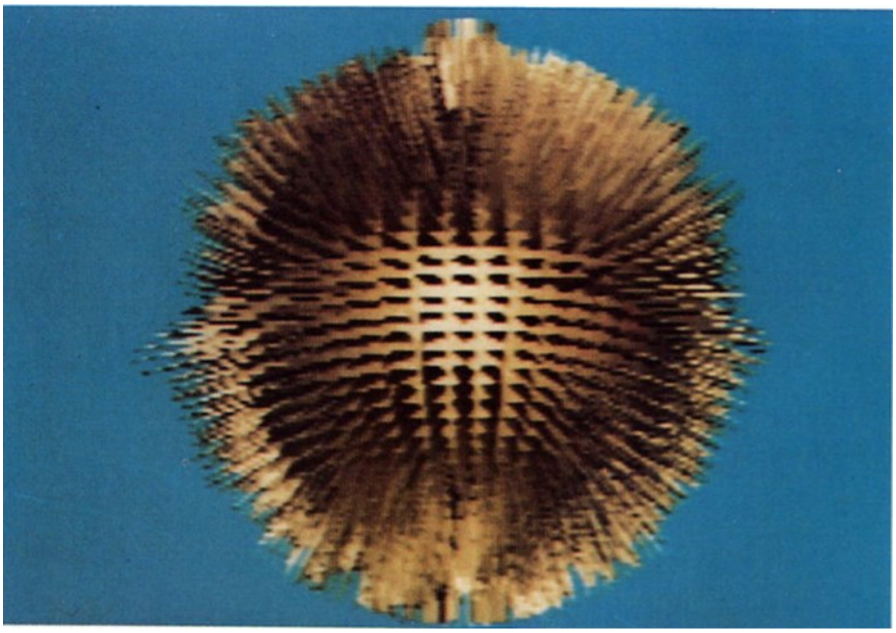


PICTURE 1. Guide hairs on left-side image and interpolated and rendered hair on the right-side (Unreal Engine 2021).

## 2.2 History of hair and fur simulations

The history of hair and fur simulation spans across multiple decades, starting from the 70’s with the advent of 3D computer graphics. Hair and fur simulation has had numerous progressions and iterations with contributions from artists and scientists alike, focusing on different aspects of hair and fur simulation, like the rendering or certain characteristics in dynamics.

The early stages for hair simulation started with just the visual representation of fur, without the dynamics. An early milestone in this area was done by Csuri et al. in 1979, in which they managed to render a primitive ball with 46,000 triangles representing fur. Aliasing, sharp, and jagged edges in geometry was an apparent issue when rendering 3D objects with complex geometry as seen in Picture 2. Despite being a simple visual representation, it was a forefront in the history of hair simulation. As hardware potential increased, so did the possibilities in the field of computer graphics.



PICTURE 2. A primitive ball with triangles. (Csuri et al. 1979).

Over the years computer scientists have made notable progress to rendering 3D fur by inventing a new texture map called texel. These texels are 3-dimensional texture maps that simulate volume by calculating density, orientation, and lighting of particles. (Kajiya & Kay 1989.) This new texel rendering method fixed the

apparent problem with aliasing, but hair and fur simulation were still missing a key component, the process of simulating dynamics.

It was not much later that one of the first hair simulations with dynamics was done by Rosenblum, Carlson, and Tripp (1991), in which they managed to simulate individual hair strands for long hair. Following this, Anjyo, Usami, and Kurihara introduced a method for modelling hair strands as cantilever beams (1992). These earliest methods for simulating dynamics did not consider one of the most challenging aspects of hair simulation, namely how hair strands interact with each other (Magnenat-Thalmann & Thalmann 2005). Even with limitations in the technology, it had progressed so far as to be employed in e.g., filmmaking.

The first live-action film to use VFX for a computer-generated character with rendered fur was The Flintstones (1994), in which a sabretooth tiger had a full coat of realistic fur (Netzley 2001, 86, 106). The scenes with the sabretooth were short, with fast movement and dim lighting (Picture 3). Creating computer generated fur was challenging and still under development at the time. Industrial Light and Magic, the company responsible for the model, had to develop specific software to generate the hair on the model, adjust its length and growth direction as well as the shading (Magid 1994, 58).

PICTURE 3. Sabretooth tiger from the film The Flintstones (Amblin Entertainment 1994).

Another early example of simulated fur in film can be found in the fully computer-generated animation, Monsters Inc (2001) by Pixar. In the film the character Sulley had more than 2,3 million hairs that needed to be rendered and had to have dynamic movement. The technical artists at Pixar needed to implement a believable method for rendering fur, especially how the hairs cast shadows on other hairs, also called self-shadowing as depicted in Picture 4. An important aspect for creating this character, was to have enough controllability for artists in the way the fur behaves in motion. The simulation didn’t always have to follow strict laws of physics but was more controllable for the artists in order to create the most impact visually and narratively. (Price 2008, 199–200.)

PICTURE 4. Sulley from Monsters Inc (Walt Disney Pictures/Pixar Animation studios 2001).

In order to tackle the challenges posed by simulating physics for hair and cloth, Pixar created a dedicated software called Fizt for the movie Monsters Inc (Price 2008, 199–200). Additional software or tools, specifically designed to solve a certain issue or technical problem in animation and filmmaking, is quite common in the industry. In another example, engineers at Disney Pixar animation studios developed a tool for simulating hair called Taz, which was especially designed to resolve how to simulate curly hair (Giardna 2021). This tool was used to create the curly red hair of Merida in the film Brave (2012). A core feature in this software was the artistic controllability. Movements in animation can be extreme, and if the

simulation were to follow them accurately, the result would be visually unappealing if a certain stylistic choice was to be maintained. (Iben et al. 2013.)

A more recent example of what can be achieved with realistic hair and fur simulation in the present-day technology can be found in the film Godzilla vs. Kong (2021), directed by Adam Wingard. In this film the computer-generated gigantic monster King Kong had fully simulated fur with more than 6,3 million hairs. An important factor was to have the fur interact with elements, like water, fire, and dirt in a convincing manner (Picture 5). This meant that the monster had to have multiple versions of groomed fur, making the process exceptionally heavy computationally. In such a massive project, iteration of the process can be an issue. To solve this, the team behind the creation of King Kong were able to utilise more modern software solutions that allowed them to split the groom into multiple smaller parts to work on them more efficiently and to preview the shading in the viewport during the fur grooming process and avoid time consuming rendering tests. (Moltenbrey 2021.)

PICTURE 5. The fur of King Kong reacting to different elements, like dust, water, debris, and snow (Legendary Pictures 2021).

The evolution of hardware and software has allowed for increasingly convincing and complicated hair and fur simulations in films and animations. Oftentimes new technology or tools had to be developed to solve certain problems or allow for more artistic control, efficiency, and interactivity during the hair grooming process. Efficiency in a production is highly valuable, especially in big budget films or

animations. The ability to lessen the workload by i.e., automating certain steps can promote this efficiency.

### 2.3 Future of hair and fur simulations

The increasingly more powerful hardware, especially in GPUs makes real-time simulation easier to implement (Yuksel & Tariq 2010, 20–21). The advantage of real-time simulation is the interactivity that is essential for game and virtual reality uses. Even in the film industry, animators and technical artists can work more in sync with each other, removing unnecessarily time-consuming steps in the production process with real-time simulation (Unreal Engine 2020).

A rapidly evolving technology currently is artificial intelligence, and it has been used in the animation industry already. An example of this is the animated film Elemental (2023), where a simulated fire was improved with NST, Neural Style Transfer, to look more stylized and have an illustrated look. This method made it possible to have a simulation deal with the dynamic movement of the fire, and the NST could enhance the overall style and look of the simulated fire. (Hoffman et al. 2023.)

At the time of writing this thesis, OpenAI introduced their new AI model called Sora, in which written prompts can be turned into video sequences (OpenAI, 2024). This new model has extreme implications for the future of animation, and VFX including hair and fur simulations. This technology has the potential to almost eradicate the need completely for technical artistry, as only a written prompt is required to create visually stunning video and animation. Even though the technology is still in its infancy, and there are issues, like keeping consistency as seen from the screenshot of OpenAI’s video of frolicking wolf pups in which another wolf pup starts to materialise from another pup (Picture 6). The exponential growth of technology means even this can change in a matter of years.

PICTURE 6. Screenshot of OpenAI’s video sequence of wolf pups (OpenAI 2024).

# 3 HAIR AND FUR SIMULATION

## 3.1 Physical properties of hair and fur

Human hair is commonly divided into three main types based on the properties of the hair. These types are: Asian, African, and Caucasian. All three have different properties and behave differently in simulations. Asian hair type is straight or slightly wavy, and African is more irregular and curlier, and Caucasian can be anything in-between these two. The internal structure is also slightly different for Asian and African hair types, as Asian hair has a very circular cross-section, but African has a more elliptical one. (Robbins 2012, 178–179.)

When examining hair strands on a microscopic level, certain characteristics are clear. Generally, the structure of a strand of hair has a three-part formation in a shape of a cylinder: an inner core called medulla, and around it is the cortex that has cells filled with keratin and it is the most prominent structure of human hair, and lastly a thin coating on top of the cortex that has a scale like structure which is called a cuticle. (Robbins 2012, 1–4.) While the inner structure is generally the same in human hair strands, the differences between human hair and animal fur strands become more evident.

The basic three-part structure is similar, but the size of the medulla is more prominent in fur than hair strands and even the properties within the medulla can vary for different types of animals (Chiang et al. 2016, 276, 280; Yan et al. 2015, 1). Medulla can also be absent altogether for certain fur types, as is often the case for wool fibres (Robbins 2012, 4; Thomas et al. 2012, 2436). The cuticle scale formation also has different variations based on animal species, as depicted in Picture 7. Not only is the inner structure of a strand of hair different for animals and humans, but so is the variation between hair types in the fur of an animal.

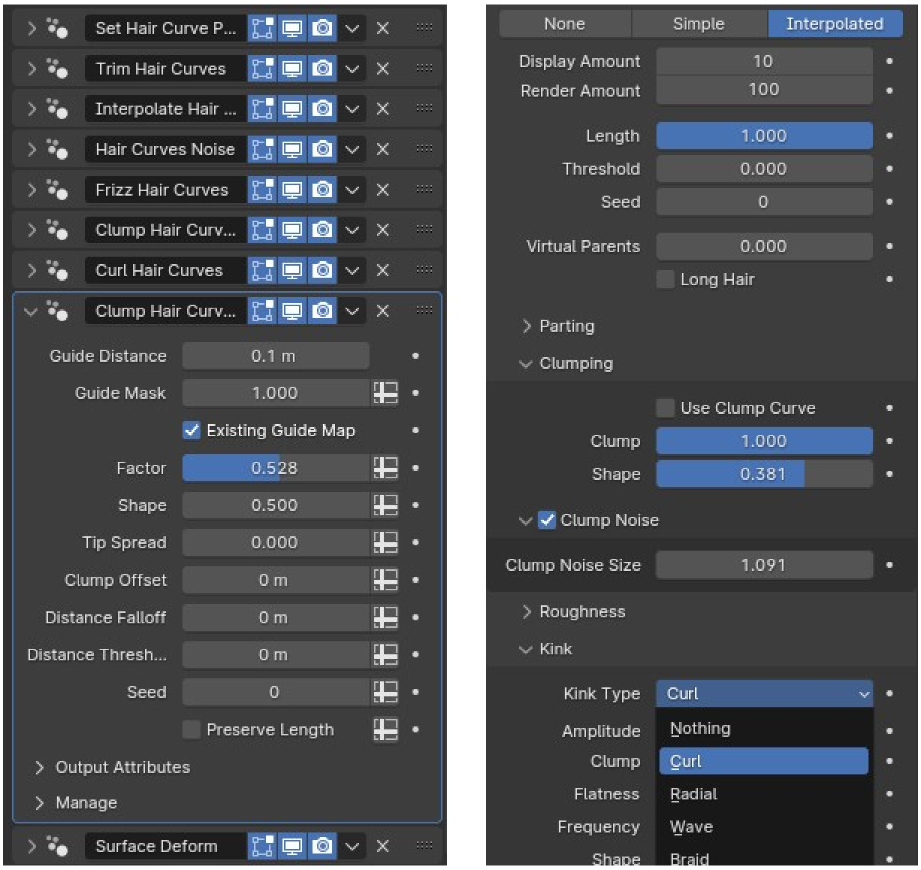
PICTURE 7. Image collage of cuticle scale formations based on electron microscope images, from left to right: cat, human, polar bear (Hautakangas 2023).

Fur can have more than one layer of different hair fibres, most commonly an undercoat and guard hairs. The undercoat has the most coverage and appears more voluminous, its strand structure is shorter, and thinner. Combined with the undercoat are the guard hairs, which instead are thicker, longer, and coarser. 3D representation of fur can be challenging, due to this layering characteristic of different types of fur fibres. (Andersen et al. 2016, 739.) This hair strand variation is undoubtedly apparent in animal fur, when strictly compared to the scalp hair of a human.

The fact that humans and animals have such an extensive variety in the type of hair fibres alone, makes the process of simulating realistic hair and fur a challenge. Every part of the fibres’ structure affects the visual and dynamic behaviour of hair and fur. Careful and meticulous research into hair properties is needed if the goal is to represent 3D hair and fur simulations. It is apparent that a co-operation of different sciences, from medical science to physics, and artistry is needed to mimic the physical properties of hair and fur in 3D representation.

## 3.2 3D hair geometry

Hair strands are naturally cylinder shaped, but representing 3D hair strands as cylinders can be unnecessary, as they are extremely small in diameter. As such a single hair strand is usually represented as a curve that has some added thickness. Hair strands can be procedurally manipulated in order to add variation to the shape, like waviness or curliness as seen in Picture 8. Randomness in hair shape is recommended, as without it the hair can look unnatural. (Yuksel & Tariq 2010.) Adjusting the general shape of hair is only part of the process, as the hair needs to be generated on a specified surface first.



PICTURE 8. Two different procedural hair settings systems in Blender: Left side is for Hair Nodes and right-side is for Hair Particles (Hautakangas 2024).

Placing each hair strand manually to a surface, such as the scalp of the head, can be too time consuming, due to the fact that humans have thousands of hair strands on their scalp alone. A Caucasian female can have between 81 000 and 121 000 hair strands in their scalp, and the amount of hair varies between humans (Yuksel & Tariq 2010; Robbins 2012, 18). For animals, the amount of

hair strands is considerably greater. For this reason, methods have been implemented that automate this hair modelling process. For example, it is common to generate a fewer number of “key-hairs” on the surface of the model and use interpolation methods that generate more hair strands near these key-hairs. There are other methods for generating hair, specifically when trying to attain certain hairstyles or accelerating the modelling process.

A common method for easier hair modelling is the wisp technique, in which hair strands are populated, and manipulated in groups in wisp form (Choe & Ko 2005, 160). Another method is the generalised cylinder, in which hair strands are confined inside a cylinder shape for easier control (Yang et al. 2000, 85; Patrick, Bangay & Lobb 2004, 115). Both of these methods are good for modelling specific hairstyles, like braids or ponytails (Ward et al. 2007, 216).

## 3.3 Dynamics

There are numerous techniques for different types of hair in hair simulation, and it is a progressive science to find different simulation models for different types of hair. Certain properties of hair can be problematic for simulation, such as collision, self-collision, and friction. Simulating hair dynamics is usually a process of balancing between realism and efficient computation.

Simulation of hundreds of thousands of hair strands individually is computationally expensive, thus it might be preferred to simulate only a fraction of the number of hairs as guide hairs, that the other hairs follow in interpolation (Chai, Zheng & Zhou 2014, 1). This is especially important when trying to simulate hair in real-time applications. Not only is the simulation of individual strands of hair a complicated process, but it also becomes even more complicated when the hairs interact with each other. Strands collide with each other and the scaled structure of the cuticle in a hair strand causes friction and static attraction (Kim & Neumann 2002, 620; Ward et al. 2007, 220). All of this must be calculated in a simulation.

There are other factors that make simulation complex. For example, preserving the volume after the hair shape is groomed and simulation is started, has been a problem in hair simulation. When the simulation begins, gravity brings the hair down, thus having the hairstyle lose its initial groomed shape. (Hsu et al. 2023, 73.) The effect of gravity affecting the hairstyle can be manipulated by changing the stiffness of the hair, but this can result in simulated motions that are unrealistic (Bertails 2008, 32). Apart from gravity, other external forces can also influence the dynamics of hair, which can be wind, or water as well as collisions from other objects and the model itself where the hair is placed.

### 3.4 Rendering

The process of rendering hair can be roughly divided into two sections, local and global properties. Local properties determine the individual illumination of each strand of hair, how light scatters, reflects, and refracts for each hair. Global properties define how shadows casted by the hair strands are rendered. Self-shadowing also creates the perception of volume in hair, and it defines the shape of the hairstyle as seen in Picture 9. (Ward et al. 2007, 226–229.) Both properties are affected by the overall structure of hair strands.

PICTURE 9. Two renderings showing the difference of how shadows add to the style and overall shape of the hair (Hautakangas 2023).

In human hair, the cuticle scale formation affects the way light is reflected from the surface of the hair strand creating a primary reflection, as well as how it transmits through the hair, as hair strands are slightly transparent. Light also refracts within the hair strand, creating a secondary highlight. (Marschner et al. 2003.) Rendering animal fur is slightly different compared to human hair. As stated before in discussing hair and fur properties, animal fur fibres can have a larger medulla, and this affects how light is reflected and transmitted within a fur fibre (Yan et al. 2015). Fur is also denser than human hair, which affects the way light is reflected and absorbed (Chiang et al. 2016). In physically based rendering models, the colour of the rendered hair is not based only on a diffuse colour but is the end result of the combination of a diffuse colour and how light scatters through the hair (Vogt et al. 2017).

# 4 FUR SIMULATION IN BLENDER

In this part of the thesis the author will explain thoroughly the process of creating a 3D model using Blender that is suitable for simulating fur. To start with, some history of Blender and why the software is so approachable is examined. The second section focuses on the step-by-step workflow of creating a base 3D model, examining some of the theory, and explaining the best practices to follow. The third section focuses on choosing a suitable workflow for simulating fur. This workflow is then evaluated based on how artist-friendly and intuitive it is, and how efficient it is to use and whether Blender is actually a viable software for simulating fur.

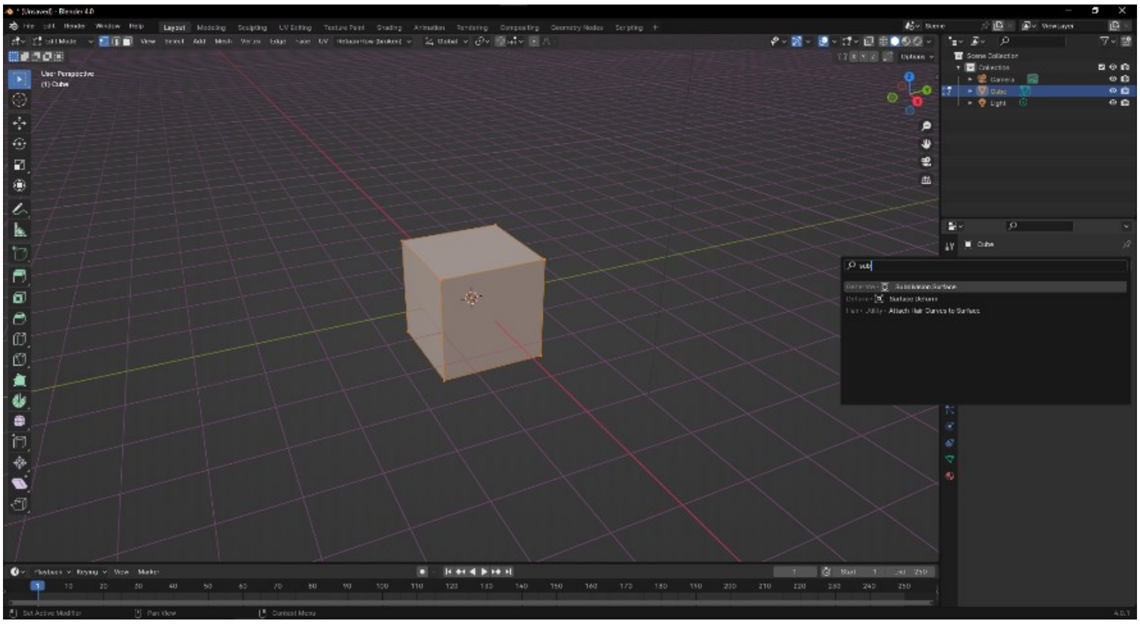
The author uses a dog as the 3D model in this project, as it is a familiar subject to create and it made the preliminary 3D modelling and subsequent processes easier and faster. The simulation process itself was done as offline simulation, which means that after the fur was generated and simulated, it was baked before the final rendered result.

## 4.1 Blender as a CG software

Blender can be seen as an easily accessible software as it is free, constantly improving, and has a large community of users. Ton Roosendaal is the original creator of Blender, and his software's first official 1.0 version was released in 1995. It was later in 2002 that Blender became officially open source under the GNU General Public License. Due to its open-source status and availability, users can contribute to the implementation of tools and functions, thus helping to improve the software continuously. (Blender History n.d.)

Blender is first and foremost a 3D software, capable of providing modelling, rendering, sculpting, rigging, animating, and simulation tools for the user (Blender Features n.d.). There are also various other functions available, such as 2D animation, VFX and video editing, but these are not an important aspect within this thesis. A Blender version 4.0 and 4.1 have been used during the making of

this thesis. The user interface can be seen in Picture 10, in which a notable change from previous versions is the added search function to the modifiers tab. Searching for specific actions is needed in software as extensive as Blender.



PICTURE 10. A screenshot of Blender's UI (Hautakangas 2023).

The fact that Blender is free and versatile with its functions and has an extensive user base that can provide additional content in the form of addons, or tutorials makes it unsurprising how popular the software has become not just for hobbyists but for industry specialists as well. Based on Blender's own published statistics of downloads per operating system from the year 2019 and 2020, there is a clear increase in downloads in i.e., Windows OS from 7,6 million in 2019 to 11,77 million in 2020 (Figure 1). A more recent statistic on the increase of Blender users on the Steam platform shows that the concurrent peak number of users is steadily increasing from 2031 users in January 2020, to 10,563 users in April 2024 (Figure 2). With this steady increase in users on different operating systems and platforms, it is clear that there is an increased interest in the software. Thus, it is worthwhile to assess specific parts of the software and its functions, to see what improvements could be made. In the case of the practical part of this thesis, an emphasis on artist-friendliness was kept as a key point.

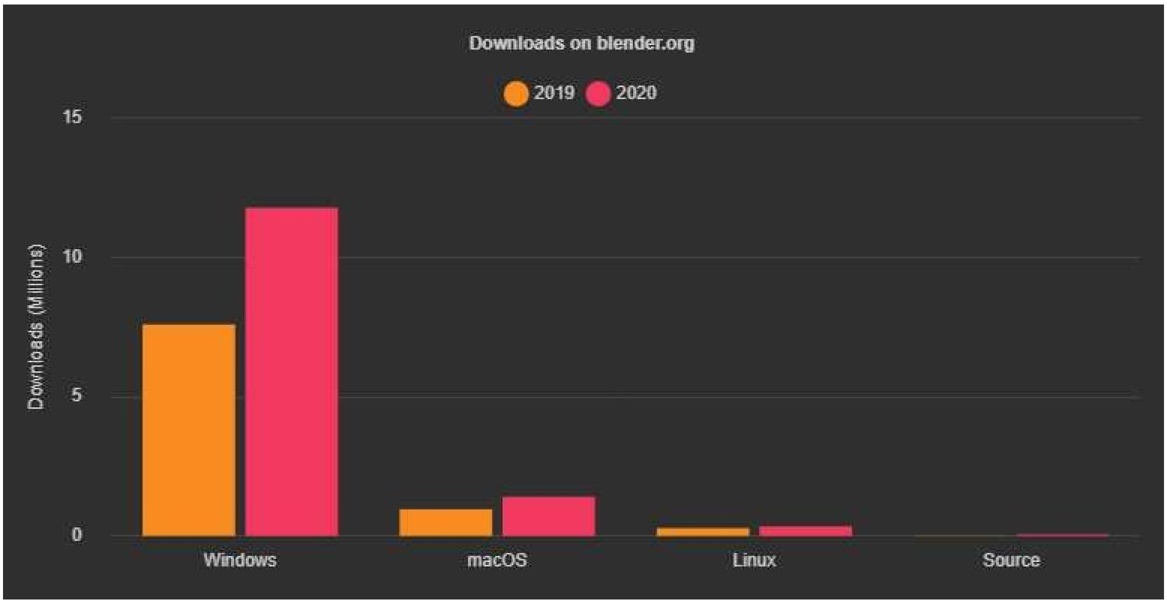



FIGURE 1. Number of downloads of Blender in different operating systems in 2019 and 2020 (Siddi 2021).


FIGURE 2. Users on the steam platform (Steam Charts n.d.).

### 4.2 Creating a base 3D model

The creation process of a character 3D model is extensive and requires general understanding of the software that is used and its techniques, knowledge in anatomy, including muscle and bone structure, colour theory, animation and its principles and various other skills. In this section this whole procedure is explained in a way that could offer insight for those interested in what things are generally advisable when creating a character model that has simulated fur.

It is important to note however, that the workflows explained in this thesis are not the only correct ones to use or might not even be the most efficient in terms of

time usage. This is simply one method of doing things. The goal is also not to create a hyper-realistic model, but something more stylized to simplify some of the work that is needed in such a project. Blender 4.0 and 4.1 versions were used, with some addons included that are described in their respective sections. It is recommended to have a basic understanding of the software and 3D modelling to follow along the process.

### 4.2.1 Reference

Most creative processes require a general understanding of the subject that is created, and it is advisable to collect and combine a reference image board that shows the subject in various perspectives, and this is especially important when creating a 3D model. In this project, a Bernese Mountain Dog was used as the subject for the model, and a reference image board was created by combining images from various Internet sources, such as Pinterest. This image board was created in a software called PureRef, created by a company called Idyllic Pixel (Idyllic Pixel n.d.). The software is lightweight and simple for handling multiple reference images.

Using these references as guides, a character sheet was then illustrated in Adobe Photoshop (Adobe 2024) for the purpose of guidance for 3D modelling. A minimum of side and front view for the character sheet is advisable, more depending on how accurate the model has to be based on the character sheet. In this case only front and side view was used, as the character sheet was to offer only a general form to follow along when modelling. As pictured in the character sheet (Picture 11) a red outline was also illustrated to show the shape of the dog without fur, as the fur itself is not modelled, but simulated later.

PICTURE 11. Character sheet with red outlines showcasing how much volume fur adds to the subject (Hautakangas 2024).

These reference images were then set-up in Blender in order to start the modelling process. The images can be added as reference images from the Add menu in Blender, there are options to adjust the opacity and viewport visibility. Side and front view are different images. The most important thing is to align the images so that the side and front views are in the correct 3D space.

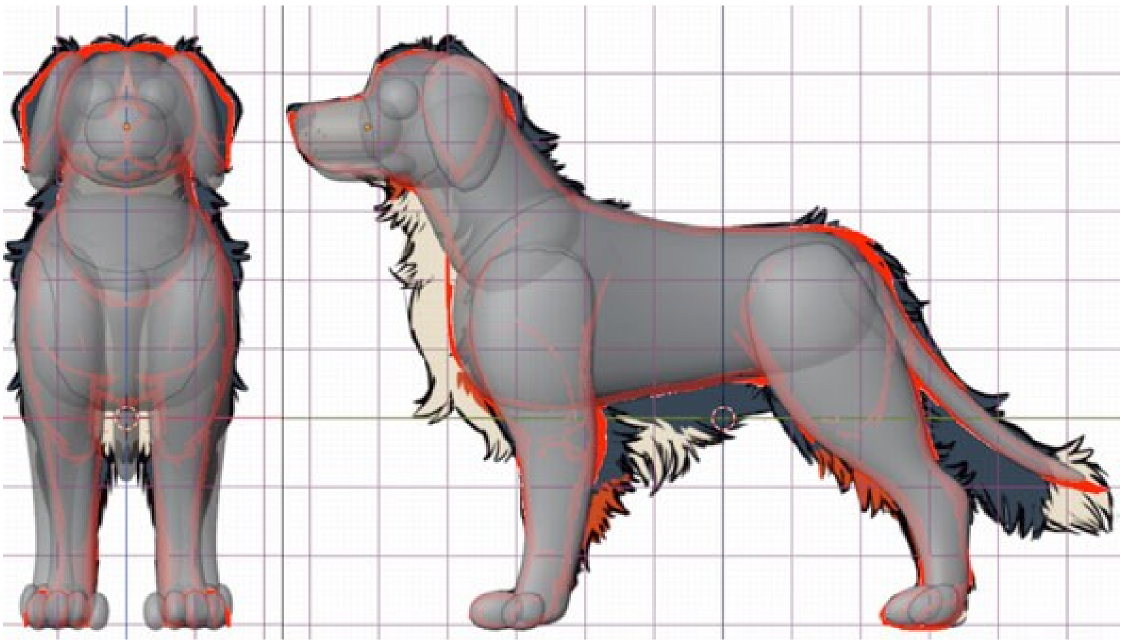
### 4.2.2 Modelling

The modelling workflow used in this project was to first create a sculpted higher poly mesh and recreate new topology, also called retopology to a lower poly mesh based on the high poly version. This method is generally used in order to have finer sculpted details, like wrinkles visible as baked texture maps instead of actual topology in lower polycount models (Beane 2012, 158). Lower polycount models also make it possible to be animated, as higher polycount models have too dense topology and thus are too computing-heavy to animate (Beane 2012, 158).

In the case of this project, this workflow was deemed to be convenient for several reasons. Even though texture baking sculpted details is not as important in this case, as most topology is hidden under the simulated fur, some finer details especially in the snout and eye areas can still be texture baked as they are more visible. Retopology of the lower polycount model gives more control over the overall topology and keeping it clean, which is especially important for animated

characters. Sculpting can also be a more intuitive workflow for an artist, as sculpting tools offer more control over mesh manipulation.

The process started with blocking out the general shape of the subject, in which the body was divided into multiple different objects, i.e., body, neck, head and legs (Picture 12). These parts were Cubes with Subdivision surface modifiers, of these objects the legs also had Mirror modifiers. Keeping the parts separated at the beginning gave more control when sculpting the mesh. The parts were later joined together as one object and using Remesh operator in Sculpting mode the parts were fused together, and the seams were smoothed out in order to get rid of any artefacts. Some parts, like the eyes, were kept as completely separate objects, in order to rig them later.



PICTURE 12. Blocking out phase, where the model consists of multiple objects (Hautakangas 2024).

When sculpting, it is generally advisable to work from bigger shapes first and details later (Spencer 2011, ch 1). The overall form of the sculpted object is important to get right at the beginning. References were used for clarifying the flow of the muscles and bone structure. Even though the subject was not aimed to be realistic, it is still preferable to follow some realism.

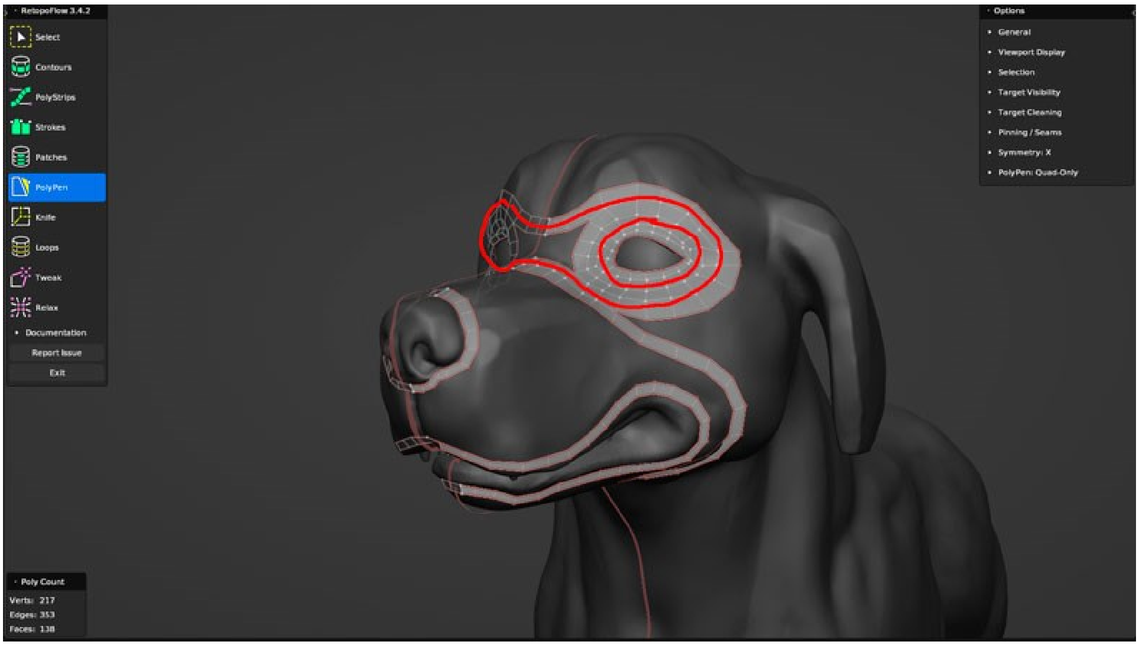
Blender has plenty of tools for sculpting, but in this case only a few of them were needed, such as the Clay Strips, Clay, Draw Sharp, Grab and Smooth. These tools offer enough control to add or move volume, smooth surfaces or add creases or crevices in such places like the snout or near the eyes. The final sculpted mesh ended up having a lot of muscle structure, which may have not been that necessary, and this increased the complexity of the mesh (Picture 13). Such a model can however be reused in other projects, so it could be considered useful.

PICTURE 13. The final sculpted mesh (Hautakangas 2024).

### 4.2.3 Retopology

Retopology was done with an add-on for Blender called RetopoFlow, which is the work of Orange Turbine (Orange Turbine n.d.). This addon essentially makes the retopology process much easier, as it adds more tools for the user to quickly create topology that follows the form of the higher poly mesh by snapping each vertex to the high polygon mesh surface.

For a clean and animatable mesh, the topology creation process needs to follow some rules. There should be a good edge flow, that is based on how the mesh moves when animated, and this can be achieved by adding loops in certain parts of the mesh (Osipa 2010, 76–77). Following these rules, a few loops were added as the starting points for the retopology process (Picture 14). It is normally good to work in as simple a resolution as possible at the start and create more topology when needed. In the case of this project, some simpler resolution could have been used at the start of retopology, which could have made the process faster.

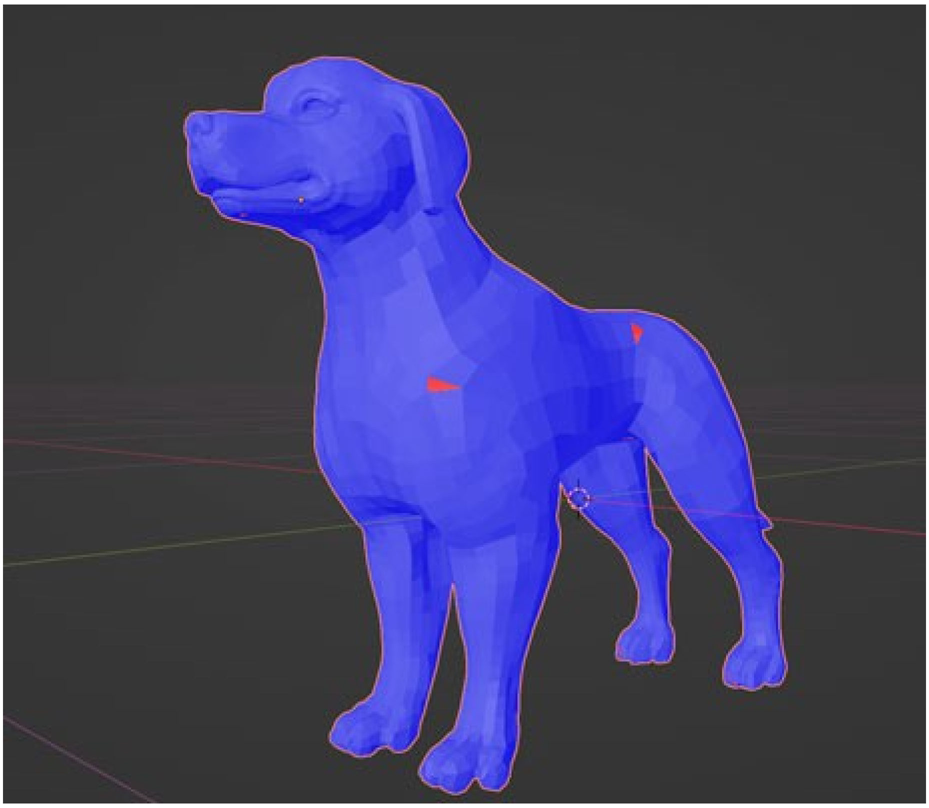



PICTURE 14. A starting point for retopology, highlighting some of the places where loops generally tend to go in the face of the character (Hautakangas 2024).

While retopology can be a tedious process, it can give a lot of insight into thinking of 3D meshes as animatable objects, if the end usage of the 3D mesh is to be implemented in animation. Placing loops in correct places forces the modeller to think of how the mesh will be animated in later stages, which can help in avoiding mistakes in future parts of the process. It is a good habit to evaluate one's process constantly and find issues and anticipate possible problems.

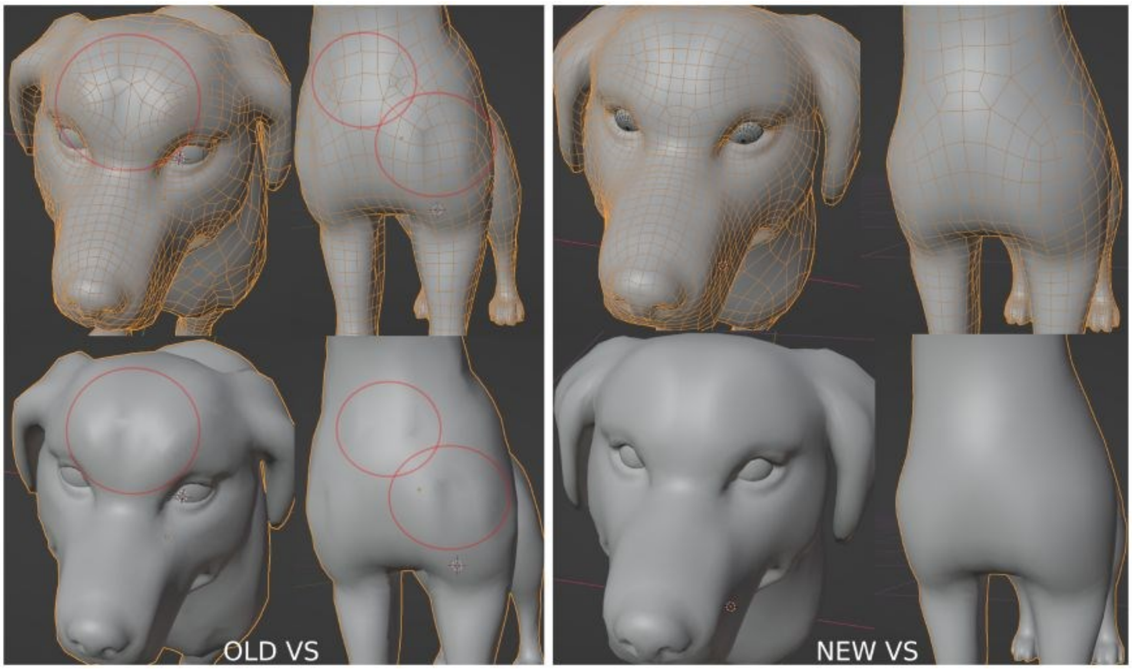
When the initial retopology was finished, a double check was done to see whether the polygons on the mesh had their normals, as in the direction of the polygon, in correct orientation as seen in Picture 15. At this point it was also clear that some fixing had to be made in the mesh. The mesh had very visible triangles (Picture

15), that is not recommended to have in an animatable mesh. For game characters and static meshes, some triangles are not a problem as the mesh is triangulated in the game engine, but for animation it can be an issue. Animated meshes are generally subdivided to appear smoother, and the subdivision does not work for triangles. (J Hill 2021.)



PICTURE 15. Triangles in the mesh. Blue colour indicates the direction of the normal is correct, and red means reversed (Hautakangas 2024).

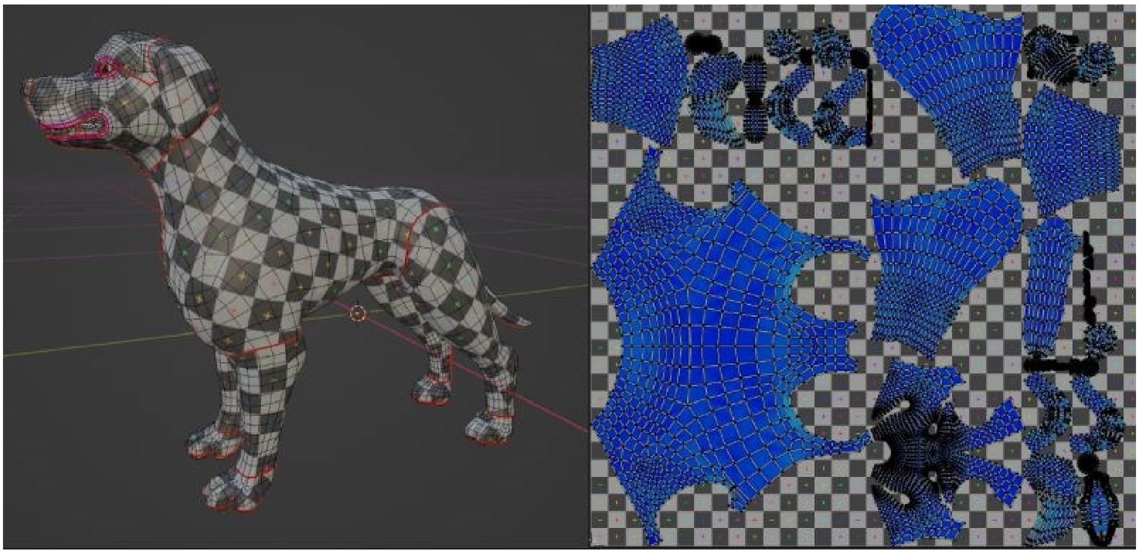
Inconsistent topology with triangles and vertices connected by too many edges can create issues in the shading and undesirable effects when animated. In order to get rid of shading artefacts and avoid possible future issues in the animation phase, some re-arrangement of polygons was made in the mesh (Picture 16). Even more refining of the topology would’ve been possible, but due to time constraints it was necessary to continue to the next phase of the project.




PICTURE 16. Left-side images highlighting problematic topology, where badly placed edge-flow causes artefacts in the shading, and the right-side images where these have been corrected (Hautakangas 2024).

### 4.2.4 UV mapping

In order to create textures and add fur to the 3D model, it needs to have UV maps. UVs are a method of depicting 3D models surface in 2D space. The name UV comes from the horizontal and vertical coordinates of the 2D image map. (Beane 2012, 160.) As the 3D object is essentially flattened to a 2D surface, it needs to have seams in edges where the mesh is opened up and laid out in a 2D surface. For organic models, which was the case for this project, seam placement was a trial-and-error process. The main rules that were tried to be followed, was to avoid stretching and overlapping UV islands in the UV map, as seen from Picture 17. Any overlapping UVs would create issues when generating and simulating fur later in the project.

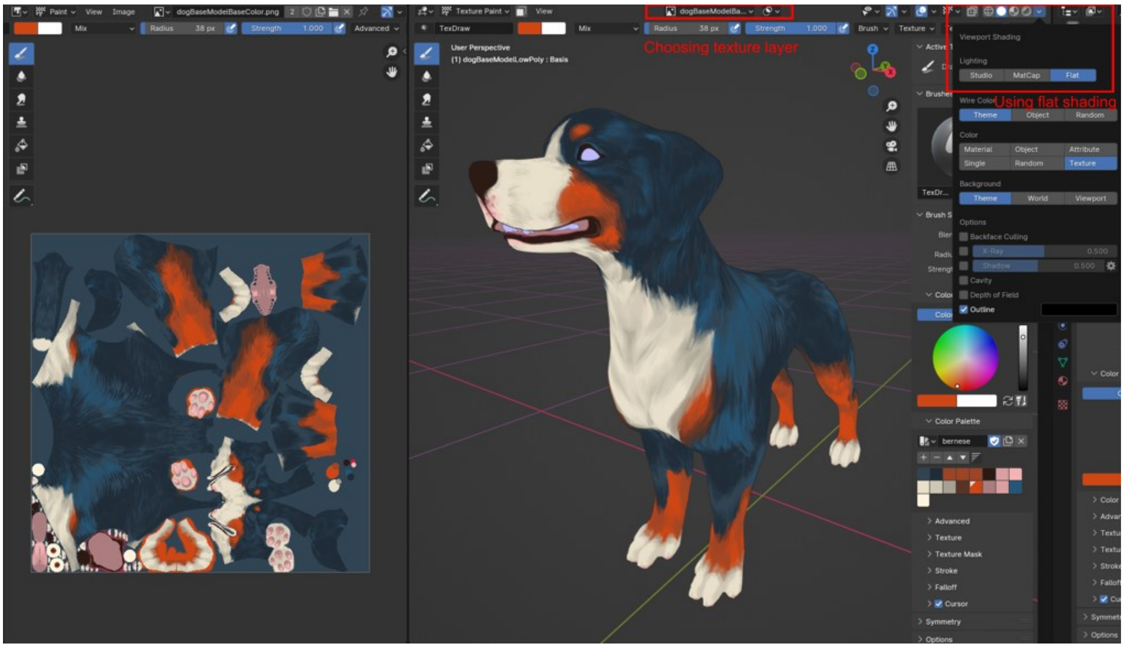


PICTURE 17. Left-side image showing seams in red colour, and how the chequered texture is displayed on the model. Right-side image is the UV map, blue colour indicates minimal stretching (Hautakangas 2024).

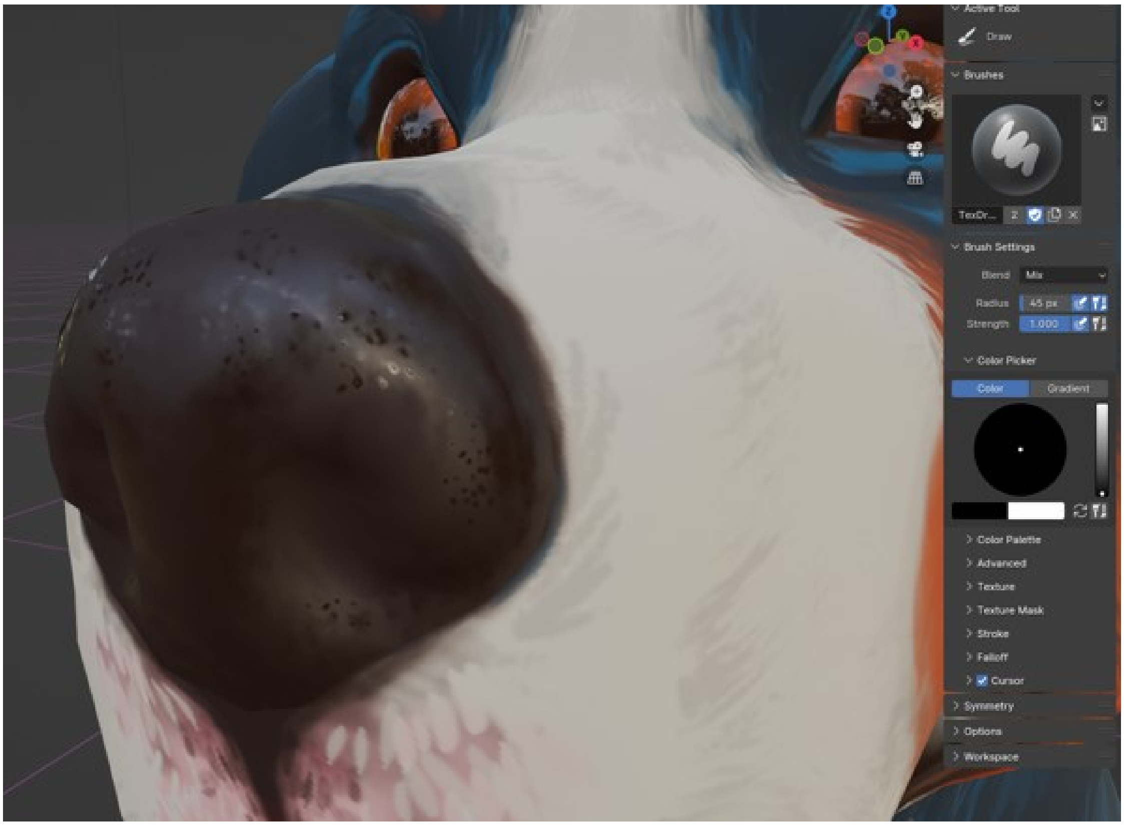
### 4.2.5 Texturing

After the UV map was finished, textures could be added to the model. In this project only a few texture maps were needed: the base colour, roughness and normal. Blender has a feature called Texture Paint in which textures can be painted straight to the 3D model with a UV map. The base colour texture map was painted with this Texture Paint mode, by first creating an image texture with high enough resolution and setting the image texture to the model itself in the Shading properties in order to see the painted texture in real time in the model (Picture 18).

The roughness texture map was also painted with this method. This map was important, as the snout, lips, mouth, teeth, and eyes are more reflective than the rest of the model. This image texture was set up as non-colour, as it only uses the alpha to determine what parts are reflective and non-reflective. Black indicates a reflective surface, and white is a rough surface. By changing the texture layer in Texture Paint mode, it was possible to see the reflectiveness in real time when painting on the model (Picture 19).

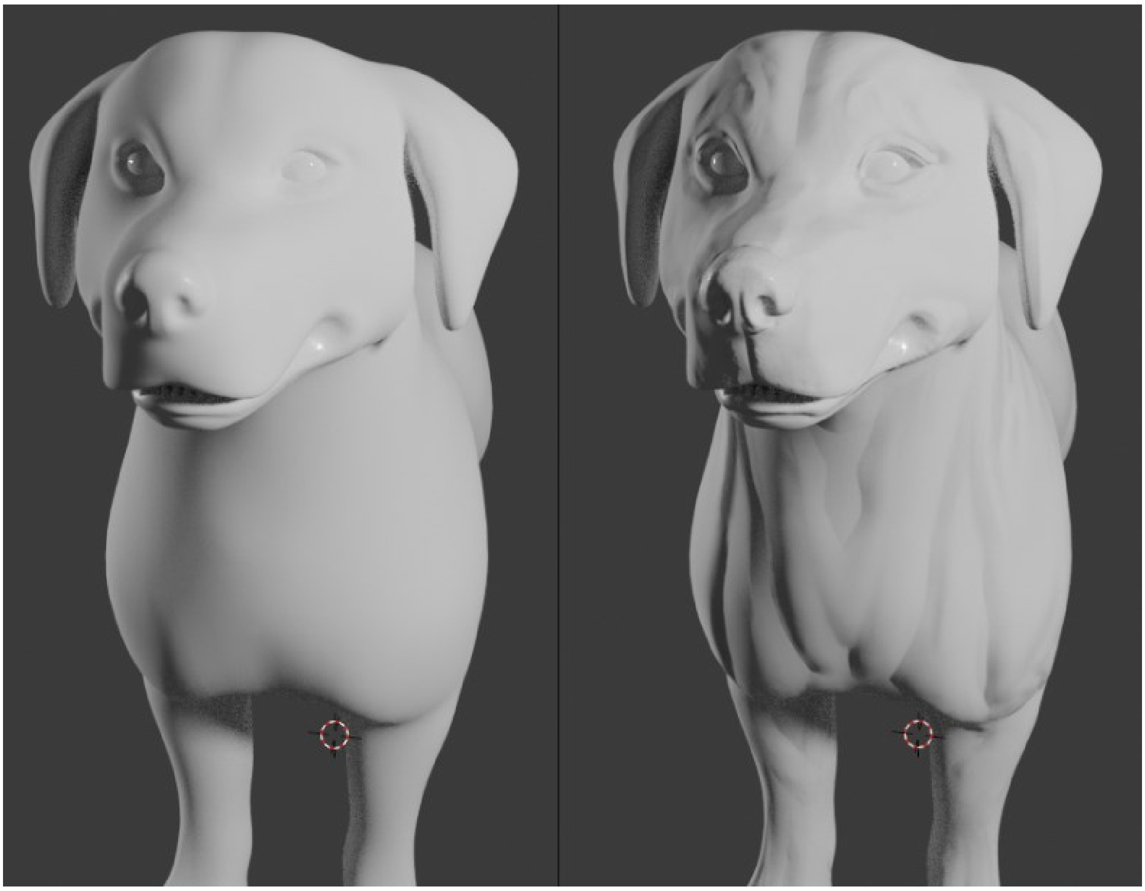



PICTURE 18. Blender viewport of the Texture Paint mode, for ease of access, different texture layers can be accessed from the menu, and for unobstructed painting the shading can be set to flat (Hautakangas 2024).



PICTURE 19. Close-up view of the snout, in which a more reflective surface has been painted on with black colour (Hautakangas 2024).

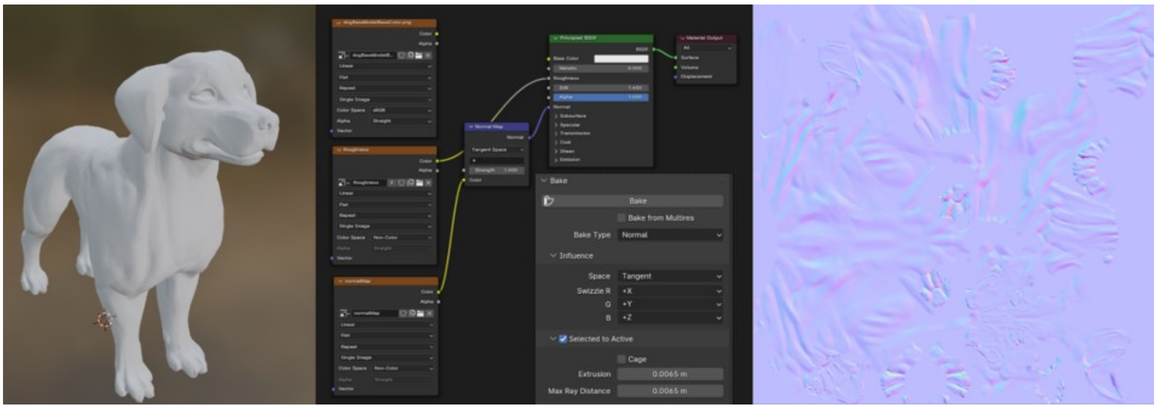
The final needed texture map was the normal map, which needed a different workflow called texture baking. The purpose of this method is to transfer details from the high polygon model to the low polygon model by using baked textures instead of polygonal data. The direction of the normals in the high polygon model is stored as RGB colour value data in the texture (Chopine 2011, 160). This creates the illusion of depth, when in reality the geometric surface is flatter than is perceived (Picture 20).



PICTURE 20. Left-side image showing the low polygon model with normal geometry, and the right-side image showing the same model with a baked normal texture map (Hautakangas 2024).

In Blender, texture baking without the use of addons required some setup. An image texture was added in the Shader menu and was then connected to a Normal Map vector and the shader itself. This image texture must be non-colour data, otherwise Blender will not read the data accurately. Both the high poly and low poly models were selected, while the low poly was the active object, and in the Render settings under Bake the Bake Type is set to Normal, and Selected to Active was used in this case (Picture 21). When the texture bake was finished,

some tweaking had to be done in the Extrusion and Max Ray Distance settings, but even after multiple tweaks, some slight editing of the image texture was done in Krita (Krita Foundation & KDE, n.d.) to smooth out any visible artefacts.

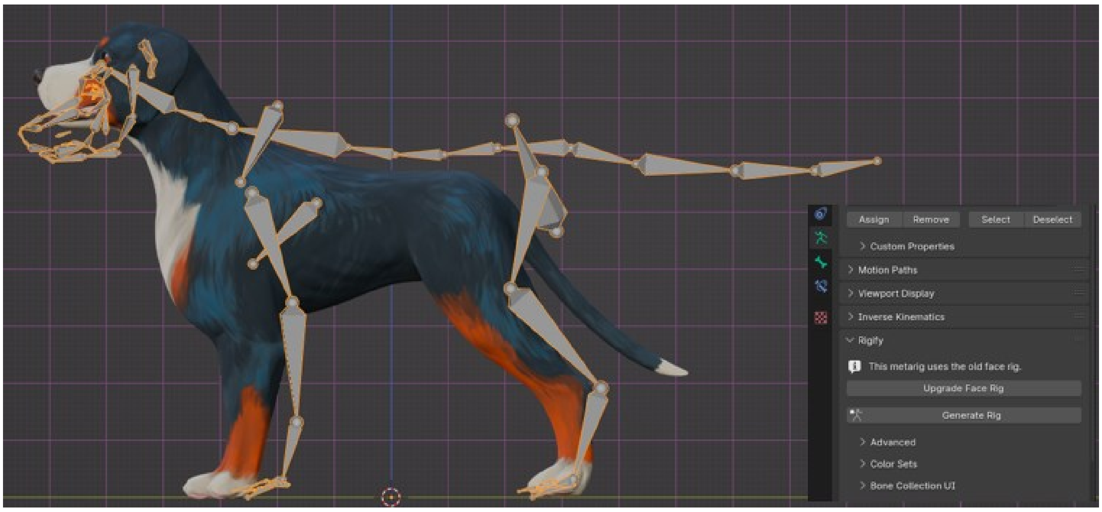


PICTURE 21. Images from left to right showing the result of the baked normals, the shader and baking setup and the texture map itself (Hautakangas 2024).

### 4.2.6 Rigging

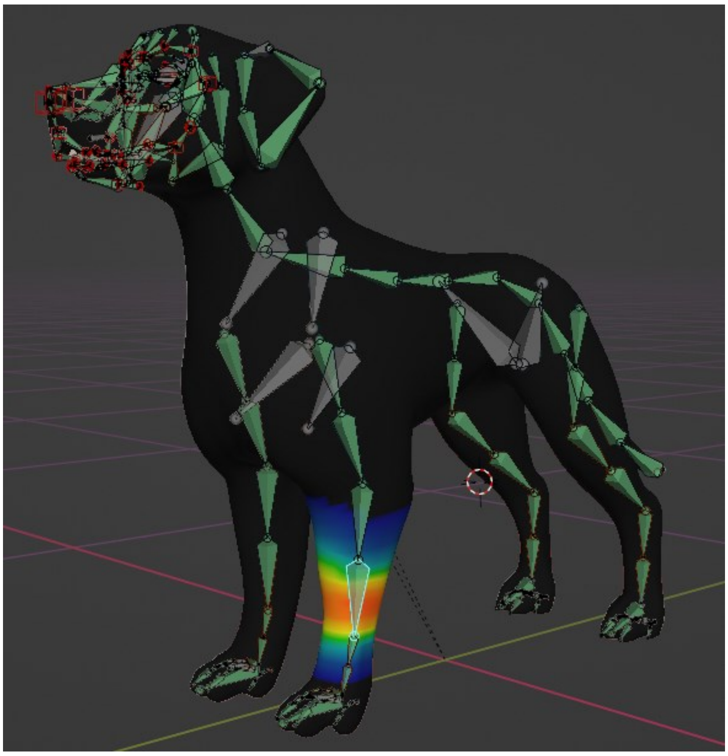
In order to animate the character model, it needed a rig first. Rig is essentially a skeleton of the model, in which movement is controlled by bones. These bones are attached to the mesh by either an automated function or each vertex influence value for each bone is edited manually. For the purpose of this project, the rigging process was aimed to be as quick and efficient as possible, which is why a template rig with Rigify addon was used. Rigify automates the rigging process by providing the bone structure and automating the rigging.

Once the wolf meta-rig was added to the scene, it needed to be adjusted to fit inside the model (Picture 22). Most important was to keep possible chained bones joints stuck together, otherwise the Rigify addon would display an error message and the rig would not be completed. Once the bone structure was placed inside the mesh, the scale of the rig was applied and the face rig was upgraded, the rig could be generated from the Object data properties (Picture 22).



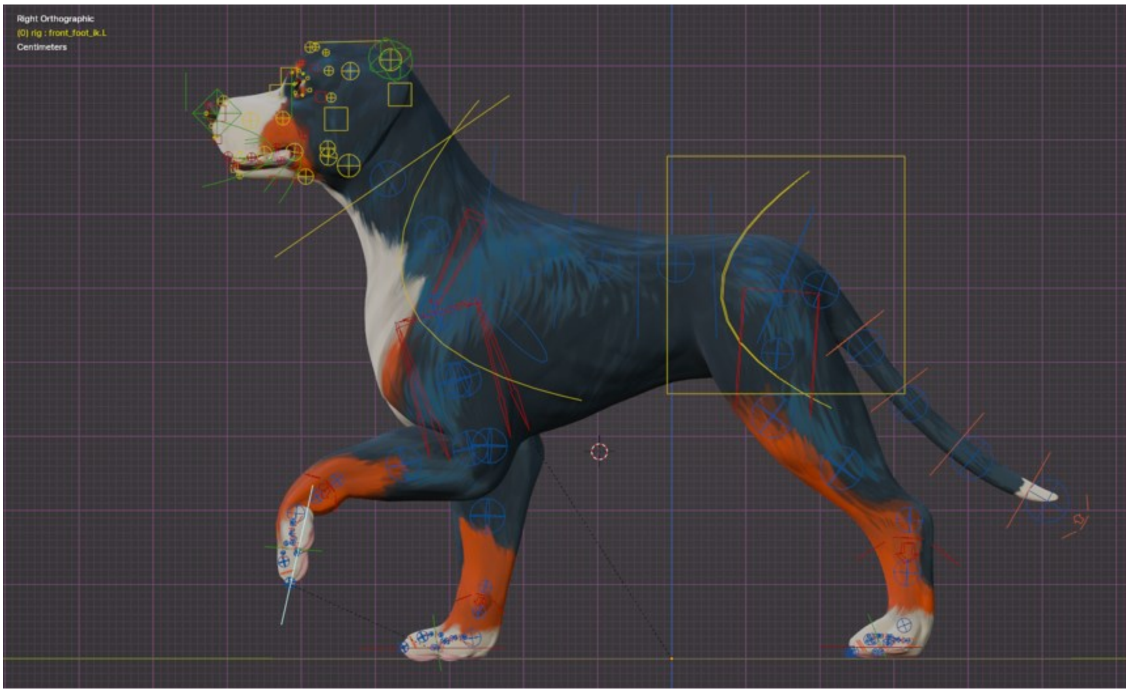

PICTURE 22. Rigify meta rig before adjusting it inside the mesh, and the Rigify application (Hautakangas 2024).

After generating the rig with Rigify, the new rig was parented to the mesh with the armature deform with automatic weights operator. Any adjustments could be made in the original metarig and re-generated later. Most of the editing was done with weight painting, in which the influence of individual bones to the mesh was fine-tuned for better mesh deformation (Picture 23).



PICTURE 23. The leg bone and how much influence it has on the mesh. Black means zero, blue colour indicates very little and red means greater deformation (Hautakangas 2024).

The great advantage of Rigify is that it automatically creates control bones for the rig (Picture 24). These control bones are used for inverse kinematics. Inverse kinematics allows one to move multiple bones with just one control bone, instead of moving and rotating each bone separately as it is done in forward kinematics (Zeman 2016, ch 6). Using control bones to animate the model is much more intuitive than animating each bone separately.




PICTURE 24. By moving and rotating the left paw IK control bone, the whole leg can be moved more intuitively (Hautakangas 2024).

### 4.2.7 Animating

A walk animation was chosen to be used in this project, as it can showcase how versatile the mesh is for animation, if there are any problems in the mesh deformation and that the fur simulation itself would have enough movement to test out the dynamics of the fur. As most of the movement was happening in the legs of the model, the focus was mostly on this part of the mesh, and not so much on the face.

As with many of the previous parts of this project, in animation using reference is also important. Various videos of especially Bernese Mountain dogs, but also

videos of dogs of various kinds were examined to see how the walk differs in different dog breeds. Specifically, how the legs move and in what rhythm was studied carefully. The back end of the dog appeared to be quite important, as according to AnimSchool (2020), the back legs control a lot of the up and down movement of the dog and the front legs are more for steering. It was essential to understand how the weight of the body transfers with each step, and taking into account that the 3D model itself might not look very bulky as itself without the added fur, so that had to be considered when animating.

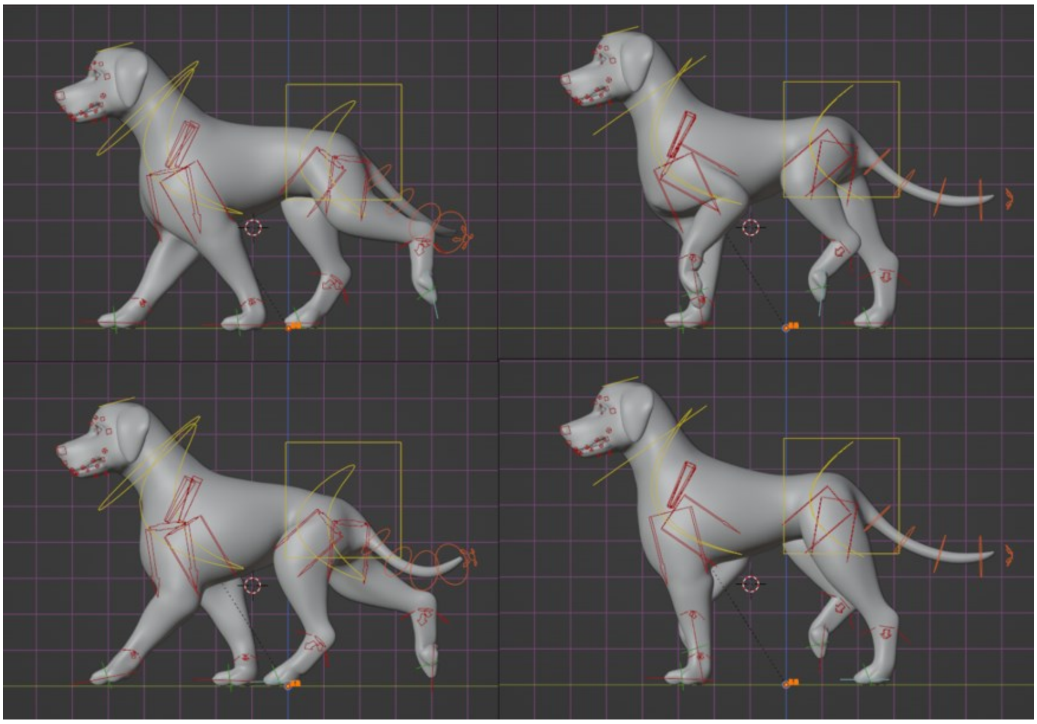
Main principles of animation, introduced by Disney animators Thomas F. & Johnston O. (1995) of which some of these were followed when animating. These principles are:

- Squash and Stretch
- Anticipation
- Staging
- Straight Ahead Action and Pose to Pose
- Follow Through and Overlapping Action
- Slow In and Slow Out
- Arcs
- Secondary Action
- Timing
- Exaggeration
- Solid Drawing
- Appeal.

Out of these principles, squash and stretch was used when animating the tongue, as the tongue is a very flexible part of the character. Arcs and slow in and slow out were used in various points in the animation, such as in the movement of the legs during walking, or when the character was turning and tilting the head. Anticipation was also subtly used on the front legs whenever the character was about to turn their head. Timing is used throughout the animation, to adjust the speed and rhythm of movement and to add weight to the walking animation. Follow through and overlapping action are greatly utilised when fur movement is simulated, but these principles were also followed for animating the tail and ears.

As for the animation process itself, pose to pose was deemed as the preferable method.

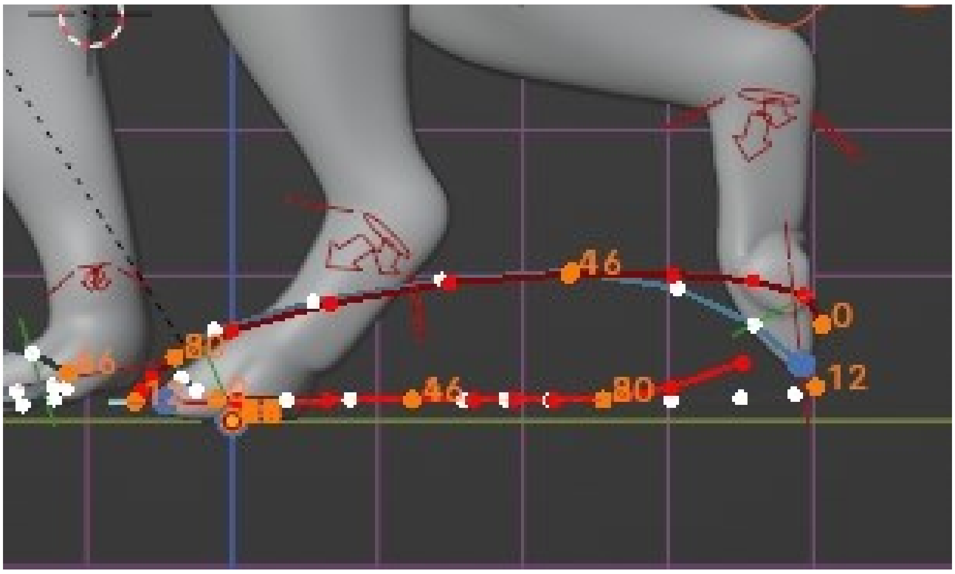
The animation process itself started with blocking out the most extreme poses, in walk cycles these are generally the contact and pass poses (Williams 2001). Only two poses were needed to start out with, as the same poses were then mirrored to the other legs on the x-axis (Picture 25). In between these poses, new poses were added to get rid of floaty and unrealistic movement and to adjust the timing. To make the animation process easier, the animation was worked in phases from bigger movements to smaller ones. First only the feet IK bones were keyframed, and after that movement to the hips were added, and so on. Only bones that were animated had keyframes, to keep things more readable in the Graph editor. The Graph editor lets the user see the animations as F-curves on a timeline, allowing to fine-tune timing and values, such as x-coordinate values of a bone.



PICTURE 25. The contact and pass pose which are mirrored to the legs on the x-axis (Hautakangas 2024).

Following the use of arcs is possible in Blender with Motion Paths, which allows the software to visualise the motion of specified bones as a curve as seen in Picture 26. Slow in and slow out can be achieved with the use of the Graph Editor, in which you can either use the automatic Bezier interpolation or edit them by

hand. The ears and tail were given follow through, overlapping actions and secondary action, based on the overall movement of the body.




PICTURE 26. Arcs visible as motion paths, dots with numbers indicate keyframes (Hautakangas 2024).

Creating an appealing animation can be tricky. In the case of this project, as the model was a mix between realistic and cartoonish, it was important to understand how much movement was to be realistic and how much exaggeration was needed. Finding this fine line between realism and cartoony can be difficult, otherwise the model and its animation can fall into the uncanny valley category, a term first introduced by Masahiro Mori (Mori, MacDorman & Kageki 2012), in which something appears uncanny and unappealing, an occurrence which can happen when trying to convey realism in created virtual humans or robotics for example. This uncanny valley phenomenon is most common in virtual humanoids, but a study by Schwind et al. (2018) indicates it can also occur in virtual animals as well.

During this project, multiple attempts at animating the same walk were made, with varying success, a walk with more cartoony style was made first after which a more realistic one was made. Later a few other short animation clips were made and combined together as a longer 300 frame animation with the NLA editor. Getting feedback was crucial in going towards the right direction in this part of the

project. Avoiding the uncanny valley was still relevant during the next phase, when work on the fur simulation was started.

## 4.3 Fur simulation

As was established within this thesis, hair and fur simulations come down to finding the middle ground between efficiency and realism. In a smaller project such as this, it could be argued that efficiency should take a slightly more prominent role, due to limited hardware capabilities. Great attention should be paid to visual appeal however, and whenever notable issues with i.e. amount of fur, stiffness of the fur, or rendering affects visual appeal, some efficiency can be sacrificed for a visually appealing outcome.

At the time of writing this thesis, Blender versions 4.0 and 4.1 have two methods for hair and fur generation, but only one method for simulation. Blender released Hair Nodes in version 3.5, which uses Geometry Nodes for generating and editing of the hair curves, which is an improvement of the old Particle Physics system and gives greater freedom for the user familiar with Geometry Nodes. Although the author was not particularly familiar with Geometry Nodes, the Hair Nodes system and the asset library were easy enough to understand.

After testing and documenting different methods for generating and simulating hair and fur (Appendix 1), a workaround workflow was established, in which the author decided to use the Hair Nodes system for generating and grooming the fur, after which the groom was converted to Particles for simulation, and some layers not being simulated at all. Using this workflow allowed for more artistic control when grooming and styling the fur, but also enabled to simulate the movement. This method has its downsides, and they are discussed in detail further in the thesis.

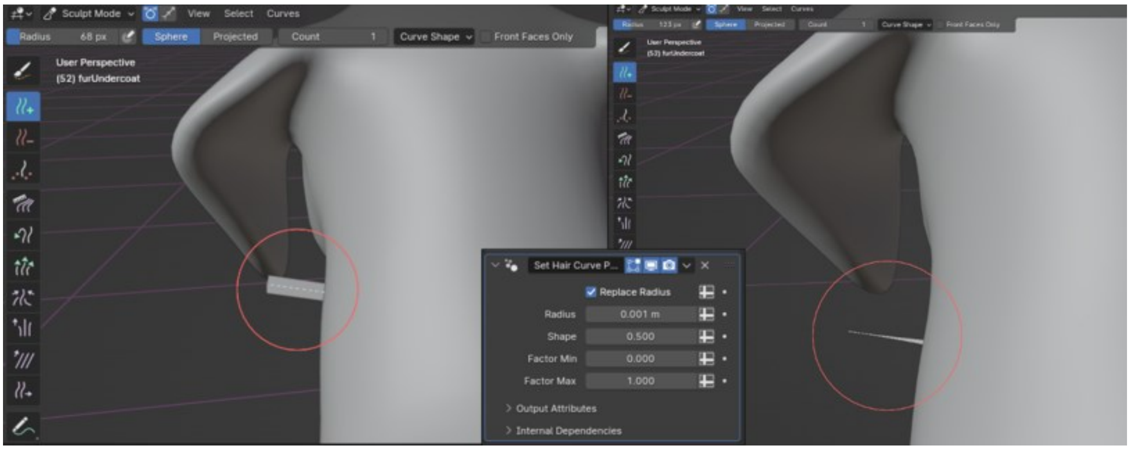
### 4.3.1 Fur grooming

Before any fur was added, some planning was needed. Keeping in mind how animal fur, and in this case dog fur is normally in layers; having an undercoat which is more voluminous and softer and guard hairs that are longer, thicker, and coarser, makes it logical to think fur in layers also when generating them in a 3D model. In addition to having fur in two layers, it was necessary to separate them even further. For easier control of the fur when grooming, and locating possible issues when simulating, the fur could be roughly separated to a body, face, ears, tail, and details. Having multiple layers of fur has the downside of increasing render and baking times however, so this complicates things.

Some settings were also tweaked before fur was added, in render settings under curves, the curve subdivisions were increased, and the viewport display was set to strips instead of strands. This was to make the fur look better when rendering, but also to look more like actual fur when previewing it in the viewport. It was also necessary to have the model be in a rest pose, and not in the middle of an animation clip, to avoid possible issues later.

Fur layers were added from the add menu, under Curves, by adding empty Hair Curves while the dog model was active. The Modifiers tab of the hair curve object is where the hair curve shape, interpolation, clumping, noise, and other modifiers can be added to adjust the amount of fur, look of the fur strands, and modify the overall groom. Surface Deform is there by default, which controls what object the hair curves are placed upon. All the modifiers were accessible from the Geometry Nodes viewport as well, but using only the basic Modifiers tab and its settings was deemed sufficient to work with.

Placement and manual grooming were done in the Sculpting mode, which has a similar UI but differs from mesh Sculpting mode. The method of using a brush to style hair is fairly similar to sculpting meshes. Hair Sculpting has different tools for adding hairs, combing, growing, and shaping them. A single strand of fur was added in the mesh in order to see the general shape and modify it with the Set Hair Curve Profile modifier by adjusting the root to be thicker and end thinner for a more hair strand look (Picture 27).

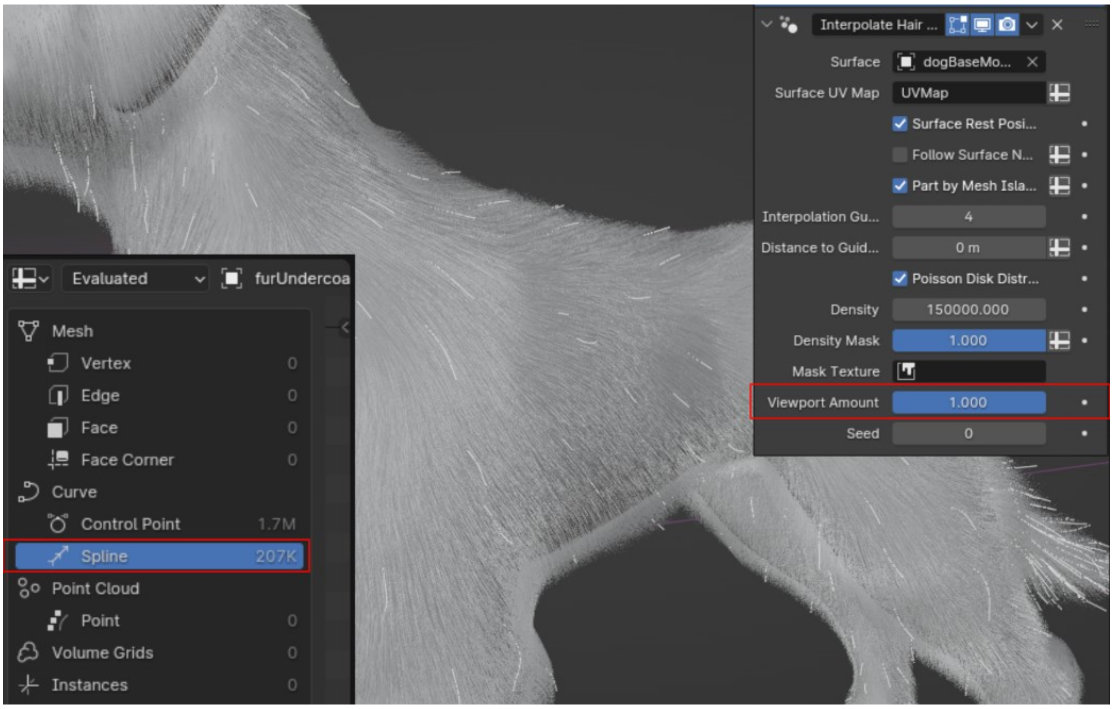



PICTURE 27. Left is a template hair curve, with modified settings on the radius and shape; the right-side curve resembles more of a hair strand (Hautakangas 2024).

As was assessed previously, when working with substantial amounts of fur it is preferred to place guide hairs and interpolate them, this is more efficient but also non-destructive when there is a need to adjust the general number of strands. To start this process, a small number of hairs were hand placed fairly evenly on the model and were then combed to have a natural flow, and to achieve a voluminous look the curves were kept slightly “puffy”. To accelerate this part, some settings in the Curve Shape were adjusted to have the curve shape follow the same flow as the previously shaped curves (Picture 28). Having x-symmetry on was also used to speed up this process.


PICTURE 28. Guide hairs on the model and settings for adjusting the shape of the curve to follow previously placed ones (Hautakangas 2024).

With some guide hairs now placed, more fur volume could be added with the Interpolate modifier. In order for the modifier to work, it needed the Surface setting to be changed to the dog model. Density was adjusted to increase the amount of fur, but to keep the Blender file more lightweight when working the Viewport Amount was changed to be of lower value. The number of hairs can be assessed from the Geometry Nodes window in the Spreadsheet, Splines specify how many hair strands are in the scene, but it only uses the viewport amount. Thus, if Interpolate Hair Curves uses a lower amount in the viewport, the value needs to be changed to 1.000 in the settings to see the true number of curves (Picture 29).



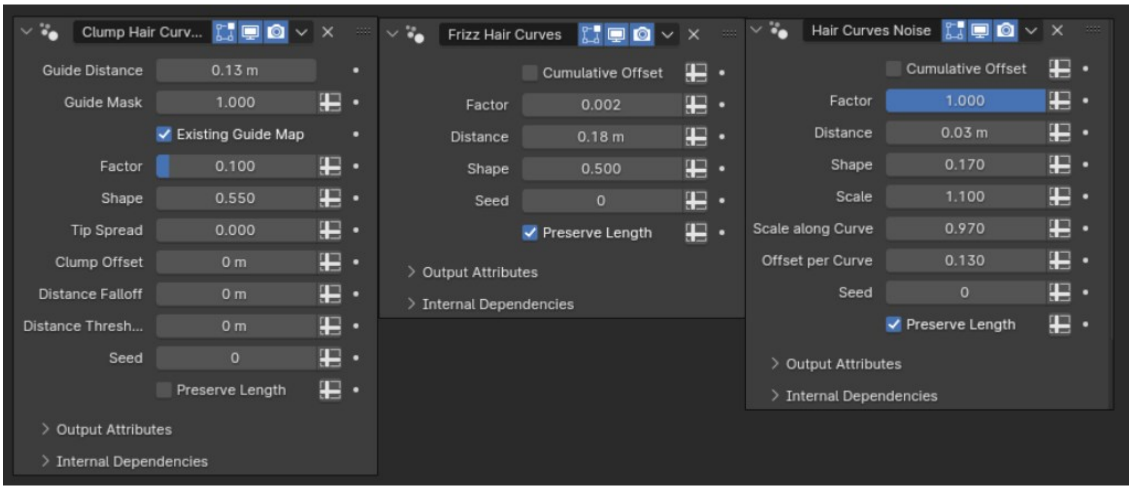

PICTURE 29. Fur interpolated on the model, and the number of curves when the Viewport Amount is set to the highest value that is used when rendering the viewport (Hautakangas 2024).

While the modifier did interpolate fur in the model and following the flow of the placed guide hairs, it placed hairs in parts of the mesh that were not desirable, such as inside the mouth, eye sockets etc. For humans, having a separate scalp mesh for hair generation is a common method to use. But as animals have fur in almost the whole body, the whole mesh is generally used. To control the generated fur placement, alpha texture maps were used. White colour is for larger density and black is zero, as such a texture map was made and added in the Mask Texture in the interpolated hair modifier settings. Adjusting the fur density

can also be done with Weight Painting to adjust vertex values, but this is not very efficient for lower poly models as it is dependent on the amount and placement of vertices.

For creating details in fur, for example the face area such as the brows or cheeks, hand painting the fur as a workflow seemed like the better approach. Painting the hair curves manually gave more control over interpolating the hairs. The curves were painted on the model using the Density brush, in which hairs are painted on an area based on the size of the brush. Density of hairs with this brush can be adjusted, but having a non-destructive method for increasing or decreasing density is usually preferred, as such a Duplicate Hair Curves modifier was added to easily adjust the number of hairs. The modifier needed to have the radius adjusted to a much smaller value, otherwise the duplicated hairs were hovering over the model and not sticking to the model itself. Even when adjusting the radius, sometimes the hairs still were not on the model itself, as such the Attach Hair Curves to Surface modifier was used.

With interpolating and manually painting hairs the overall outcome was still too uniform, and to correct this several different modifiers were used to increase clumping, frizz, and noise of the fur. Clump Hair Curves modifier guides the hairs to form clumps, which is a natural occurrence due to static, friction, and oiliness of hair. Frizz Hair Curves modifier adds randomness to the hair curves, making them look less as straight uniform lines. Hair Curves Noise modifier randomises the direction where hair curves are pointing at. All the aforementioned modifiers have multiple settings for fine tuning, but most modifiers share a few same settings such as the factor and shape (Picture 30). Adjusting the values on the modifiers was seen in real time on the fur, which made it intuitive to experiment different styles.



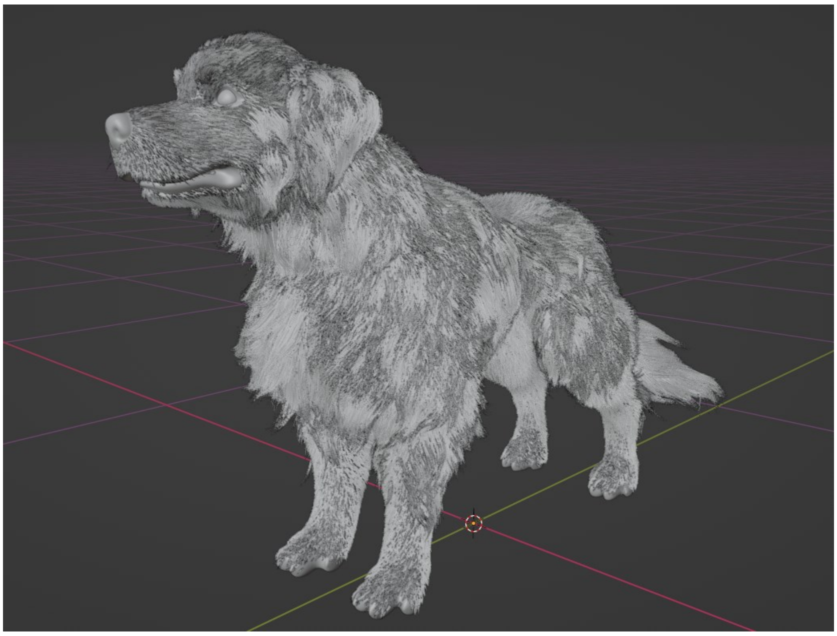

PICTURE 30. Modifier settings. Factor affects how much overall effects take place and shape affects how much the modifier affects the curves (Hautakangas 2024).

The fur grooming process was repeated for multiple different fur layers. Choosing the best workflow for each layer was dependent on the purpose of the fur: when larger areas had to be covered, the easiest method was to use guide hairs and interpolate them, and when creating details, hand painting and duplicating hairs was preferred due to having more control of the flow and placement. After the fur grooming process was finished, the model ended up having 9 different layers of fur, of which 5 were simulated and 4 were kept as static layers not having simulated movement (Table 1).

TABLE 1. Fur layers, their hair types, generation method and whether they were simulated or not.

| **Fur layer** | **Hair type** | **Generation** | **Simulation** |
|---|---|---|---|
| Ears | Undercoat | Interpolation | no |
| Ears details | Guard hairs | Manual | yes |
| Face | Undercoat | Interpolation | no |
| Face details | Guard hairs | Manual | no |
| Face cheeks | Guard hairs | Manual | yes |
| Whiskers | Vibrissae | Manual | yes |
| Body | Undercoat | Interpolation | no |
| Body details | Guard hairs | Manual | yes |
| Tail | Guard hairs | Manual | yes |

The styling process took multiple attempts in order to find a suitable look for the character model. Critically evaluating the outcome after each attempt made it clear that it was important to keep the fur more stylized rather than very realistic, as the model itself was more stylized. This was achieved by adjusting the hair shape to be larger in diameter than a realistic hair would be, as well as shaping the overall groom to be more controlled. This evaluation process can be found more in detail in the Appendix 2. The finished groom can be seen in Picture 31, in which the character model has a very moderate number of curves for an animal.



PICTURE 31. All the fur layers with approx. 110k curves in total (Hautakangas 2024).

### 4.3.2 Converting hair nodes to particles

Once the fur groom was finished, the layers needed to be set up for conversion to particle physics. The reason conversion is needed is because at the time of writing this thesis, Hair Nodes do not have a simulation system. Converting them to particle physics was a workaround, until a better system would be implemented in Blender.

Converting the layers is a destructive process, as the layers need their modifiers applied completely, thus removing the ability to edit them, i.e. adjust the density or curviness. For this reason, a separate Blender file with finished fur groom was kept, and if any edits needed to be made to the fur, it was done in this separate Blender file. An edited fur layer could be appended to another file if needed.

Only the layers that were simulated needed to be converted, and layers for creating fur volume were kept as their original curve objects, without applying any modifiers. These fur volume layers had to have the Surface Deform modifier at the bottom of the modifier list, in order for the curves to follow the mesh when animated. For better visibility and faster simulation, these layers were also kept hidden from the viewport.

The conversion was started by applying all modifiers in object mode, beginning from the top layer in the modifiers tab. The reason modifiers need to be applied first, is because any modifications to i.e. amount of fur as interpolation or duplication, or curve shape like frizz or noise is not transferred otherwise. The viewport amount of generated fur was set to the highest value before applying in order to convert the desired density of fur. Fur strands could be also interpolated later in the particle properties, but depending what fur layer is used, the interpolation might not work as intended, i.e., fur being interpolated on the whole mesh instead of just the tail.

After applying all modifiers, it was checked whether any bugged hair strands were visible, as was the case in some layers. These hairs were removed in the sculpt mode by using the Delete brush. Making sure no more potentially problematic hair strands remained, all strands were selected to be active in sculpt mode and from the Curves menu Convert to Particle System was chosen. Once the conversion was completed, the hair curve object remained in the scene, and was hidden from the viewport and render views. Conversion also reset the size of the hair strands, which were adjusted again in the particle properties menu.

### 4.3.3 Simulation with Particle Physics

The overall workflow for offline simulations is to bake simulations first and do a final render afterwards. In this case baking means essentially to store simulation data in memory or disk, to preview them faster on the timeline. Depending on the quality of the simulation, baking can take a long time and requires plenty of disk space, which is especially the case when having multiple simulations.

Most relevant sections in the particle physics menu are Hair dynamics, Cache, Render, Hair shape and Field weights. In order for simulations to work, Hair dynamics is set to active. Cache is where simulation data settings can be adjusted, and if the cache is saved straight to disk which might be preferable instead of temporary memory. Render menu allows you to change the material of the hair or fur. Field weights have values on how much outside forces can affect the simulation, such as wind or gravity. These effectors can be added from the add menu to the scene, which have their own settings for fine tuning.

Hair dynamics have a few main sections for adjusting simulations: collisions, structure, and volume. Other settings include Quality steps that increase the overall quality of simulation with the cost of disk cache size of the bake. Pin goal strength affects the general stiffness of the hair particles. These hair dynamic settings can be saved as presets and load them to each separate simulation. For similar fur using the same settings could be considered useful, but in certain cases it might be preferable to have more or less stiffness in specific fur.

In order to start working on the simulation, one layer was chosen, and its hair dynamics values were adjusted. The goal was to have the fur be fairly stiff, as to not start to droop and lose its shape immediately after starting the simulation, and to also have realistic simulated movement. The default values were quite close, but especially pin goal strength and quality steps needed to be adjusted. To prevent the fur from falling inside the mesh itself, the dog object was given collision from the physics menu.

At first only a few frames were baked to see if any issues would happen, such as exploding hair which can be quite common. This occurs generally on the first few

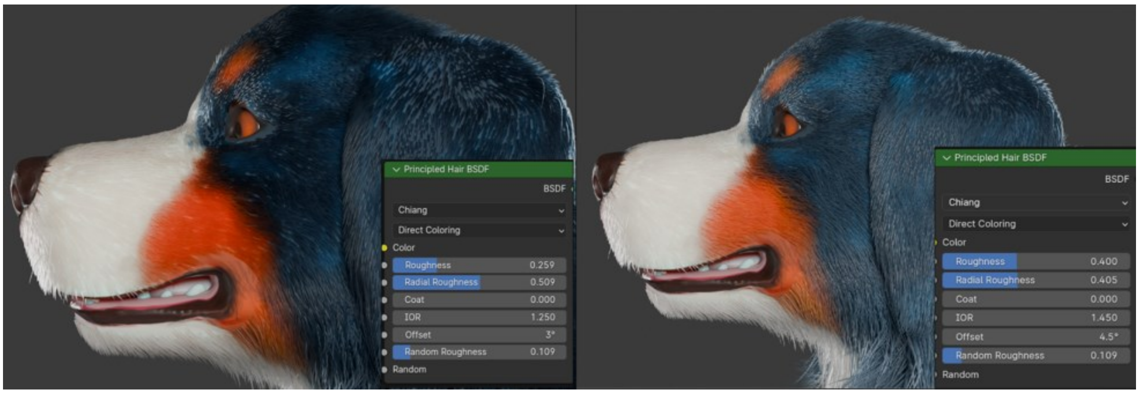
frames of simulation, where the hair strands erratically expand in a manner which resembles an explosion. Exploding hair did happen multiple times while testing, and even when trying to adjust collision distance and impulse clamping, nothing seemed to work. So, in the end, the dog model had to have the collision taken off. Keeping the pin goal strength at a higher value kept the fur stiff enough as to not look unrealistic, even without the lack of a collision from the mesh itself. To actually see some simulation, the model needed to be moving, thus an animation as action was appended from another Blender file and set as active animation from the action editor.

Similar testing was done on all the rest of the simulation layers, and when the values were deemed sufficient and no issues were apparent, a full bake on all layers was done. The animation clip was 300 frames, and the bake had to be done for each simulation, so 5 different bakes in total, that took approximately 40 minutes time and 3,8 gigabytes of disk space. Each baked simulation was verified to have no issues and were ready for rendering the final animation.

### 4.3.4 Shading and rendering

Setting up the materials in the shader can, and in many cases, should be done even earlier during the process, especially when reviewing the overall look of the finished fur groom to see how it would look in a final render, hence this is what was done during this project. However, to keep the structure of this thesis coherent, it was left at the final section of this chapter.

Similarly, as was done with the dog model itself after the texturing process, specific material was set up for the fur. In the shader properties, Principled Hair BSDF shader was chosen for the fur, and the same diffuse texture used for the model itself was kept as the colouring. As the goal was to not achieve a realistic material for the fur, but rather keep the style consistent, adjusting the values were done multiple times to find a pleasing outcome (Picture 32). Ideally, the fur was to look shiny but also soft.




PICTURE 32. The hair shader, left-side fur is too dark while the right-side fur looks better and does not clash with the diffuse texture (Hautakangas 2024).

Some additional tweaking was also done on the material of the dog model, as it appeared too plastic. To fix this issue, Subsurface value was set to higher. This allows the light to slightly travel through the model, as light would shine through skin for example. Placement and strength of lighting also affects how much Subsurface has an effect, therefore the lighting was adjusted as well. Light placement was based on the three-point lighting technique, in which a key light, fill light and rim light is used. This method lights the object evenly, bringing out enough details, avoiding stark shadows, and framing the object to stand out from the background (Beane 2012, 234). The colour of the lights was also adjusted for a more visually interesting effect. To maximise the advantage of lighting, a simple plane object was added underneath the model, but kept as a holdout, which makes the plane invisible in renderings. The purpose is to offer bounce lighting to the dog model, to distribute lighting more evenly and increase the perception of depth (Picture 33).

PICTURE 33. The effect of bounce lighting is very apparent on the hind legs of the right-side image (Hautakangas 2024).

As the rendered animation was a simple and short one, keeping the camera stationary in the scene seemed like the best course of action. As such, the model was kept in the middle of the frame, from a slight angle, to showcase the project as a whole. The lack of a background makes the framing and compositing simpler, as the focus is only on the object itself and nothing else. Leaving out the background was a choice by the author, as the model was the main point of this thesis. The background was made transparent from the Render settings under Film.

To render the animation, some settings were edited for a slightly shorter render time. Cycles render was used, and as the scene did not have any other objects other than the model itself and the invisible ground object, the max samples used in rendering were adjusted to a lower value of 512 with a Noise Threshold of 0.0018, and denoising set to use the OptiX denoiser which is more suitable for animation. Additionally, the amount of light bouncing in the Light Paths was set to a maximum of 8, and individual values adjusted to smaller values as well. With these settings a single frame took approximately 40 seconds to render, and the whole animation 2,2 hours on a hardware setup as seen from Table 2. The finished render of the animation can be found in the Appendix 3.

TABLE 2. Hardware setup.

| System | Desktop PC |
|---|---|
| Processor | 11th Gen Intel(R) Core(TM) i7-11700K @ 3.60GHz |
| RAM | 32 GB |
| OS | Windows 10 |
| Graphics Card | NVIDIA GeForce RTX 3070 |
| Storage | SSD (500 GB) |

### 4.4 Evaluating the workflow

To evaluate this workflow of combining the Hair Nodes system with the Hair Particle Physics system, it was determined how much artistic control and ease of access as well as efficiency was present. This evaluation was subjective and can differ based on overall knowledge of hair simulations and Blender, available hardware, various technical skills as well as other experiences of the person doing the evaluation. It did however give insight to the author, in how to work with hair and fur simulations in the future, and whether Blender is a suitable software for such work. As the workflow consists of roughly three parts; fur grooming, conversion to particles, and simulation, the evaluation process can be divided into these sections as seen from the simplified flowchart in Figure 3.

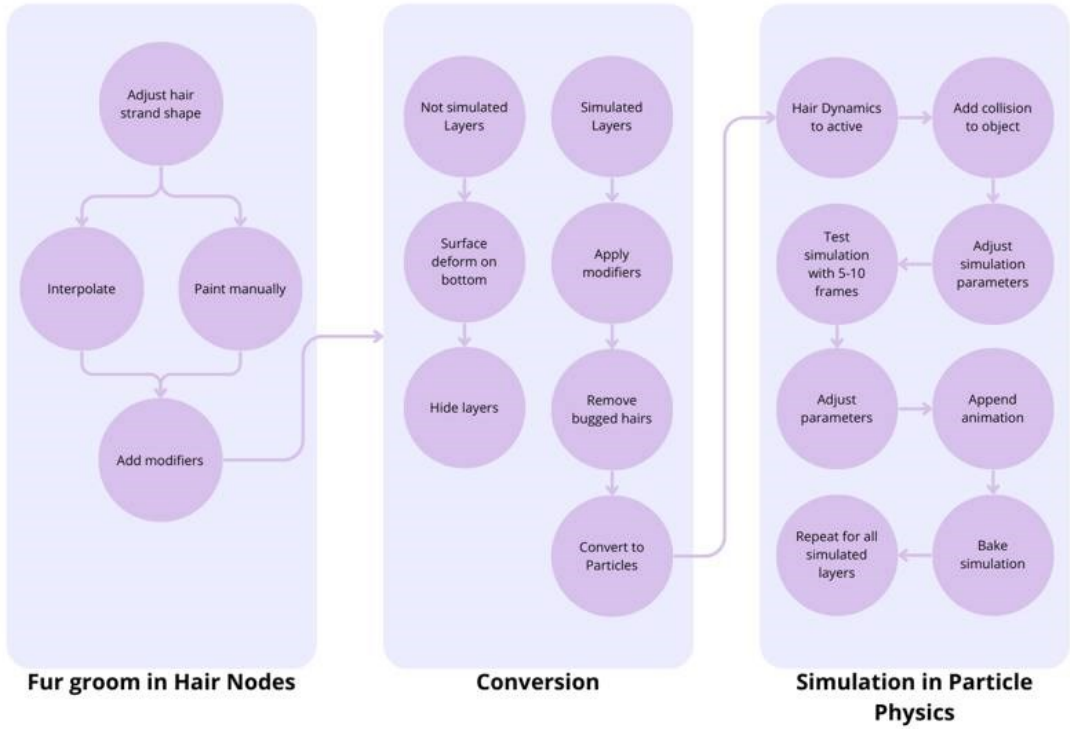



FIGURE 3. Flowchart of the workflow (Hautakangas 2024).

The fur grooming process by using the Hair Nodes system was generally intuitive as an artist: having different modifiers for procedurally generating and altering the appearance of fur was easy to understand and control. To see the adjustments made in the fur in real-time was helpful. The only issue with the modifiers was at times the naming conventions of the settings was not clear at the start, but needed testing to see what happens with each parameter. Having the ability to paint hairs by hand, and style them was similar to sculpting meshes, and the familiarity of this was not only artist friendly but made working faster and efficient.

The biggest issue with this workflow was undoubtedly the process of converting Hair Nodes to particles. Not only was the process quite cumbersome, having to do multiple steps in a certain way, and failing to do so, possibly breaking something later when simulating, but was also a very technical process for an artist. Some steps, such as applying modifiers first in order for simulations to work did not make sense as an artist, as it was a more technical detail that was not visible. This part was clearly redundant, but unfortunately a necessary step in order to work on the simulations later. The destructive process of applying all modifiers and losing the ability to later adjust them in-between simulations increased inefficiency.

Working on the simulations was reasonably straightforward, as there were not an excessive number of different parameters to adjust in order to create believable simulations. For an artist this can be seen as an advantage, but also a disadvantage if more control over the simulations was needed. But as the model and its animations were quite simple, the available settings were adequate. Whenever issues arose, such as exploding hair, it was not immediately clear what was causing them, and whether that was because of using converted Hair Nodes. This led to cutting corners when it comes to realism with the simulations, as i.e. the collision had to be taken off from the mesh to avoid hair explosions. Another method of favouring efficiency over realism was to simulate only some fur layers, which can work in a simple and stylized model, but not as much when trying to portray realism. One other issue was the fact that not being able to use interpolated hairs while simulating massively impacted the performance. As with generating interpolated hairs, even the simulation can benefit from a smaller amount of guide hairs that only need to be simulated, and the rest only follow the guides.

In conclusion, the workflow can be seen as an alternative, albeit a bit arduous method for simulating simple and stylized characters with fur, until a compatible system of simulation for Hair Nodes is implemented in Blender. If the goal is to create very realistic hair or fur, in appearance as well as simulation, this method is not recommended. The inefficiency can greatly hinder the process, which could be disastrous in a large project with strict deadlines, as baking and rendering take multiple hours to do.

# 5 DISCUSSION

As complex and extensive as creating fur simulations in Blender is, it is completely possible to do. Stylized and simple characters with fur are more feasible to create, especially when limited by a hardware setup and available time. The used workflow of combining Hair Nodes with Particle Physics proved there can, at least in part, be an artist friendly workflow in creating fur simulations. This was at a cost of efficiency though. Prior knowledge in simulations and understanding the technical aspects behind it can make the process much easier, as one can pinpoint e.g. what the cause for a specific problem is. It was important to understand the need to balance between attainable realism and efficiency, and that it can differ for every project.

To say that Blender is not viable for more realistic hair or fur simulations would be untrue, as testing or creating it in practice was not in the scope of this thesis, and it would require more knowledge in simulations in general to make an assessment. The practical project itself also lacked an environment for the character, which is crucial when it comes to realistic representation, as lighting, reflections and other aspects within this environment affect the overall outcome. As no other software for simulations were also not discussed in detail or tested, it would be unfair to judge how viable Blender would be in such a case, as it would require comparing it to other industry standard software. When it comes to finding a workflow that is artist friendly, it is also very subjective. All experiences and knowledge, and personal preferences can affect the outcome. As such, the evaluation was purely based on the author’s own experience.

When it comes to the overall outcome of the practical project, plenty of experience was attained about fur simulations in general and being able to critically think and evaluate one’s process constantly and do improvements when it’s necessary. The project itself proved it is possible to use the aforementioned workflow especially for stylized characters, even if it does have some redundancies as a consequence. Plenty of tutorials about fur and hair simulations in Blender exist, but very few of them combine the two existing systems as was done in this thesis. The reason behind this is most likely the inefficiency and potential unstable

results. More artist friendly workflows for fur and hair simulation would become available when a compatible simulation system for Hair Nodes is implemented in Blender. It was unfortunate that such a system was not yet available at the time of writing this thesis, as the Hair Nodes system has great potential as a tool.

Furthermore, as automation and realism are at the core of simulations, and a very recent implementation of OpenAI's (2024) AI model Sora brings these to a new unprecedented level, it begs the question – Will offline simulations, that are used in animation and film become obsolete in the media industry? Is it possible, or even beneficial to keep improving a system that is very reliant on hardware, and costly to create, when AI can potentially create a similar result with a fraction of the time and costs? If in the future AI is capable of keeping a consistent style in animated characters for example, and a character's animated or simulated movement could instead be controlled by a written prompt alone, this could definitely decrease multiple hours of manual work on character creation and animation. The impact of AI is already enormous – as well as the possibility to use it as a tool in the future – thus further research on the subject is needed.

## REFERENCES


Adobe. 2024. Adobe Photoshop. Version 25.6. [Computer Software]. Adobe.

Andersen, T. G., Falster, V., Frisvad, J. R., & Christensen, N. J. 2016. Hybrid fur rendering: combining volumetric fur with explicit hair strands. The Visual Computer, 32 (6–8), 739–749. Read on 07.11.2023. https://doi.org/10.1007/s00371-016-1252-x

AnimSchool. 2022. It's time to walk the dog – How to Animate a Quadruped. Released on 16.3.2022. [YouTube Video]. Watched on 12.03.2024. https://www.youtube.com/watch?v=e5mn7WxjMZg

Anjyo, K., Usami, Y., & Kurihara, T. 1992. A simple method for extracting the natural beauty of hair. Computer Graphics (ACM), 26 (2), 111–120. Read on 12.12.2023. https://doi.org/10.1145/142920.134021

Beane, A. 2012. 3D animation essentials. 1st edition. Indianapolis, Ind: J. Wiley & Sons.

Bertails, F. 2008. Realistic hair simulation animation and rendering. ACM SIGGRAPH 2008 Classes, 5–108. Read on 24.10.2023. https://doi.org/10.1145/1401132.1401247

Blender. N.d. Features. [Website]. Read on 04.12.2023. https://www.blender.org/features/

Blender. N.d. History. [Website]. Read on 04.12.2023. https://www.blender.org/about/history/

Brave. 2012. Direction: Mark Andrews & Brenda Chapman. Production: Walt Disney Pictures, Pixar Animation Studios. Production country: United States.

Chai, M., Zheng, C., & Zhou, K. 2014. A reduced model for interactive hairs. ACM Transactions on Computer Systems, 33 (4), 1–11. Read on 14.11.2023. https://doi.org/10.1145/2601097.2601211

Chiang, M. J.-Y., Bitterli, B., Tappan, C. & Burley, B. 2016. A Practical and Controllable Hair and Fur Model for Production Path Tracing. Computer graphics forum. 35 (2), 275–283. Read on 24.10.2023. https://doi.org/10.1111/cgf.12830

Choe, B., & Ko, H.-S. 2005. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. IEEE Transactions on Visualization and Computer Graphics, 11 (2), 160–170. Read on 02.11.2023. https://doi.org/10.1109/TVCG.2005.20

Chopine, Ami. 2011. 3D art essentials: the fundamentals of 3D modeling, texturing, and animation. 1st edition. Oxford: Focal Press.

Csuri, C., Hackathorn, R., Parent, R., Carlson, W., & Howard, M. 1979. Towards an interactive high visual complexity animation system. Computer Graphics (New York, N.Y.), 13 (2), 289–299. Read on 16.11.2023. https://doi.org/10.1145/965103.807458

Elemental. 2023. Direction: Peter Sohn. Production: Walt Disney Pictures, Pixar Animation Studios. Production country: United States.

The Flintstones.1994. Direction: Brian Levant. Production: Amblin Entertainment. Production country: United States.

Giardna, C. 2021. Making Sure Characters Don't Have Bad Hair Days: Three female engineers will be recognized by Academy SciTech Awards for hair simulation developments. Hollywood Reporter, 426, 76–76.

Godzilla vs. Kong. 2021. Direction: Adam Wingard. Production: Legendary Pictures. Production country: United States.

Hoffman, J., Hu, T., Kanyuk, P., Marshall, S., Nguyen, G., Schroers, H., & Witting, P. 2023. Creating Elemental Characters: From Sparks to Fire. Proceedings - SIGGRAPH 2023 Talks. Read on 04.12.2023. https://doi.org/10.1145/3587421.3595467

Hsu, J., Wang, T., Pan, Z., Gao, X., Yuksel, C., & Wu, K. 2023. Sag-Free Initialization for Strand-Based Hybrid Hair Simulation. ACM Transactions on Graphics, 42 (4), 1–14. Read on 03.11.2023. https://doi.org/10.1145/3592143

Iben, H., Meyer, M., Petrovic, L., Soares, O., Anderson, J., & Witkin, A. 2013. Artistic simulation of curly hair. Proceedings - SCA 2013: 12th ACM SIGGRAPH / Eurographics Symposium on Computer Animation, 63–71. https://doi.org/10.1145/2485895.2485913

Idyllic Pixel. N.d. PureRef. Version 1.11. Idyllic Pixel. [Computer Software]. Idyllic Pixel.

J Hill. 2021. EVERYTHING You Need to Know About Topology. Released on 13.08.2021. [YouTube Video]. Watched on 11.03.2024. https://youtu.be/6Kt0gW3_kio?si=15iYwbTgeypLfakg

Kajiya, J. T., & Kay, T. L. 1989. Rendering fur with three dimensional textures. Computer Graphics (New York, N.Y.), 23 (3), 271–280. Read on 16.11.2023. https://doi.org/10.1145/74334.74361

Kim, T.-Y., & Neumann, U. 2002. Interactive multiresolution hair modeling and editing. ACM Transactions on Graphics, 21 (3), 620–629. Read on 17.04.2024. https://doi.org/10.1145/566654.566627

Krita Foundation & KDE. 2024. Krita. Version 5.2.2. [Computer Software]. Krita Foundation & KDE.

Magid, R. 1994. ILM Meets The Flintstones: Effects house uses computer technology to enhance cave dwellers' world. American cinematographer. 75 (7) 58. American Society of Cinematographers. Read on 23.04.2024.

Magnenat-Thalmann, N., & Thalmann, D. 2005. Virtual humans: Thirty years of research, what next? The Visual Computer, 21 (12), 997–1015. Read on 12.12.2023. https://doi.org/10.1007/s00371-005-0363-6

Marschner, S. R., Jensen, H. W., Cammarano, M., Worley, S., & Hanrahan, P. 2003. Light scattering from human hair fibers. ACM Transactions on Graphics, 22 (3), 780–791. Read on 11.12.2023. https://doi.org/10.1145/882262.882345

Moltenbrey, K. 2021. Clash of the Titans. Computer Graphics World, 07/2021, 18–26.

Monsters Inc. 2001. Direction: Pete Docter. Production: Walt Disney Pictures, Pixar Animation Studios. Production country: United States.

Mori, M., MacDorman, K. F., & Kageki, N. 2012. The Uncanny Valley [From the Field]. IEEE Robotics & Automation Magazine, 19 (2), 98–100. Read on 27.03.2024. https://doi.org/10.1109/MRA.2012.2192811

Netzley, P. D. 2001. Encyclopedia of movie special effects. New York: Checkmark.

OpenAI. 2024. Creating video from text. [Website]. Released on 15.02.2024. Read on 11.03.2024. https://openai.com/sora

Orange Turbine. N.d. Power up with Blender. [Website]. Read on 27.03.2024. https://orangeturbine.com/

Osipa, J. 2010. Stop Staring: Facial Modeling and Animation Done Right, Third Edition. 3rd ed. Indianapolis, Ind: Sybex.

Patrick, D., Bangay, S., & Lobb, A. 2004. Modelling and rendering techniques for African hairstyles. ACM International Conference on Computer Graphics, Virtual Reality and Visualisation in Africa, 115–124. Read on 09.11.2023. https://doi.org/10.1145/1029949.1029971

Price, D. A. 2008. The Pixar touch: the making of a company. 1st ed. New York: Alfred A. Knopf.

Robbins, C. R. 2012. Chemical and Physical Behavior of Human Hair. 5th ed. Berlin, Heidelberg: Springer Berlin Heidelberg. Read on 13.11.2023. https://doi.org/10.1007/978-3-642-25611-0

Roda, C. 2022. Real Time Visual Effects for the Technical Artist 1st ed. CRC Press. https://doi.org/10.1201/9781003009795

Rosenblum, R. E., Carlson, W. E., & Tripp III, E. 1991. Simulating the structure and dynamics of human hair: Modelling, rendering and animation. The Journal

of Visualization and Computer Animation, 2 (4), 141–148. https://doi.org/10.1002/vis.4340020410

Schwind, V., Leicht, K., Jäger, S., Wolf, K., & Henze, N. 2018. Is there an uncanny valley of virtual animals? A quantitative and qualitative investigation. International Journal of Human-Computer Studies, 111, 49–61. Read on 13.03.2024. https://doi.org/10.1016/j.ijhcs.2017.11.003

Siddi, F. 2021. Blender by the numbers – 2020. [Website]. Released on 07.07.2021. Read on 30.05.2024. https://www.blender.org/news/blender-by-the-numbers-2020/

Spencer, S. 2011. ZBrush Character Creation: Advanced Digital Sculpting. 2nd ed. Hoboken: John Wiley & Sons, Incorporated.

Steam Charts. N.d. Blender – Steam Charts. [Website]. Read on 30.05.2024. https://steamcharts.com/app/365670#All

Thomas, A., Harland, D. P., Clerens, S., Deb-Choudhury, S., Vernon, J. A., Krsinic, G. L., Walls, R. J., Cornellison, C. D., Plowman, J. E., & Dyer, J. M. 2012. Interspecies Comparison of Morphology, Ultrastructure, and Proteome of Mammalian Keratin Fibers of Similar Diameter. Journal of Agricultural and Food Chemistry, 60 (10), 2434–2446. Read on 13.11.2023. https://doi.org/10.1021/jf204811v

Thomas, F. & Johnston, O. 1995. The illusion of life: Disney animation. 1st Hyperion ed. New York: Hyperion.

Unreal Engine. 2020. The Making of ‘Meerkat’ Real-Time Hair & Fur Short | Spotlight | Unreal Engine. Released on 10.12.2020. [YouTube Video]. Watched on 12.12.2023. https://www.youtube.com/watch?v=0ixtpcqb85I

Unreal Engine. 2021. Hair at a Strand Level | Grooming your Creations in Unreal Engine. Released on 22.01.2021. [YouTube Video]. Watched on 12.12.2023. https://www.youtube.com/watch?v=__BScFPJy3E

Vogt, R., Marschner, S., Iben, H., Palmer, S., Doncaster, C., Ameziane-Hassani, O., Crowe, D., & Morand, G. 2017. State and future of hair in feature film and visual effects. ACM SIGGRAPH 2017 Panels, 1–2. Watched on 11.12.2023. https://doi.org/10.1145/3084847.3084848

Ward, K., Bertails, F., Tae-Yong Kim, Marschner, S. R., Cani, M.-P., & Lin, M. C. 2007. A Survey on Hair Modeling: Styling, Simulation, and Rendering. IEEE Transactions on Visualization and Computer Graphics, 13 (2), 213–234. Read on 07.11.2023. https://doi.org/10.1109/TVCG.2007.30

Williams, Richard. 2001. The animator’s survival kit. London: Faber and Faber.

Yan, L.-Q., Tseng, C.-W., Jensen, H., & Ramamoorthi, R. 2015. Physically-accurate fur reflectance: modeling, measurement and rendering. ACM Transactions on Graphics, 34 (6), 1–13. Read on 13.11.2023. https://doi.org/10.1145/2816795.2818080

Yang, X. D., Xu, Z., Yang, J., & Wang, T. 2000. The Cluster Hair Model. Graphical Models, 62 (2), 85–103. Read on 02.11.2023. https://doi.org/10.1006/gmod.1999.0518

Yuksel, C., & Tariq, S. 2010. Advanced techniques in real-time hair rendering and simulation. ACM SIGGRAPH 2010 Courses, 1–168. Read on 12.12.2023. https://doi.org/10.1145/1837101.1837102

Zeman, N. B. 2016. Essential Skills in Character Rigging. 1st edition. [Online]. Oakville: A K Peters/CRC Press.

## APPENDICES

Appendix 1. Fur simulation testing

Hautakangas, A. 2024. Thesis – Fur simulation testing. Presentation. Released on 12.12.2023. Updated on 17.04.2024. https://docs.google.com/presentation/d/1JlYKYGruESDychZNN_do_2zIXkphLMSq5Kg0J9w-02o/edit?usp=sharing

Appendix 2. Fur simulation – Achieving an appealing style

Hautakangas, A. 2024. Thesis: Fur simulation – achieving an appealing style. Presentation. Released on 21.03.2024. Updated on 25.03.2024. https://docs.google.com/presentation/d/1QTNemW8OC2NEZeQLBU3RidGUlttyN-fK-Kstm26cA3I/edit?usp=sharing

Appendix 3. Final rendered video of the project

Hautakangas, A. 2024. Stylized Bernese Mountain dog – fur simulation. YouTube Video. Released on 28.03.2024. https://youtu.be/a5VyxClgQbE