Kerttu Aalto

RUSHER - CREATING AN ENEMY DESIGN FOR A GAME

Bachelor’s thesis
Degree programme in Game Design

2019

South-Eastern Finland
University of Applied Sciences

South-Eastern Finland
University of Applied Sciences

Author (authors)	Degree	Time
Kerttu Aalto	Bachelor of Culture and Arts	March 2019

Thesis title
Rusher – Creating an enemy design for a game 35 pages

Commissioned by

Joensuu Games OSK

Supervisor

Brenda Jiménez Rodriguez

Abstract

The main purpose and objective of this thesis is to create an enemy design for a game project. It will go through the process of creating a 3D-model of an enemy called “The Rusher” for post-apocalyptic First Person Shooter game developed by Joensuu Games OSK.

The thesis includes discriptions of all necessary phases of character design for the particular enemy; concept art, modeling, sculpting, retopology, fur and import.
The first part of the thesis consists of introduction, which is followed by a short synopsis of types of enemies in games. The second part starts with reference materials, concept art and the modeling process, followed by sculpting and unwrapping. The last part is about creating a fur and importing the model to game engine.

The thesis was described and demonstrated with pictures of the work phases from the different programs used during the process.

The conclusion of the thesis is fully modeled, animation-ready enemy design that can be imported to the game and create a consistent description of the workflow for creating 3D-models of game characters.

Keywords

3D modeling, 3D sculpting, Retopology, Concept Art, Character Modeling, Enemy Design, Unwrapping, Texturing, Game Engine Optimization, Game Design.

CONTENTS

1 INTRODUCTION

When playing a video game, it is easily noted that almost every game has an enemy or enemies that the player will have to conquer in order to complete the game. It is of utmost important that the enemy fits the environment and overall appearance of the game to make it seem believable to exist in that particular setting.

The subject of this project is to create an enemy design for an existing idea for a game project by Joensuu Games OSK. It will be made using Blender and Substance Painter and imported to Unreal Engine. The project, named Rusher, is an enemy design for PK-170, a post-apocalyptic First Person Shooter (FPS) game.

The process as a whole in the project can be summarized in eight steps: Concept, Model, Sculpt, Retopology, Unwrap, Texture, Fur and Import (Figure 1). At the end, the purpose of the project is to create an animation-ready, detailed and fully textured enemy design with fur that can be used with the game engine’s physics.

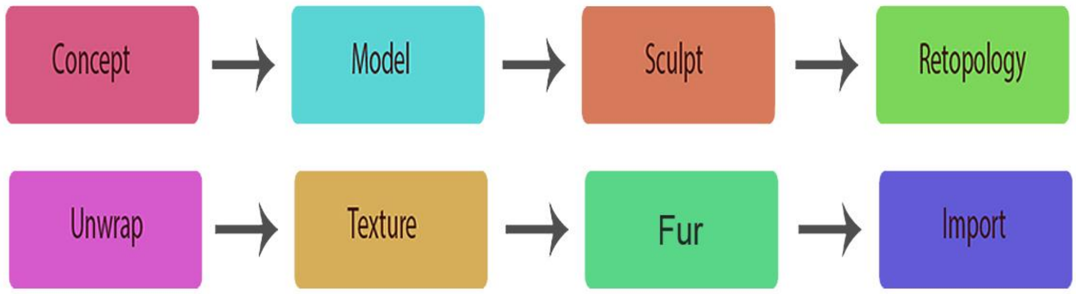


Figure 1. Project progress chart

This project will be concentrating on the design of the enemy, not behaviour or other gameplay-related elements. It will, however, be imported to the game engine and tested how the engine will receive it.

OPTIMIZING FOR GAME ENGINE

When designing a 3D-character for a game, optimizing is a very important element to consider, even before actually the creating progress. It is not only enough that the design looks good, but that the game engine can handle it effectively, keeping the game fast while having several models at once on display (Tshabangu 2016).

This is why before starting the project (and in process) there are certain aspects that need to be consider:

1. What is the maximum amount of polygons the model can have?
2. Is the model constructed so that it can be animated?
3. Can the model be made more low-poly than it already is without sacrificing too many details?

If the model is not optimized correctly, it can cause the game to run significantly slowly or crash it completely. A poorly optimized model is also difficult to animate, for example.

2 ENEMY DESIGN

2.1 What makes an enemy design

As mentioned before, most games have enemies. They are characters or creatures that are hostile towards the player have to be either destroyed or avoided in order to proceed in the game. Their design usually supports their behaviour; they might have imposing external appearance, signalling they are not friendly towards the player character. Enemies in video games can be roughly divided in to four major groups: basic, support, elite and boss (Bycer 2014).

Basic enemies are the most common, with very basic skills and they mostly just exist as a simple obstacle for the player. Support enemies have more slightly more skills, and are more challenging than the common enemies. Elite enemies are larger and can do more harm, but are still considered just common enemies. Bosses are the strongest, grandest and most unique enemies, usually reserved for special occasions, such as end of the level. They are usually fought one-on-one, with occasionally weaker common enemies beside them (Bycer 2014).

This information is important to remember, since the design of the enemy is largely based on its role in the game; common enemies that are seen multiple times during one playthrough should have more simplistic appearance than boss enemies that are going to appear only once. The more common the enemy, the more times it is going to appear in the game.

Although the basic idea of the game is that all enemies are going to be quite challenging due to the post-apocalyptic survival elements, The Rusher is categorized as a common/support enemy. It is something frequently seen in the game world, but it still poses a significant threat to the player. The design should have simplicity to reflect their role in the game, but should look threatening enough so that the player knows this is an enemy not to be underestimated.

2.2 Moodboard

The Rusher is an already existing idea created for the game. The aim was to create an enormous rat enemy living in the desolate nuclear wasteland. It was decided beforehand that it has a spotty fur and strong upper body. Other than that, everything else was to be left for the artist to decide.

Creating a mood board (Figure 2) was the first step. It is a collection of pictures which are used as an inspiration for the concept art, the next phase in the project. It is useful reference material for other than concept art, for example details and textures for the 3D-model.

Several pictures are of other game enemy designs with similar themes, most notably Rat ogre from Warhammer: Vermintide 2 (Fatshark 2018), Giant rat from

Kingdoms of Amalur: Reckoning (38 Studios 2012) and Mole rat from Fallout 3 (Bethesda Softworks 2008). There were also pictures of real-life vermins.

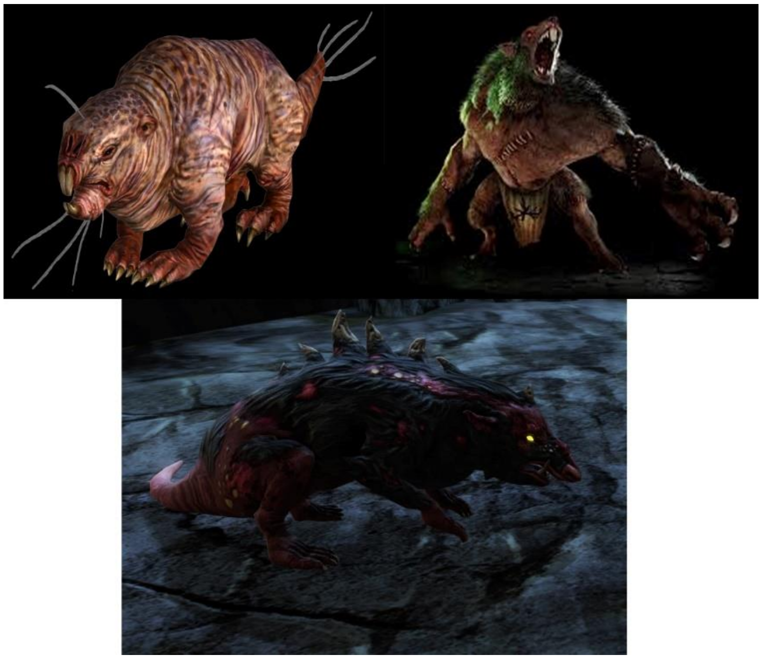


Figure 2. Moodboard: Mole Rat (Bethesda softworks 2008), Rat Ogre (Fatshark 2019) and Giant Rat (38 Studios 2012)

All the enemies from existing games and Rusher have several features in common; larger than normal size, ragged fur, visibly diseased skin, big teeth/claws and other abnormalities not usually seen in normal rats. These are very common design choices for vermin-type enemies.

3 CONCEPT ART

3.1 Detailed concept art

When the moodboard is established, it is used as an influence and reference the concept art. The main aspects in the concept art of Rusher should be the basic attributes (posture, shape) and the most important details (such as claws, teeth, scars etc.)

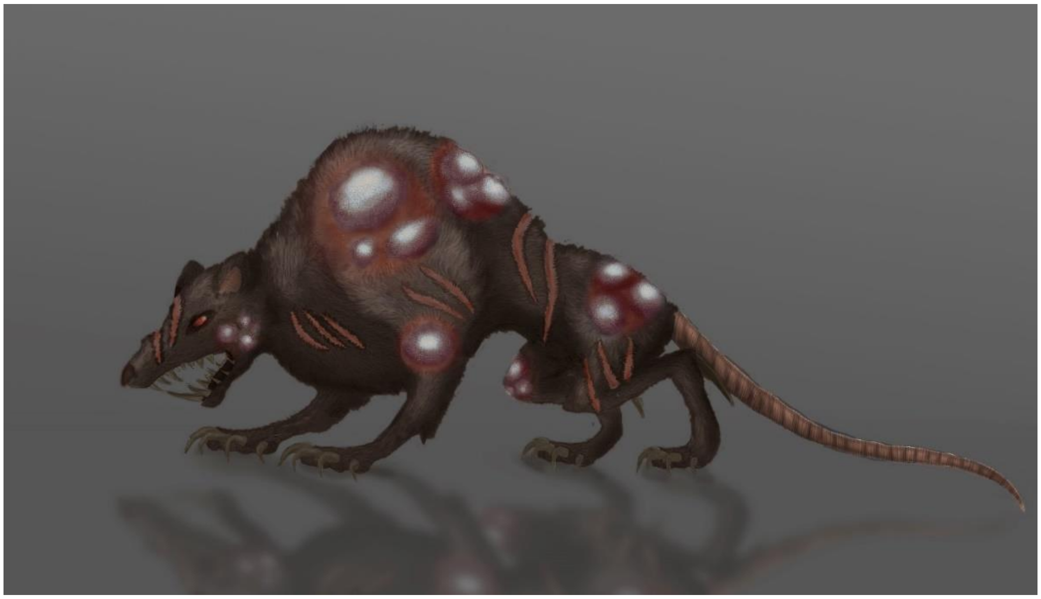


Figure 3. Concept art

The ready concept art (Figure 3) has a lot of the before mentioned aspects from the mood board references. There were some aspects that were changed from the original idea; the rusher has a lot more fur and it has darker colors. It also has a bit of a canine-inspired design to make it seem more intimidating.

3.2 Simplified guideline art

While the concept art is very detailed reference, it might be too complex to use as for 3D-modeling purposes, because it might be too distracting for optimal workflow. It also shows only the side perspective of the design, and for modeling, multiple angles of the intended subject are essential. Some modelers do sometimes work with only one reference (or none at all), but it is highly recommended to always have at least two, preferably more.

This problem can be solved by creating a simplified, line art-only version of the character from different perspectives (Figure 4). This step is not strictly necessary or the only effective way, but can help understanding and constructing the model.

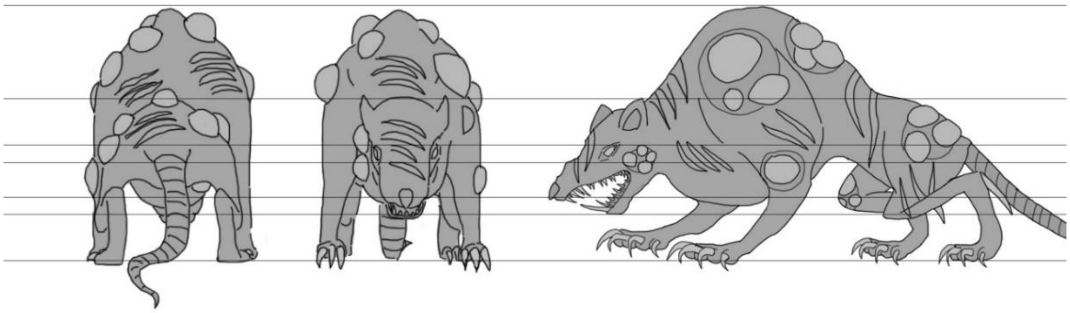


Figure 4. Modeling guideline drawings

Beside the line art, there are also two other elements notably present that were added later; color and the lines over the drawing. These are for further helping with the modeling progress by showing the details and scale. They should also help with keeping the model symmetrical on all sides. It also shows the places of the scars, tumors and other details.

4 MODELING

After the concept part, the construction of the basic model can be started. The basic 3D-model will be quite low-poly, which means the model has a small amount of polygons, making it lowly detailed but easy for the program to handle. It is going to be the base for the much more highly detailed higher poly version created later. The basic model should capture the basic structure and shape for sculpting details later on (Simons & Dinges 2013, 36).

The first part of this project is importing the reference images to Blender as background images. Before starting, it is to be mentioned that Blender has two viewports, perspective and orthographic. The orthographic port is more important of these two for now, because it allows the user to see reference images that are imported to the program.

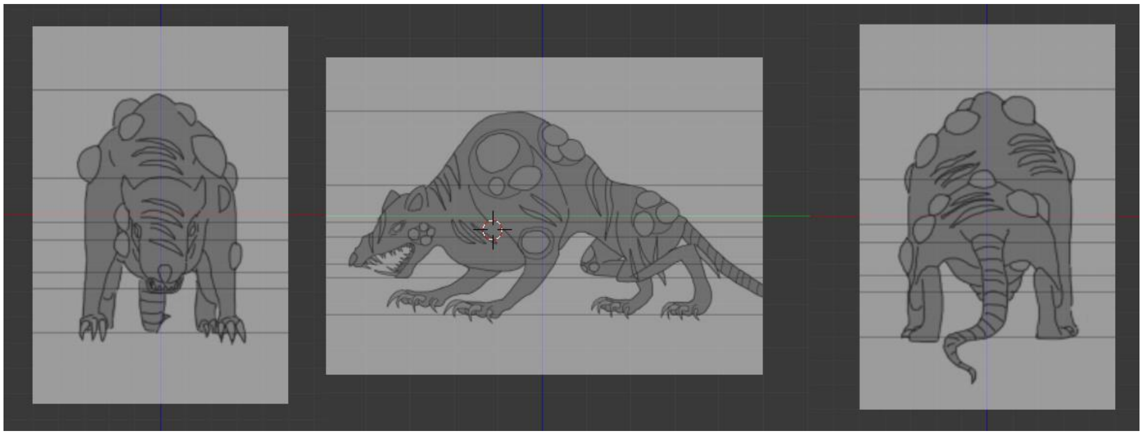
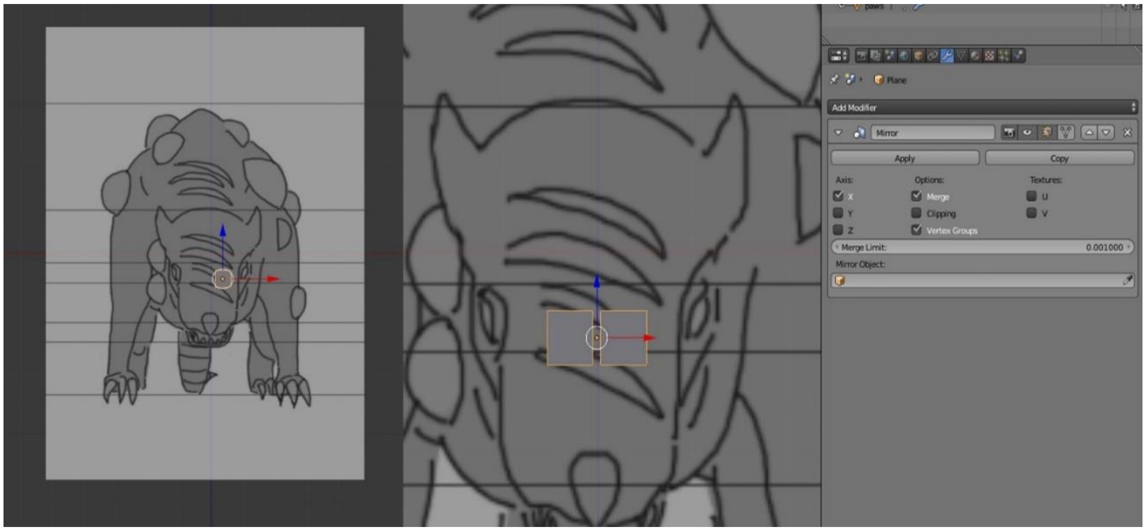


Figure 5. Viewports

Each background image is assigned its own viewport in orthographic mode (Figure 5). Because of this each reference picture will be seen from their respective view (in this case front, back and left). However, they will not be seen in perspective mode.

There are many ways to start building the model. There is no one right way to do it, and the best way to know what works for a 3D-artist is just to try different choices. In this project just a basic mesh plane is used as a starting point.



Mirror

Axis: ☑ X ☐ Y ☐ Z

Options: ☑ Merge ☐ Clipping ☑ Vertex Groups

Textures: ☐ U ☐ V

Merge Limit: 0.001000

Mirror Object:

Figure 6. Starting plane and mirror-modifier

The plane will be rotated 90 degrees so it shows up in front viewpoint and placed near the center of the picture. This will be the start of the model.

After the plane has been placed in the correct place, Mirror-modifier is activated (Figure 6). Modifier is a feature in most 3D-programs that adds different effects to the model in a way it does not corrupt the original model. It will project a duplicate of whatever is modeled on the chosen axis. It is helpful when creating a model, as it keeps it symmetrical and saves time because there is no need to model the other side of the model by hand.

The planes are also set in wireframe mode. It is a mode where the mesh is invisible besides the outlines. It is usually used when there is a need to see the possible reference image underneath.

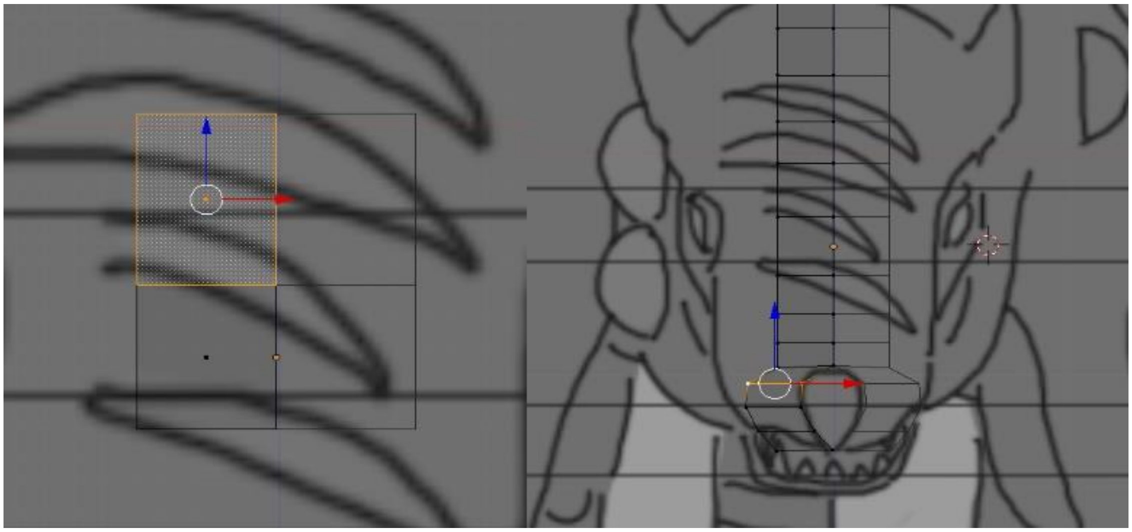


Figure 7. Extruding the basic shape of the head

When the plane is placed in the correct place, it will be extruded from the already existing plane. This will make the planes connected and create a unified mesh. The plane is extruded several times to the height of the head (Figure 7) to help create the general shape. The nose and other details will be added later, since at this point the simple shape of the head is more important.

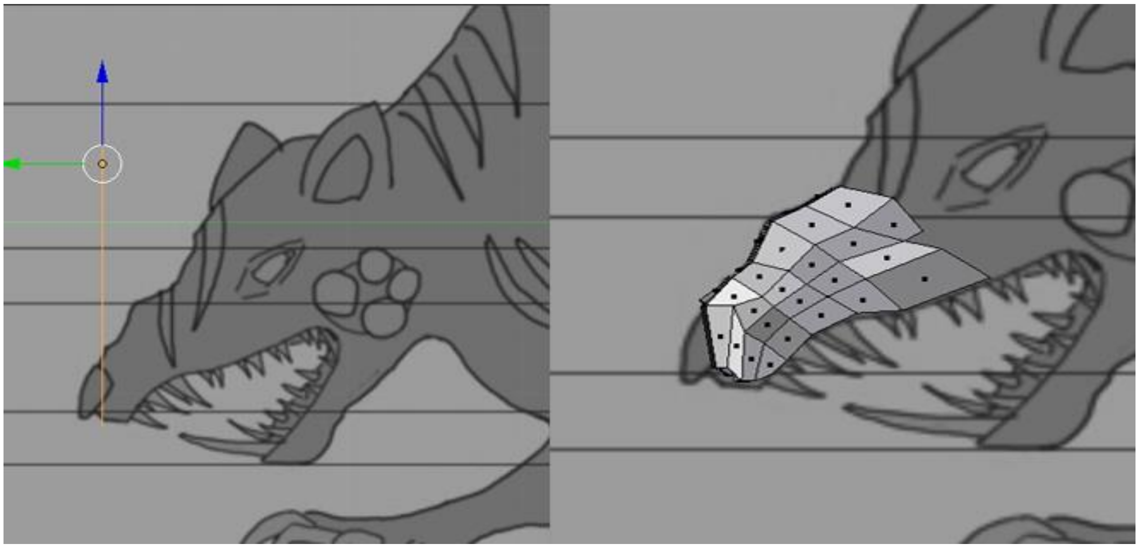


Figure 8. Arranging the mesh to match the left viewport

When the height is correct, the next step is to go to the left side view and arrange the mesh so it matches. When the planes are in correct order in both views, they can be extruded again. With extruding and arranging the mesh planes, the model will start to have a shape matching the reference pictures (Figure 8). Everything else for the model is created in a similar way, including the details such as ears and paws.

The model is starting to have the shape desired, and it is time to add another modifier to it, Subdivision Surface. The modifier is used to subdivide the mesh in to smaller faces giving it a smoother appearance (Figure 9). In the model, the amount of subdivision is on level 2. The amount of subdivision is basically limitless, but it is not recommended to go over 3 because of performance issues.

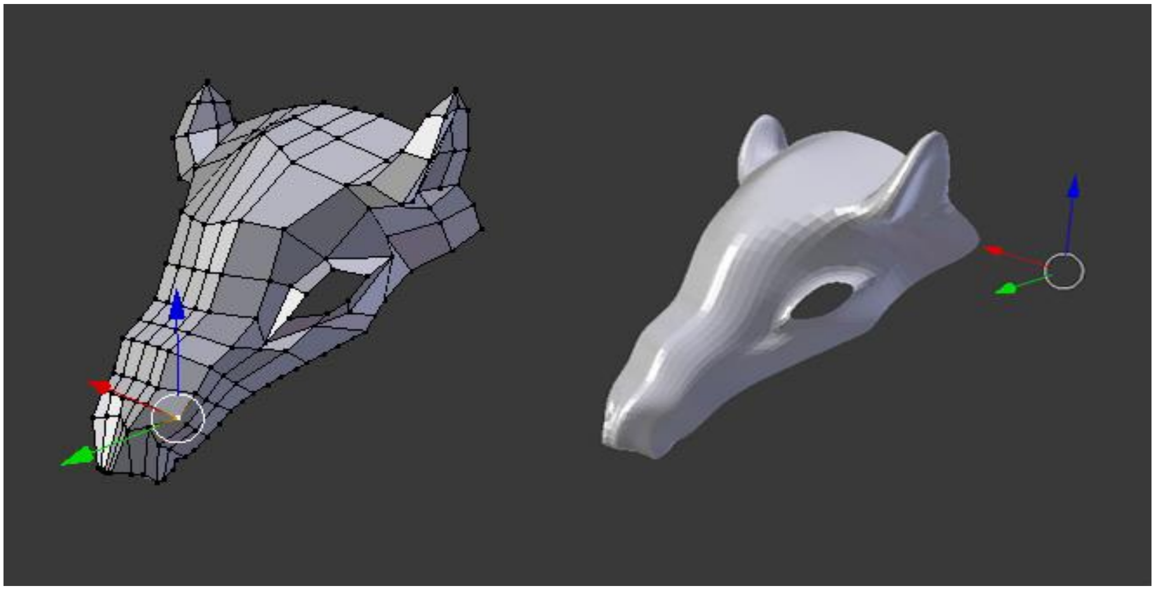


Figure 9. Subdivision Surface

Rest of the body is modeled in a similar way, using extrusion and occasionally other tools, such as the loop cut tool that can create new edge loops to the model.

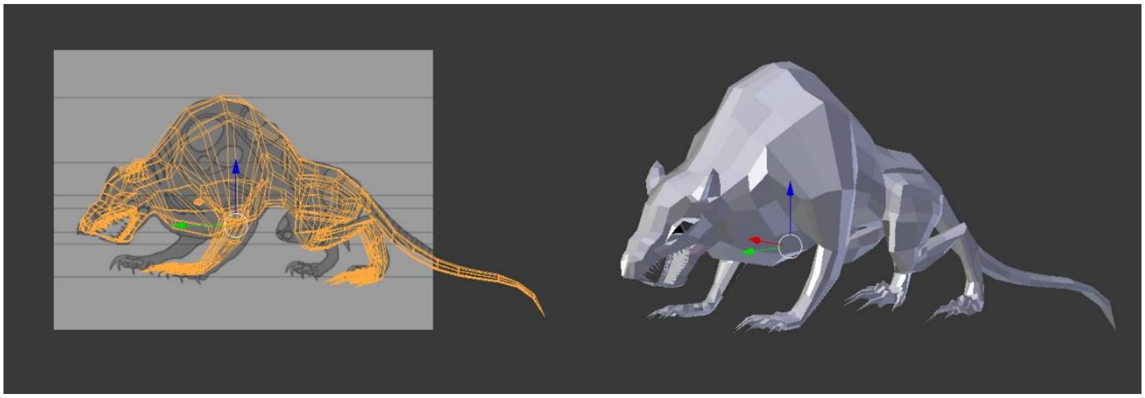


Figure 10. Wireframe and solid model

Also some parts of the model (such as feet) were modeled individually and connected together in the end to ease the progress. Subdivision surface and mirror are added to the model, resorting in a simplified, and quite low-poly version of the design that is ready for sculpting (Figure 10).

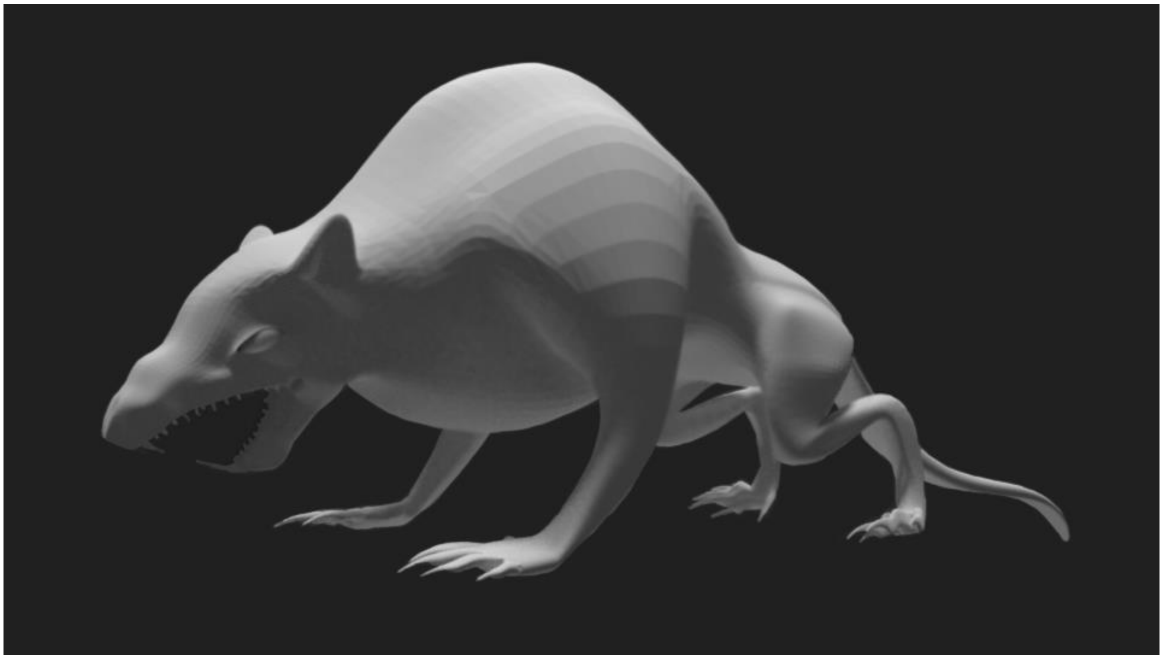


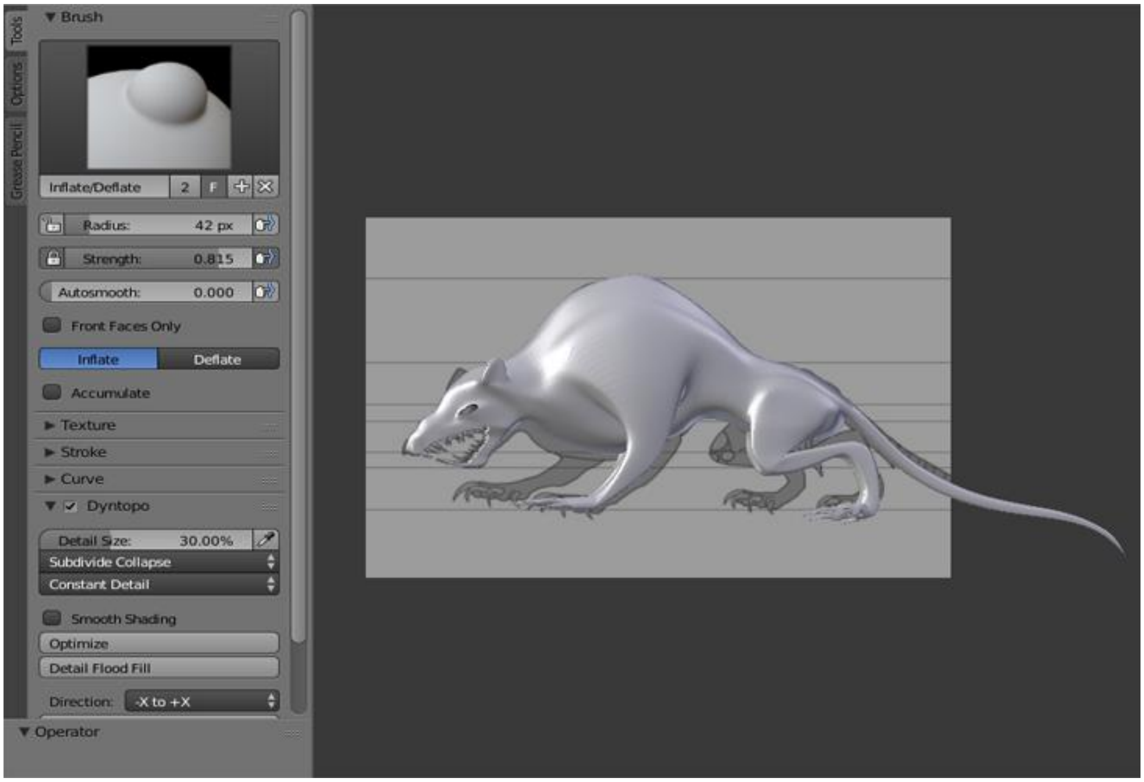
Figure 11. Rendered model

The mesh is rendered to have an idea what it will look like (Figure 11). Render is a term used for a 2D-image of the model created from the 3D scene. What the final image looks like depends on various variables, such as scene's camera angle, created light sources and materials, which is definition of the quality the model is made of, attached to the model (Blender 2017).

5 SCULPTING

After the basic low-poly model is constructed, the more complex designs (such as muscle, scars and other details) can be started to add to the model by sculpting. Sculpting is a method of shaping the model in a direct and intuitive way like when shaping clay, and many 3D-artists find sculpting to be more natural way of creating a model rather than moving individual vertices and polygons (Mullen 2012, 87).

Sculpting can be used to create whole models from the start, but because in many scenarios, including this particular case, the model was already pre-created by polygon modeling, sculpting tools will only be used to adding details that would be difficult to add with basic modeling techniques.



Radius: 42 px
Strength: 0.815
Autosmooth: 0.000
Detail Size: 30.00%
Subdivide Collapse
Constant Detail
Direction: -X to +X

Figure 12. Sculpting start

Blender does lack some advanced methods from some high-end programs that specialize in sculpting, but is very useful because it is integrated to the program’s other tools (Mullen 2012, 89). Therefore, it is highly recommended to be used for when the model only needs some basic sculpting (Figure 12).

The sculpt mode has a lot of different brushes to choose from, each one with their own unique function. This same window also has the brush properties, such as radius, strength, smoothing and add/subtract. There are also other attributes that can be customized, such as stroke and curve that have an effect on the results.

Such as in the polygon-modeling, modifiers can also be used in sculpting mode. In this project’s case, it was not used because of the model’s asymmetrical design. In any other case, this can save some time and effort when creating/modifying a model via sculpting.

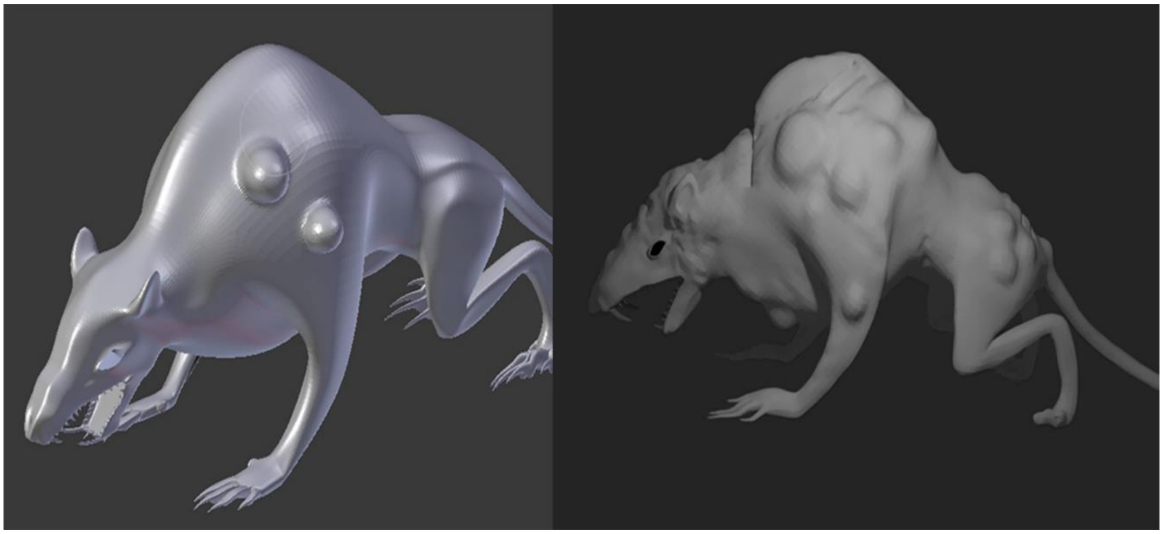


Figure 13. Brush example (Blob) and finished result

In this project, the brushes mostly used beside the most common Draw were Blob, Inflate/Deflate and Grab; Blob was used to create tumors, Deflate scars and Grab visible muscles or other details (Figure 13). After this work phase, the model has all the details needed.

6 RETOPOLOGY

6.1 The importance of retopology

The model is now modeled and sculpted, and it is looking highly detailed. However, the model has around 300000 polygons in it, making it high-poly as a result. While this is still a manageable amount, it is quite heavy on the programs and could be made more low-poly for better performance. This process is called retopology.

Retopology is a method where a new geometry is created to an already-existing mesh while using the original as a model, maintaining its basic shape to create a more functional topology (Simons & Dinges 2013, 95). It will be done in top of the original model with snapping it to its mesh.

Lower polycount and cleaner topology are important to a model, especially for the purpose of rigging and animation; a sculpted model is usually too high-poly and lacks clear topology and cannot be used.

6.2 Retopology of the model

When the sculpted project was ready, the retopology was to be done with a program that would instantly create a new topology for sculpted mesh. When a program like this was not found or proved unsuccessful in the context of the task, the only option was to create the topology by manual means.

The first part of the retopology progress is to create a single mesh. This will be the starting point of the new topology. There is no absolute rule what sort of mesh is acceptable, but the simplest and easiest to work with is plane (Figure 14). It is then rotated to be on the same level as the surface of the original model; this will be the starting point.

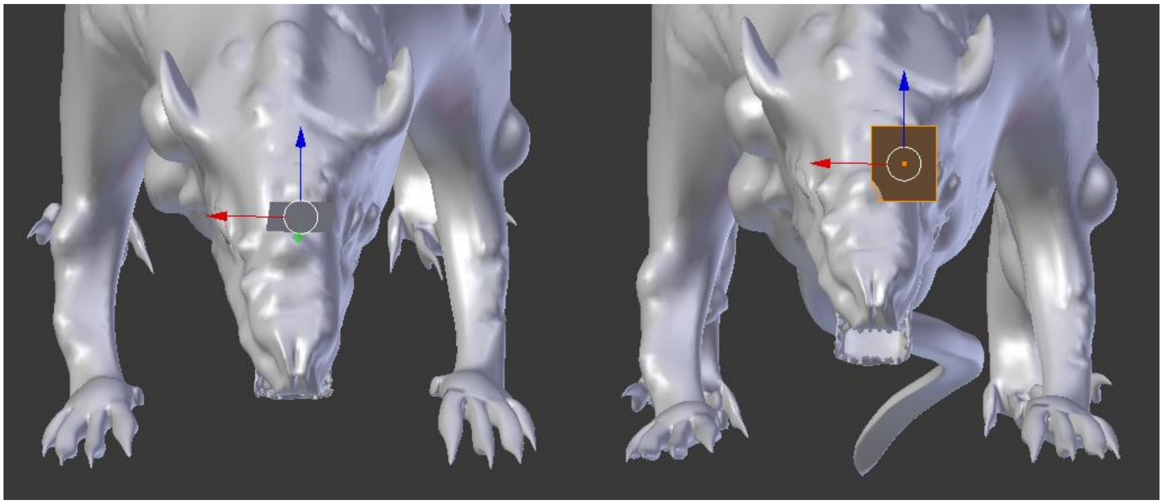
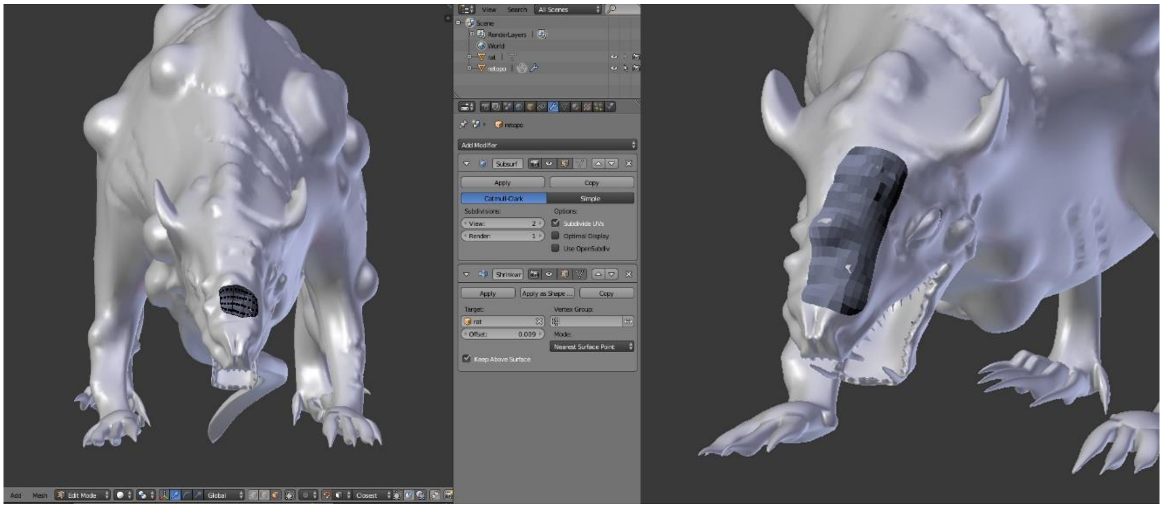


Figure 14. Mesh plane

Next, the snapping tool will be enabled. With snapping and the setting to project individual element to the surface of the object this will make the mesh plane snap to the sculpted model's surface.

Modifiers are an essential part in retopology, particularly Subdivision Surface and Shrinkwrap that in all simplicity does the same as the snapping tool to the sculpted mesh when chosen as a target. The major difference is that Shrinkwrap also uses subdivisions created by Subdivision Surface-modifier (CG Boost 2017).



Subsurf

Catmull-Clark

Subdivisions:
View: 2
Render: 1

Options:
☑ Subdivide UVs
□ Optimal Display
□ Use OpenSubdiv

Shrinkwrap

Target: rat
Offset: 0.009
Vertex Group:
Mode: Nearest Surface Point
☑ Keep Above Surface

Figure 15. Retopology Modifiers and resulted topology

As in the case of the basic 3D modeling, the mesh is extruded and it will snap automatically to the surface of the high-poly model (Figure 15). Often used mirror-modifier is not added to the list of modifiers because of the asymmetrical design of the original model, as that would make it symmetrical and therefore inaccurate. The finished topology, lacking some details such as claws and teeth, does demonstrate the finished result and the fundamental idea of the progress (Figure 16).

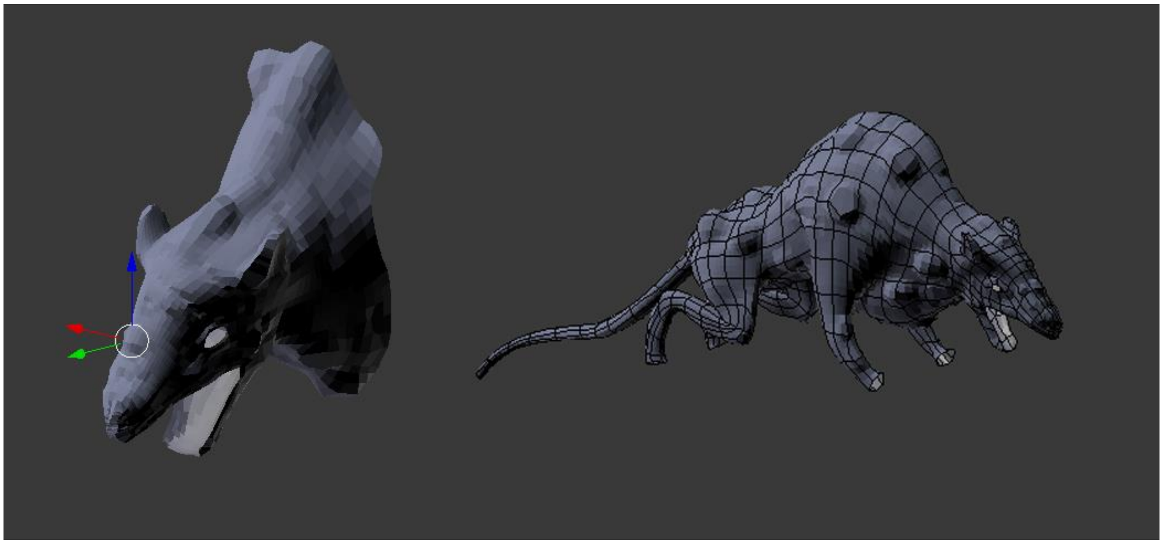


Figure 16. Retopology head and body

The most effective way to work with retopology is to try creating the same details as in the original with larger polygons (Mullen 2012, 104). Rationing polygons is vital in the stage of retopology.

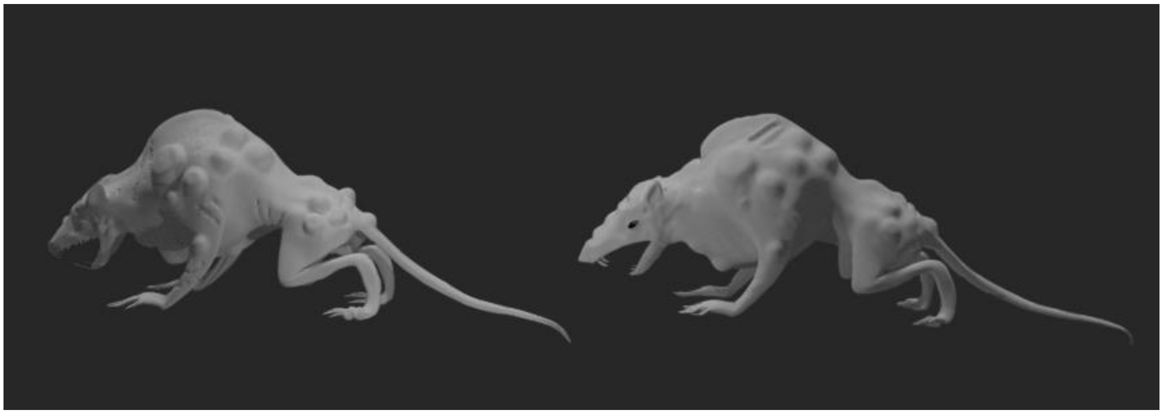


Figure 17. High-poly and low-poly model comparison

The retopology should not differ from the original model, aside from lower polygon count. Some minor changes might happen during the progress, if it is necessary. The comparison shows the exact nature of this information; some small changes, such as the shape of the nose and the neck were transformed to suit the lower-poly model (Figure 17).

7 UNWRAPPING AND TEXTURING

7.1 Unwrapping

With the new improved topology, the next part of the process is unwrapping the model for texturing.

The unwrapping progress starts with marking the seams. The easiest and the most common way to explain unwrapping is to imagine the model is made of paper; the seams are cutting the mesh, making it a flat shape (Blender Guru 2017). This is a very simple but effective rule when preparing a model for unwrapping.

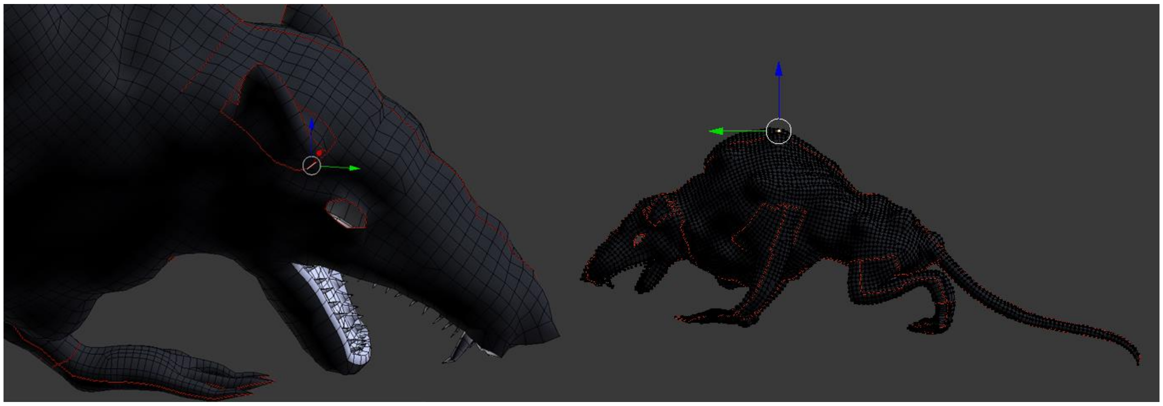


Figure 18. Seam markings

The mesh has seams in all the edges that are necessary such as legs, ears, torso and neck (Figure 18). Also the mouth is a different mesh all together and will be unwrapped and textured separately, and later connected to the main mesh. When unwrapping a model, there should be the least amount of seams as possible, because too many seams can make the model too complex to texture (Blender Guru 2017).

7.2 Texturing

The unwrapped model is displayed in UV/Image editor (Figure 19). It has a placeholder texture created by Blender. It is meant to give the idea of how the unwrapped model works as a texture.

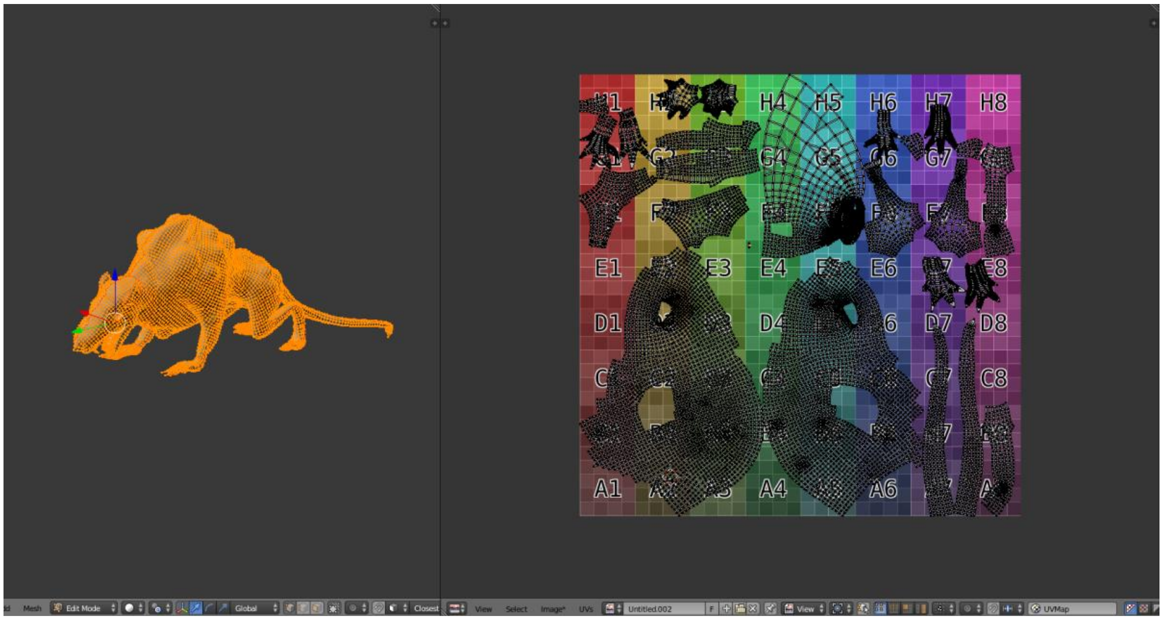
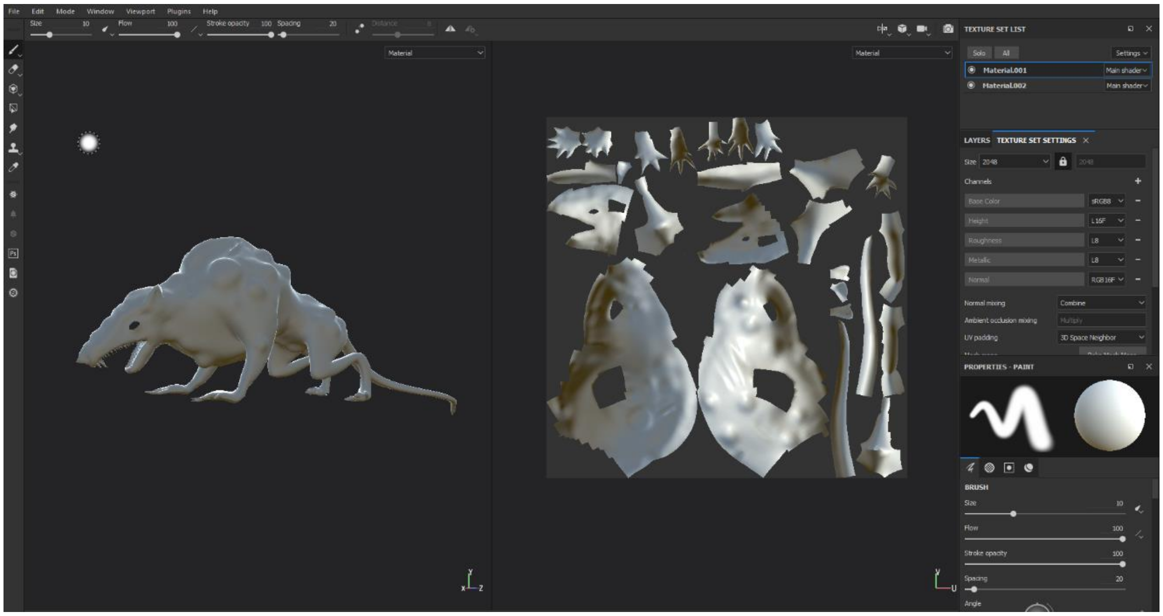


Figure 19. Unwrapped mesh

The model is going to be textured in Substance Painter. It is a program that allows the user to import the original meshes beside the UV maps for texturing and rendering (Allegorithmic 2019). This provides an opportunity to paint directly on to the mesh. This will create maps, a two-dimensional image that is applied to the surface of the model to add texture, color or other details (Slick 2019).



Material.001 Main shader
Material.002 Main shader

Size 2048
Channels
Base Color sRGB8
Height L16F
Roughness L8
Metallic L8
Normal RGB16F
Normal mixing Combine
Ambient occlusion mixing Multiply
UV padding 3D Space Neighbor

Size 10
Flow 100
Stroke opacity 100
Spacing 20
Angle

Figure 20. Overview of the model in Substance Painter

Substance Painter projects the mesh both in 2D (with UV map) and 3D view (Figure 20). Both views can be used to paint on, and it is mostly a matter of preference which one is used. It can be observed that the program registers all changes to either one viewpoint in both of them (Figure 20).

The process for creating the basic color map, known also as diffuse map (Polycount Wiki 2015), is explained simply as just painting over the mesh with color that was needed for each part (Figure 21). Substance Painter has various different brushes to use from, each with unique properties and textures. However, diffuse map is only the basic color of the model and does not include any sort of information about for example roughness and height of the mesh. It is basically a flat image attached to the surface of the model (Blender Guru 2015).

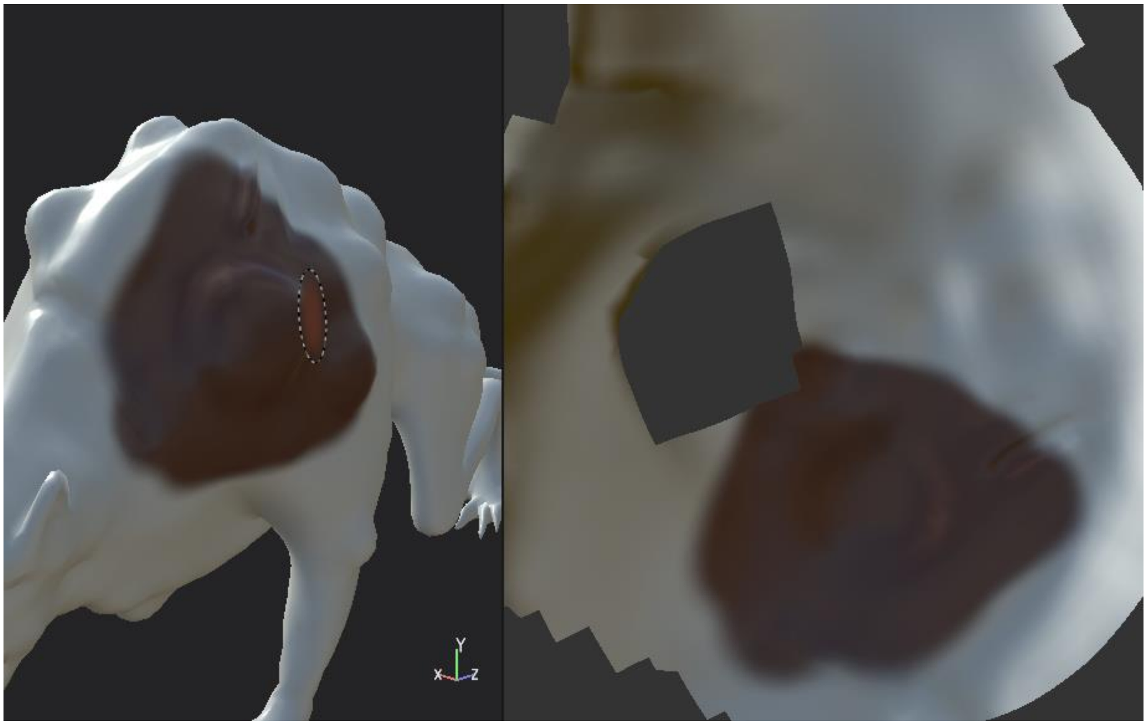


Figure 21. Demonstration of the brush tool

Aside from the diffuse map that was created, the project is also going to need at least four different maps; normal, height, roughness and mixed.

All of these maps serve a different purpose in the mesh (Figure 22); normal map is a map that is a base on what the surface of the model looks, recreating the details of the surface that could not be achieved by modeling or sculpting (Slick 2019). It is created to give the model a high-resolution looking surface without being high-poly, height map is to control the appeared height of polygons by marking them in black and white to mark the high and low points, creating an elevated surface. Roughness gives the surface of the model the representation of the material of the surface and mixed map shows how the light is reflected on the surface of the model (Polycount Wiki 2015).

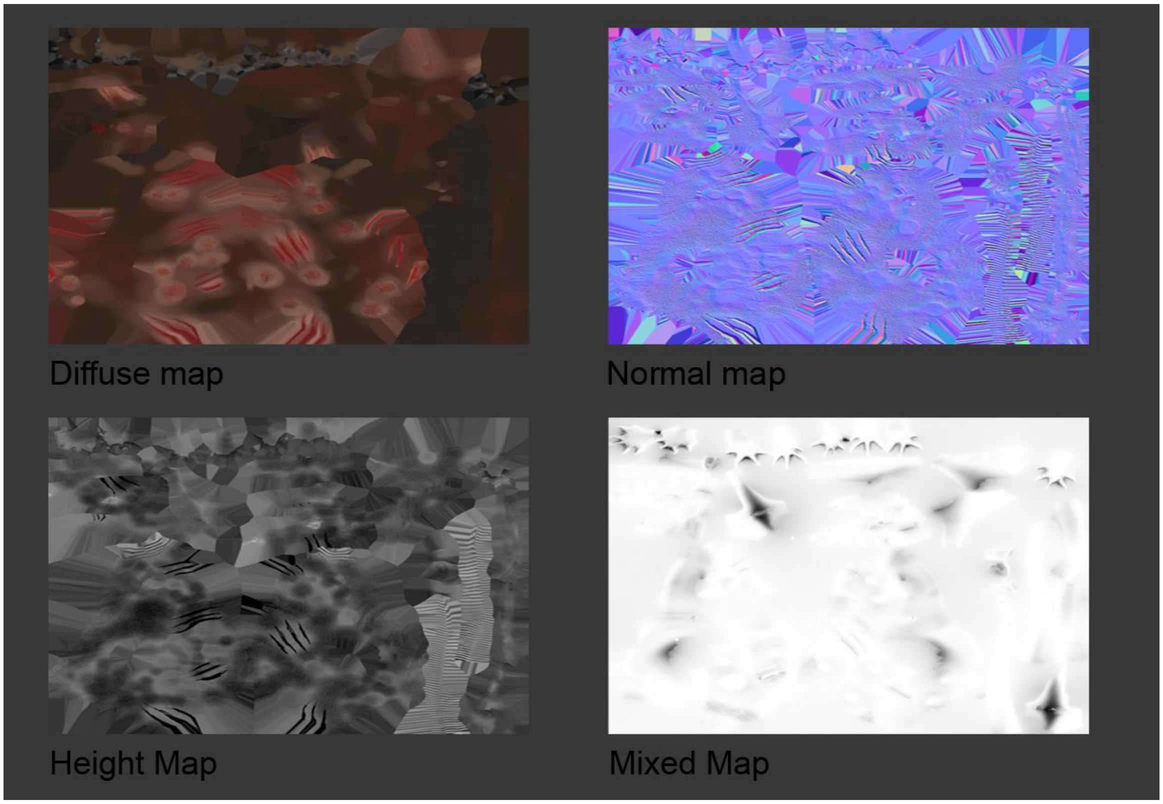


Figure 22. Different texture maps

There are two ways of preparing these maps; texture baking and texture painting. Baking is a process where details from a high-poly model are transferred in to a lower-poly model via texture maps. It creates the map automatically by projecting the high-poly in to the low-poly mesh. Most texturing programs, such as Substance Painter, have support for this method.

In the case of the mesh of the rusher, texture painting was used to create the maps. It is a method were the normal map is usually created in 2D painting software (such as Photoshop) manually. In Substance Painter, this work phase was made possible by layers specifically created for different maps. Such as with diffuse, the map was directly painted on to the mesh of the model. The only actual map that needed to be created was height map, and from that the program created automatically the rest.

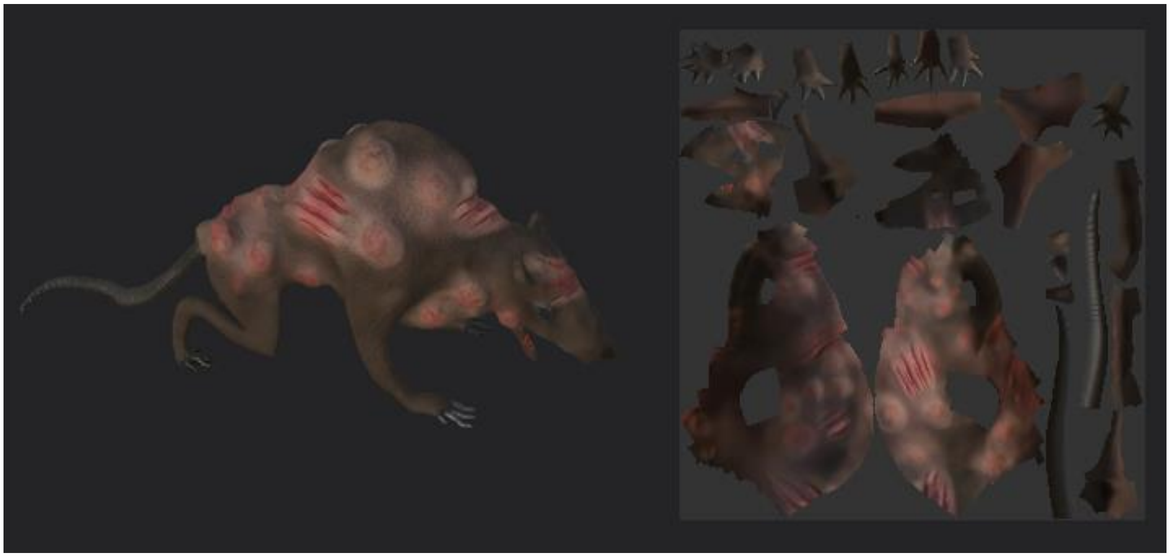


Figure 23. Ready texture

When the model is textured in Substance Painter (Figure 23), it will be exported back to Blender. The textures are saved as individual PNG-files.

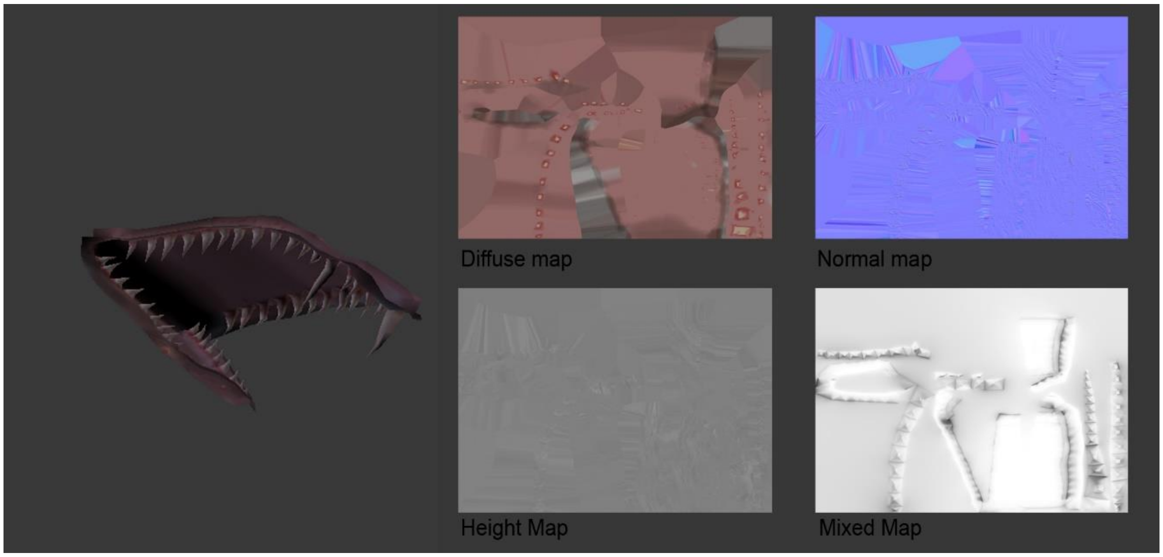


Figure 24. Mouth texture maps

As the mouth was a separate mesh, it was textured individually (Figure 24). It was later imported back to Blender as a combined file with the rest of the body and attached to it, while keeping its original textures contained.

7.3 Node editor

The textures are going to be imported back to Blender, and attached to the model with the Node editor (Figure 24). The editor makes it possible to attach all the texture maps to the model, making them stack and make the model look more detailed.

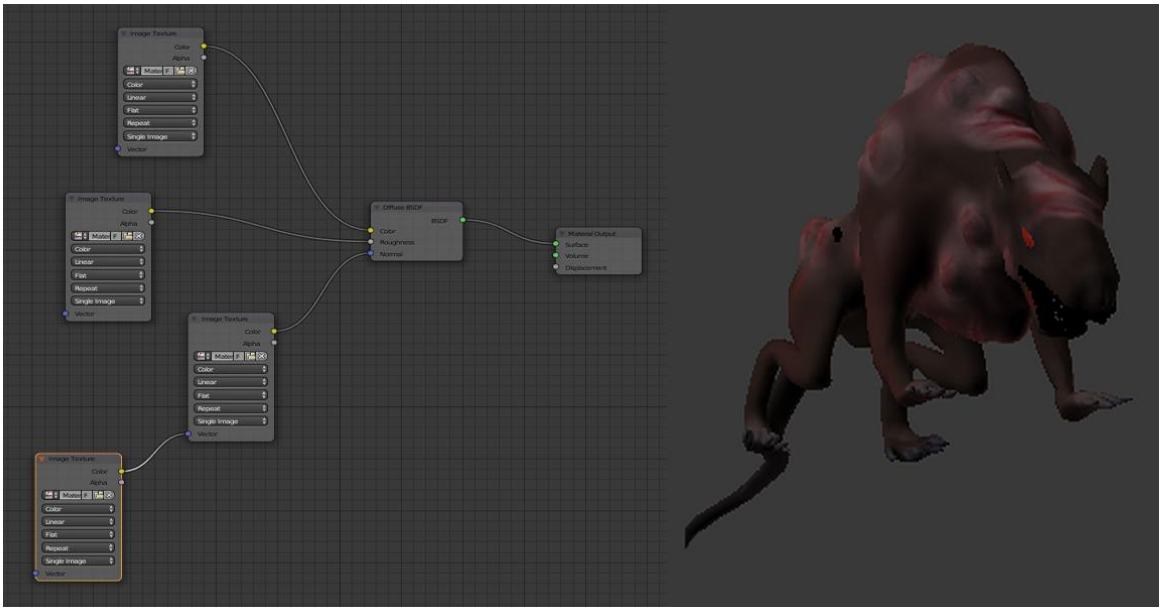


Figure 24. Node editor

The first texture map, diffuse, is attached to the Diffuse Bidirectional scattering distribution function-node (BSDF). This node is usually automatically created when the model is first opened in node editor. The other already existing node is Material Out that displays the materials and textures on to the model (Blender 2017).

Diffuse BSDF inputs the color, roughness and normal maps. They are to be attached to the Diffuse BSDF in their corresponsive places; diffuse to color, roughness to roughness and height to normal. Normal texture map is used for normal, but height can be used instead, because it has similar attributes. Normal map is not needed if height map is created.

8 FUR

The concept art of Rusher shows that the creature has a long and scraggly fur. It needs to be created to look and act like actual hair. In situations like this most 3D programs have already existing system for creating hairs.

Since the project is going to include hair that reacts to possible physics in the game (such as wind, gravity etc.), it needs an additional program for that. The model needs to be exported to a different program in order to do that. Blender can also create hair via particle system, but it does not have an official plug-in for the particular program about to be used.

The program in question is NVIDIA Hairworks. It enables users to simulate and render professional-quality hair and fur (NVIDIA 2016), and is often used in major game releases.

The model needs to be exported out of the program as an FBX-file in order to import it to first Autodesk 2018 and after that to Hairworks. FBX is a file form that transfers the geometry, animation, texture and material information between most 3D-applications (Plowman 2016, 75). It is imported to Unreal Engine 4 as a 3D-model, but it is also used in Hairworks as a base for the fur.

Because of the need to use the official plug-in, the model is transferred to Autodesk's 3DS Max 2018. The first part of the process is adding a Skin-modifier, as it needs to be attached to the skeleton of the model. Skeleton is a term used from a bone system that is jointed linkage of bone objects that are used in animation. A skeleton is necessary in animating, especially if the model has continuous skin mesh (Autodesk Knowledge Network 2014).

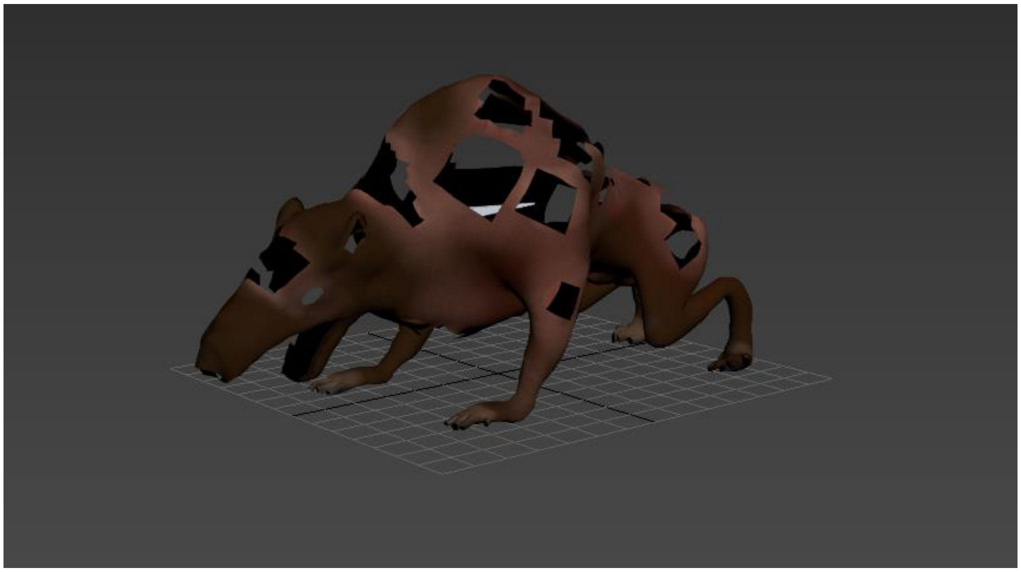


Figure 25. GrowthMesh

Since Rusher does not have animation, it is not going to have a whole skeleton attached to it. Instead, it uses a single bone for skinning. The model is also modified to remove the parts that are not meant to have hair in them (mouth, tumors, etc.). The mesh is only used as a base for the hair and does not need to have all parts of the mesh attached to it. This simplified mesh is called GrowthMesh (Figure 25), a copy of the original mesh that is for strand generation (NVIDIA 2016).

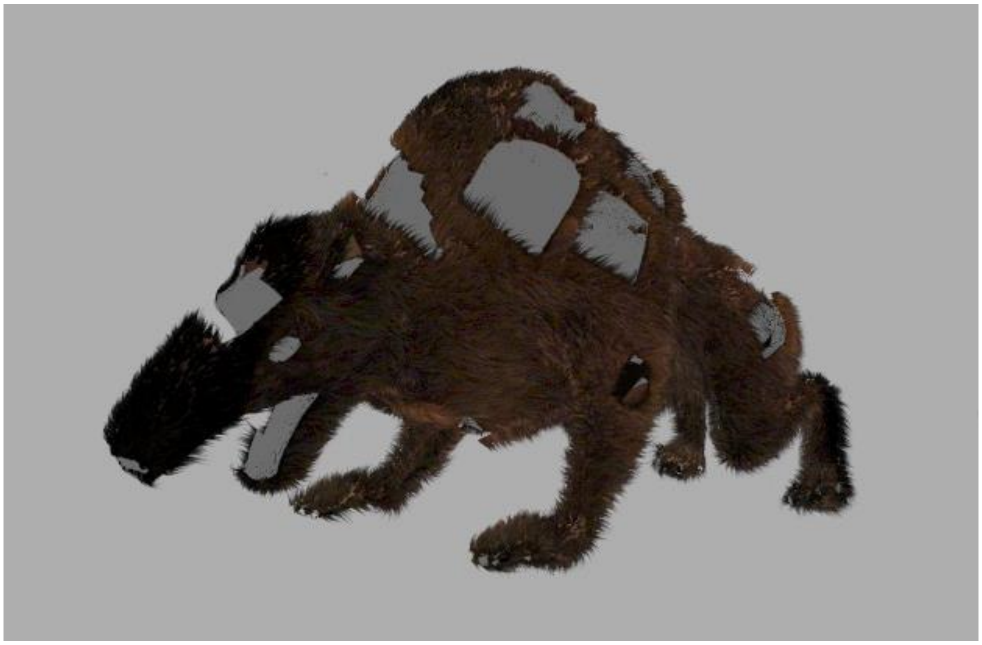
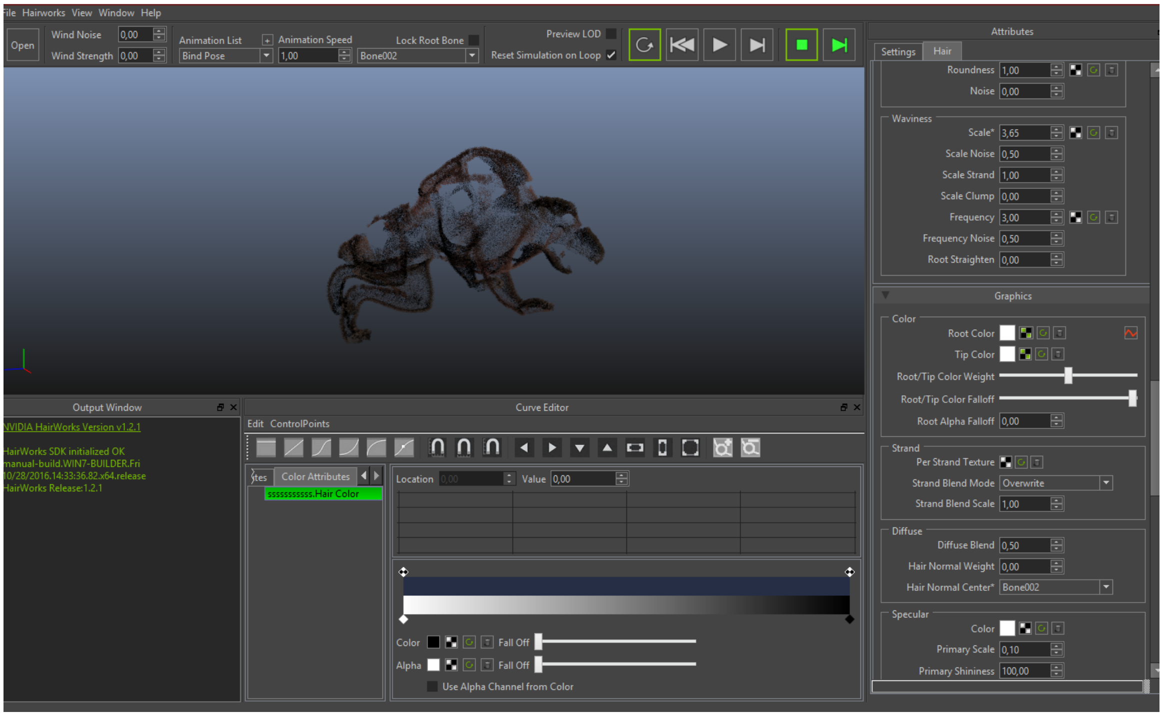


Figure 26. Hair and Fur-modifier on GrowthMesh

In 3DS Max, the Hair and Fur (WSM)-modifier is added to the model (Figure 26). This creates a hair to the mesh that cannot be selected but can be edited with the various tools presented in the program (Autodesk 2018). The fur is changed to what the desired look is. It needs to be converted from hairs to guides so the program can read it. The resulting guidelines are used in the Hairworks-modifier.



Wind Noise 0,00
Wind Strength 0,00
Bind Pose
Animation Speed 1,00
Lock Root Bone □
Bone002
Preview LOD □
Reset Simulation on Loop ☑

```
NVIDIA HairWorks Version v1.2.1

HairWorks SDK initialized OK
manual-build.WIN7-BUILDER.Fri
10/28/2016.14:33:36.82.x64.release
HairWorks Release:1.2.1
```

sssssssssss.Hair Color
Location 0,00
Value 0,00
Color ■
Fall Off
Alpha □
Fall Off
□ Use Alpha Channel from Color

Roundness 1,00
Noise 0,00
Scale* 3,65
Scale Noise 0,50
Scale Strand 1,00
Scale Clump 0,00
Frequency 3,00
Frequency Noise 0,50
Root Straighten 0,00
Root Color □
Tip Color □
Root/Tip Color Weight
Root/Tip Color Falloff
Root Alpha Falloff 0,00
Per Strand Texture
Strand Blend Mode Overwrite
Strand Blend Scale 1,00
Diffuse Blend 0,50
Hair Normal Weight 0,00
Hair Normal Center* Bone002
Color □
Primary Scale 0,10
Primary Shininess 100,00

Figure 27. Hairworks-modifier and Viewer

Adding the Hairworks-modifier enables the exporting of the guidelines as HairWorks-file (.APB, .APX) in to Hairwork’s Viewer (Figure 27), along with the FBX-file of the model. The hair can be further modified in the viewer in the same basic manner than in 3DS Max. After the exporting, it can be resaved as a combined FBX /Hairworks-file and send to game engine for further customization (NVIDIA 2016).

9 IMPORT TO GAME ENGINE

The model is imported to the game engine, Unreal Engine 4, for a final look. It contains the textures and possible other information, such as animation.

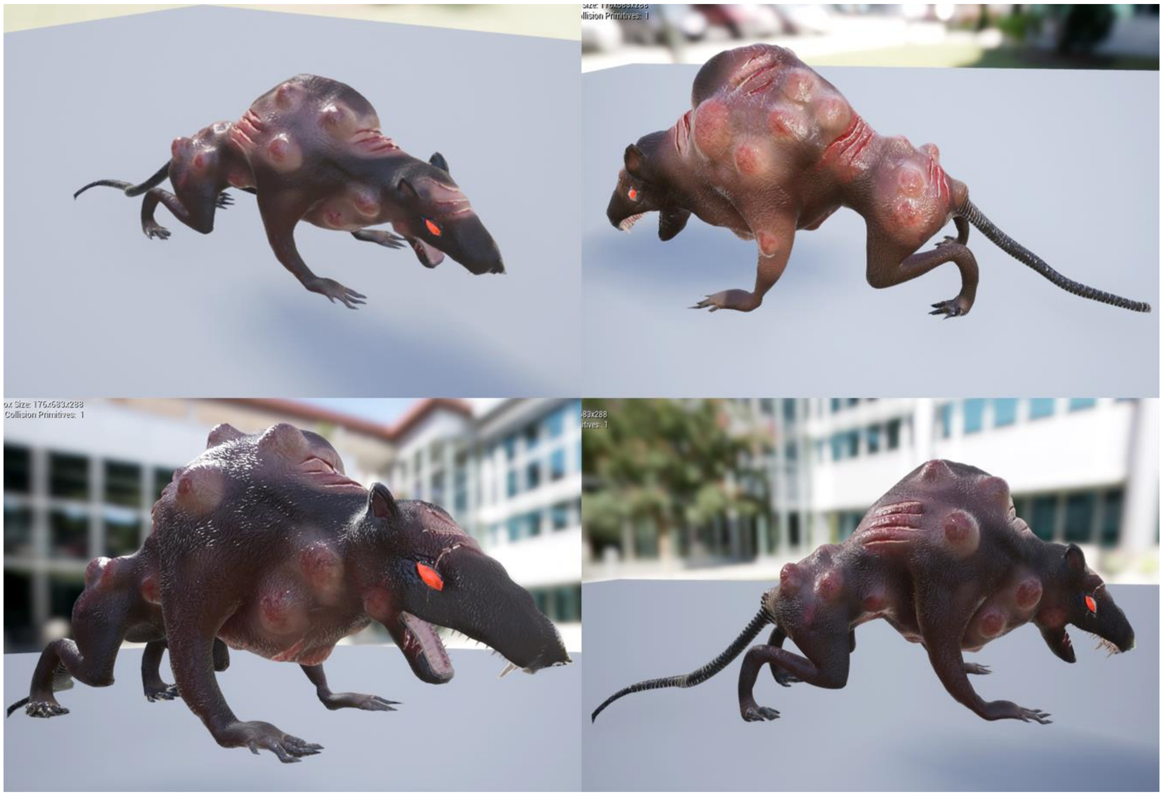


Figure 27. Unreal Engine 4 import

Unreal Engine 4 does not support NVIDIA Hairworks automatically, but needs to be modified in order for it to receive the export. The original source code is needed in order to achieve the result needed. Unfortunately the particular part of the project was removed from the thesis because of the technical difficulties and time restrains. The original model with textures was, however, imported as a test on how it would look in the engine (Figure 27).

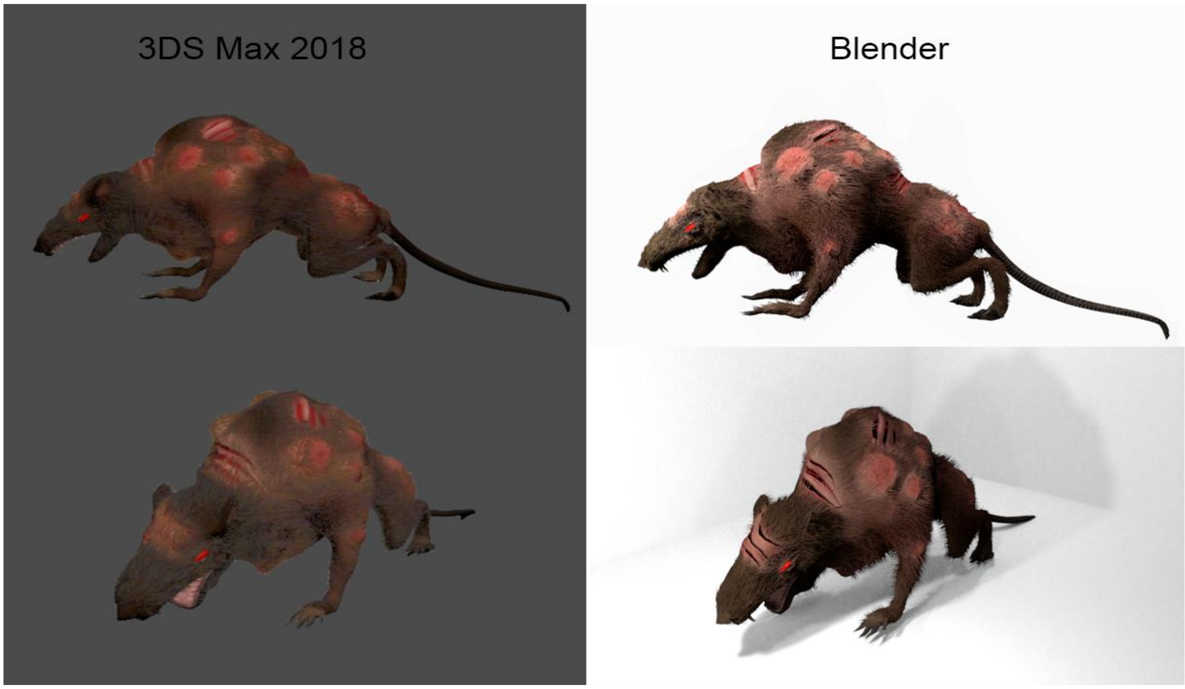


Figure 28. Blender and 3DS Max renders

Aside from the engine export, the model was rendered in the programs it was created, Blender and 3DS Max (Figure 28). The renders give the basic idea of how the final version of the model is supposed to look, even if it was not achieved in the game engine.

10 CONCLUSION

Creating a 3D-model of a character for a game project is a combination of many aspects. The process is a sum of various work phases, each having their own complexities. It is not sufficient that the character looks good, but whether it is functional and correctly optimized, otherwise it is going to be problematic at a later stage of the working progress. This is why it is highly advisable to plan ahead on a project.

The basic understanding of the programs is one of the most essential skills to have in a project such as this, since it is important to know what program to use in which part of the progress.

The author did successfully model, sculpt and texture a character, although the last part of the project that included the import of the model to the game engine with fur-physics was erased for the amount of game engine-related issues. The possibility that this issue could have been handled better with better time management skills exists, but as it was just a minor element from the project as a whole, it was relatively acceptable.

The process as a whole was useful for the author, since it provided a great amount of practical experience on all the aspects of the project. It gave a new insight on how the process of creating 3D-models for a game, and a basic guideline for projects of this nature.

REFERENCES

Bycer, J. 2014. Three Components of Great Action Game Design: Enemy. Available at: http://game-wisdom.com/series/three-components-of-great-action-game-design-enemy [Accessed 3 March 2019]

Tshabangu, V. 2016. Optimizing 3D Models for games. Available at: https://vtsdemozone.wordpress.com/2016/02/18/optimizing-3d-models-for-games/ [Accessed 2 April 2019]

Mullen, T. 2012. Mastering Blender. New York: Wiley.

Plowman, J. 2016. 3D Game Design with Unreal Engine 4 and Blender. Birmingham: Packt Publishing.

Simons, B., Dinges, T. 2013. Blender Master Class. California: No Star Press.

NVIDIA. 2016. NVIDIA HairWorks 1.2.1 documentation. Available at: https://docs.nvidia.com/gameworks/content/artisttools/hairworks/index.html [Accessed 7 March 2019]

Autodesk. 2018. Hair And Fur Modifier (World Space). Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/3DSMax-Simulation-Effects/files/GUID-4BCFC5E4-266F-43AA-A552-77CC241F9D1E-htm.html [Accessed 7 March 2019]

Allegorithmic. 2019. Substance Painter Documentation. Available at: https://support.allegorithmic.com/documentation/spdoc/substance-painter-20316164.html [Accessed 25 March 2019]

CG Boost. 2017. How to set up Retopology in Blender (Tutorial EN). [video online] Available at: https://www.youtube.com/watch?v=2hEHtKH55Us [Accessed 20 March 2019].

Blender Guru. 2017. Blender Intermediate UV Unwrapping Tutorial. [video online] Available at: https://www.youtube.com/watch?v=scPSP_U858k [Accessed 21 March 2019]

Polycount Wiki. 2015. Texture Types. Available at: http://wiki.polycount.com/wiki/Texture_types#Specular_Maps [Accessed 21 March 2019]

Slick, J. 2019. Surfacing 101: The Basics of Texture Mapping. Available at: https://www.lifewire.com/texture-mapping-1956 [Accessed 2 April 2019]

Blender. 2017. Blender 2.79 Manual, Render Introduction. Available at: https://docs.blender.org/manual/en/latest/render/introduction.html [Accessed 27 March 2019]

Blender Guru. 2015. The Basics of Good Texturing in Blender. [video online] Available at: https://www.youtube.com/watch?time_continue=40&v=rzXNZkEoTAk [Accessed 27 March 2019]

Autodesk Knowledge Network. 2014. 3DS Max. Bones System. Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-E6164716-CFA9-4DE9-9976-F8A58850461F-htm.html [Accessed 28 March 2019]

Warhammer: Vermintide 2. 2018. Fatshark.

Kingdoms of Amalur: Reckoning. 2012. 38 Studios.

Fallout 3. 2008. Bethesda Softworks.

LISTS OF FIGURES OR TABLES