Bachelor's thesis

Information technology

Healthcare Informatics

2017

Juuso Salminen

CONSTRUCTING A 3D MOTORCYCLE MODEL FOR GAME DEVELOPMENT USAGE IN UNREAL ENGINE 4

TURKU AMK

TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR’S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information technology | Healthcare Informatics

2017 | 35

Juuso Salminen

CONSTRUCTING A 3D MOTORCYCLE MODEL FOR GAME DEVELOPMENT USAGE IN UNREAL ENGINE 4

The purpose of this thesis was to examine the process of creating a 3D model of a motorcycle from the concept stage to the game ready object. The model was designed for a current generation game engine (Unreal Engine 4) and was created using the newest techniques. The complete process was first studied and then used in practice. Creation consists of modeling high and low poly models, baking them down, UV-mapping and texturing.

Theoretical part focuses on learning: modeling high and low poly, retopology, uv-mapping, texturing. How to make the best possible outcome on all these areas.

Practical part consists of high poly modeling; making the base high poly model using the techniques learned during the theoretical part. Creating the low poly model from the high poly, using different techniques, decimation and retopology. UV-mapping the low poly model and baking it down. Texturing the final model. And presenting renders of finished product.

Modeling was done using Maya 2017, chosen for being the most intuitive and easiest to use compared to other options. Some cloth parts of the model were created in Marvelous Designer 5.

The finished product is high quality, highly detailed asset ready to be placed into a game engine.

KEYWORDS:

3D modeling, Maya, Substance, texturing, Unreal Engine

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma | Hyvinvoititeknologia

2017 | 35

Juuso Salminen

MOOTTORIPYÖRÄN 3D-MALLIN RAKENTAMINEN PELINKEHITYSEEN UNREAL ENGINE 4:SSÄ

Tämän opinnäytetyön aiheena oli tutkia moottoriopyörän 3D-mallin luomista konseptista valmiiksi peliin lisättäväksi tuotteeksi. Malli tehtiin nykyisen generaation pelimoottorille (Unreal Engine 4) ja luotiin käyttäen uusimpia alan tekniikoita. Prosessi opiskeltiin aluksi läpi, jonka jälkeen se tehtiin käytännössä. Työ koostuu korkea- ja matalapolygonisen mallin mallintamisesta, niiden yksityiskohtien siirtämisestä, UV-kartoituksesta sekä teksturoinnista.

Teoriaosuus keskittyy oppimiseen; mallintamiseen, retopologiaan, UV-kartoitukseen, teksturointiin sekä miten eri osa-alueilta saadaan parhaat lopputulokset.

Käytännönosuudessa teoriaosuudessa opittujen käytäntöjen avulla mallinnetaan lopullinen malli. Luodaan korkeapolygonisesta mallista pelimottorin vaatima matalapolygoninen versio käyttäen eri tekniikoita, kuten retopologia ja desimointi. UV-kartoitetaan matalapolygoninen malli ja siirretään korkeapolygonisen mallin yksityiskohdat siihen. Lopuksi teksturoidaan lopullinen malli ja luodaan siitä esittelykuvat.

Mallintaminen tehtiin käyttäen Maya 2017, joka valittiin sen intuitiivisen käyttöliittymän ja helppokäyttöisyyden takia. Osa kankaita sisältävistä malleista luotiin käyttäen Marvelous Designer 5:ttä.

Lopullinen tuote on korkeatasoinen ja yksityiskohtainen malli, jonka voi ottaa käyttöön pelimoottorissa.

ASIASANAT:

3D-mallinnus, Maya, Substance, teksturointi, Unreal Engine

CONTENT

PICTURES

LIST OF ABBREVIATIONS

Baking	Tranfering attributes of high poly to low.
HP	High poly. Mesh which's polycount is typically high.
LOD	Level of detail. Also a name for a llower poly mesh.
LP	Low poly. Mesh which's polycount is typically low.
Maya 2017	3D modeling program
Marvelous Designer 5	Program to simulate cloth.
Ngon	Polygon with 4+ vertices.
PBR	Physically based rendering.
Poly	Polygon. Made from edges, creates a face in 3D space.
Primitive	Shape created in modeling software.
Prop	Asset in a game world used to dress up a scene.
Quad	Polygon with 4 corners, or vertices.
SP 2	Substance Painter 2. Texture authoring tool.
Subdiv	Subdivide. Term for the action of smoothing a mesh.
Tris	Triangle. Split polygon with 3 vertices.
UE4	Unreal Engine 4. Game engine.
UV	2D texture of a 3D models surface.

1 INTRODUCTION

This thesis focuses on creating a 3D-model for a game engine, following the creation from choosing and creating the concept art, into modeling and texturing and what happens in between. Final model will be a part of a personal game project.

The theoretical part focuses on learning good practices of 3D-modeling, baking and texturing. All of these are explored with the intention of finding a good, flexible pipeline to use on this, and in future projects. Basic pipeline consists of highpoly modeling, lowpoly modeling, map transferring between these two and texturing. Everything learned will be the lastest, current generation workflow to ensure future functionality. In theoretical part, differences between highly detailed essential models are compared with those of less importance, usually props.

Practical part delves into the modeling itself, using only techniques learned during the theoretical part. Firstly creating high poly model, then creating low poly model from the high poly. After that, using different tools and manual deletion of excess edges, all while considering the edgeflow of the object. After modeling, 2D-maps are created and then the final model will be textured.

Closing chapter condenses the project, its successes and pitfalls. Also reflects on what was made well and what needs to be considered before the next project using the same workflow.

2 3D MODELING FINDINGS

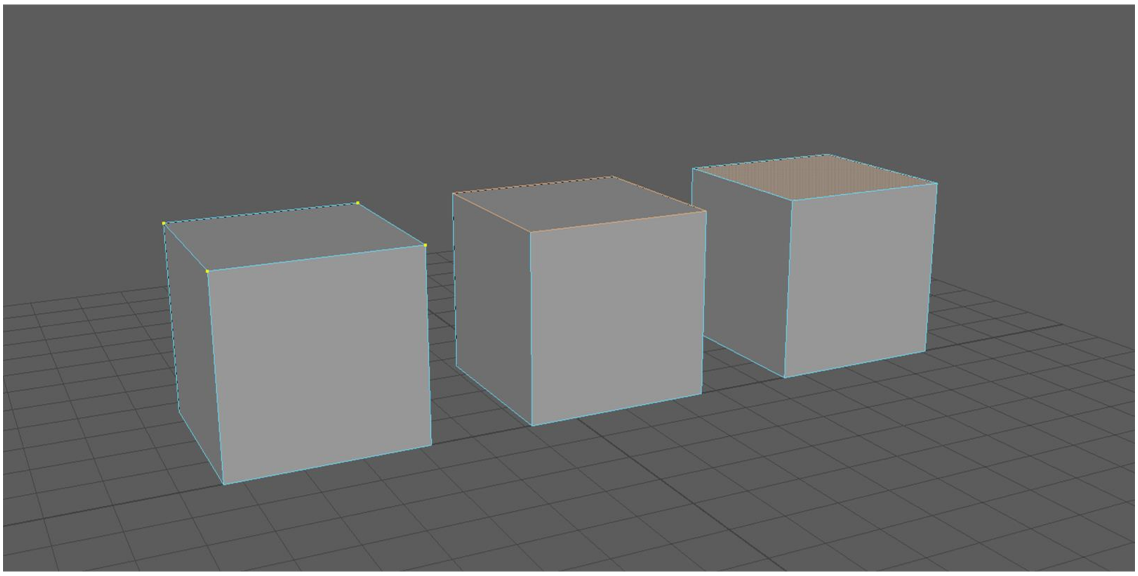
Work on this thesis was started by learning the basics. All information was found online; literature, forum conversations and videos were all used during the learning and theory phase. Mostly using tutorial videos and doing them while watching was found to be the best way to learn. Mistakes were easy to spot and the tempo of working could be adjusted. A lot of information was gathered, and all that information later on slowly molded together as more and more techniques were learnt and used in conjunction with each other.

2.1 3D modeling basics

3D modeling has evolved fast during last few years, mainly because new and improved hardware. This has had bigger effect on videogames than in ordinary modeling work and games can reach new heights. Personal computers can render more polygons and handle more and bigger texture maps. This has led to big competition between big companies as developers try to push the hardware to it’s limits creating more immersive game worlds.

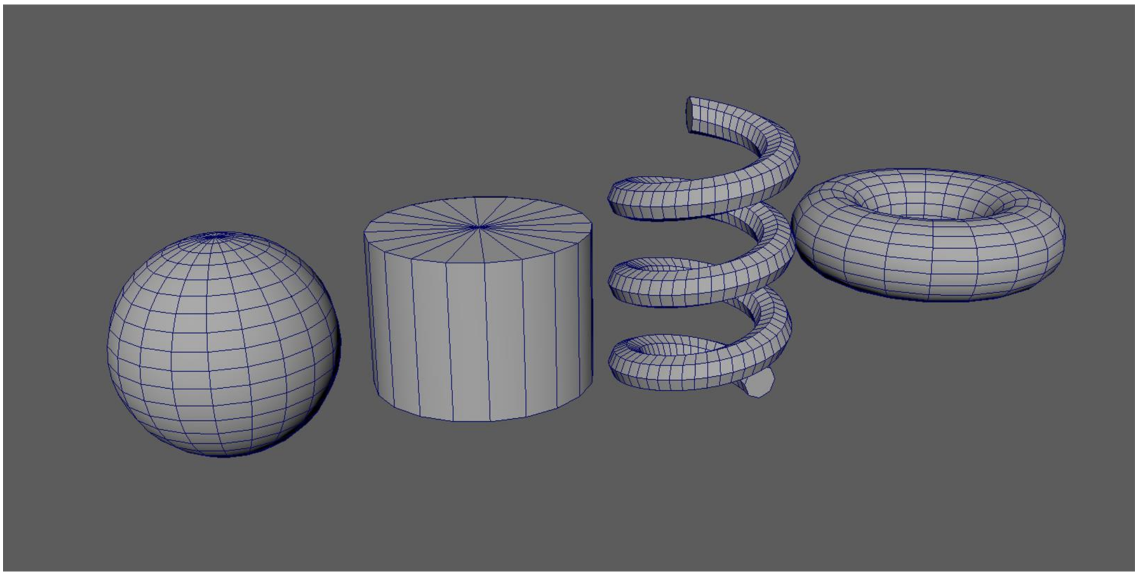
2.1.1 Basic knowledge of modeling

Usually 3D models are created out of polygons. Polygons are planes in 3D space that have 3 or more corners (vertices) connected by edges creating a face, as shown in picture 1. By connecting faces together different shapes can be created, from a basic 6 face box to very complicated models. (Flipped normals, 2017.)



Picture 1. Vertices, edges and a face selected.

Most modeling programs have primitives, or basic shapes, to help get started with the modeling process. These primitives are basic shapes like cylinders, boxes or helixes, some of these shown in picture 2. Despite coming in multiple shapes, primitives usually need to be manipulated to gain the wanted result. In addition of creating new primitives, existing ones can be combined, cut, bridged or other ways modeled to the wanted shape. (Autodesk, 2016.)



Picture 2. Sphere, cylinder, helix and torus.

2.1.2 Limitations in modeling and the target platform

When modeling for a game engine instead of just rendering images there are limitations to consider such as the amount of polygons, and usage of texture maps. Different platforms can handle more than others, PC usually being the one able to handle the highest. It is a good idea to limit models to a certain polycount to ensure the engine and platform can handle the model. (JordanN, 2014.)

Engine's load can also be helped by using certain meshes, which reduce mesh's polycount when far away or when having small screen size, also known as LOD meshes. Smaller texturemaps can also be used in combination with LOD meshes. This allows objects to be beautiful up close and use less resources when far away, allowing the game to look beautiful but require less computing power. (Unreal Engine documentation, 2017.)

2.2 Concept art and design

Concept art is usually the first procured item when starting a new model. Concept art usually consists of multiple views of the object with a size refenrence of the side. Concept is usually used to match dimensions and get a clear vision of how to engage the modeling process. Concept art is something a modeler can choose to use, but it is not necessary. (Flipped normals, 2017.)

2.2.1 Why is concepting important

Concept gives a guide on general size, proportions and might be helpful when creating the most basic mesh, also known as blockout. Blockout phase consists on making just the basic shapes of the model, a crude version of the soon to be finished project. When starting with a blockout, it is easy for the modeler to quickly adjust different parts of the models, that may have been found not to work with the desired outlook of the model. When the blockout is finished it is easy to move on to the next step, high poly model. (DiPaola, C., 2015.)

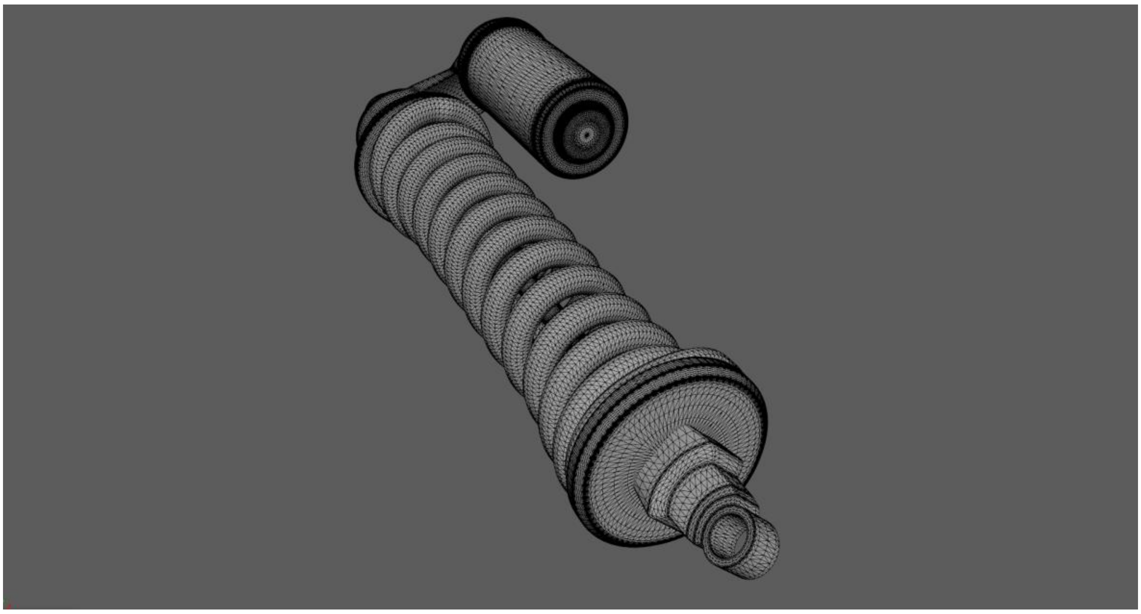
2.3 High poly modeling

High poly modeling means to create intricate, very detailed model of the modeling target. This can be made straight from concept or from an earlier created blockout mesh. Since high poly models are highly detailed they can go over millions of triangles. Depending on the use of these models they can be used as is, but in video games they are baked down to lower poly meshes. Otherwise they would require immense power from the hardware to render correctly and would affect the runspeed of the game to an unplayable stage. (Unity documentation, 2017.)

2.3.1 High poly modeling basics

High poly modeling is modeling without limitations. The created models are either used as is for rendering or baked down for games. This means that models can have as much detail as possible. Without limits, high poly models' only limiting factors are the skills of the modeler, time restrictions and the level of detail wanted. Picture 3 shows a high poly model of a spring assembly.

Creating hard surface models like weapons and vehicles usually incorporates only the core modeling tools. When creating these models, soft edges and round, smooth surfaces are created by using subdivisions. Subdividing a mesh divides each face to 4 therefore smoothing out sharp edges and other transitions between edges. (Polycount, 2017.)



Picture 3. Wireframe of a smoothed spring assembly.

High poly models, in addition to traditional modeling techniques, can be created by sculpting. Sculpting is a modeling method that usually incorporates a drawing tablet of sorts. Using traditional sculpting tools in digital application can create more organic, realistic result compared to normal modeling tools. Sculpting is usually used on characters, cloth and other organic models. (Raitt, B. and Minter, G., 1998.)

2.3.2 What to keep in mind in high poly modeling

When creating the high poly model few things need to be considered. Most important are poly types and edge flow. For basic high poly model quads are recommended, but not needed. You can use triangles, but the problem with them is that they can create artifacts when the mesh is smoothed. Good edge flow ensures good smoothing and bakes. (Raitt, B. and Minter, G., 1998.)

2.4 Low poly modeling

When moving from high poly to low poly the target of the modeling work changes. In high poly the target was to create as detailed model as possible, whereas in low poly modeling you want to create as simple model as possible.

2.4.1 Low poly modeling basics

As said, low poly mesh is basically the high poly represented with as few faces as possible. Removing edges from high poly mesh can cause unwanted results and should be done carefully. Usually most small detail can be removed since it will be baked on the normal map. Support loops and other manipulators affecting the smoothing of the high poly can be decimated without too big of an impact on the silhouette and therefore reduce the polycount drastically. The model still requires certain amount of similarity to the high poly, or the main shapes will be lost. A good guideline for low poly creation is to affect the silhouette of the model as little as possible. This changes if the model has to reach specific polylimit in terms of the engine or project.

2.4.2 Retopology

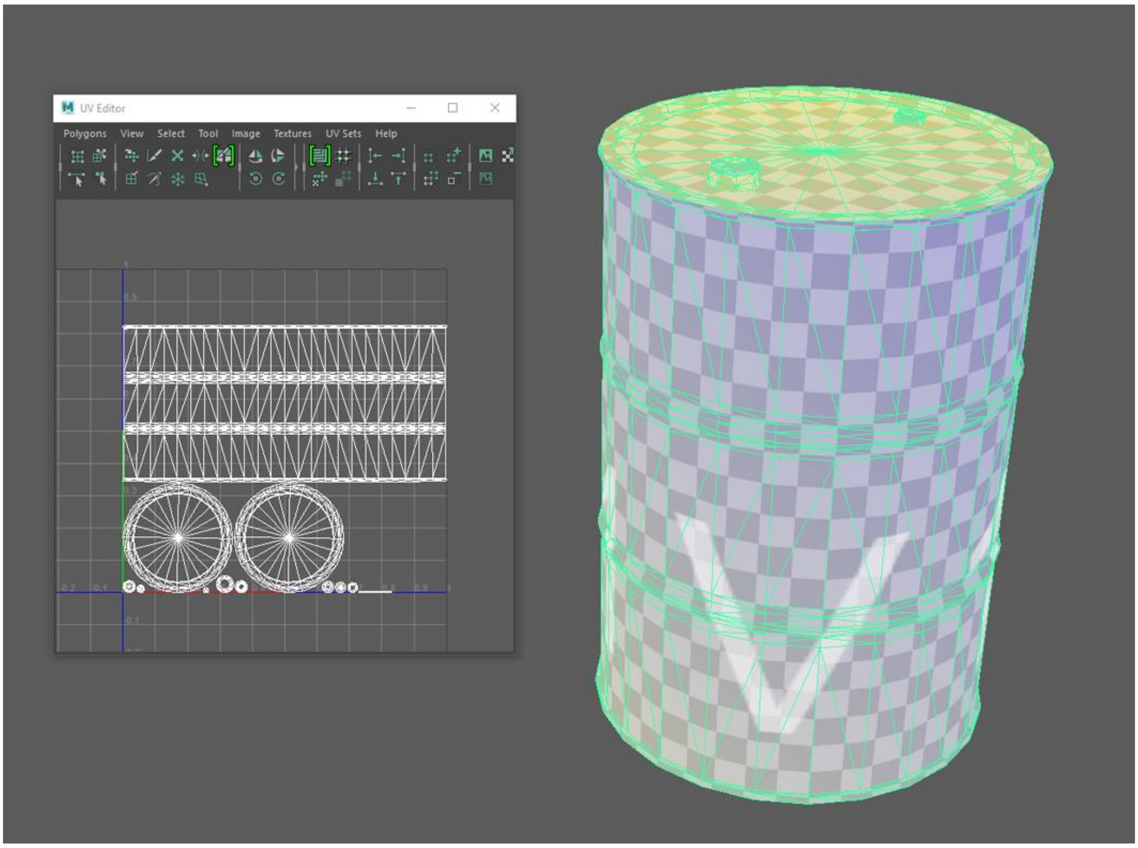
Retopology is act of creating new polygons on an existing surface. Mostly used when creating low poly models from complicated organic meshes. Different tools allow users to draw vertices on the highpoly surface and then connecting them to create new faces. These retopologized meshes allow for same application as normally decimated low poly meshes. (Autodesk, 2016.)

2.5 UV mapping

UV mapping is the act of turning the 3D model into 2D. The surfaces that create the three dimensional model are cut into specific parts, specified by the model's topology, and then folded flat into an 2D image. (Modo, 2014.)

2.5.1 UV mapping basics

Creation of UV maps has many rules and most of them carry the same weight when it comes to creating clean, good looking textures. Having as little UV islands as possible with as little seams as possible. Unfolding isles to minimize distortion and unfolding different isles in relation to others to keep texture density even. Picture 4 shows a barrel that’s UV maps are unfolded and arranged on the grid. Some case by case rules are UV island orientations, overlap of duplicates and padding size. (Modo, 2014.)



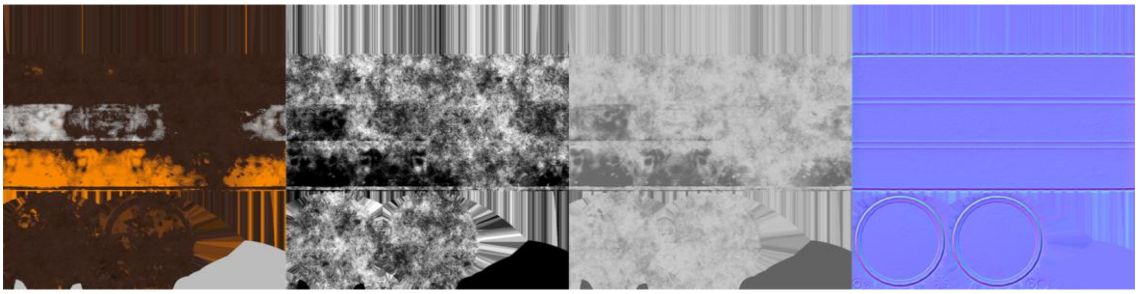
Picture 4. Uvs of a barrel.

2.6 Texturemaps

Without texturemaps 3D models would just be single color blocks of intricate shapes. Textures are applied on the unwrapped, unfolded and optimized UV maps of the object. Texturemaps are what give models color and other detail.

2.6.1 Texturemap basics

There are multiple different texturemap types and multiple uses for them. They all affect the same set of UVs but create different effects. Textures can be created in a image editor like Photoshop with different source pictures, drawn by hand or created in specific texturing applications like Substance Painter or Quixel Suite. Newest texturing style is PBR or Physically Based Rendering that uses specific maps to create realistic renderings of the textures. It uses 3 maps as a base but more maps can be added to increase complexity or add features. (Unreal Engine, 2017.)



Picture 5. Base color, metallic, roughness and normal maps of a barrel.

As shown in picture 5, maps differ multiple ways. Base color is just the color of the material, shown on a RGB map. Second one is a greyscale map of metalness. This map determines which areas of the material are metals and which are not. Next similar map is a greyscale map of roughness. It determines the sharpness or blurryness of the reflection of the material's surface. Last one, normal map, is a RGB map of depth and detail. Using the map's details in a game engine it creates illusion of detail, that isn't actually there.

2.7 Differences between drivable motorcycle and static prop

Drivable vehicles usually need to be highly detailed as they can be closely inspected by the player. Being in the focal point of the game this model especially needed the high details and was created as so.

Static props are objects that clutter the game world and give life to the environment. They can usually be less detailed as the player wouldn't pay too much attention to all items around the game world. They need to be detailed enough to be viewed closely, but since they will have no actual use for the player they usually are ignored by the normal player. Having lower polycount and lower textures the model save power of the engine for other feats, while still creating depth and visual richness.

This model was split into multiple parts to be able to have enough texture space and also to be able to divide the model into multiple pieces. These pieces can then be used as separate objects. For a static prop, 1 texturemap would have been enough and the mesh could have been a singular piece. To counter the polycount of the model, LOD groups were planned. This allows the models to act as static props but also as the player driven motorcycle. This is en extra gained by the need to make the LOD levels anyway, so it allows it to be used as a prop as well. In addition of using LODs, the model can be reduced in polycount by removing multiple parts of the model for the prop version. The impact on the engine is minimal with the LOD meshes enabled, but they are not a part of this thesis. Only the creation of the main high poly and low poly mesh is discussed.

In the case of just creating a basic static prop of a motorcycle the model wouldn’t need as much detail as a key character of the game. Static mesh would need biggest and the most noticeable areas detailed, but most of the engine details could only be done with the texture. Despite being noticeable prop in the game the player wouldn’t pay much attention on the prop while exploring the game world and with having lower polycount and smaller textures it would give smaller load on the engine.

In this case the motorcycle acts as a player crafted drivable vehicle so both sides of these 2 types of modeling needed to be considered. Creating detailed mesh with LOD levels counters the weight of the models on the engine, while still having the beautiful highly detailed mesh as the drivable vehicle.

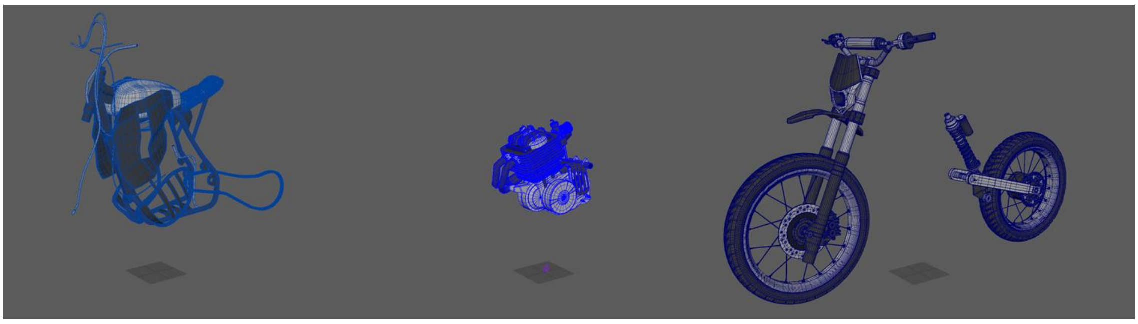
3 3D MODELING

3.1 Reference pictures and creating base of design

The aim was to create a motorcycle model to be used in Unreal Engine 4. The base of a real motorcycle was used at the beginning to ensure the realistic scale. Only the frame and most of the engine parts were created from existing pictures, other additions were created on the fly using some references. The theme of the modeled motorcycle was "worn out old motorcycle built from multiple bikes into one". No actual multidimensional reference pictures were used, only one picture was taken into the modeling program at the beginning to scale the base frame correct.

Since this model was meant to be one of the most iconic pieces of the gameworld, it's polycount and texture usage was planned accordingly. Polycount limit was set to 150 000 triangles. Based on the information gained during the theory part of this thesis this limit was found out to be realistic and forgiving even if the limit was not met. Model has 3 different parts, and each of them will have 4 texture maps. Model was planned to use LOD groups, but they are not discussed in this thesis.

Motorcycles having multiple moving parts and different pivots for multiple pieces, it required planning to divide. The division was required to make sure the model would make the best use of the produced textures as well as help in the future when the model was textured. With clear splits on the structure of the model, most problems with animation, usage and engine import would be taken into consideration. Inspecting different types of motorcycle the division became clear quickly. 3 groups were made; front and back forks with wheels, frame and engine, show in picture 6. Later a different group was added to accessories, which are not required on the model, but bring depth to it and some gameplay elements.



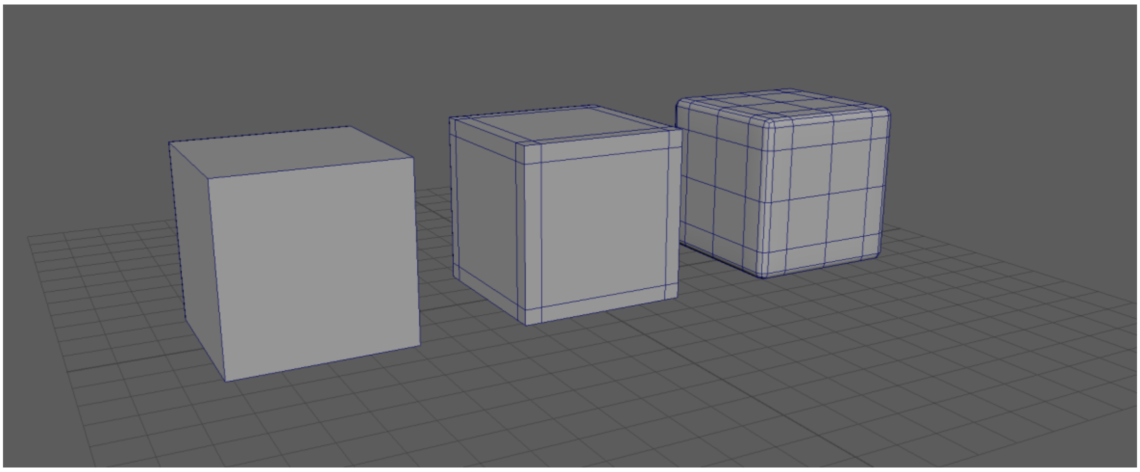
Picture 6. 3 main groups, engine, frame and both forks.

With these splits of the model and the image of the finished product in mind, the modeling phase started. First the blockout model was created on top of the only reference picture used in Maya, making sure it was scaled in the real world scale. The blockout was a simple mesh with the base of the frame, tank, fenders, forks and wheels. Other pieces based on real life products were sometimes quickly referenced, but nothing was modeled straight based on their specific dimensions or volumes. At this point the readability of the model was considered and some things were changed to create more readable and interesting model.

3.2 Creating high poly model

With the blockout model in the modeling program, creation of the high poly model started. This step of the modeling consisted mostly of creating completely new objects on top of the existing blockouts, or manipulating the existing blockouts to create the wanted shapes.

The technique used most in this modeling project is called hard surface modeling. It is used, as the name implies, to create models that have hard surfaces and sharp transitions between edges. Mostly weapons and vehicles use this technique and it was used mostly on this motorcycle too. The mesh is created and smoothed to gain the final result. The smoothing is controlled with additional edges called support loops. These edges are added around the existing perimeter edges to create sharper, smoother edge transitions after smoothing. Technique demonstrated in picture 7.



Picture 7. Base mesh, base mesh with support loops, smoothed mesh.

When dealing with something of a more organic nature, sculplting can be used. Sculpting uses traditional tools to manipulate vertices, faces and edges. Often compared to working with clay in a digital environment using digital tools to sculpt. There are many programs that can do sculpting, and there are basic sculpting tools built in Maya too. These tools were only used to add small details, since most of the model could be built with using only the hard surface modeling.

In addition to sculpting, cloth, hair, fur and the like can be simulated. Simulation can be done in Maya or in a external program. For the objects that were created as accessories for the model, carry bags and straps were created in Marvelous Designer 5. It is a clothing program that simulates cloth within set restrictions for the wanted results.

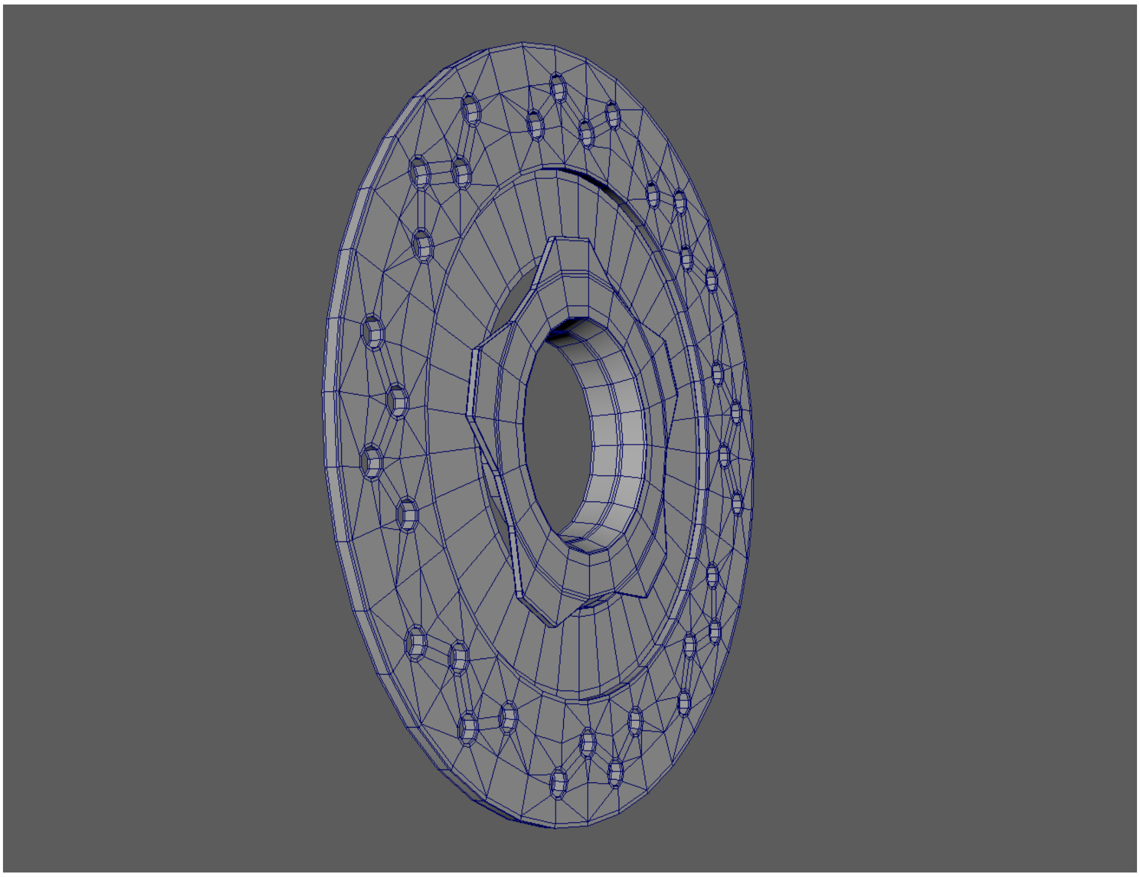
3.2.1 Hardsurface modeling

The frame, created from multiple pieces in the blockout, was created from scratch as a mostly single piece to gain the wanted detail and similarity of the real life version. With faces sharing edges in the places they meet, it allowed for a more realistic smoothing and the ability to fake welding seams. Still, some parts of the frame were detached from the base model, mostly where it was not required to create good seams or smoothing.

The final model of the gas tank on the other hand was created from the blockout model. The created blockout was close to the wanted shape and was easily manipulated further to gain wanted level of detail. The hole under the gas cap was added, and the indent under the tank, where it sits on the frame. With theses additions and some added geometry the model was smoothed and the final result was met.

One of the hardest and most complicated parts was the engine. It has multiple pieces of geometry close together, with intricate transitions, smooth and sharp edges. The bottom base of the real world equivalent is created from 3 casted parts, left and right face and the middle that connects them. It was important to model all these 3 parts as they are in real world since it is one of the most interesting parts of the model. Both sides were modeled as their own and so was the middle part. Both sides have more round shapes added on top of them, and they were all created as one mesh to simulate the casting process in the modeled piece as it is in the real world counterpart. The sides and the middle were later merged together, since the seams didn't bring much interesting detail and could be added in the texturing phase. The cylinder of the engine has a basic underlying mesh that has all the cooling fins and components on the top just crashing into it. By crashing meshes correctly together makes the modeling faster and easier and can also reduce polycount. It was easy to use in this part, since it is very busy and has a lot of nooks and crannies that cannot even be seen.

All models were created so that their high poly mesh was previewed in Maya using a "smooth mesh preview" function. Object ready to be previewed is shown in picture 8. This allows the modeler to see the final result, while still having the low poly mesh. When having these both, it is easy to create the wanted 2 results, high and low poly meshes. By duplicating these, and actually subdividing the other mesh, and removing support geometry from the other, both high and low poly models are created fast and are sure to match each other closely.



Picture 8. Unsmoothed brake disc with support loops.

Tires were created on a plane that was later on bent to shape. The main shapes of the pattern were cut into a plane, some were created separately and added onto the main part later. Wanted parts were then extruded out to give depth to the pattern. Same simple piece of pattern was duplicated to create a long piece of repeating geometry, that was then bent to create the shape of a wheel. The rounded sides of the tire were added later.

Some models allow the usage of techniques that ease and simplify the modeling phase. For example, since the front fork is symmetrical, mirroring could be used. Only needing to create the other side of the forks fastens the creation process and leaves more time for the more complicated pieces. Mirroring can be used just to duplicate the mesh along the preferred axis, but also be used with instancing that allows the manipulation of the original and the mirrored side at the same time.

When creating the t-bones, the pieces that connect both fork struts, mirroring was used. In addition basic duplication was used to create the top part from the bottom one. By not just duplicating the mesh and calling it done, the duplicated mesh was manipulated in size, while still being efficient and using the same topology as the original one.

3.2.2 Sculpting and cloth simulation

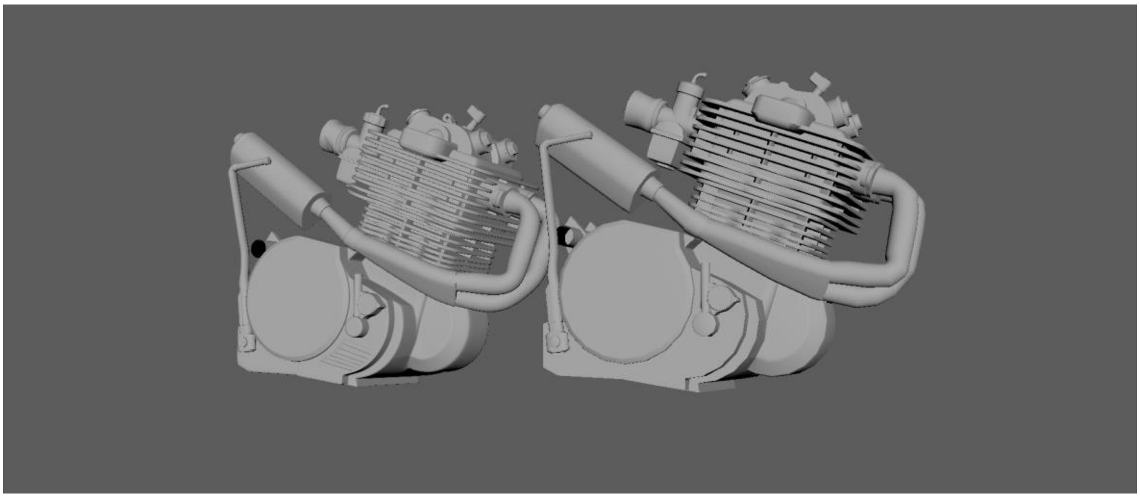
This hard surface workflow was not usable on all the parts of the model. The cloth and the straps on the seat are organic and the earlier way of modeling wouldn't have created very realistic or good results. Instead, they were created using in Marvelous Designer. It is a cloth simulation program that allows users to create different clothing to be used in other programs. Creating the patterns, sewing them together and adding different manipulators to affect the simulation can be all created in the program and then be exported out to be sculpted or used as is, in another program.

3.2.3 Details

In terms of small detail modeling, the motorcycle is lacking. It is mostly constructed using large clean pieces of geometry that create the whole. Most of the detail work was located in the engine, in terms of nuts, bolts and small shapes. On the right side of the engine small bulge lines were modeled. They were later removed from the low poly and only baked on the normal map. Nuts and bolt were duplicated around the model, and by duplicating and not modeling each individual piece, texture space was saved. Negative side of duplicating them around the mesh was that they all look the same and do not have the texture fidelity they could, but since they are extremely small in the whole scope of the model, they do not require it. Most of the other detailing work, like the weld seams on the frame, were done during the texturing phase.

3.3 Creating low poly model from high poly

Creating low poly mesh from the high poly is rather easy but time consuming. By making a duplicate of the mesh that would be subdivided to create the high poly model, it is ensured that the base shapes of the low poly match the high poly and therefore create the best possible bakes. By removing the support loops that would limit the deforming of the edges before smoothing the mesh, the geometry of the model does not change, but the polycount reduces drastically. Depending on the model and the mesh, some of the other edges can be removed or at least vertices can be merged to reduce polycount while still having the mesh remain close to the high poly. For parts like the engine’s right side bulge lines, they can be removed completely and just leave the flat faces on the same place in the low poly mesh. By using this method the details are baked on the normal map, and the polycount is reduced. Picture 9 shows 2 models, high and low poly, that were created from the same base.



Picture 9. High and low poly models.

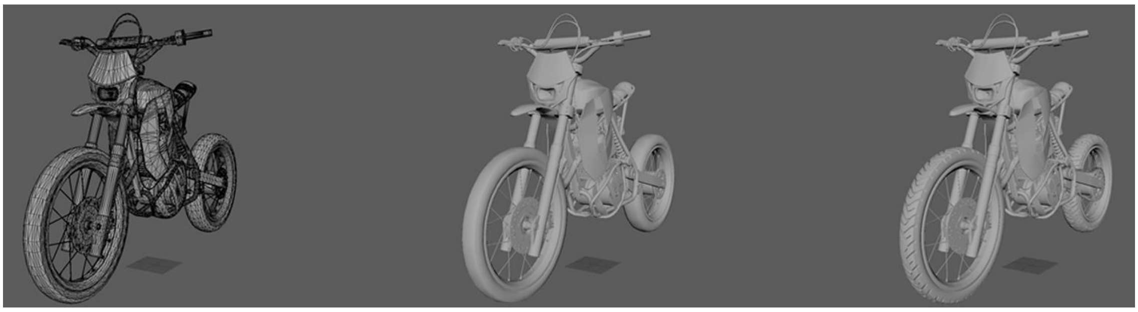
While removing all possible edges is generally a good thing to do, some places might require them in place to bake the normal map correctly. This area is opened up more in a later part of the thesis. Sometimes when a project requires the model to be within certain polycount limits, some geometry might be required to be removed completely. This causes less fidelity in the model itself, since the details won’t be as readable in the texture as they would be as an actual geometry in the model itself.

When creating the low poly mesh in addition to it, also the basis of the baking cage is created. This cage is used to calculate the differences between the low and high poly meshes. The baker uses the low poly mesh with it’s edge smoothness to determine how the rays are cast down from high poly to low poly mesh. Depending on the angle these rays are cast the details of the high poly are calculated on the low poly mesh. With incorrect edge smoothness or edge placement the baked normal can be warped or distorted.

3.3.1 Mesh decimation

Decimating the mesh manually can be time consuming but rather simple. This basically means removing all edges possible from the model, like support loops and other edges that might affect the mesh only slightly. Flat and close to flat surfaces can only have the minimum amount of vertices to create the shape, when other parts of the model that have smooth transitions around the edges can have a small bevel or even nothing between them in the low poly model. Doing this the polycount is dropped to the game engine friendly limits. End result of decimating the high poly mesh shown in picture 10.

During the decimation process also the UV mapping needs to be considered to optimize UV shells' cuts to right places to create the best bakes possible.



Picture 10. Wireframe, low poly and high poly models.

3.3.2 Retopolgy

Retopology is another way of reducing polycount of a specific mesh. Most commonly used on organic, complicated models. Retopology tools allow user to create completely new mesh using the high poly mesh as a guide underneath. Maya's retopology tool is called Quad Draw, and was tool of choice in this project. When used, it places vertices on user determined locations on the high poly mesh. By placing 4 vertices on the mesh they can be connected with edges to create a face. This allows the topology to support the important parts of the high poly geometry by allowing more geometry on certain parts to increase readability, and reduce the usage on parts that might be hard to see, or those lacking detail.

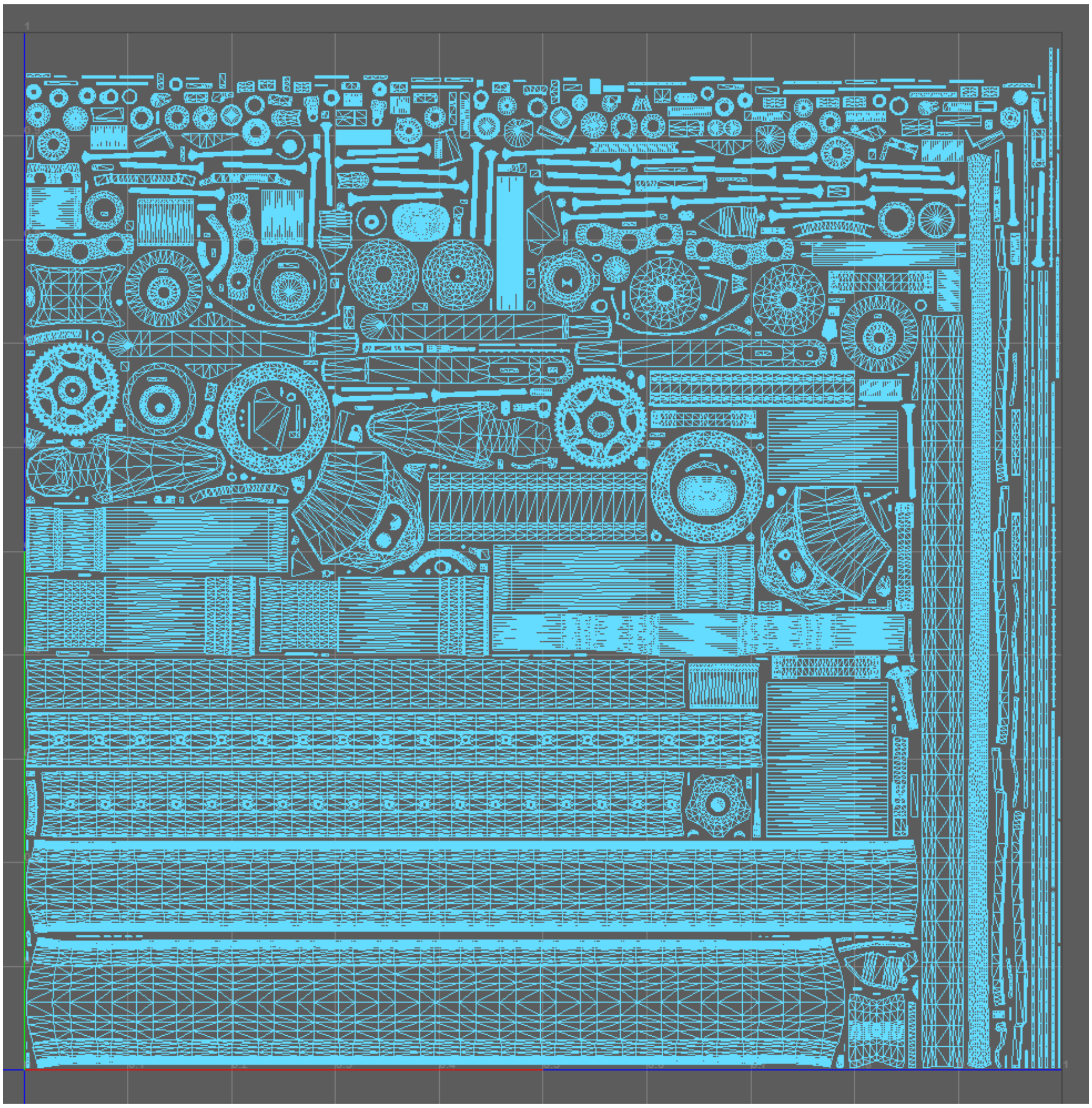
3.4 UV mapping

UV mapping is the process of folding the 3D model down to 2D space. This 2D space is a texture map with a size in power of 2. Most common are squares, usually 1024x1024 or 2048x2048 pixels, but they can also be non squares. All faces require their own space on the UV map, and in most cases they cannot overlap each other. To achieve the end result, the low poly mesh is cut up into pieces that create UV shells. For example a cube's specific edges can be cut and the cube can be laid flat in 2D space. For more complicated objects like parts of the motorcycle, there needs to be multiple UV shells so that the whole model can be arranged on the specified UV map.

To fasten the UV mapping, different kinds of projections can be applied on models to create fast UV layouts. Shells created by using these basic projections can work, but usually require manual adjustment. When working with hardsurface models, good places to cut shells are the hard, sharp edges of the model. This way seams hide better and this also helps the baking process.

When models are laid out on the 2D space, their borders and inner edges can be warped. Different tools are used to fold, relax and adjust them to their correct shapes to represent the 3D model the best they can on the 2D space.

After the shells are folded correctly they can be arranged on the UV map, as shown in picture 11. It is important to take as much of the UV map space as possible to get the highest quality textures. Since there cannot be overlap with different shells, some space is always empty because of different shapes, but it is also important. Allowing some space between the shells allows the textures to bleed over the shells' lines, reducing the possibility of edge artifacts and other problematic occurrences.



Picture 11. UVs of front and rear forks folded and arranged.

3.5 Exploding meshes for baking

In this thesis the traditional way of exploding the meshes was not required. A new technique was used that was implemented in the latest version of Substance Painter 2. Traditionally all meshes were required to be moved away from each other to allow the rays cast by the baker to only contact the wanted parts of the meshes. By using this new technique allowed by Substance Painter, meshes weren't moved at all, but instead were named accordingly. The baker matches only the high and low poly mesh that are named the same. Their only difference in naming convention was the postfix, in low poly "_low" and in high poly "_high". Using this naming convention fastens the process and also eliminates the need to explode the mesh for baking and then putting it back together after the bake. The bake process itself takes longer but overall is a timesaver.

3.5.1 Baking maps

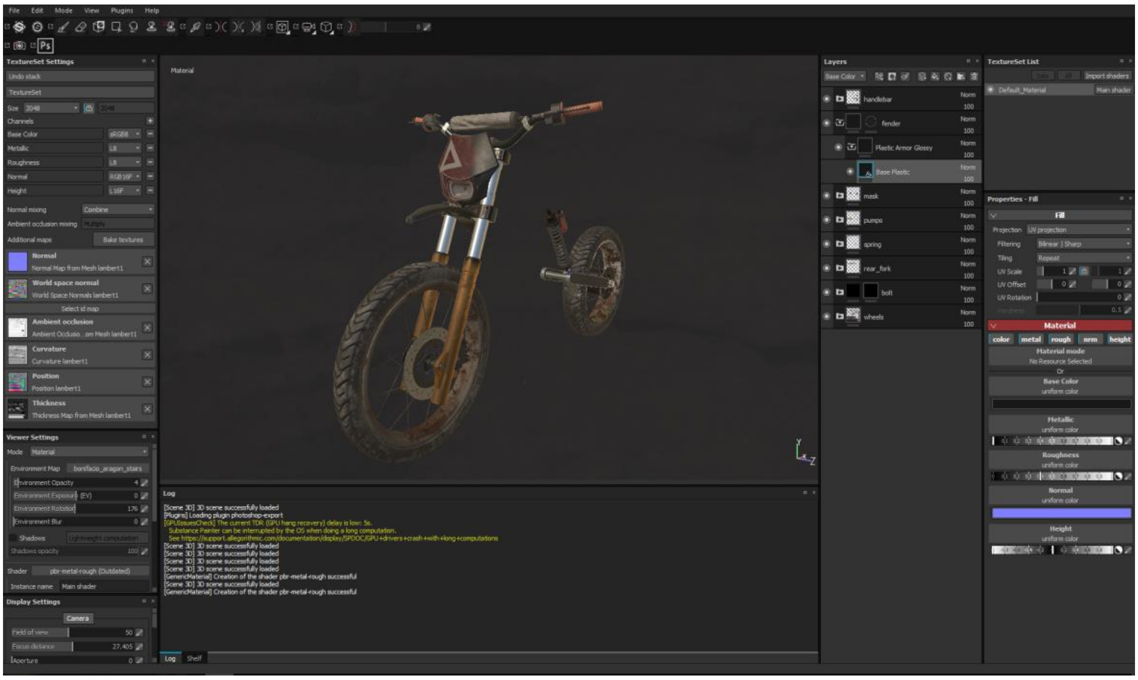
All the maps created were done in Substance Painter 2. Using the high poly as a base for the calculations different maps were created. These namely being normal, world space normal, ambient occlusion, curvature, position and thickness. Basic baker settings that were changed were texture map resolution to gain the intended texture size, antialiasing to get sharper images and dilation, which controls the texture bleed over the shell borders.

These maps allow to create different masks, effects and details while texturing. Not requiring much hand painting, these maps can be used to create realistic effects for dirt, occlusion and general wear and tear.

3.6 Texturing workflow; PBR

PBR, or physically based rendering imitates real life material using different maps. Real life values are used to calculate the realistic look of different materials.

Meshes were imported into the Substance Painter 2 and normal, world space normal, ambient occlusion, curvature, position and thickness were baked to help with the texturing process. Using the materials available in Substance, different parts of the meshes were then masked and a corresponding material was assigned to them. This created the base textured model that was then tweaked further to gain the wanted look on the models.



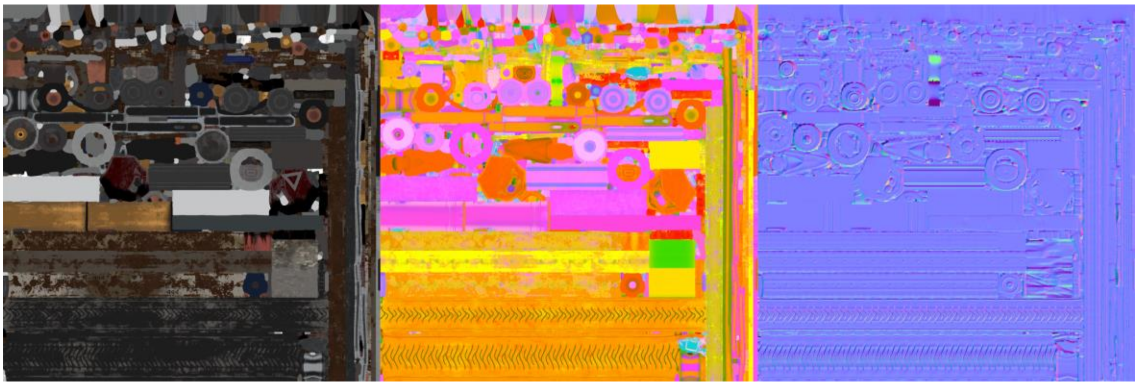
Picture 12. View in Substance Painter 2.

Same steps were repeated for all 3 different parts of the model and later on the accessories. Picture 12 shows almost finished texturing on 1 of the models.

3.6.1 Final maps

While normal, world space normal, ambient occlusion, curvature, position and thickness were created to be used in the texturing process, they are not used in engine for the final product, except normal map. They are only used in the texturing tool to affect the final textures.

Different maps are exported out of the texturing tool depending on the target platform. Since the target for this project was Unreal Engine 4, 3 maps were produced. Unreal Engine can use single map for multiple purposes, using different packing methods. This allowed to create less maps with smaller computing footprint.



Picture 13. Base color, OcclusionRoughnessMetallic and Normal maps.

Base color. This map has the basic color information and also the opacity channel mapped on the texture's alpha channel.

OcclusionRoughnessMetallic. 1 map combining 3 different maps packed for use by the engine. Red, green and blue channels are used for ambient occlusion, roughness and metallic accordingly.

Normal. Has the normal information of the high poly model baked down into the low poly.

The model did not require other channels to be exported and only a total of 3 maps were exported for each part of the model, as seen in picture 13.

3.7 Fixing possible errors

Creating the model from the basic blockout model to the finalized textured model many clearly identifiable steps are taken. Always when moving to the next step the model was checked for possible errors. Biggest errors were found when baking normal information from high to low poly. This caused the need to do the baking multiple times. When doing the error checking instantly when moving to the next step, they are easy to fix and eliminate the possibility of an error carrying all the way to the finished product, which then takes multiple steps and a lot of work to fix.

Unreal engine has beautiful different view modes, which can help with error checking. Seeing the model in its final location, different problems may occur. By cycling through different modes problems can still be easily fixed.

Picture 14. Model in Unreal Engine 4. Only Detail lighting shown.

3.8 The final model

The final model was modeled in Maya, textured in Substance Painter 2 and put into Unreal Engine 4. The limitation set in the beginning of the work were met, model has around 130 000 triangles, that clearly fits the target set in the beginning. With the packing of texture maps for Unreal Engine 4, each part also used 1 less texture map. Together totally using 9 2k texturemaps. The model is now ready to be rigged and animated and be put into a use in Unreal Engine 4. For the beauty shot, the model was posed leaning against a pole. Pictures 14 and 15 show the final renderers of the model and picture 16 shows the scene assembly in Unreal Engine 4.

Picture 15. Final beautyshot of the model.

Final render only has a simple scene, to show the model in engine. The final wanted result was not a beautiful render, but a working model. No special effects were used, just a basic cubemap, reflection capture, and basic lights.

Picture 16. Scene in Unreal Engine 4.

4 CLOSING CHAPTER

The target was was to model and texture a game ready 3D model of a motorcycle to be used in Unreal Engine 4. Work consisted of modeling in Maya 2017 and texturing in Substance Painter 2. Some of the modeling work was done in Marvelous Designer 5, where models were then imported into Maya for final touches.

Modeling alone was a time consuming process. At the beginning of the project only the very basics of modeling were known, but many techniques and tricks were learned after starting and while doing the project. Learning curve in any 3D modeling application is steep and constantly learning new things lead to the need of redoing old models to get to the wanted level and to gain the optimal results.

Texturing was something completely new, and learned completely from the start for this project. Like 3D modeling, it is easy to get into, but to get the results to the industry standard level required time, try and error and many different sites and tutorials.

The requirements of the engine were clear and were taken into consideration since the beginning. Since the target was to get the model into the engine and not animate or play with it, the skills to do so were low. However the future use of the model and the requirements to do them were observed at the beginning and the project was done so that it does not require any modifications later on.

The time spent doing this project allow the use of the accumulated skills to be used in the future while modeling, texturing and working in the engine.

REFERENCES

Flipped normals. (2017). Introduction to 3D.

Available at: https://flippednormals.com/introduction-3D/ [Accessed 27.2.2017]

Autodesk. (2016). Create polygon primitives interactively.

Available at: https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-6D21314A-54AD-41D4-AFC0-AAED13CD50A6-htm.html [Accessed 27.2.2017]

JordanN. (2014). Polycounts in next gen games.

Available at: http://polycount.com/discussion/141061/polycounts-in-next-gen-games-thread [Accessed 27.2.2017]

DiPaola, C. (2015). Block out 3D scenes.

Available at: http://www.3Dartistonline.com/news/2014/11/maya-tutorial-block-out-3D-scenes/ [Accessed 27.2.2017]

Unity documentation. (2017). Optimizing graphics performance.

Available at: https://docs.unity3D.com/Manual/OptimizingGraphicsPerformance.html [Accessed 27.2.2017]

Unreal Engine documentation. (2017). Creating and using LODs.

Available at: https://docs.unrealengine.com/latest/INT/Engine/Content/Types/StaticMeshes/HowTo/LODs/ [Accessed 27.2.2017]

Polycount. (2017). Subdivision surface modeling.

Available at: http://wiki.polycount.com/wiki/Subdivision_Surface_Modeling [Accessed 27.2.2017]

Raitt, B. and Minter, G. (1998). Digital Sculpture Techniques.

Available at: http://www.theminters.com/misc/articles/derived-surfaces/index.htm [Accessed 27.2.2017]

Autodesk. (2016). Create new topology with Quad Draw.

Available at: https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-03933C1F-1BA7-4E7E-AAB5-31FB0E6A77B2-htm.html [Accessed 27.2.2017]

Modo. (2014). Working with UV Maps.

Available at: http://modo.docs.thefoundry.co.uk/indie/help/pages/modotoolbox/WorkingWithUVmaps.html [Accessed 27.2.2017]

Unreal Engine. (2017). Physically based materials.

Available at: https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/PhysicallyBased/ [Accessed 27.2.2017]

EarthQuake. (2012). You are making me hard.

Available at: http://polycount.com/discussion/107196/youre-making-me-hard-making-sense-of-hard-edges-uvs-normal-maps-and-vertex-count [Accessed 1.3.2017]