Georgi Babushkin

ANALYSING PIPELINE FOR A REALISTIC 3D GAME-READY WEAPON

Bachelor's thesis

Culture and Arts

Game Design

2023

South-Eastern Finland University of Applied Sciences

South-Eastern Finland
University of Applied Sciences

Degree title	Bachelor of Culture and Arts
Author(s)	Georgi Babushkin
Thesis title	Analysing pipeline for a realistic 3D game-ready weapon
Commissioned by	-
Year	2023
Pages	40 pages
Supervisor(s)	Brenda Jiménez

ABSTRACT

The objective of the thesis is to analyse the process of creating a realistic 3D weapon to be implemented in a console or a computer game by making a model of an AKSU-74 assault rifle from scratch. The pipeline used for that included pre-production, modelling, UV-unwrapping, baking, and texturing. Due to scope and time limitation the study's focus is on the modelling.

To understand how modern game weapons are made the paper examines basics of each step and overall design principles from a game weapon artist's perspective. These topics include hard-surface modelling, weapon mechanics, material behavior, and optimization. This is done to give readers understanding of how one step in the pipeline affects consecutive steps, and how all the processes work together.

The asset is made by mesh-based modelling and the software used to analyze the pipeline is Blender, Marmoset Toolbag, and Substance Painter. As the paper aims to examine the general process of 3D weapon assets creation, it will feature primarily content from beforementioned software. However, alternative methods will be presented from different weapon artists' articles.

The pictures of making an AKSU model following a modern pipeline are added in each step to better present the context to the readers, and to provide one of the many possible workflows.

The results of the research achieve a working game-ready weapon asset that can be implemented in an FPS game, and its quality can be scaled according to the needs of game development.

Keywords: 3D, modelling, weapon, game-ready, realistic, pipeline

CONTENTS

1 INTRODUCTION

The research question of this paper is “What is the workflow for a realistic 3D game weapon?”. The method for research is a case study of modelling a weapon from an image. By examining techniques often used for making weapons for FPS games, the paper will present a workflow with the steps necessary to go from a still image to a three-dimensional digital object file. The objective will be to make a realistic AKS-74U assault rifle with a holographic sight, tactical flashlight, and silencer attachments. The paper will often reference technical aspects that are not explained previously, because making 3D assets for videogames requires understanding of its every stage. These stages affect each other constantly in various aspects of design and technicalities, which is why it is difficult to present them in a chronological order.

Studying different artists’ recommendations on a weapon workflow reveals a pattern of actions and a checklist of requirements for a modern asset in game development industry. To meet the industry standards and make an effective realistic in-game prop, various weapon workflows can be usually separated into three main sections, making the base model, preparations for texturing, and texturing. The texturing part is examined briefly due to limitations in time. In base model section the paper examines reference gathering and how it affects the base modelling process. The section also covers basics of modelling with tools and methods commonly found in various 3D software editors, resulting in a low poly model. Preparations for texturing section goes over the UV unwrapping, making a high poly model and baking. Outcome of that section is a UV map and a normal map from high poly baking. Last texturing section utilises all the previous stages work for texturing, and its outcome will be a set of realistic weapon textures.

As the study is not commissioned, the scope is set according to game industry standards for a First Person Shooter (FPS) in 2023. The scope is set according to statements in game art articles made by weapon artist. Because the goal of research is an asset for videogames, the criteria for outcomes of the case study research weapon will be its quality of optimization and visual fidelity. By the end of the analysis, the paper should give a preview into weapon art specifically, and a broad overview on 3D asset creation. Examination of the weapon workflow presents readers with one of the possible workflows. An ideal workflow solution does not exist because each artist and development studio have their own standards and requirements for game assets based on style, time, and available resources. Additionally, the thesis research is important for beginner 3D modellers, as it encapsulates basic but necessary information to get started on asset creation.

2 THE BASE MODEL

The following chapter will discuss aspects of 3D modelling that are important for the current weapon project. Understanding the foundation of what a 3D model is will help to explain the decisions for later actions. Every 3D model made from a mesh consists of certain components that make it render on the screen. The mesh appears to be a web like structure of dot, line and surface components that will be discussed in this chapter. The information is taken from the Blender Manual, among other 3D related sources, but is not specific to Blender.

2.1 Mesh structure

The simplest part of a mesh is a vertex. It appears as a dot in 3D editors. Vertices can be stored as an array of coordinates for the object built out of them. Two Vertices can be connected, forming an edge, which looks like a line. It is always connected by at least two vertices, and the more vertices are connected that way, the more intricate that line can get, potentially forming a different shape entirely. Lastly, if three vertices connected by edges are formed into a triangle shape, a surface between them can be formed, called face (Blender Foundation 2023a).

A closed connected chain of vertices and edges with a face between them is called a polygon. They are what is used to build the surface of an object and by default, polygon faces will render their surface only from one side. Triangular polygon face’s surface is always flat, and therefore easy for game engine to calculate. While three vertices and edges are the minimum required amount to form a face, the maximum can technically be whatever the artist makes. Common types of polygons are polys, quads made of four vertices, and everything above that is called a n-gon (Polycount 2015).

For game development purposes, however, a polygon should not contain more than four vertices and edges. This is because upon exporting the model faces will be triangulated, which is the process of connecting vertices by an edge. In case of a quad, it will be a predictable connection that will turn it into two triangles. N-gons, however, will be connected in whatever way is the easiest, which leads to unpredictable geometry deformations. It should be noted that Faces between Edges and Vertices are not mandatory, and an area without a face will simply not render, as is shown in Figure 1.

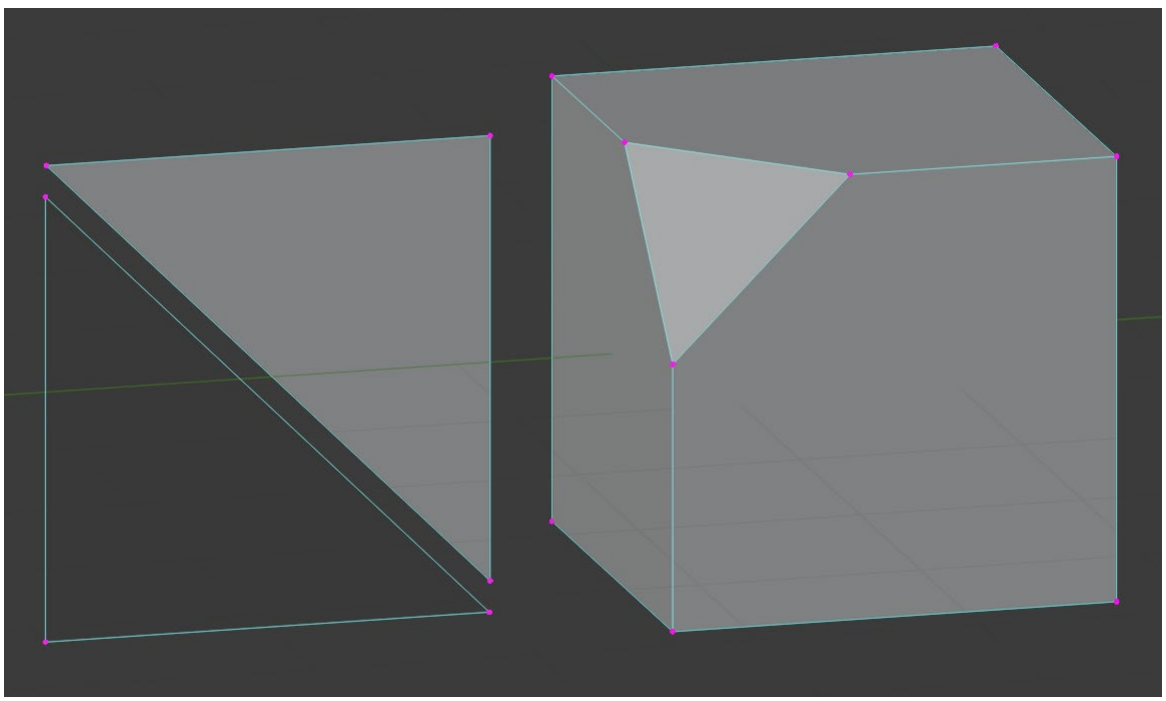

Figure 1. Polygons and n-gons

Figure 1 shows three objects and how their surfaces are rendered, with backface culling enabled in Blender to show transparency. On the mesh to the right a polygon face is formed on the edge of a cube to show flat surface. The other three sides of the cube become n-gons.

2.2 Normals

According to the Blender Manual, normal in 3D mesh is a direction that is perpendicular to the surface of a Face on which they lie. That direction also controls how light bounces off the meshes surface, and by default models have hard edges with flat lighting according to the normal’s angle, but there is an alternative light shading way. By enabling Auto Smooth in Blender on a mesh, it is possible to render edges smaller to a specified angle as smooth transitions that resemble a single surface (Blender Foundation 2023a).

In his article on the topic of normal maps and baking, Lemos (2020) further explains that the process of making smooth transitions between polygons is called setting smoothing groups, and it communicates to the 3D editor how soft or hard transitions are to be perceived. During the modelling process this principle can be used to make smooth low polygon surfaces, as is shown in Figure 2.

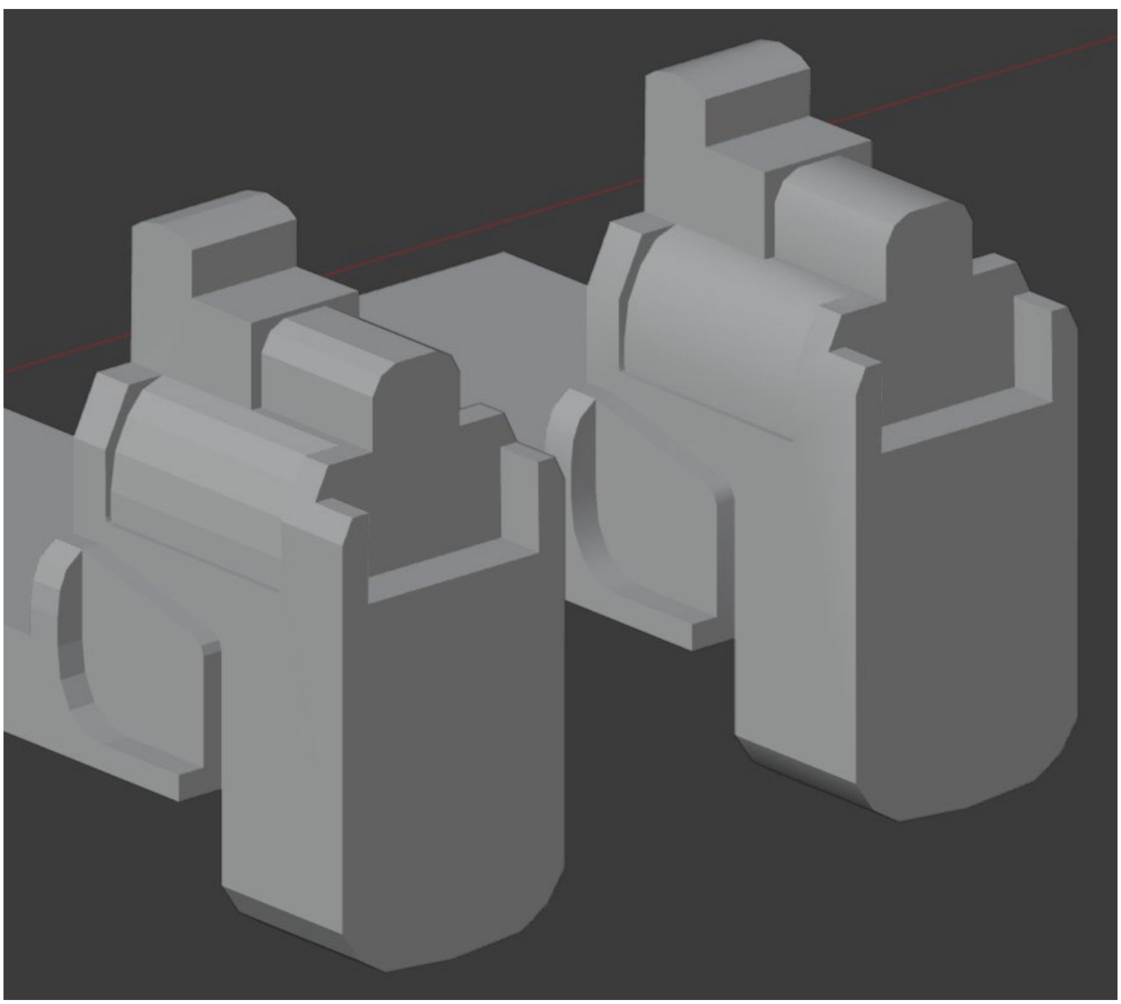

Figure 2. Part of the model with flat shading (left) and auto smooth shading (right)

As Figure 2 shows, certain parts of the mesh are smooth while others are hard. The principle behind this feature is important to understand for later baking stages, as a fake smooth surface consisting of two polygons is less resource intense than a modelled smooth transition of two hundred polygons. The concept is linked directly to baking and helps with adding detail to the model with no performance cost.

2.3 High and low polys

The whole process of modelling for real-time game environment is based on the concept of detailed high polygon meshes and simple low polygon meshes. This is due to how highly detailed surface information of a model can be transferred to a low poly version of itself, which drastically reduces the workload of a real-time engine. To summarize, high poly models are detailed and dense with geometry, which makes them realistic when rendering. Low poly models are simple models, with only the necessary geometry, which is easy to calculate and render in real time environments.

According to an article on the topic of high and low polys in 3D modelling (2022) published on Queppelin, an artist does not have a clear number of polygons required to differentiate a low poly from its high counterpart. The standards and circumstances for each model can be the deciding factors in this question, as a model made for a mobile game can be considered high poly for its purpose but would be a low poly in a computer shooter, for example. As technology advancements in 3D field are made, what is considered high poly today can be a low poly in the future (High poly vs Low poly… 2022).

Because each videogame studio will have their own polygon amount standards for different 3D game assets, the sought-out quality is making use of each polygon to represent accurate shapes for the players. Each in-game model has its own purpose and is made to accommodate that with various level of polygon detail. For example, for mobile games' characters Unity engines documentation suggests a number of polygons between 1500 to 4000, and the more characters are on the screen, the fewer polygons each should have (Unity 2017). Shooter games can have much higher polygon numbers but must be heavily optimized in their own technical aspects to perform with high number of frames per second.

The importance of 3D meshes polygon count, however, must be addressed for the purposes of making game assets. As mentioned before, higher number of polygons affect in-game performance negatively. The more polygons a model has, the harder it is for a game engine to perform. Upon export, the model's polygons are always split into triangles, also called triangulation. This is due to graphic hardware rendering triangles faster. Because of that, game artists often refer the number of the model's triangular polygons specifically as the poly count (Polycount 2021). Polygons, however, can be comprised of any number of vertices and form any shape, and still count as one polygon, illustrated in Figure 3.

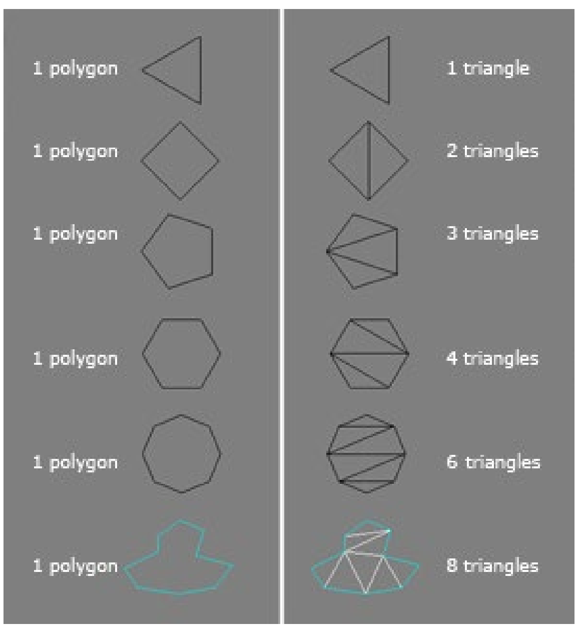

Figure 3. Visualization of difference in the number of polygons and triangles by a user Michael "cryrid" Taylor (Polycount Wiki 2021)

Interestingly, on the same Polycount Wiki page (2021) there is a statement that the polygon amount does not affect performance directly. The true performance hitter is the number of vertices that the polygons are comprised of. This is due to already mentioned smoothing group changes between polygons being treated as a physical separating breaks in the model's surface by game engine renderers.

For breaks to occur, the vertices on the break point need to be duplicated. The duplicated broken-down mesh is then sent in parts to the graphics card. Large number of vertices that reduce performance also comes from overuse of UV seams and material changes (Polycount 2021).

The objective definition of a low and high poly is not necessary for this research, but what is necessary is the understanding of key factors that differentiate a model in such way. As mentioned before, the low poly is an optimized mesh of a high poly, and the way its geometry is built must meaningfully contribute to the shape of the model, but still have a lower polygon amount than its high poly. The number of polygons for each project can vary to anything from one polygon to tens of millions, and one such case is the model made during this research shown in Figure 4.

Figure 4. A low poly on top and high poly on the bottom with statistics

While the high polys detail in Figure 4 might not be visible at a first glance, it is nonetheless a required part of the workflow for realistic weapons. The high number of polygons on the edges of the rifle create a synthetically manufactured look that weapons in real life have due to their making process. This is one more of the considerations that a weapon artist keeps in mind when creating their low and high polys, with a plan on how to transfer the detail information on a low poly via the process of baking. This topic will be revisited in the baking section of the thesis.

2.4 Topology

Within a 3D mesh, the combined collective of connected and distributed Vertices, Edges, and Faces forms a cage looking structure that will render its shape, which is called Topology. Two models can look the same when rendered but have different Topology, according to Turbosquid's 3D resources (Turbosquid 3D 2023). A simple demonstration of that is in Figure 5.

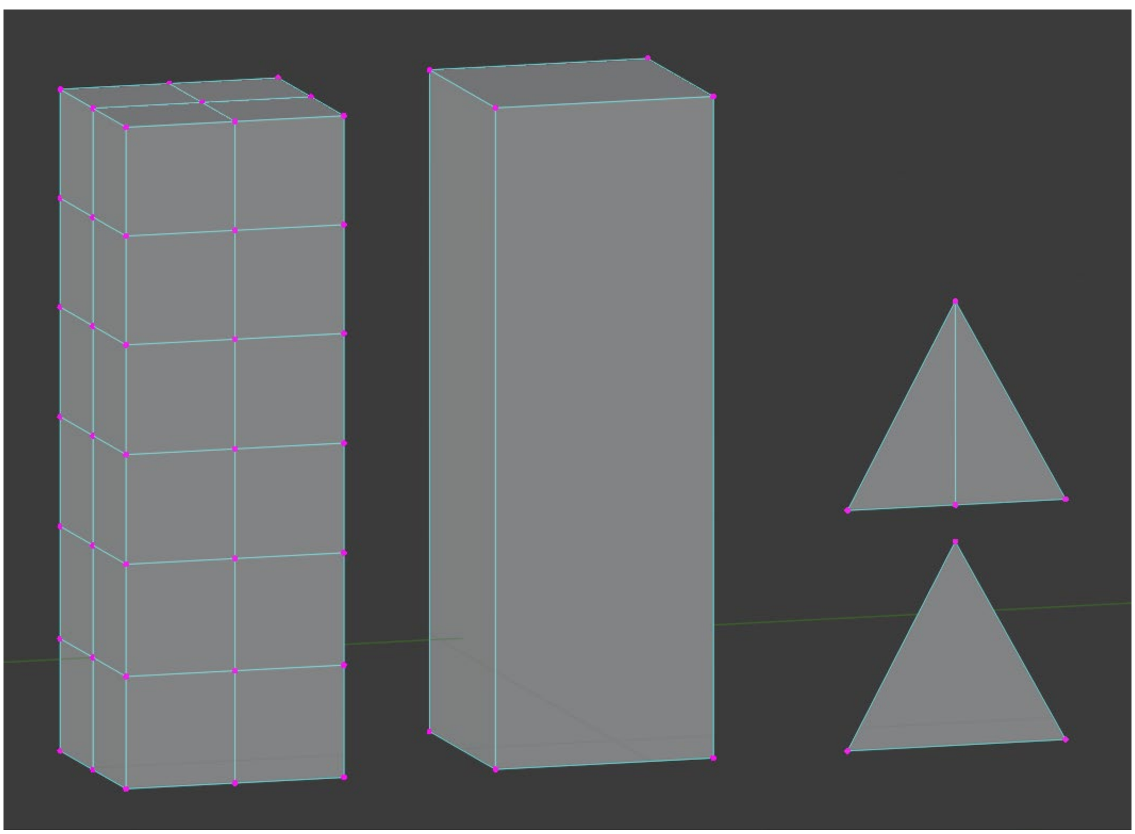

Figure 5. An example of identical models with different topology

In videogames, models have different topology based on how much deformation they need to have. Dynamic models that will move and perform actions, like characters, must have Topology consisting of rectangular quads, spread as evenly as possible. The quad requirement is also due to triangles and Vertices branching in five or more directions, also called Poles, causing a pinching effect, as mentioned in the Polycount Wiki's page for character base meshes (Base Mesh 2020).

While quad topology is not necessary for making static non-deformable meshes, such as weapons, it is good to follow as a guideline for clean topology. Sometimes additional geometry that might not affect the shape of a model can be used to support it on later high poly stages of the workflow. Most importantly, however, is the number of polygons of the elements that are closest to the viewer in a hypothetical FPS game. That means that the closest and most prominent elements will require denser Topology even in a low poly, while areas that are not visible or areas that do not have many details can be optimized with lower number of polys.

2.5 Modelling for hard surfaces

The following section will examine common hard surface modelling techniques used for our weapon workflow in Blender. It will start by explaining the topic of box modelling, and how its philosophy is used for the paper's weapon creation workflow. The section will then present each most used tool for the thesis's weapon and explain their purpose.

In the article What Is Box Modeling (2023), it is explained that the idea behind the method is to turn a primitive shape into a complicated model. The original primitive does not have to be a box, but the more it corresponds to the model that is to be achieved in the end, and the simpler its original topology, the better results are. This is because it is simpler to add complexity to a primitive than to reduce it. The required steps are usually identifying primary shapes of the reference and making a rough draft with simple shapes. After the big shapes come the details, which can be chamfered corners on the model, insets of various shapes, repeating patterns that would be left by manufacturer in a real-life item. Lastly the resulting meshes topology is optimized to reduce unnecessary topology and eliminate n-gons from the model (What Is Box Modeling? 2023). All these steps can happen either simultaneously or in steps while working on the model, as will be shown in Figure 6.

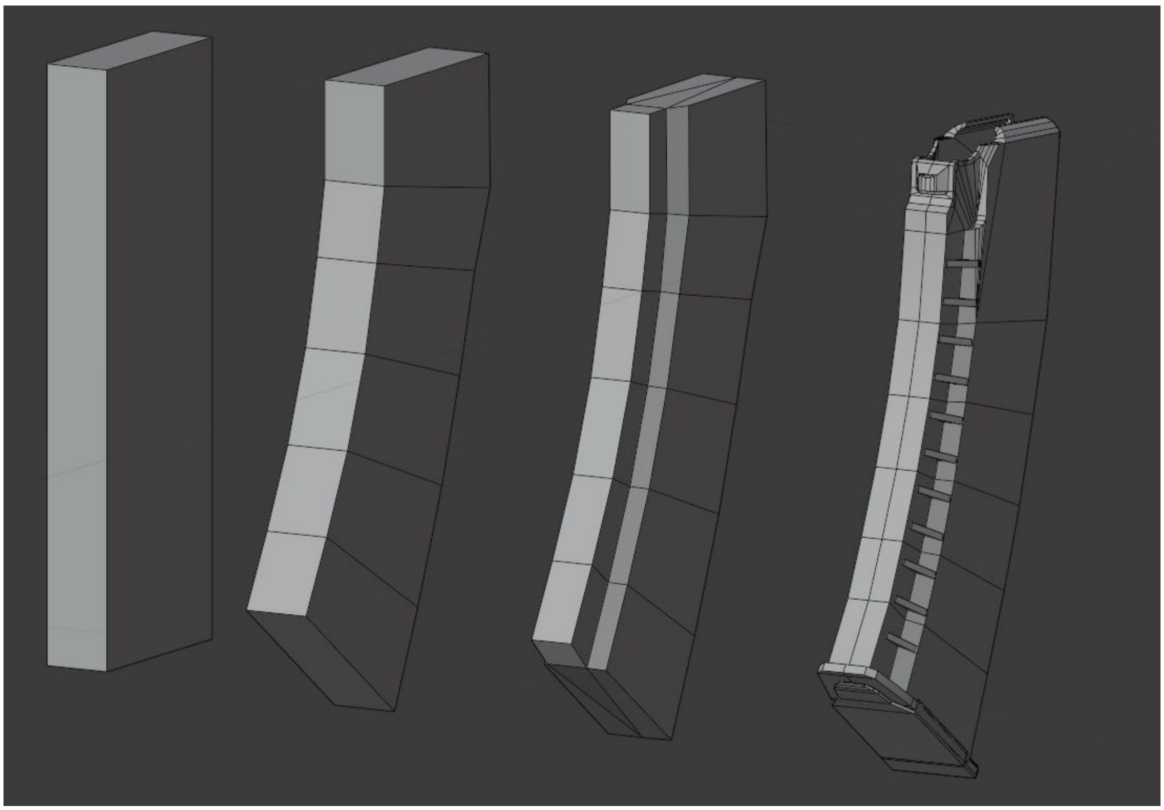

Figure 6. Example of box modelling turning a rectangle into a weapon magazine

To explain further how the rectangle in Figure 6 is made into a magazine the paper will examine hard surface modelling principles and methods. The focus will be on question like how the methods can be used, why they are used, and their fundamental properties. In the scope of the research, the examined modelling program is Blender, but the same concepts can be found under different name in most similar 3D software suites. The full list is comprised of the following topics:

- Object origin
- Mirror modifier
- Edge loops
- Bevel edges
- Extrusion

- Inset faces
- Merge
- Subdivision modifier
- Booleans

The first concept to understand is the object origin, which is created with every new mesh and assigned as an individual origin point in the center of the meshes mass. These are also called pivot points and can also be freely altered to any other coordinates in the editor. Multiple objects can share origins with same coordinates, and of course share an origin as part of one mesh (Blender Foundation 2023b).

Origins make it possible to perform mesh transformation according to an anchored point in 3D space. This is important to understand for object transformation, and to utilize congruently with procedural tools that will be presented next. Figure 7 from the Blender manual shows how origin points affect transformation.

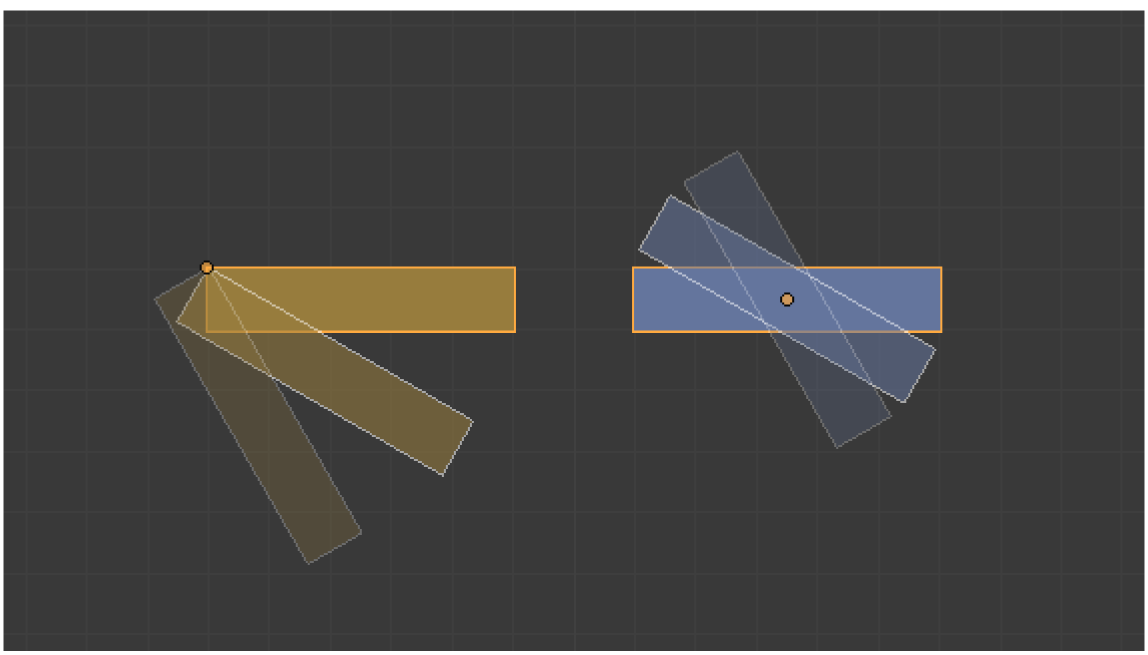

Figure 7. Showcase of how rotation is different according to its origin (Blender Foundation 2023c)

Often origins enable precise transformations that can make the model look more synthetic. Adjustments like these are part of the detail that simulates the resulting asset to look like the weapons done in real manufactories and workshops.

The mirror modifier is what makes the natural continuation to the topic of origins, as it mirrors the object along one of the users defined local X, Y and/or Z axes across the object origin (Blender Foundation 2023d), shown for better understanding in Figure 8.

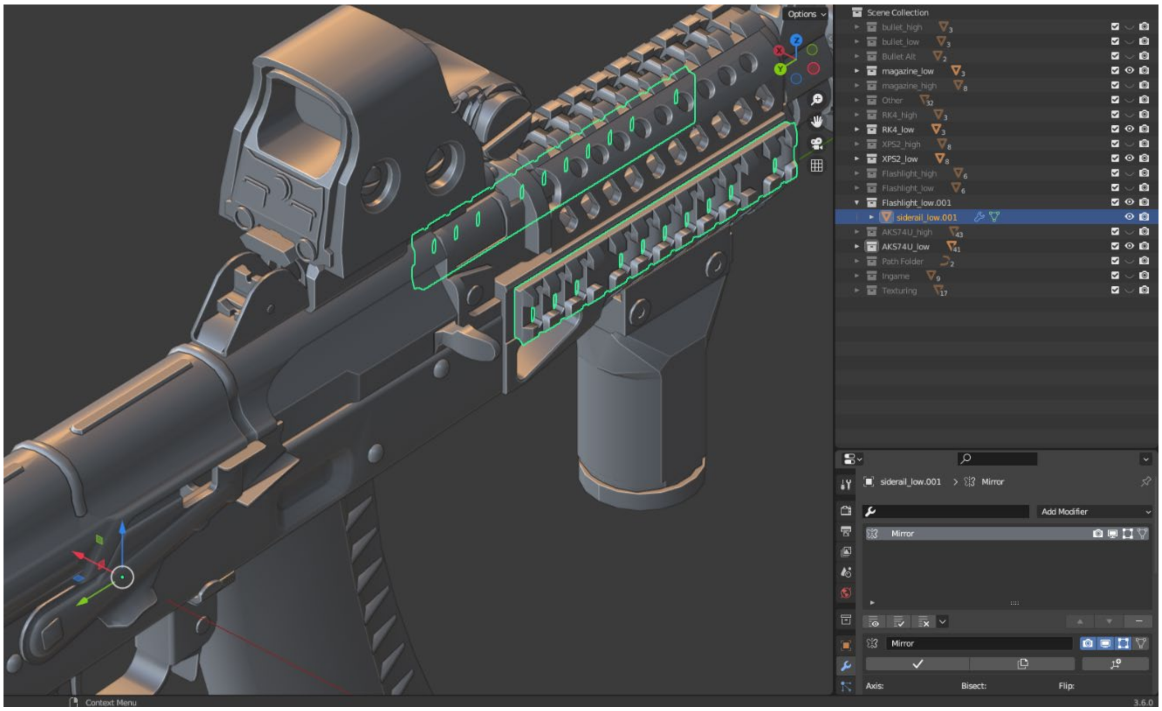

Figure 8. Outline of a mirrored picatinny rail with the origin in the center of the model.

While it is not the only way to use mirroring, it is one of the easiest ways of creating precise symmetrical objects and showing the made changes on the mirrored side in real time.

Edge loops are edges that are connected to each other in a continuous track from the first vertex and to itself and are not broken by poles. They can be selected to form an edge loop. The full loop can only occur when a selection has one possible path forward along the loop, and if there are more possible variants the selection stops there. The same principle applies to face selections, forming face loops (Blender Foundation 2023d), as can be seen in Figure 9.

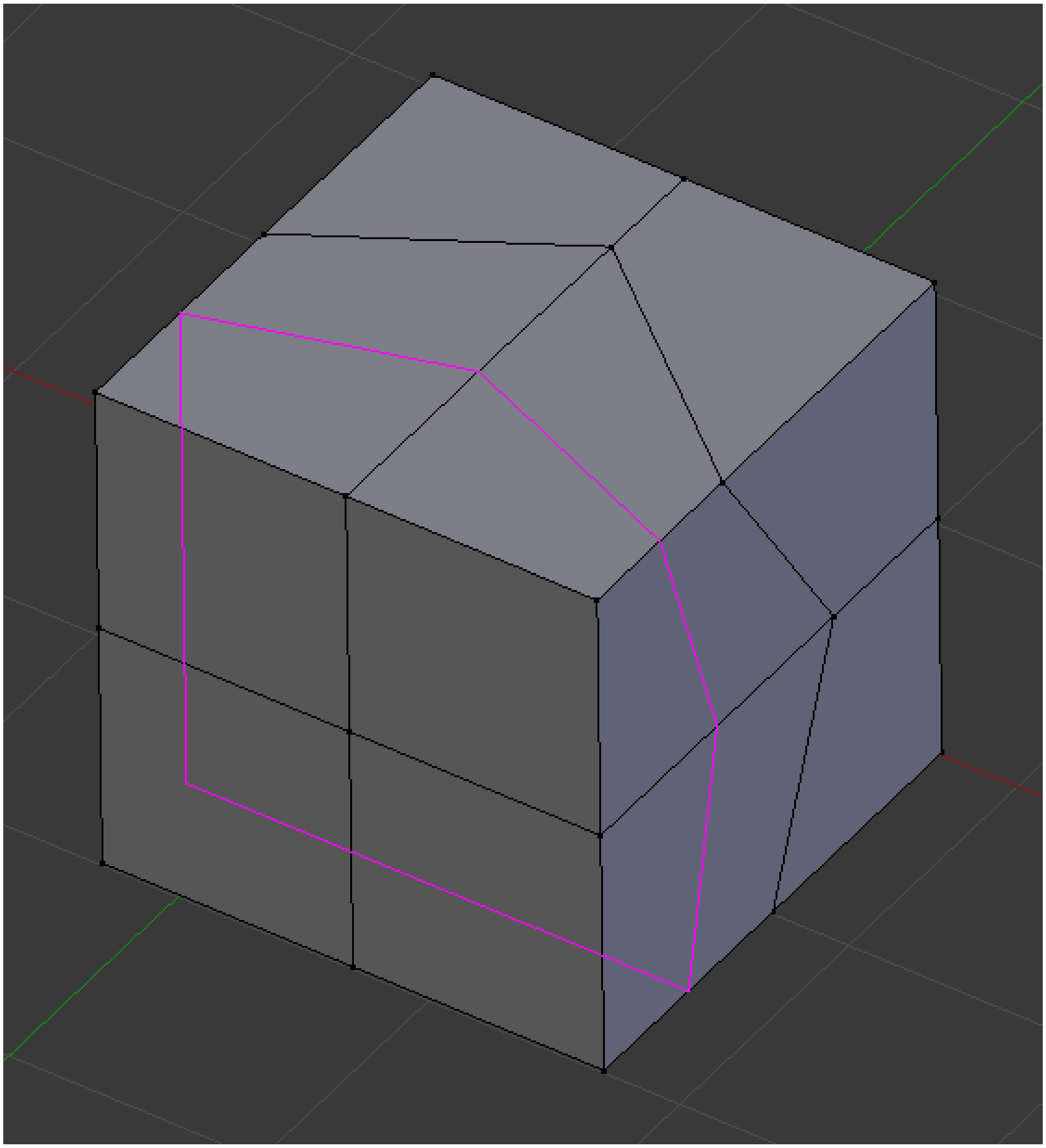

Figure 9. A preview of edge loop cut with a previously cut edge loop (Blender Foundation 2023d)

Between the edge loops users can make a new loop cut, forming an additional edge loop. This can be done horizontally or vertically, as shown in the previous example of turning a rectangle into a magazine. Adding edge loops made it possible to increase the topology for the rectangle box, enough to change its silhouette to a magazine.

Bevel edges tool allows Blender users to chamfer an object’s edges or vertices. The object can be chamfered in one or multiple segments of faces. An example is shown in Figure 10.

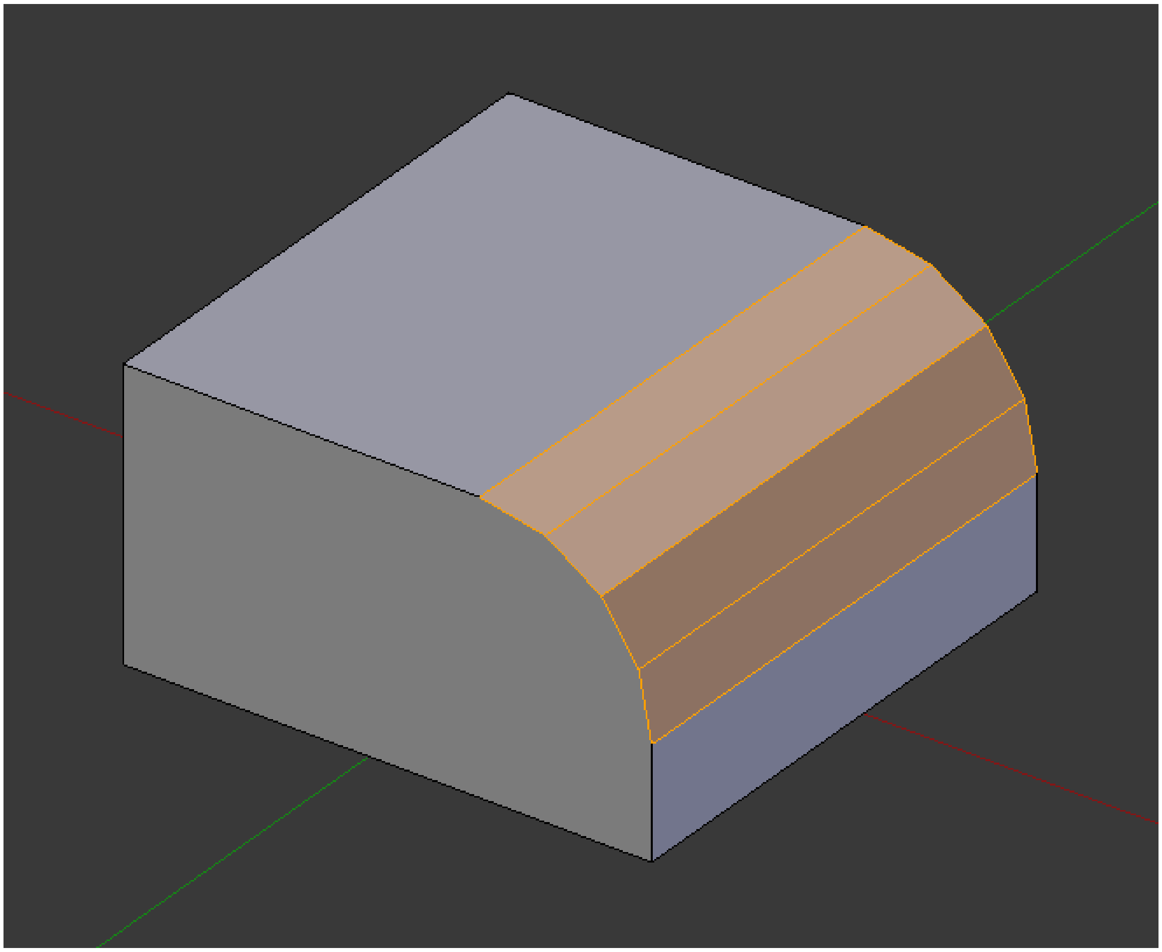

Figure 10. A bevel of four segments (Blender Foundation 2023f)

The tool works on edges with precisely two adjacent faces and will detect edges in vertex and face selections (Blender Foundation 2023f). It is important for the realism aspect of the workflow, as perfect sharp edges would look too unreal and boring.

Extrusion tools in Blender allow the users to duplicate vertices and keep them connected to the geometry of the mesh, in turn allowing users to add new forms. Figure 11 displays it in action.

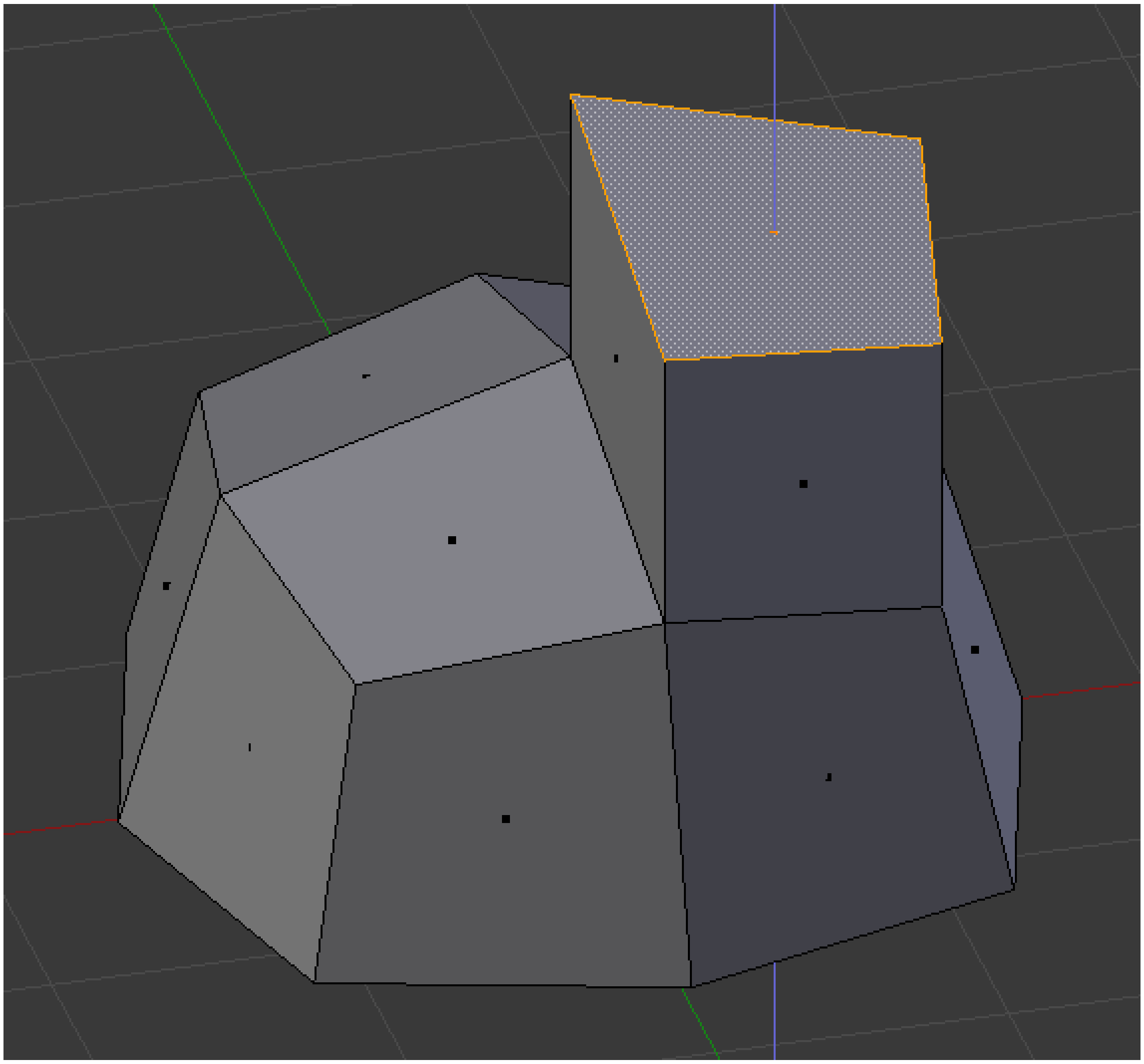

Figure 11. A face extruded along the Z axis (Blender Foundation 2023g)

The manual stresses the significance of this tool, even calling its importance paramount. Extrusion can also be done inside the model, crating holes, and other complex geometry. This tool can be locked according to the editor’s 3D coordinates axis for better precision or made to extrude along the normal of the mesh (Blender Foundation 2023g).

Assisting extrusion is the inset faces tool. Much like edge loops, it creates loops relative to the selected geometry. Taken from Blenders manual and shown as Figure 12.

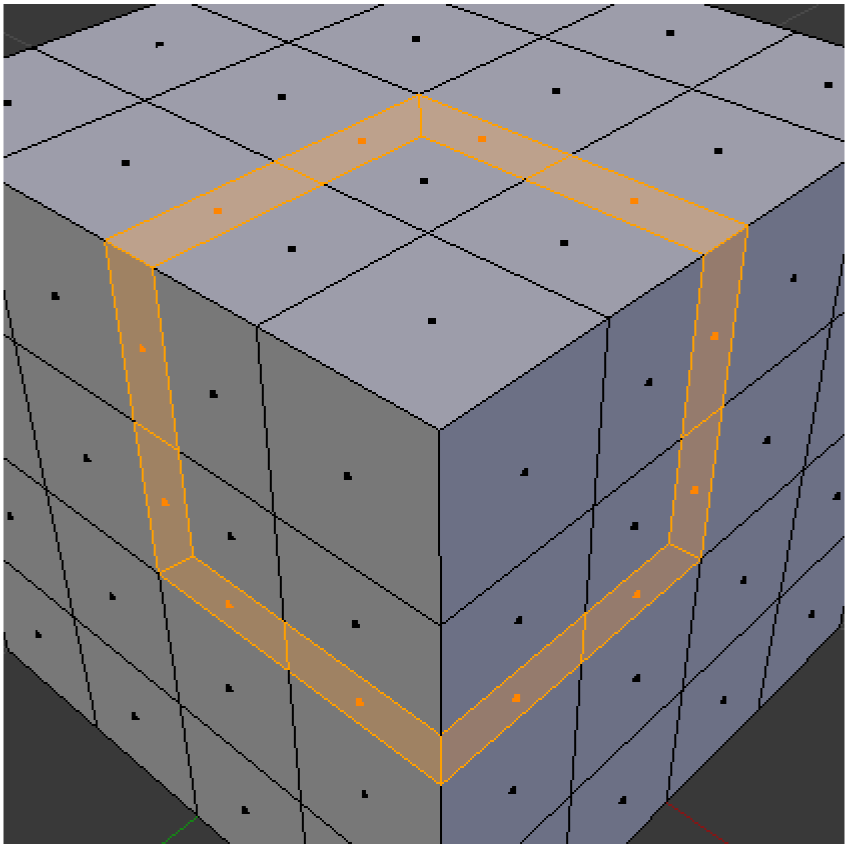

Figure 12. Inset in a selection (Blender Foundation 2023h)

The selection can be as complicated as the user chooses it, but for anything to happen it must at least have a face (Blender Foundation 2023h). Throughout the workflow this tool is used in combination with various extrusions, bevels, and subdivisions.

Merge tool allows to merge multiple selected vertices into one vertex, dissolving or deleting some vertices, edges and faces in the process. The tool can be adjusted to choose the location of the last vertex to be merged at. The choices are center, 3D cursor, median center of selected vertices, and finally first and last vertices selected (Blender Foundation 2023i). In the paper's case study model this tool is used mostly for optimization purposes, to simplify the mesh's topology for real-time rendering.

Subdivision modifier is one of the main tools used for creating high resolution models. The modelling method that utilizes subdivisions is called subdivision modelling, or sub-d method for short. Many methods of sub-d modelling exist, but the one utilized in this thesis case study is the Catmull-Clark subdivision, which is an algorithmic division of a polygons surface into smaller areas based on its closest vertices. Invented by Edwin Catmull and Jim Clark in 1978, this method is famous for its use in Pixar animations (Dassault Systems, n.d; Blender Foundation 2023j). Illustration of the process is shown in Figure 13.

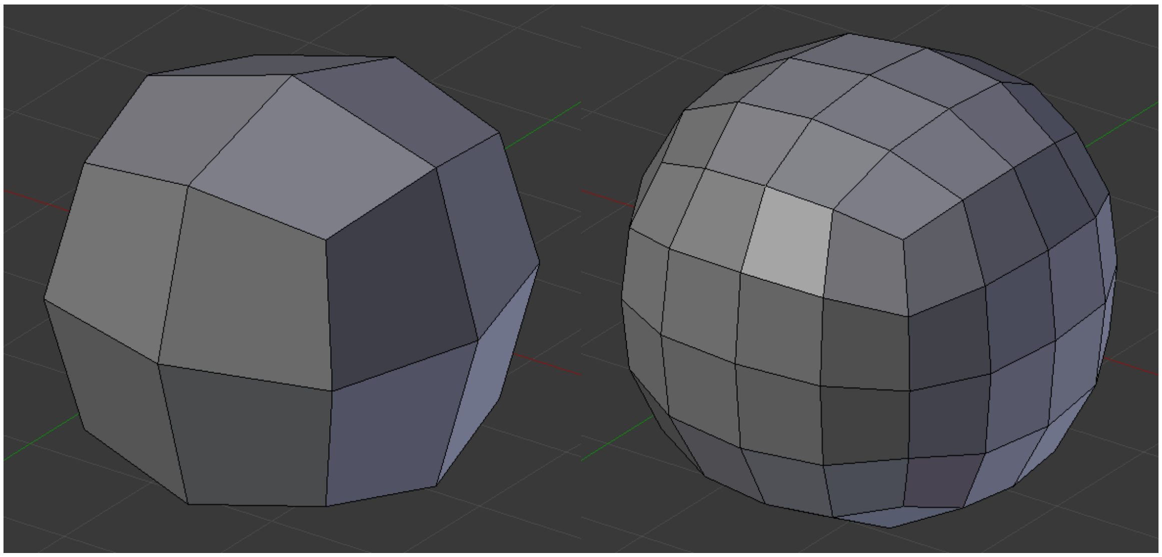

Figure 13. Mesh before and after subdivision (Blender Foundation 2023j)

The example shown is a basic subdivision of the first step, but it can be tuned further to subdivide each of the new segments and so on. Often clicking the subdivide button more than necessary results in Blender trying to calculate a model with millions of polygons, long loading waits and crashes. Subdividing the model is used in thesis project, and its role will be expanded later in the explanation of making the high poly for the case study model.

Another important tool is the Boolean operation. It can use one mesh to cut, unify or create insets into another mesh. These operations require at least two meshes, and result in one, as can be observed in Figure 14.

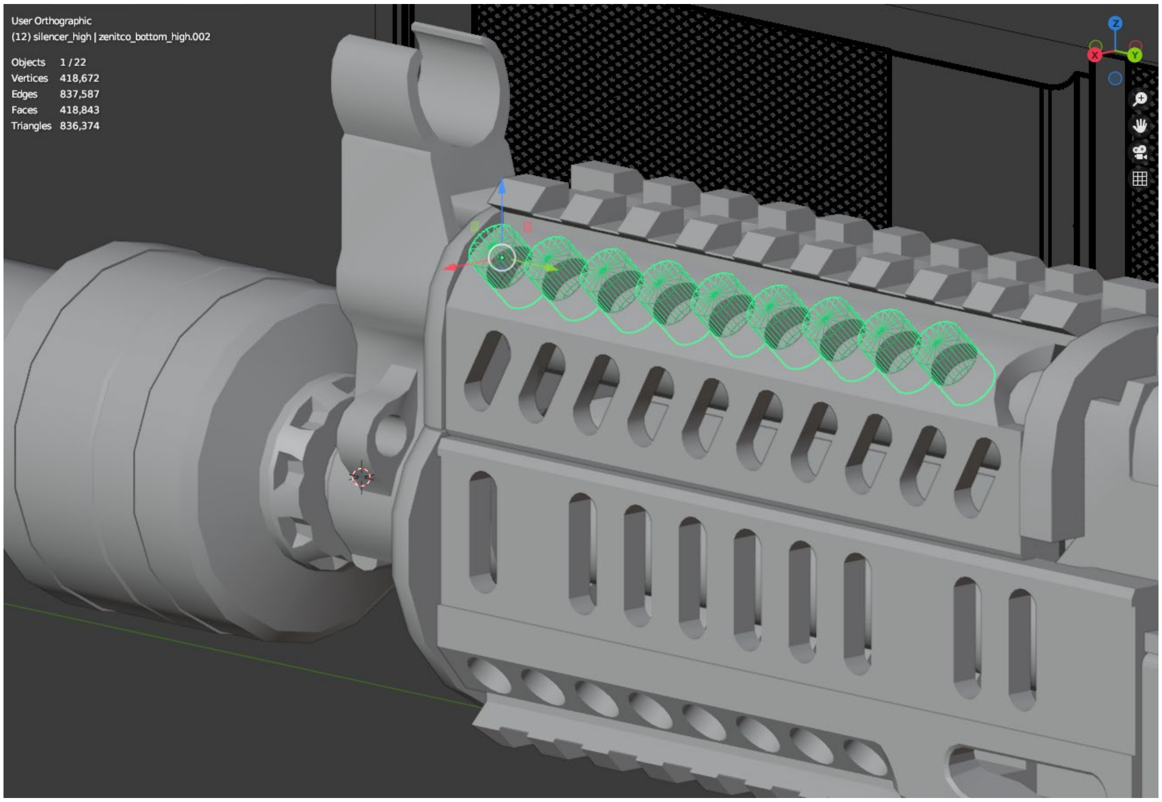

Figure 14. Boolean modifier creating repetitive detail at an angle.

As the Blender manual (2023k) states, the Boolean can perform operations that are either too complex or time consuming to do by manual mesh editing. Often in 3D editing suites Booleans can be non-destructive and preview changes in real time. Essentially, they can make artists' job easier and more accurate. In an interview for a Weapon Art Tips article (Burton 2020), lead weapon artist of C77 Entertainment Patrick Sutton states that he uses Booleans for the beforementioned attributes, as it allows to correct mistakes in realistic weapons and iterate on shapes in fictional ones.

2.6 References

In his article on creating an assault rifle in Substance 3D painter, Resenberger-Loosmann (2023) recommends to study and compare high resolution real-life photos of the weapon and its parts. To see the width of the objects, technical specifications, and to understand how the weapon is manufactured, Wikipedia articles are a recommended source, along with forums and web stores containing images of the guns they are selling (Creating an Assault Rifle… 2023). Photos also have their own caveats, as they are almost never taken orthographically, resulting in depth that skews the perspective. Inspecting other artist' 3D work is to be avoided but it is possible. However, doing so requires critical approach as examining other artists works limits one to the quality of the inspected art.

Additionally, the geometry of the online assets may not be completely accurate, and often 3D models can have multiple purposes. One such case are movie props that have higher detail standards and are too heavy to run in a real-time environment (3D Modeling in Games vs Movies… 2019).

In the best-case scenario, a 3D artist would have access to the real weapon itself, or an accurate replica that could be at least field stripped. Even better, shooting the gun could further help to understand not only the function of different parts, but to see how a 3D model could be prepared later for rigging and animating.

To control the dimensions when modelling and preserve realism of the weapon, the artists can make a collage with relevant photos and information that they need for any specific details that they wish to keep close at hand, showcased in Figure 15.

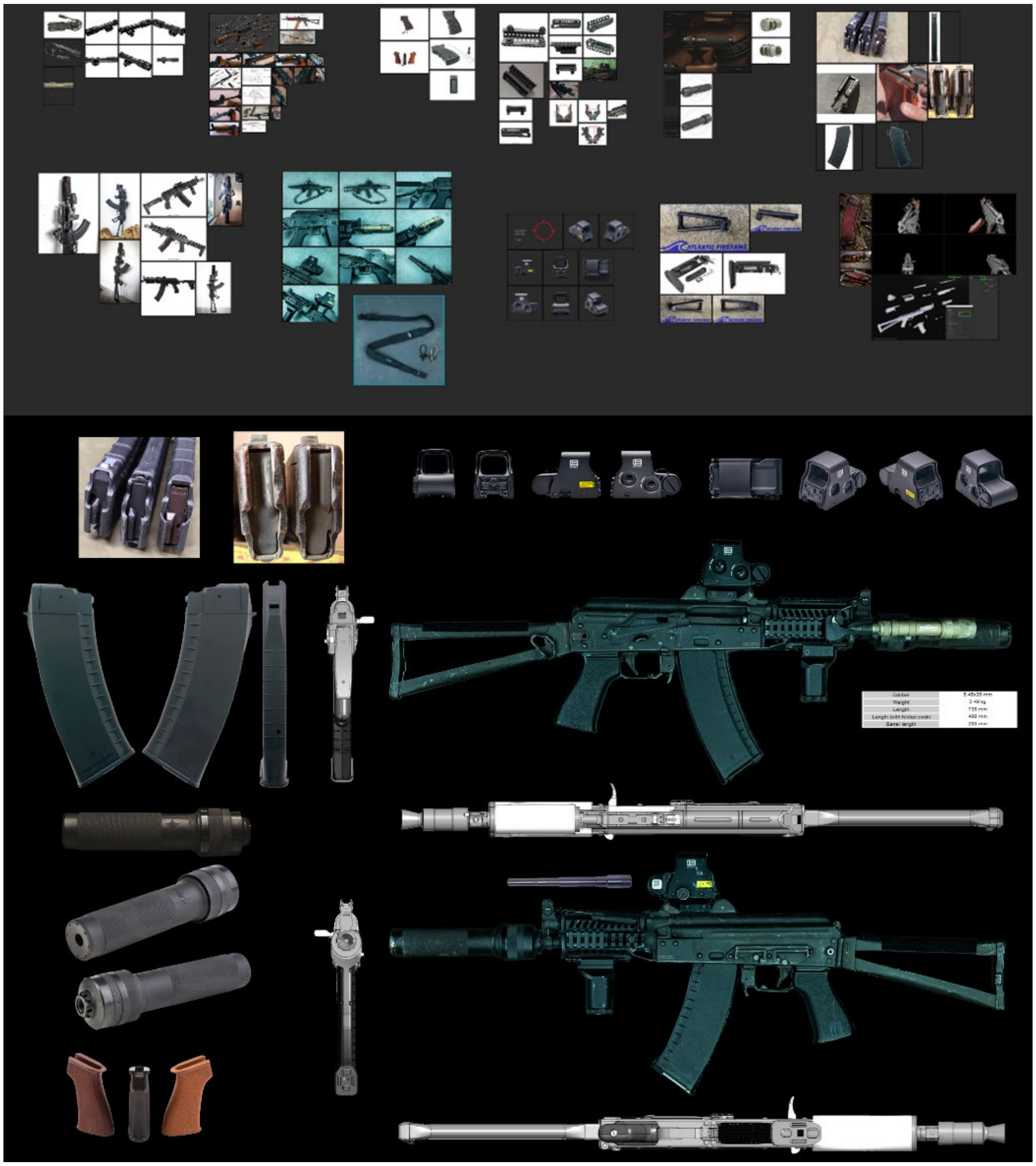

Figure 15. Reference gathered in Pureref (top) and assembled as a single image (bottom)

A large number of references is not a must have, and sometimes can even have detrimental consequences when modelling realistically. The relevant information is isolated in the collage as an attempt to visualize the weapon and its parts from top, bottom and both sides. It utilizes both realistic photographic and digital media assets and fulfils an idea of a cheat-sheet for the artist to refer to.

Good references from different angles and formats, such as video and still images, not only help to translate their shape into the 3D editor, but to create a lifelike surface material for the asset in later stages, as the Figure 16 demonstrates.

Figure 16. Close up images of AK rifles with different material and lighting conditions (Zenitco n.d; T.Rex Arms 2021)

The figure also shows how closely similar materials behave under different lighting conditions. Considering the physical behaviour of the weapons surface can enable the artist to create textures that can communicate a story and gameplay meaning to the players. Such example could be a heavy edge wear, that would communicate both how extensively the weapon was used and does it need potential repairs.

2.7 Realistic dimensions in 3D

The thesis attempts to recreate the AKSU rifle in 3D with accurate dimensions. This is done because keeping realistic scale helps throughout the whole process of modelling, with both big main shapes and small details. Incorrect relationship between scales of each part of the weapon will look off-putting to both a gun enthusiast and a person who never held a gun before. With imaginary props this issue is somewhat avoided, but realistic assets that are often seen in movies and videogames has less room for inconsistencies. According to Kevuru Games Lead Weapon Artist Ivan Dikhtyar (2022), because of the way weapons function, there is not much room for artistic expression when making realistic weapons.

Gathering information on weapons' dimensions from Wikipedia was mentioned before in the thesis, and is corroborated by previously mentioned weapon artist Sutton, which stresses that looking for correct information on weapon cartridges and using that as a reference to measure the rest of the weapons' parts is a way to avoid unrealistic dimension proportions during the workflow. (Burton 2020). After analysing these statements, it seems that the most notably important elements to keep scaled correctly are the length and width of the whole weapon, length of its barrel, and the dimensions of the cartridge and ammunition that it uses. Due to 3D editors being able to simulate dimensions precisely as in real world Figure 17 will demonstrate a relatively simple way to set up such references in Blender.

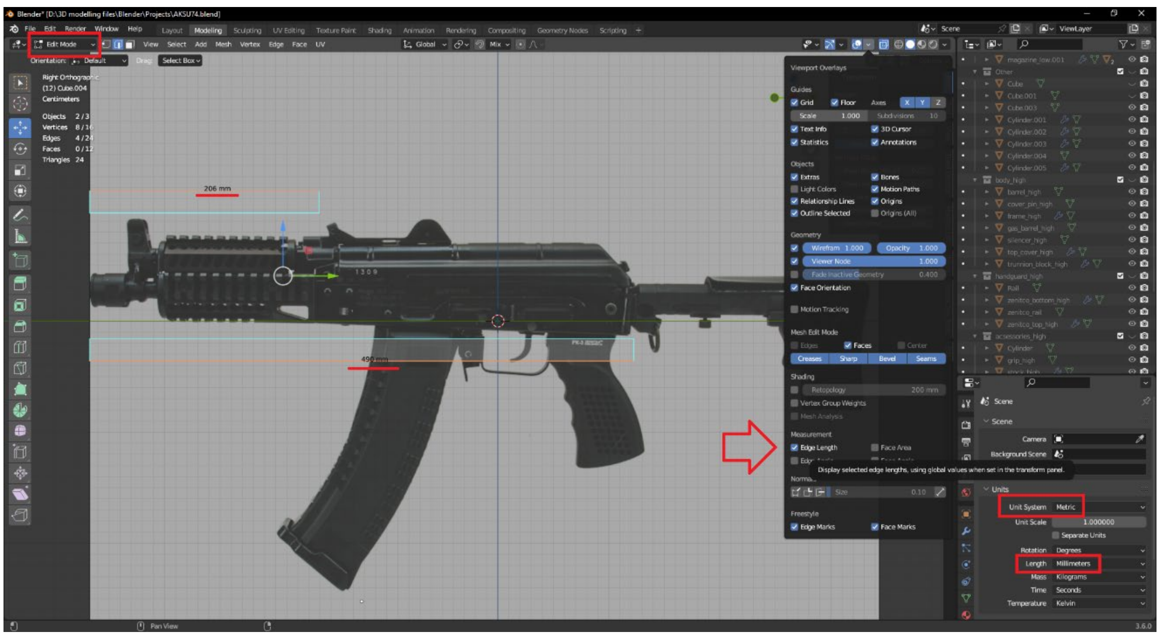

Figure 17. Screenshot of modelling view in Blender 3.6.0 with relevant tabs open.

In Figure 16 the Edge Length is enabled in the Edit Mode Viewport Overlays options. Additionally, in the Scene tab the Unit System is set to Metric, with Length displayed in Millimetres. The image is then scaled according to the dimensions of the rectangle's length. With the right dimensions the first block-in parts can be made.

Even if it is impossible to get the dimensions of each part from every angle, knowing one of the parameters and having access to high resolution visual references can help to approximate the rest of the unknown dimensions, exampled in Figure 18.

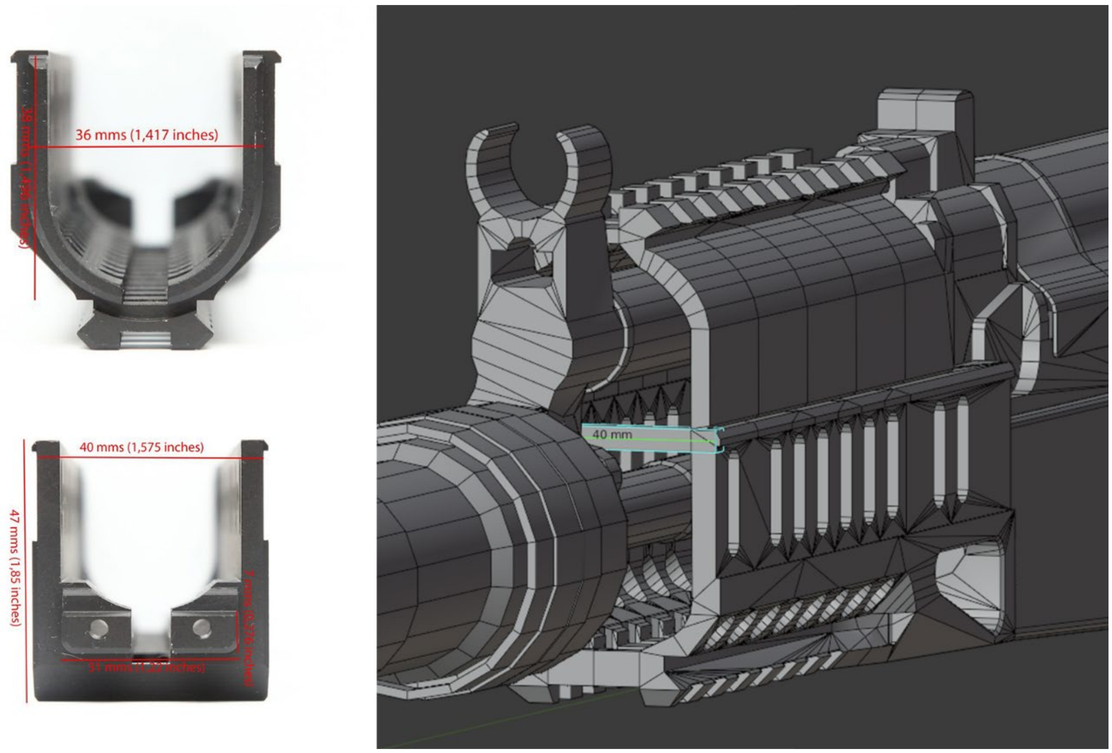

Figure 18. Dimensions of real handguards, and how they influence the 3D shape (Ivan the Bear n.d.)

Figure 18, a 40-millimetre rectangle is placed between the rails to help measure the position of both handguards in relation to the rest of the weapon. The width of similar Zenitco lower handguards is measured in millimetres from back and front. Dimensions for the reference's handguards match their dimensions, according to the reseller (Zenitco n.d.).

2.8 Low poly result

Following the design and modelling principles, as well as utilizing the mentioned tools for hard surface modelling, the case study low poly model results can be examined in Figure 19.

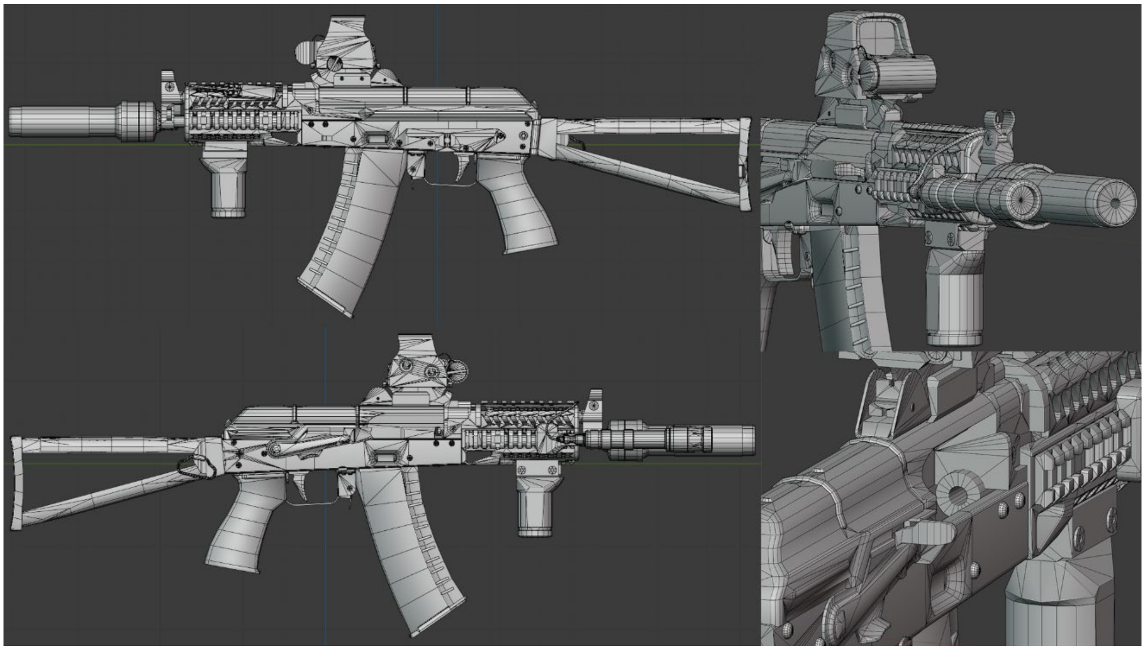

Figure 19. A low poly of an AKSU-74 assault rifle

The model is one thousand polygons shy of forty thousand, and the results turned out to be too low poly for a FPS asset, and too high poly for a low poly asset. The low polygonal shape is most visible in cylindrical parts such as suppressor, front handle, and grip. The standards achieved on this stage have more in common with the assets during the seventh generation of consoles in 2005, such as Ps3 and Xbox 360. This can be remedied by remodeling the mentioned parts, or keeping the model as is and hiding the imperfections by smart texturing. The shape of the weapon however is believable and looks like its real-life counterpart.

3 PREPARATIONS FOR TEXTURING

In this section the thesis will examine the texturing preparations steps of the weapon workflow and why these steps are taken. These consist of methods that go into UV unwrapping, making a high poly version of the asset, and consecutively baking said asset.

3.1 UV unwrapping

According to Thomas Denham’s article on Concept Art Empire (n.d.), at its core UV unwrapping, also called UV mapping, is a flattened-out information of a 3D models surface. The U and V stand for horizontal and vertical axis, since the usual X, Y and Z coordinates are used for 3D modelling. The finished model is seamed along the edges according to the modeler's needs, and the resulting pieces of model's surface turn into UV isle, also called UV shells. Because a mesh's mapped surface information must be two dimensional, this mapping process unwraps it, creating UV shells for textures to be applied on top of them (What is UV Mapping & Unwrapping? n.d). Simple visual explanation of the process is seen in Figure 20.

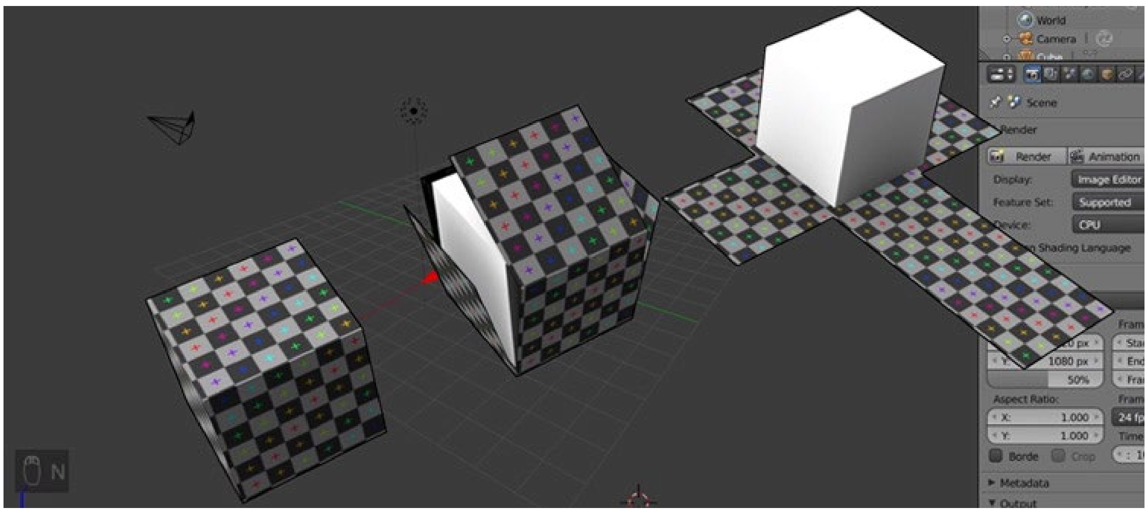

Figure 20. Visualization of a cubes UV unwrapping (Blender YouTube channel 2017)

While the actual process of correctly seaming and unwrapping complex geometries is one of the more complicated and technical parts of any model creation, the results of the process itself can be compared to unwrapping a paper origami model to paint on it between the creases on a flat surface.

The difficulties arise when seams need to be made on flat surfaces due to the UV stretching otherwise, and when packing the seamed UV shells the texture atlas. For the last part it is important to keep the atlas resolution in mind, because when texturing, the background color of the texture atlas could bleed into UV shells, creating unwanted seams on the texture (Polycount 2017). To avoid that, each atlas resolution has its own suggestions for UV shell padding, illustrated in Figure 21.

Figure 21. A 512x512 texture down sampled progressively (Polycount Wiki 2017)

In the figure, the texture has no edge padding to prevent color bleeding from the unused black areas. According to the Polycount Wiki (2017), the suggested minimum paddings for texture atlases are as follows:

- Two pixels for 256x256
- Four pixels for 512x512
- Eight pixels for 1024x1024
- Sixteen pixels for 2048x2048

The common practices of UV unwrapping for game weapons, among other topics, can be found on Michal Kubas web publication Gun modelling for FPP games. The most notable advice for the research case study model that was followed are mirroring UV's and scaling unseen UV shells to save texture space (A guide to weapon modelling… n.d.).

The web page has more information on creating optimized weapon assets in all its aspects related to game production that was deemed outside of the current research scope. However, it is still important to examine the mentioned UV mirroring in that was used, with an example in Figure 22.

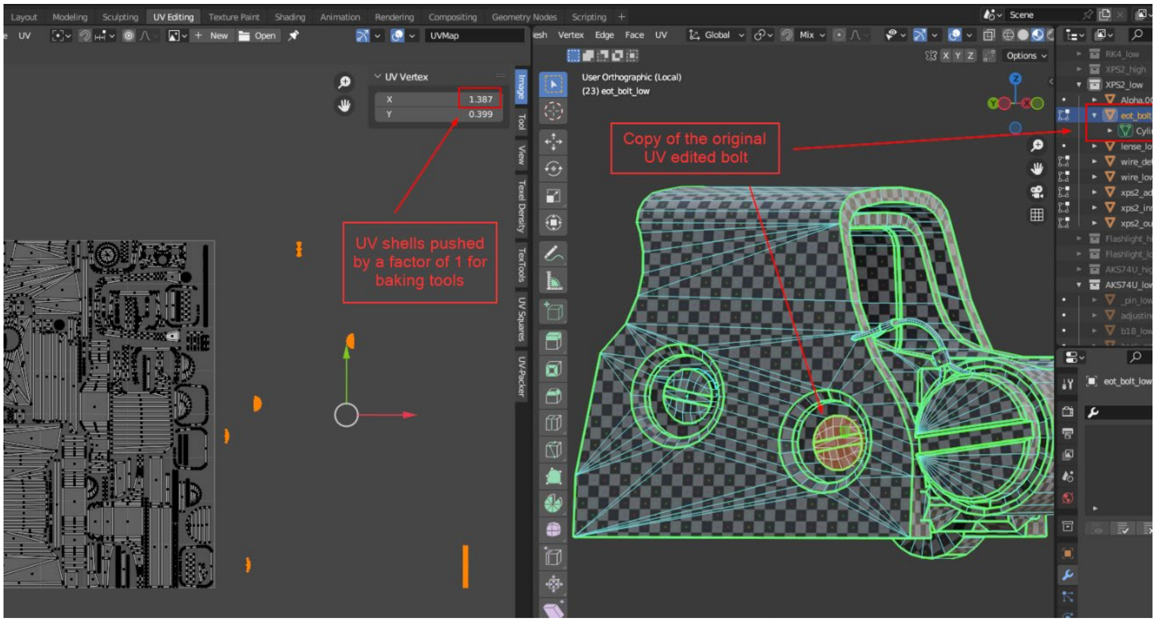

Figure 22. Copied UV shells pushed one unit on the X axis on the UV map

The copied shells are being ushed by one unit, because the copied model of the bolt is a repetitive element that can share UVs with the original model, save overall texture space for other elements and be virtually indistinguishable in the final game-ready result. Leaving the copied UVs over the originals can result in unwanted baking artifacts. Because of the way how baking software interacts with UV maps, if the copied or mirrored UV shells are moved any whole number to the side, they can be safely left in their new positions after the bake, and still be correctly mapped. Easy way to mask the nature of the copied bolts UVs is to rotate it in a different position, exposing angles that are not seen on the original model (Polycount 2021).

The scaling of unseen UV shells happens after unwrapping the whole model, and manually selecting the shells that need to be scaled, after what the UV map is packed again utilizing the extra space on the texture atlas. The result of the UV unwrapping process is showcased in Figure 23.

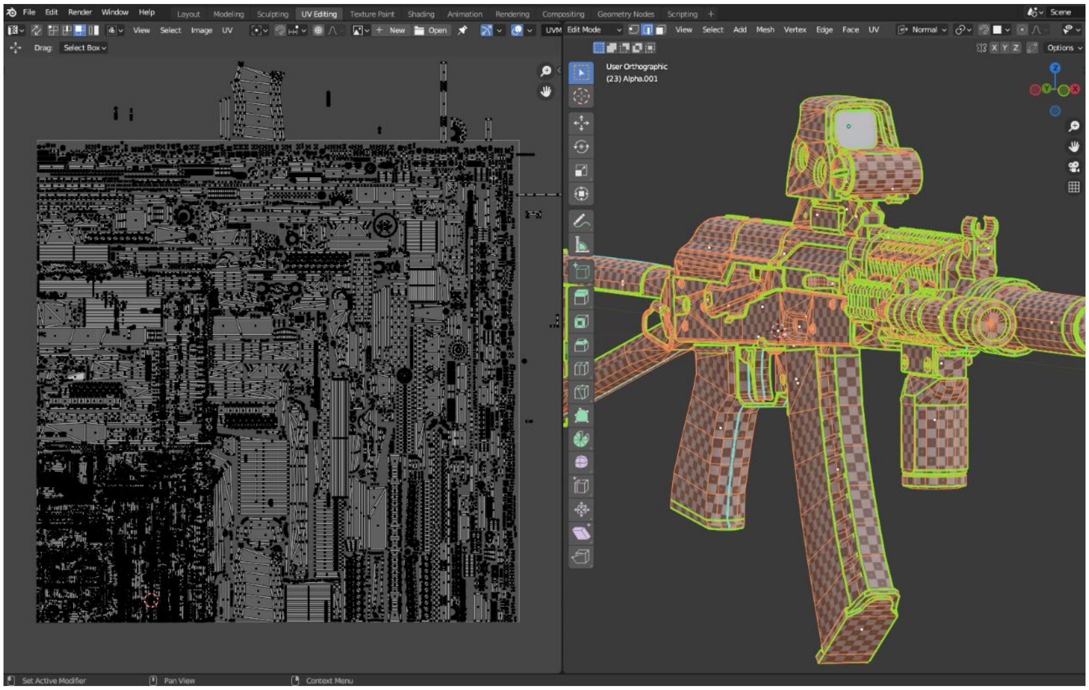

Figure 23. Final UV map and the low poly side by side

The pixel per unit ratio, also called texel ratio, is set to be uniform across the model which creates the uniform preview checker pattern. Finally, the smoothing groups are set to go along the UV seams, for the later baking to work correctly.

3.2 High poly process

For creating the high poly as part of the weapon pipeline, the method was referenced from a YouTube video High-Poly Modeling Techniques in Blender. In it the author explains what modifiers in Blender can transform the low poly to a high poly according to weapon creation needs. The method and its effects are illustrated step by step in Figure 24.

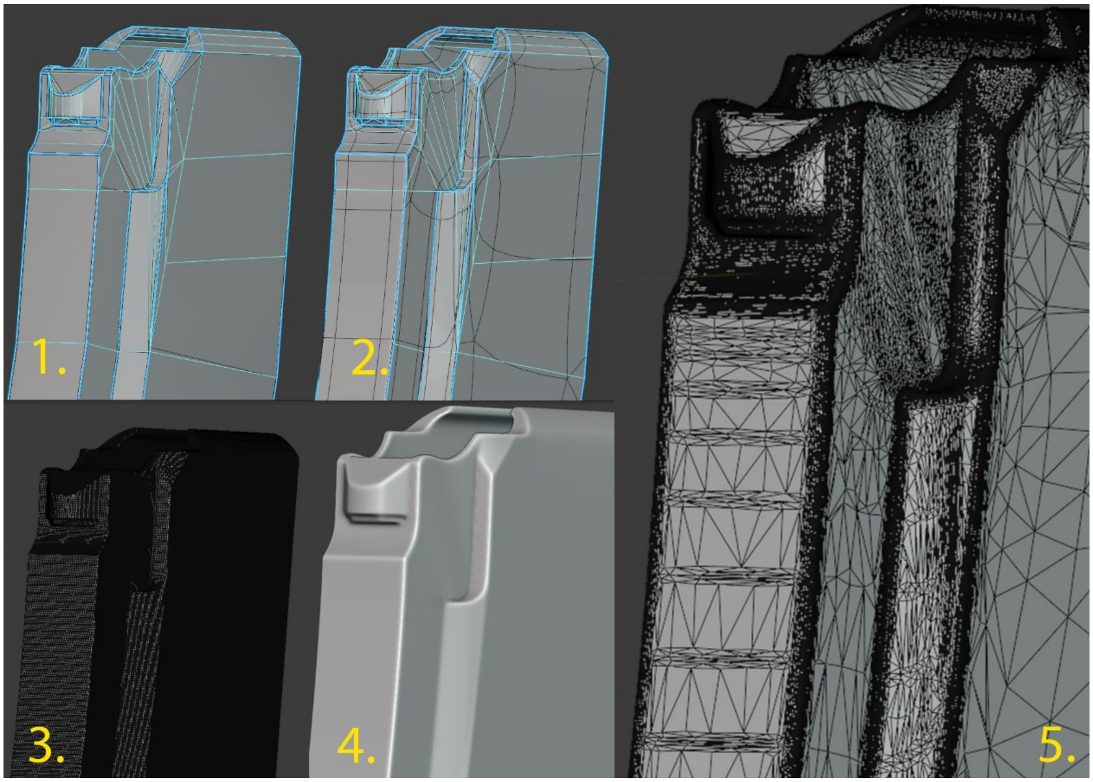

Figure 24. Magazine model turned from low poly to high poly in five steps

For this method to work, the author showcased a series of actions that turns the low poly model procedurally to a high poly. The example is used in Blender, but various software has similar solutions. The sharp edges are first marked to be beveled in step one, so that the subsequent subdivision in step two could hold the original form. In step three the model is remeshed into millions of one by one voxels, that create the realistic edge beveling. Remeshing often result in an incremental ladder like structures on various angled surfaces, which are corrected by a smoothing modifier in step four. As this creates unnecessary topology throughout the whole model, in step five the model is decimated to preserve most of its high poly remeshed topology on edges and deformations of the geometry. This optimizes the mesh but keeps its look identical to the remeshed version (High-Poly Modeling Techniques in Blender 2021).

Similar workflow can be found in Tim Bergholz's Revolver Tutorial (2021), but without relying on edge beveling. The tutorial also goes over the whole process of creating a game ready revolver model (Bergholz 2021).

The last example of making a high poly for weapon models is a workflow in Zbrush. In the Gameartist article, Jose Sanz shares his process of upscaling a low poly imported from different software in his weapon creation process. The steps are essentially the same as in Blender methods, illustrated by the author in Figure 25.

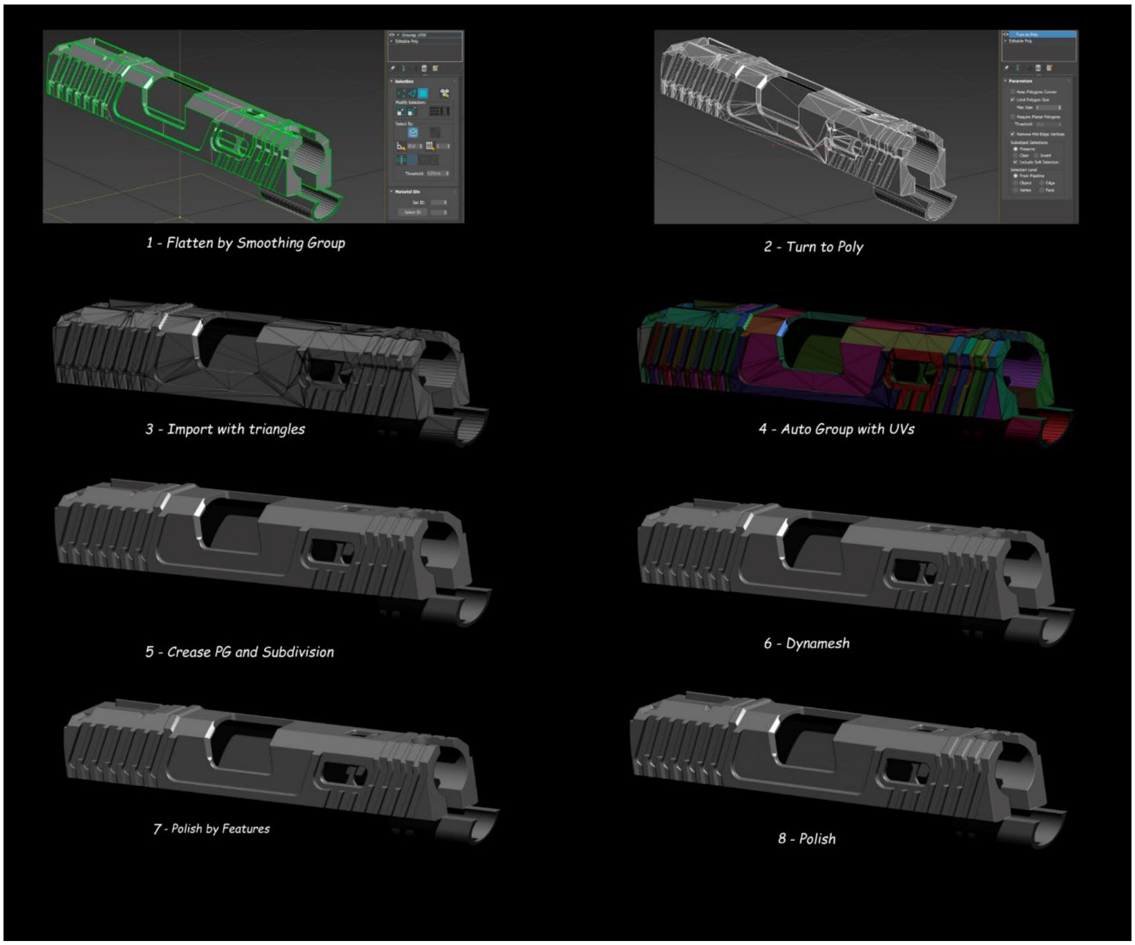

Figure 25. Eight steps taken to make a realistic model in Zbrush (Sanz n.d.)

The variety of the process is examined to show that none of methods is the best one. While software like Blender excels at manual modelling, generating large number of polygons is slow for them, and can lead to crashes. On the opposite spectrum, Zbrush handles millions of polygons at once like it is nothing but does not have an intuitive hard surface modelling solution. All the methods, however, have the advantage of being procedurally generated, which makes it easy to keep consistent level of polish throughout the final model. The remeshing also eliminates the pinching effect left buy the subdivision of the model.

3.3 High poly result

After applying the modifier stack on all parts of the AKSU model, the high poly was finished. For the weapon workflow it is needed for its details, particularly along the edges. While there are various methodologies for creating these details, no other faster and less resource intensive approaches are mentioned in all the examined articles on weapon creation pipelines. The result is examined in Figure 26.

Figure 26. High poly from different angles

The final polygon number for the high poly is 4,714,976 polygons. The original low poly is just below forty thousand. It is possible to model in more little details on this stage, such as knurling effect of the many little faces on the suppressor’s surface, and insets that are created between tightly adjusted parts of the rifle’s construction, beneath the bolts, for example. These details can also be added in Substance Painter 3D when texturing and should be done so due to how baking works.

3.4 Baking

Detailed dense geometry is usually required to achieve high levels of realism, but it can be optimized by modern 3D techniques to not use the resource intensive geometry of the high poly mesh. This can be achieved by a process called baking, in which the high poly mesh related information is saved into a texture file, also called a bitmap, that is transferred onto a low poly mesh to create fake detail (Wilson n.d.). The method used for the research weapon also uses the UV seams in the low poly for distributing the smoothing groups correctly. By baking the high poly, the following maps are generated for subsequent texturing:

- Ambient occlusion, which looks like shadows between parts
- Normal information, surface level details like indentations and bumps
- Direction, which indicates where is up, down, left, or right on a texture
- Curvature, which detects different types of edges and cavities
- Position, which tells the relative position of the geometry inside a normalized cube

To bake in Marmoset Toolbag both high poly and low poly are exported as one object, with their geometry being inside of each other's. This set-up is shown in Figure 27.

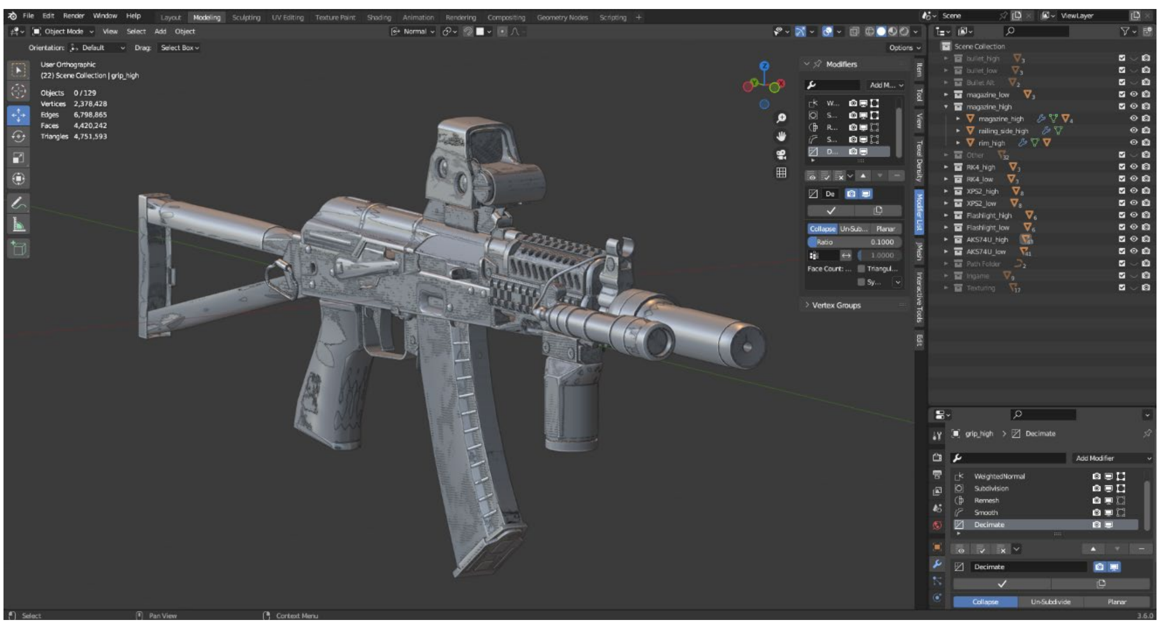

Figure 27. High and low poly both visible at the same time

The unique way to export may seem unintuitive at first, but combining everything in one file has the benefits of staying organized in the file folders.

To show an example of what baking can do, small detail is baked into the magazine in Figure 28.

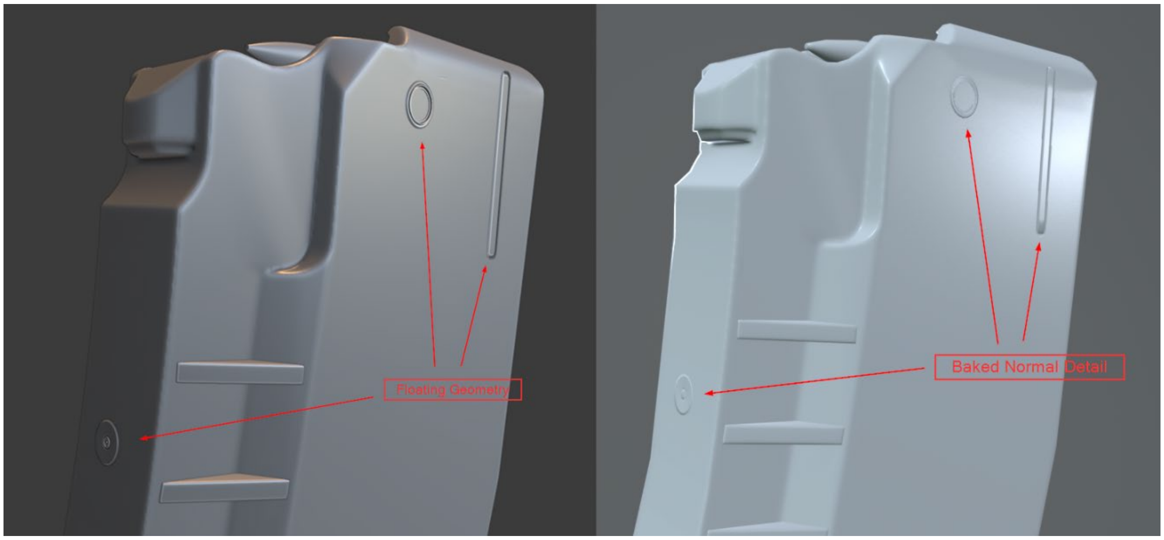

Figure 28. High poly detail baked into the low poly mesh

In this preview, the advantage of baking detail overall in the weapon creation pipeline is explained visually. The model on the left is drastically higher in numbers of polygons compared to than its counterpart on the right, but they look and behave almost identically.

4 TEXTURING

Everything done so far in the workflow was for texturing, as bad texture quality can downgrade a well modelled asset, and good textures can upgrade a poor model. To make a believable model it is recommended to study the gathered reference from the point of surface material behaviour. At this stage the artist should pay attention to the little details occurring in the surface and try to understand why they are occurring. Such detail may be colour variation that occurs in metal surfaces, the way it reflects light, and physical reasons like oxidation or rotting on wood materials. This makes the material believable and tells a story about it to the viewer. One way to keep track of everything that the textures are supposed to contain by the end of texturing is making notes of all perceived aspects in the reference's surface.

Resenberger-Loosmann (2023) proposed that simple repeatable materials can be done in ten layers in Substance Painter. In his ten-layer theory, almost any material would be generated from ten layers that will express colour, roughness, height, edge highlights, and lastly finer detail. Basic colour must not be pure black or white, and not be too dark or bright, for highlight and darker tones to be clearly visible. Roughness layers control the reflectiveness of the material. Height details for adding the feel that the model can be touched, like damage scratches on the edges of metal surfaces. Edge highlights are added using curvature generators that utilise the baked curvature maps. Highlighted edges add to the assets variety in colour and make edges look interesting. Finer details can include sharpening the contrast between textures, tiny hairs and so on. Final polis adds one last pass of high-fidelity realism to the texture. The philosophy behind texture creation is discussed more in depth in the original Resenberger-Loosmann’s article (2023).

The result of following these principles for texture creation as much as it is possible due to time constrains is showcased in Figure 29.

Figure 29. Texturing view in Substance Painter 3D

In the Substance Painter, a first pass of a black metal with light edgewear and subtle surface variation is applied. Due to time limitations, it is applied universally on almost all surfaces, and its roughness is altered for the flashlight to match Resenberger-Loosmann's advice on texturing.

5 CONCLUSIONS

The resulting model and its textures serve as solid starting base for the polishing part in the hypothetical FPS weapon work pipeline. In many cases, the referenced articles would mention that the initial results are usually treated as first rough passes. These passes are not used as a final hero asset for their games but can give a good presentation of the feel of the asset. It should be noted, however, that different studios can have different standards, and for some intense productions such artistic exploration is not possible.

In the current state, the model showcased throughout the study is held back mostly by a lack of detailed textures, as well as minor details in overly optimized parts of cylindrical geometry. It is difficult to purpose what could be done to have a better result, as realistic FPS weapons cannot sacrifice any aspects of their workflow to prioritize texturing over modelling, for example. Each step is so integral that it needs to be done right, or it will hinder the overall presentation. One possible way to salvage an underwhelming FPS asset is to turn it into an asset viewed from a third person, as the already existing geometry can be unsubdivided easier than re-modelling new geometry with detailing.

REFERENCES

Blender Foundation. 2023a. Structure. Web page. Available at: https: https://docs.blender.org/manual/en/latest/modeling/meshes/structure.html [Accessed 17 November 2023].

Blender Foundation. 2023b. Object Origin. Web page. Available at: https://docs.blender.org/manual/en/latest/scene_layout/object/origin.html [Accessed 17 November 2023].

Blender Foundation. 2023c. Individual Origin. Web page. Available at: https://docs.blender.org/manual/en/latest/editors/3dview/controls/pivot_point/individual_origins.html [Accessed 17 November 2023].

Blender Foundation. 2023d. Mirror Modifier. Web page. Available at: https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/mirror.html#index-0 [Accessed 17 November 2023].

Blender Foundation. 2023e. Loop Cut. Web page. Available at: https://docs.blender.org/manual/en/latest/modeling/meshes/tools/loop.html [Accessed 17 November 2023].

Blender Foundation. 2023f. Bevel Edges. Web page. Available at: https://docs.blender.org/manual/en/latest/modeling/meshes/editing/edge/bevel.html [Accessed 17 November 2023].

Blender Foundation. 2023g. Extrude. Web page. Available at: https://docs.blender.org/manual/en/2.79/modeling/meshes/editing/duplicating/extrude.html [Accessed 17 November 2023].

Blender Foundation. 2023h. Inset Faces. Web page. Available at: https://docs.blender.org/manual/en/2.79/modeling/meshes/editing/duplicating/inset.html [Accessed 17 November 2023].

Blender Foundation. 2023i. Merging Vertices. Web page. Available at: https://docs.blender.org/manual/en/latest/modeling/meshes/editing/mesh/merge.html [Accessed 17 November 2023].

Blender Foundation. 2023j. Subdivide. Web page. Available at: https://docs.blender.org/manual/en/2.79/modeling/meshes/editing/subdividing/subdivide.html?highlight=subdivide [Accessed 17 November 2023].

Blender Foundation. 2023k. Boolean Modifier. Web page. Available at: https://docs.blender.org/manual/en/2.79/modeling/modifiers/generate/booleans.html?highlight=boolean [Accessed 17 November 2023].

UV Mapping & Unwrapping - Blender Fundamentals. 2017. Blender. Video clip. July 25. Available at: https://www.youtube.com/watch?v=V6OXSR5Ynyc [Accessed 17 November 2023].

Polygon. 2015. Polycount Wiki. Web page. Available at: https: http://wiki.polycount.com/wiki/Polygon [Accessed 17 November 2023].

Edge padding. 2017. Polycount Wiki. Web page. Available at: http://wiki.polycount.com/wiki/Edge_padding [Accessed 17 November 2023].

Base Mesh. 2020. Polycount Wiki. Web page. Available at:

http://wiki.polycount.com/wiki/BaseMesh [Accessed 17 November 2023].

Polygon Count. 2021. Polycount Wiki. Web page. Available at:

http://wiki.polycount.com/wiki/Polygon_Count [Accessed 17 November 2023].

Texture Baking. 2021. Polycount Wiki. Web page. Available at:

http://wiki.polycount.com/wiki/Texture_Baking [Accessed 17 November 2023].

Lemos, C. 2020. Tutorial: How Normal Maps Work & Baking Process. Web page. Available at: https: https://80.lv/articles/tutorial-how-normal-maps-work-baking-process [Accessed 17 November 2023].

High poly vs Low poly in 3D Modeling explained in simple terms. 2022. Queppelin. Web page. Available at: https://www.queppelin.com/high-poly-vs-low-poly-in-3d-modeling/ [Accessed 17 November 2023].

Topology. 2023. Turbosquid 3D Resources. Online documentation. Available at: https://resources.turbosquid.com/checkmate/checkmate-product-presentation/topology/ [Accessed 17 November 2023].

Unity Documentation. 2017. Unity. Web page. Available at: https://docs.unity3d.com/560/Documentation/Manual/ModelingOptimizedCharacters.html [Accessed 17 November 2023].

What Is Box Modeling? 2023. Information Transformation Services. Web page. Available at: https://it-s.com/what-is-box-modeling/ [Accessed 17 November 2023].

A simple guide to the basics of subdivision modeling. N.d. Dassault Systems. Available at: https://www.3ds.com/store/cad/subdivision-modeling [Accessed 17 November 2023].

Resenberger-Loosmann, M. 2023. Creating an Assault Rifle in Substance 3D Painter and Unreal Engine 5. Web page. Available at: https://80.lv/articles/creating-an-assault-rifle-in-sunstance-3d-painter-and-unreal-engine-5/ [Accessed 17 November 2023].

3D Modeling in Games vs Movies: A battle of the budget? 2019. 3D Ace. Web page. Available at: https://3d-ace.com/blog/3d-modeling-in-games-vs-movies-a-battle-of-the-budget/ [Accessed 17 November 2023].

B-19N Rail. n.d. Zenitco. Available at: https://en.zenitco.ru/catalog/weapon-upgrade/b-19n-rail/ [Accessed 17 November 2023].

B-11U Rail. n.d. Zenitco. Available at: https://en.zenitco.ru/catalog/weapon-upgrade/handguards/b-11u-rail/ [Accessed 17 November 2023].

Tarkov AKS-74U Krinkov. 2021. T.Rex Arms. Available at: https://help.trex-arms.com/hc/en-us/articles/4683331400471-Tarkov-AKS-74U-Krinkov- [Accessed 17 November 2023].

Dikhtyar, I. 2022. How to Improve Your Skills in the Field of Weapon Art. Mistakes and Life Hacks. Web page. Available at: https://www.linkedin.com/pulse/how-improve-your-skills-field-weapon-art-mistakes-life-hacks-/ [Accessed 17 November 2023].

Burton, A. 2020. Weapon Art: Tips for Design, Texturing and Presentation. Web page. Available at: https://80.lv/articles/weapon-art-tips-for-design-texturing-and-presentation/ [Accessed 17 November 2023].

Zenitco Parts Compatibility. n.d. Ivan the Bear Tactical and Outdoor Gear from Russia. Available at: https://ivantactical.com/zenitco-parts-compatibility/ [Accessed 17 November 2023].

Denham, T. n.d. What is UV Mapping & Unwrapping? Web page. Available at: https://conceptartempire.com/uv-mapping-unwrapping/ [Accessed 17 November 2023].

Kubas, M. n.d. A guide to weapon modelling for FPP games. Web page. Available at: http://piratportfolio.com/fpp_eng/ [Accessed 17 November 2023].

High-Poly Modeling Techniques in Blender. 2021. Tosmo. Video clip. May 29. Available at: https://www.youtube.com/watch?v=kBgYwSDC3wI [Accessed 17 November 2023].

Bergholz, T. 2021. Revolver Tutorial - what's in it and why Blender's remesher workflow is amazing. Web page. Available at: https://chamferzone.com/projects/NxAoLz [Accessed 17 November 2023].

Wilson, J. n.d. The Toolbag Baking Tutorial. Web page. Available at: https://marmoset.co/posts/toolbag-baking-tutorial/#toolbagsetup [Accessed 17 November 2023].

Sanz, J. n.d. AKS-74u MAXIMOV. Web page. Available at: https://gamesartist.co.uk/aks-74u-maximov/ [Accessed 17 November 2023].