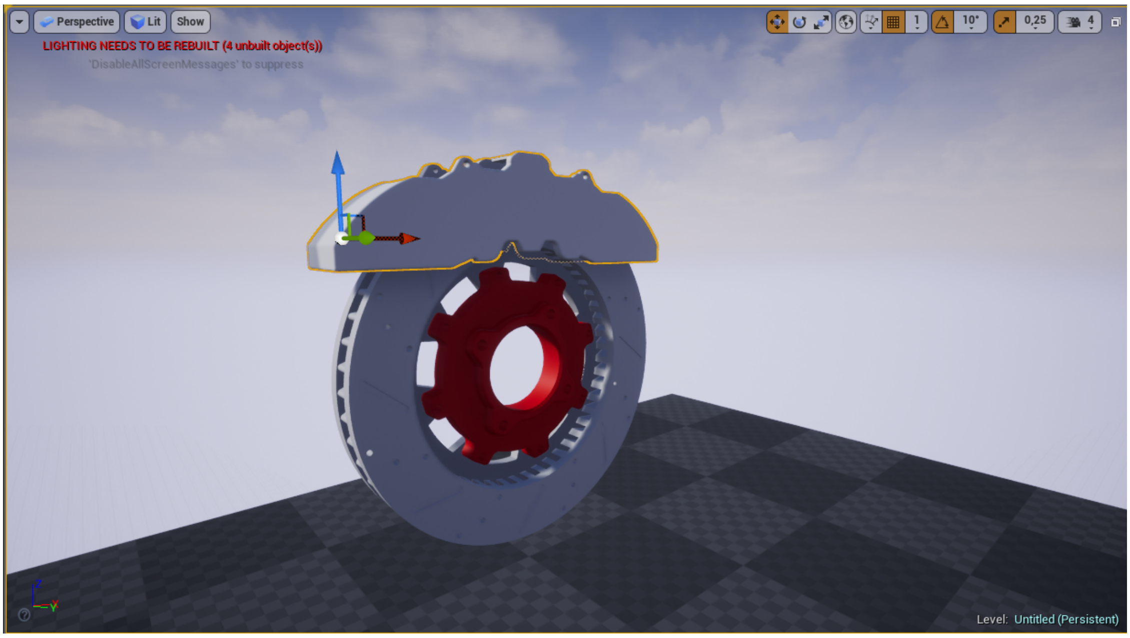



# Interactive visualization of CAD data in real time rendering

## Using game engines to visualize engineering data in virtual reality

Master's thesis in Product Development

Filip Kleverud

Department of Industry- och material Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis 2018:IMSX30

# Interactive visualization of CAD data in real time rendering

## Using game engines to visualize engineering data in virtual reality

Filip Kleverud

Department of Industry and Material Science
*Division of Product Development*
Chalmers University of Technology
Gothenburg, Sweden 2018

Interactive visualization of CAD data in real time rendering
Using game engines to visualize engineering data in virtual reality
Filip Kleverud



Supervisor: Mikael Litendahl, Alten Sweden
Examiner: Håkan Almius, Industry and Material Science

Master's Thesis 2018:IMSX30
Department of Industry and Material Science
Division of Product Development
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Break disc CAD model converted in Unreal Studio

Typeset in LaTeX
Printed by [Name of printing company]
Gothenburg, Sweden 2017

Interactive visualization of CAD data in real time rendering
Using game engines to visualize engineering data in virtual reality
Filip Kleverud
Department of Industry and Material Science
Chalmers University of Technology

# Abstract

Consumers are getting more demanding regarding product quality, safety and design. As a result the product development process are getting harder. Complexity leads to higher competence and knowledge needed leading to more people involved. Time to market are reducing meaning more complex and safer products need to reach the market and make profits faster than the year before. This development will probably continue and thus new and more effective ways of product development is needed. Virtual reality is on such technology that have the possibility to revolutionize the product development process. The question is how will it do that and how do companies make use of this promising technology and how do they incorporate it into their existing development pipeline without disrupting current workflows.

This rapport have researched one possible solution to these questions, the use of game engines in the product developoment process. These software have near to photo realistic graphics and can show these graphics in real time with quality improving for every game that is released. They have also made use of the virtual reality technology for several years more than the engineering sector have. Thus they have templates and ready to use projects for exploring virtual worlds. The problem is that CAD data rarely is compatible with the data these game engines accept and can import. This rapport have investigated a few software that solves this problem by converting the CAD data automatically to game engine compatible file formats. Thereby opening up engineering data for virtual reality use.

The rapport found PiXYZ and Unreal Studio to be good candidates for engineering use. They can optimize the data for different computer hardware and demands. Unreal Studio can import large files into ready to use virtual reality projects where the data can be programmed to be interacted with by utilizing an easy node based visual programming.

Keywords: game engines, virtual reality, Unreal Engine, Unity, PiXYZ, product development, real time rendering, gamification, visualization.

## Acknowledgements

I would first and foremost like to thank my mentors during this thesis, Håkan Almius and Mikael Litendahl. They are both visionaries in the visualization field for seeing this as a possible thesis subject. Without as much as a sigh they have answered all my doubts and curious questions. Truly a big thank you. I also want to thank the employees at the design and visualization department at Alten Sweden for letting me be around and learn by you. It has been an interesting experience to see a design studio working with engineering. You have been more than kind to let me into your group.

Thanks to to all!

Filip Kleverud, Gothenburg, June 2018

# Contents

# 1
# Introduction

This chapter presents why this master thesis will handle the subject of transferring CAD data for visualization in virtual reality. Also the aim and limitations of this thesis will be presented.

## 1.1 Background

Alten is a consultancy firm within the engineering sector with over 23000 engineers in several different areas, mostly within automotive[1]. As product development is their core activity they are always looking for ways to improve their product development process. One such way is Virtual Reality (VR) which is getting mature and stable enough to view digital models rendered in real time as if they where real products in VR, with the right software. As [2] describes these rendering capabilities are best experienced with so called head mounted devices short HMD, goggles that are placed on the head with two screens projecting images in front of each eye. With several commercial HMD's now available for around 700€, like HTC Vive or Oculus Rift the interest for viewing CAD models in virtual reality is increasing. Alten want to see the capabilities of using VR for visualization and interaction in VR.
Several 3d modeling and animation software exist and there are several that have incorporated VR in their software in one form or another. But since there is a difference in just viewing the model in VR and working with some sort of interactive animation, and thereby involving programming, the decision can be hard. Especially since programming can take more time to learn than a company workflow provides. Gamification is a buzzword often used in describing the need to incorporate more interactivity in services and product development, since games are in its essence interactive. Game development software is often originally 3d game engines with real time rendering and graphic optimized performance. It often comes with support for VR view as well as interactive capabilities [4] and have real time realistic lightning and textures[6]. As games are getting more and more realistic so have Gaming softwares graphics. Recently gaming softwares have developed plugins or add-ons allowing engineering industry to utilize gaming software. Commercial real time rendering engines have been looking into CAD export plugins as means to visualize CAD data in better graphics and possiblity for VR [3, 7, 68]. Both automotive [8], life science [9] and architecture [10] are looking at the possibility to use game engines for visualizing complex geometry and now Alten want to see the possibilities.

## 1.2 Aim

The aim of the thesis will be to research the capabilities of the real time rendering engines in handling CAD data. It will compare what other softwares capable of real time VR rendering offers and what is the state of the art technology in rendering and VR. It will also handle what the workflow would look like used in full scale. In the end a scene in one real time rendering software will be produced where a model from a CAD software is exported using these CAD conversion plugins. This scene will be programmed after a fitting case has been decided together with Alten to show the capabilities of Unreal and conversion software.

## 1.3 Limitations

This thesis will not make any deeper research into or make use of more than one real time rendering softwares then as time to learn more software does not exist. Of the same reason only one CAD software is to actually be used. Software with only realistic image rendering capabilities will not be compared neither will software without what is deemed "realistic rendering" capability. As there is a lot to cover in the field of programming, virtual reality and computer graphics this thesis will introduce the terms needed to utilize the workflow described in this thesis.

## 1.4 Questions

1. Why would use of game engines for virtual reality be beneficial for the product development process?
2. Why can game engines and VR be used for engineering applications?
3. What is state of the art technology concerning realistic real time rendering software and VR?
4. What is the difference between CAD models and Tesselation models

## 1.5 Methodology

### 1.5.1 Workflow

In [11] an iterative workflow is described . This workflow starts with an idea with functionality to be implemented. This prototype is internally tested to be deemed satisfactory, if not two things can happen. The product is evaluated and decided if to be worked further on or be discarded. Othervise the product is tested externally and evaluated again with new input and reworked.

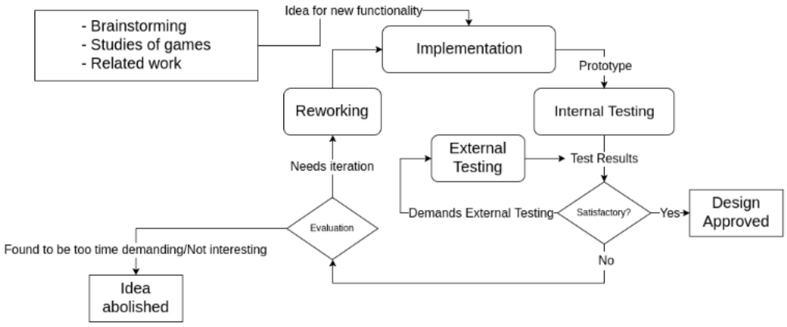



**Figure 1.1:** Iterative workflow chart

This workflow fits the aim for this thesis as the outcome of the conversion between softwares has to be analyzed to determine what steps need refinement.

### 1.5.2 Design of Experiments

Experiments in this thesis will be the file transfers between the CAD software and Unreal Engine. The process of exporting the model from the CAD software and importing it into the real time rendering engine is performed. After which the evaluation is done to determine what is satisfactory concerning materials, the mesh (triangles and vertices) of the model, time of transfer, consistency of results as well as other factors that is deemed important.

### 1.5.3 Data Collection

Data collection is done by reading papers and reports about virtual reality and earlier real time rendering engine projects at Chalmers. Other softwares will be researched trough respective homepage as well as papers describing similar cases as this thesis. Materials and lightning in Unreal is important as VR enables to see details and for this Unreal Engine Documentation is read [6].

## 1.6 Schedule

Initial schedule is presented in a Gantt Schart in Appendix A

# 2

# Theoretical Framework

This Chapter will present the world of computer modelling and graphics, virtual reality and design engineering and its history. Earlier research in these subjects will also be presented with the results and the discussions held in those papers.

## 2.1 Computer Modeling history

Computer modeling are divided into two major fields, CAD or 3d modeling. These can sound similar and might be hard to distinguish. They both essentially produce a computer model, that based on numbers run through the computer end up projected on the computer screen. Computer models started to emerge as early as the 1950s [13] as a promising way to simulate simple 2D drafts. 1960 MIT started researching the possibility to automate engineering design processes. As more researchers started their own projects the technology grew and by 1970 several start-ups were developing their own CAD softwares.

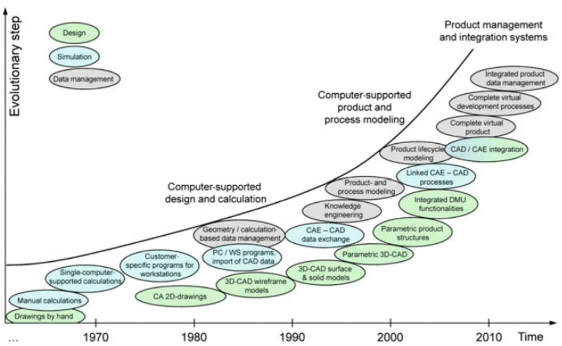



**Figure 2.1:** Historical overview of CAD development [15].

The CAD technology was developed for simulation of engineering processes but the technology was still lacking many of the basic attributes taken for granted today. By the middle of the 1960s, algorithms was invented to translate, rotate and scale objects on the screen in three dimensions. Cyrus Levinthal wrote in Scientific America 1966 that "It is too early to evaluate the usefulness of the man-computer combination in solving real problems of molecular biology. It does seem likely, however, that only with this combination can the investigator use his 'chemical insight' in an effective way", describing the early process of simulating molecules with CAD. When the groundwork of getting a screen to show models and later manipulate and combine these models existed researchers focused on graphics. In the 1970s the first efforts to simulate real life objects and scenarios began. First out was solid modelling, exact mathematically represented 3D models with information about whats inside the models. Following this different ways of combining different primitives, objects like spheres, cubes or planes, into more complex models where made.

But solid models was not needed to model things that should look realistic, only the surfaces mattered then. For this implicit surface modeling where used. Models where here built up by tiny triangles, or faces as they were called, connected with each other forming the model. To make it look more realistic shaders were used to "paint", or to shade, the faces with colours and light. Later algorithms were made to shade models in such a way that height and displacements were faked, making model surfaces look more complex than they really were.

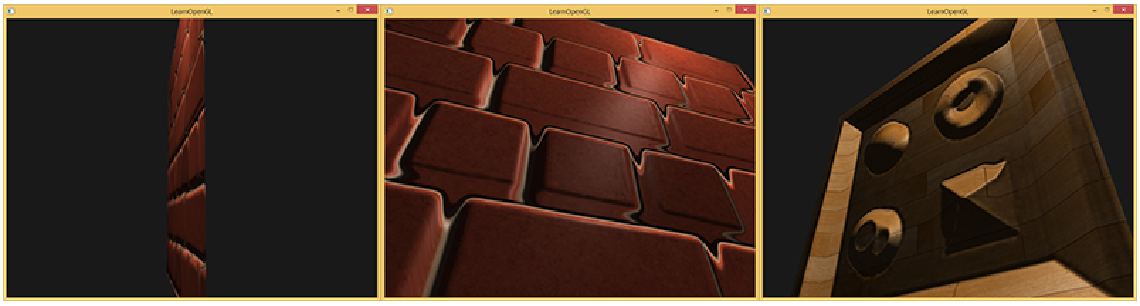


**Figure 2.2:** Example of faking geometry [14].

During the years since graphics began to evolve it has begun to approach photo realism. Several different techniques are used to produce these images. Techniques that mimics the rays of light hitting a surface, flow of water or air blowing at a tree, all using efficient algorithms. These algorithms has gotten so optimized that computer games today are drawing these images at around 100 frames a second [12] at screen resolutions that involves millions of pixels.

## 2.2 Computer Graphics

Computer Graphics (CG) is a vast subject with too many areas to cover in detail for one report. However, to be able to justify the use of one CAD software over the other, and to explain the differences between CAD and 3d modelling and explain the reasoning behind the usage of game engines for visualizing CAD data despite these inherent differences, a high level coverage of key subjects within rendering has to be made. There is certain things a CAD software does that a game engine will never be able to do and the opposite, which will be described in this chapter. There is also certain softwares specialized in rendering of one type of product but not others. Some are good for just one picture while others renders over 100 pictures every second. A few of the key reasons for these differences will be explained here.

### 2.2.1 Rendering

According to "game engine architecture" rendering a three dimensional scene is a factor of calculating different steps [12]. First a virtual scene is created where models positions are represented in mathematical form with vertex coordinates. If the

(a) (b)

**Figure 2.3:** a) Shows a hoto realistic visualization of a scene and b) a model with culled triangles thanks to wrongly aligned normals

model have animations applied these are also calculated and stored and then the physics and potential Artificial Intelligence, ai. In the case of rendering for visualization purposes a virtual camera is positioned with virtual light sensors that helps determining what pixels on the computer screen to light up. Light are emitted from light sources and model surfaces have properties applied telling the computer how the surfaces reacts to the light. The goal with these kind of visualizations are usually to achieve a high degree of photo realism as in 2.3. The degree of photo realism is where real time rendering, that game engines use, differs from still framed rendering, that is used for entertainment and visualization purposes. The level of photo realism is heavily dependent on the light transport models, the algorithms used to determine how light reacts when hitting a specific surface. Real time rendering software often have to produce 30 to 120 frames per second while a still frame render software can render a picture over several hours or days. This leaves vastly different requirements on the software.

The faces building up the non-parametric 3d models are shaded individually and then algorithms determine which color the pixel that are in that specific face should have. An important factor is the normals of the faces. The normals help determine the faces behaviour when struck by light rays. These calculations is also helped by tangents and bitangents. What the normals and tangents determine can be for example how the light rays bounce of the face when lit by the sun, what is the shininess, the roughness or the texture. When the normals are facing one way but light is hitting the surface from the backside of the face (the side the normal is not facing), that side is culled, meaning it is not drawn on the screen, see 2.3.

This is done to save computer performance as every face not drawn to the screen is less computation which leads to less work load. Game engine architecture writes that light is normally either absorbed by the surface, reflected of the surface or transmitted through and usually refracted. Photo realistic rendering software takes the first three into account. Colours is represented with colour channels (rgb) with a fourth channel the alpha channel handling level of opacity of the faces. To help artists and engineers a normalized coordinate system called UV space was defined where the model coordinates are transformed into 2d space. to make lightning and texture calculations, see 2.4.

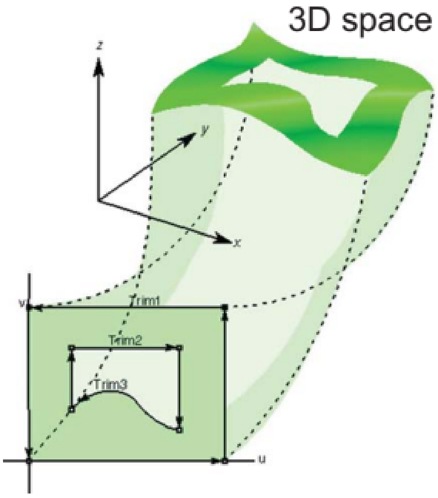



**Figure 2.4:** UV coordinates of a 3d model in 2d space [17].

The face attributes are simplified, discretizied approximations of what the surface or model look like, called flat shading. The normals and tangents are used to determine what each face looks like, unaware of what their neighbour looks like. To make the model more realistic the values of the faces are interpolated over the whole surface using gauraud shading and Phong shading 2.5. This smooths the colour of the faces and blends between neighbours to create a continuous shading.

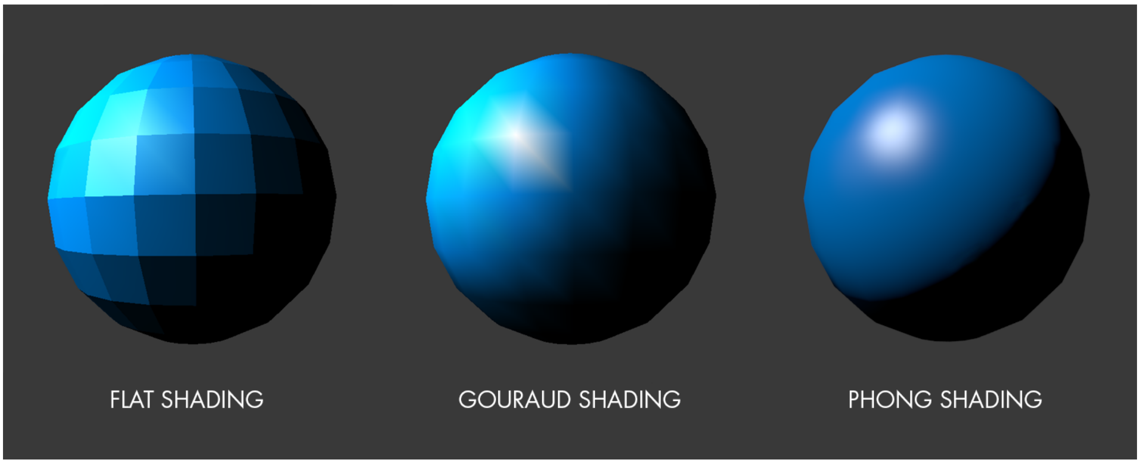



**Figure 2.5:** The different models of shading [16].

When needed texture maps can be used to project additional lightning information onto the surfaces. Resulting in the difference between a jaggy highlight were the face boarders are clearly visible, as the middle picture in 2.5. To the smooth transition of the highlight in the phong shading model.

True realism can only be achieved when accounting for indirect lighting, the light-rays that bounces of between one model and another. This differs from the local direct lightning that is the lightning directly applied to a surface. Lightning models that take both of these into account are called Global Illumination, GI, models. GI models can have different focus, either on making realistic shadow, making the interplay of light between models as realistic as possible or providing a holistic realistic lightning scenario where many different optical phenomena are taken into account, like the ray-tracing lightning model does. Ray tracing means shooting out millions of rays of light over a long period of time, calculating every path the specific ray takes across the scene, like a real sunbeam. Ray tracing are what is usually used in most software in the visualization industry as it makes the most realistic lightning scenario. But has the drawback that it takes longer for a scene to render. The process of calculating how light strikes surfaces are called to bake light. Referring to that the lightning information is "baked into" the polygons of the objects. Of course the lightning and the objects has to be static, othervise this light would change.

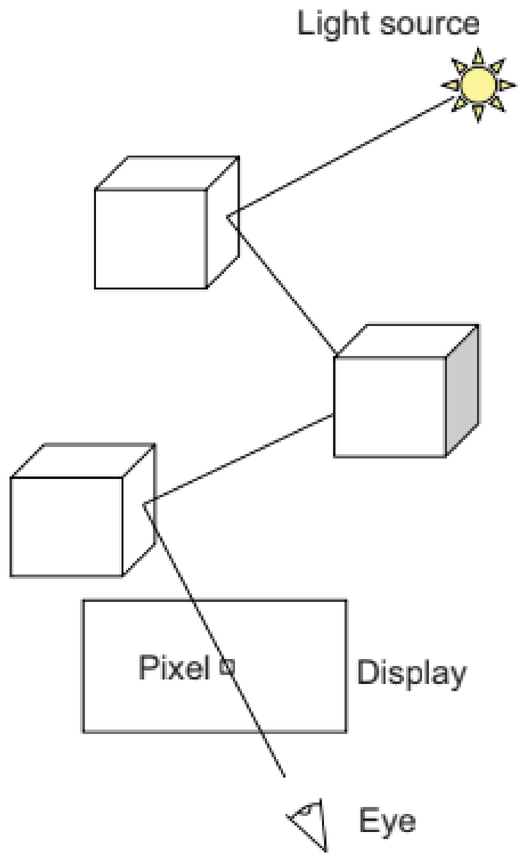



**Figure 2.6:** Ray-tracing beam travelling from a virtual light source and bounce between objects and finnaly reaches a pixel on the virtual camera or screen.

The process mentioned earlier where algorithms calculate which face of the model a pixel on the computer screen overlaps is called rasterization. The faces are fragmented into smaller pixel sized squares, so that the pixels on the screen that overlaps that face have the same colour that the overlapping face has. 2.7 shows this rasterization applied to a triangle. If the virtual camera is further away from an object, less pixels share the same space, this can make an image look blurry and unclear. To fix this the density of the texture mapping can be adjusted to fit the number of pixels occupying the face. Meaning less information has to share the same screen space.

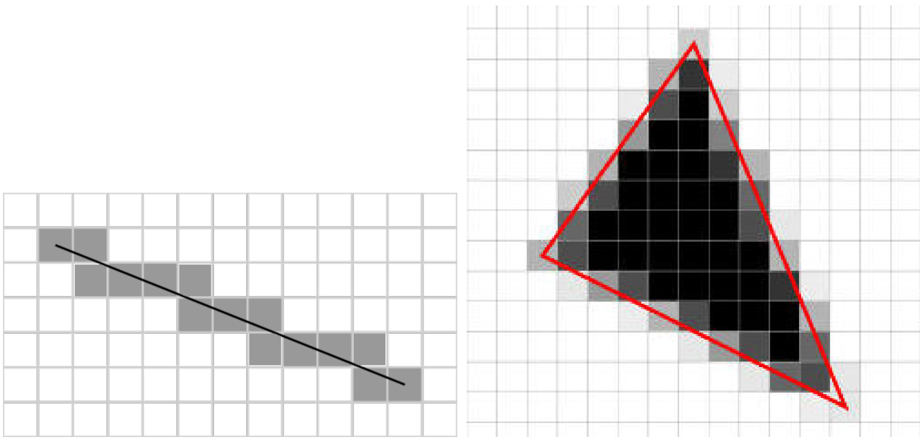


**Figure 2.7:** The rasterization process [19, 20].

When the models of the scene are rendered there can be after effects layered on top of this information. Like decals (small simpler pieces of geometry) or post processing effects like saturation, bloom effects or even hair and fur on characters and animals. After this the sky, particles or water can be rendered in with different techniques that is simpler or harder to model in depending of the level of detail needed. To further enhance the quality of the frame, lines and edges that are jagged are anti-aliased [21]. Anti-aliasing interpolates between points on edges to produce a smoother line and thereby creating a more realistic model. [17]

### 2.2.2 Software

CAD and 3D modeling software are used in different ways and for different applications. 3D modeling software can directly manipulate and even add to the number of faces building up the surfaces of the model, but has no information of whats inside and they are imprecise representations of the surfaces the represent. CAD models on the other hand are exact representations, can directly be used for manufacturing of the part and contains information like density or E-modulus [17] but are harder to make soft bodies of. The 3D modeling software where developed to a large extent for the entertainment industry, movies and games [13]. This put pressure on optimizing graphics so that the models looked as realistic as possible. A movie is usually played at 24 frames per second, meaning an animated movie consists of around 170000 frames. These must be rendered looking realistic. In games 30 to 100 frames a second is not unusual, and in games a lot of calculations must happen at the same time in real time, meaning the AI shall be controlled, lights calculated, objects in movement animated etc [12]. Clever algorithms are used too make some of these function at the speed required, like clothing [22].

#### 2.2.2.1 CAD software

A CAD model is projected to the surface of the screen by different algorithms. A solid model consists of points, curves, surfaces and solids and they all are algorithms representing the model. For the curves, parametric models are used which gives exact representations of the models. CAD systems use specially developed curves for modelling solids. The most important are Bézier curves, B-Splines and Non-Uniform Rational Bézier Splines or NURBS. These gives larger control over the surfaces and also provides more technically interesting forms. These curves are basically controlled, in different ways, by manipulating control points. Moving a control point - moves the surface, either by translation or rotating tangents. Bézier curves are controlled this way. In 2.8 a typical use of these manipulations is shown. The control points are constrained in the end points to adjoin with other curves.

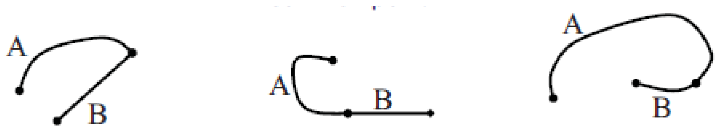



**Figure 2.8:** Example of manipulation of Bézier curves control points [17].

B-Splines are a further development of the Bézier curve. They are based on a polygon controlled by a number of control points. Contrary to the Bézier curve it is not affected by all control points meaning it allows for better control of the curve. The NURBS was created as the two former curves had trouble representing conical and circular forms. NURBS provided the engineer with even more control over how the models could look. Nurbs is the most common form of model representation within CAD. When the models become too imprecise, decomposition of models is used. Decomposition of the solids gives a better resolution near borders to avoid jagged edges. Like pixels, decomposition define three dimensional pixels inside the solid model called voxels. These are cube octrees created when dividing a cube into smaller chunks.

#### 2.2.2.2 Visualization and 3d rendering Software

For visualization by industrial designers and entertainment industry direct modeling and manipulation of surfaces or patches is most used, see 2.9.

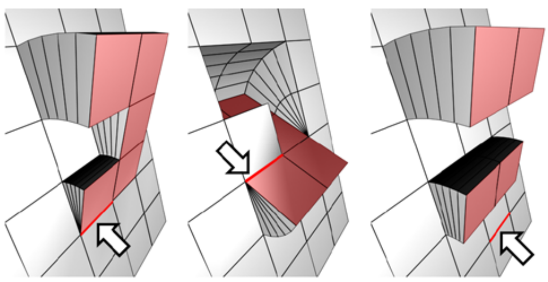


**Figure 2.9:** Direct modeling manipulation of the polygons faces [23].

These have, unlike the solid CAD models, just surfaces built up by faces. By using faces a more optimized model is possible as well as providing the user the possibility to scale down the Level Of Detail, LOD [12]. Faces are similar to the simulation of FEM models in CAD where the same triangle surface representation is used to get an approximation of where the surface would have been. Though FEM models still possess the information of whats under the surface while the models of 3d modelling software does not. An example of how FEM and 3D modeling are approximating surfaces is seen in 2.10.

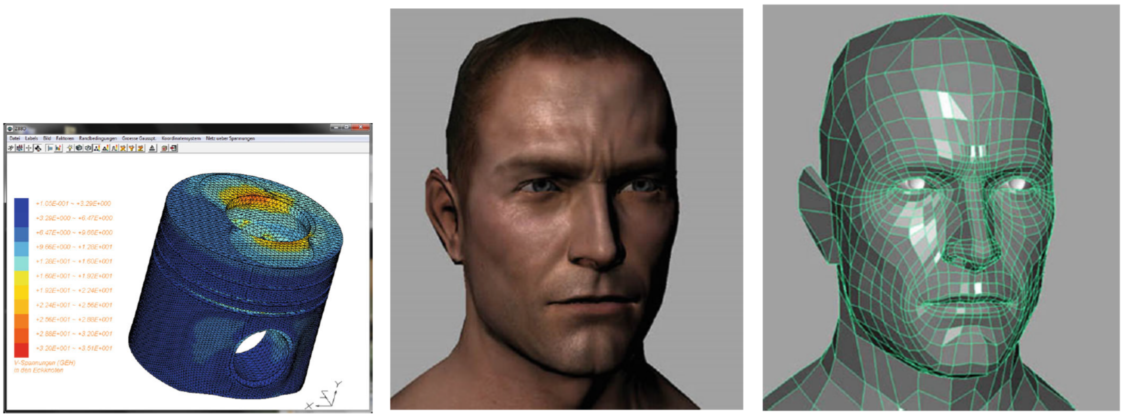


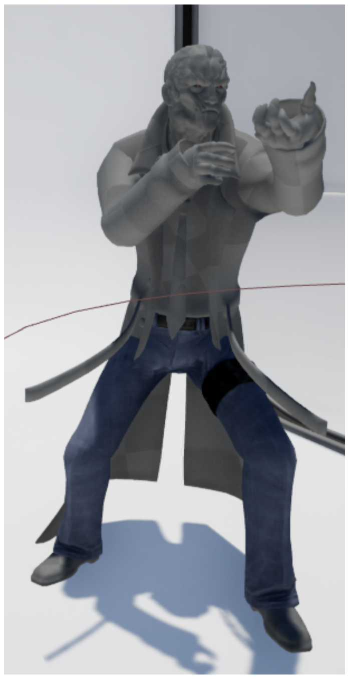
**Figure 2.10:** Example of how faces are used in FEM to the left and 3D modeling to the right [13].

Surface models are easier to use when creating smooth complex forms, such as 2.11, which is necessary when creating soft bodies like humans or cloth [17]. It is the surface model representation that allow for direct face manipulation meaning individual triangles and vertices can be translated or rotated. Visualization software also provides the possibility for more animation features than the CAD system, which are more focused on solid body assembly simulations [17]. 3ds Max and Maya are two industry standard 3d rendering software [66, 24, 25]. They include features like rigging a body with muscles and making animated limbs automatically affect the movement of connected limbs, as in 2.11.

#### 2.2.2.3 Game Development Software

When games started being developed they consisted of 2d shapes on a background [26]. They were largely done of pure interest and to test mathematical algorithms. Over time game development evolved and games such as Doom and Quake pushed the boundaries of what was possible to do on a computer [13]. As hardware evolved so did the software and with the software the demand for better graphics rose from customers and today a game development studio can spend hundreds of millions of dollars on one game [27]. The graphics are reaching levels of realism not too far from the photo realism or ray-tracing. As the cost of developing a AAA title have sky rocketed so have the need to make profits of said title. The Forbes article [27] mention a game on a 240 million USD development cost will often have to make at least 350 million USD to please stakeholders. To make the process of developing a game as lean as possible, and to significantly improve time to market and the number of people being able to work on a game, game developers utilized a game engine dependent workflow [12]. Meaning the company developed a user friendly architecture where more people than programmers can develop the game.


(a)



(b)

**Figure 2.11:** a) shows a smooth surface model and b) a rigged character, when the hand is moved the arm follow as if there were real muscles

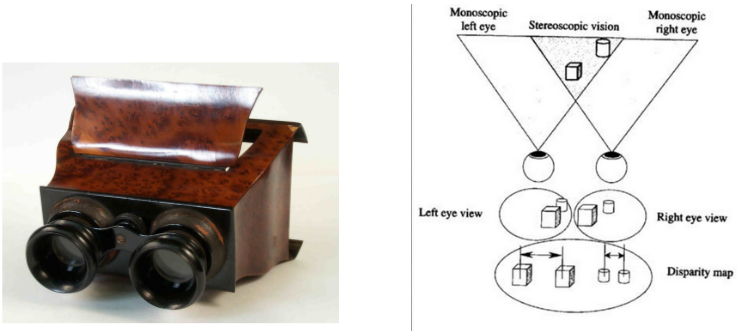
A game engine is like a CAD software but for placing models and other game related activities. A game engine usually includes a graphics processing pipeline, physics, integrated programming compiler, AI development, vfx, material developer, camera animation and an editor to work in. Instead of every programmer having to send a complex set of code many times to several different people, the designers can now also work on a game simultaneously. When building up a level, usually referred to as a scene, it is built up by actors. Actors can be the virtual cameras, lights, models of people, the model meshes like chairs or a curtain etc. Simply put anything that can be placed in the world. These can then, like in the 3d and visualization software, be built up with materials and shadered in a desired way. Camera actors and the animation system often work similar to the visualization software system with drag and drop functionality into the editor. Meaning that instead of every actor having to be programmed in separately they can just be dragged and dropped. Programming a game engine takes time [12] but the gains outweigh the costs. In short, [28] states that a game engine is a set of tools allowing developers to rapid development of a game or an application. Today both automotive [8], life science [9] and architecture [10] are looking at the possibility to use game engines for visualizing complex geometry in real time.

#### 2.2.2.4 Programming and Visual Programming

3d modelling software is good for visualization of products. The possibility to ray trace creates realism on a scale not achievable otherwise so far. CAD software on the other hand is good for making the model work for manufacturing as well as providing the possibility to use FEM simulations to test durability. But these softwares lack the possibility to make an application interactive and realistic. What programmable software contributes to these two cases is the possibility to incorporate interactivity in the scene. To place an object and with the click of a button change the colors of the walls in a architectural study or change the rims on a car. The programming is often either c#, C++ or Python. A problem arises when the design engineers or managers, that are the people most likely to need the interactive visualization, lack the proper programming knowledge. Traditionally engineers and managers are more focused on mechanical design and project management. To make the process of bringing interactivity to a scene easier, a form of visual programming has been introduced in recent years [29, 30, 31]. The engineering sector has tools like Mathwork Simulinks block based scripting that allows for simpler and faster prototyping. Children are learning programming through visual scripting language Scratch [32]. Where puzzle-pieces looking blocks of code can be moved around and connected with fitting blocks to create smaller programs. Game engines has also introduced this type of node based scripting and Epic games Blueprints system has been known to be able to create whole applications in just blueprints. This allows for more than programmers to utilize real time rendering and interactivity. By using the visual programming the application will be less optimized than if the code was written by the user. Meaning applications might be slower and demand a more powerful computer the more visual code is included. But as a trade off between not being able to interact with the models and use the visual programming systems available it is a good solution.

## 2.3 Gamification

The first game was made in a research lab on a cathode ray tube in the 1940s[33]. Since then several games has been developed and the industry today has a larger turnover than the Hollywood film industry [34]. Games contains many genres ranging from action games to city building and puzzle games. The average gamer is in his 30s and most is over age 36, men are more common gamers but the margin between men and women are slimming, reports Entertainment Software Association [35]. The average time a person is gaming is around 15 hours a week [36]. According to a study by naval research from Kenan-Flagler Business School summarized in [33] it was found that people that were playing games had up to 20 percent better cognitive and perceptual skills than non gamers. There were also reports about better retention rate of information, higher declarative knowledge and better understanding och complex problems. Gamification is a term used to describe the process of making a task as engaging and interesting as games can be. Meaning involving engaging tasks to solve problems and a reward system to keep interest.




(a) (b)

**Figure 2.12:** a) The first stereoscopic viewer b) how each eye percept stereoscopic view [17]

## 2.4 User study

For the user study a qualitative approach was chosen. As Ullrich and Eppinger suggest in the book Revolutionizing Product Development [37] a Small sample size is sometimes preferable. This can be in cases where qualitative data is needed, the test or survey is performed early in the concept development process or required investment to develop and launch the product is relatively small. To evaluate the software and the workflow for using it few qualitative studies were to be performed with lead users.

## 2.5 Virtual Reality

### 2.5.1 Virtual Reality History

The term Virtual Reality has become synonymous with entering a computer generated world through the use of either Head Mounted Display (hmd) or surrounding screen projections. A virtual world is then projected on screens in front of the eyes. Mikael Litendahl described virtual reality as a diving experience. Where the virtual world is entered like an ocean and the user get to experience this world as she or he wants. In VR the user can also control hands, feet or other parts with the help of trackers who's position is constantly read with infrared sensors. The user can interact with objects in these virtual worlds through these trackers. The thought of entering virtual worlds has inspired people and engineers since the 1800s when stereoscopic viewers was the newest intention to hit rich homes [38]. The technology was based on each eye observing one image each, slightly offset resulting in a 3d effect, see 2.12. These devices was purely static images without moving parts. The virtual reality industry started taking its current form in the 1900s with the emergence of electricity and computers. 1929 the first flight simulator was introduced, the "Link trainer" in 2.13. The Link Trainer consisted of motors simulating pitch, jaw and smaller turbulence and was used to train pilots during world war 2.


**(a)** **(b)**

**Figure 2.14:** The immersive cinematic experience "The Sensorama"

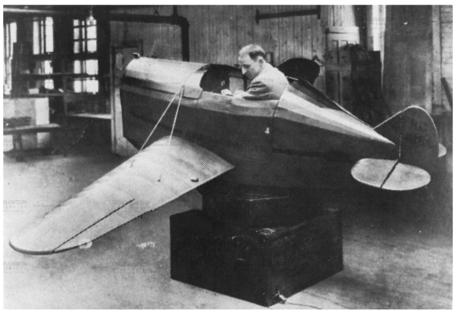


**Figure 2.13:** The Aerospace Link trainer flight simulator.

With the introduction of computers and computer screens actual virtual reality could take form with the 1950 introduction of the Morton Heiligs Sensorama 2.14. The Sensorama was a cabinet with a stereoscopic 3d display in front of the user. Surrounding the user was walls with speakers, smell generators and the user was sitting on a vibrating chair. The Sensorama was an immersive experience for all the senses. Morton Heilig continued in the 1960s with the first HMD, though the device was purely a sterescopic screen with no motion tracking technology. This came 1961 thanks to two Philco Corp. engineers developing the Headsight. The Headsight had two screens, one for each eye and motion tracking technology. Though the device was built to move a camera so when the user moved the head the camera mimicked the movement 2.15.

(a)

(b)

**Figure 2.16:** a) The Sony virtual reality headset b) a VR arcade experience from the 1990s

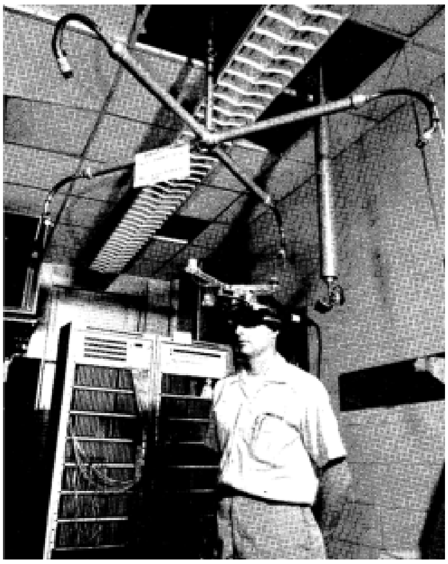


**Figure 2.15:** The first headset with motion tracking technology.

The time between 1960 and 1990 saw many steps being taken in the virtual reality field. The emergence of the wireframe graphics provided the hmd's the ability to project computer graphics instead of just tv screens. The tracking technology got better and motions tracking for other parts than just the head was introduced. In 1987 Jaron Lanier founded the Visual Programming Lab and coined the term "virtual reality" and developed and sold several VR devices including haptic gloves. VR technology existed as a hype technology during 1990 to 1995 and arcades game halls and several devices and business was created. But the technology was too young and immature yet to reach the larger population as it was to expensive, slow and graphics was not good enough yet. Sega was one company developing VR goggles with head tracking, stereo sound and lcd-screens. The hmd never got released as technical difficulties stopped it. 1995 came the Nintendo Virtual Boy, a portable gaming console and a first in the gaming market with an hmd built into the console. The Virutal boy flopped and made low sales, much thanks to the display colour which was in red and black only, the weight of the headset and low software support. In 2010 Pamer Lucky invented the Oculus Rift, an hmd with 90 degree field of view and high fps and good motion tracking technology [39] 2.17. This kickstarted a new wave of virtual reality technologies. Luckys business was later bought by Facebook, a sign that big companies was interested in developing the technology and that it was here to stay.


**Figure 2.17:** The Oculus development kit 1, the kickstarter for the modern commercial virtual reality.

### 2.5.2 Virtual Reality Hardware

Virtual reality is the term chosen to describe the experience of travelling to virtual worlds through the use of hardware. This hardware can differ from the head mounted displays described in the previous chapter to a room with multiple large screens projecting the image. There are usually trackers available for tracking hand, feet or full body. There is also trackers for objects for example mounted on nutrunners for manufacturing or ergonomics studies 2.18 [40]. Head mounted display versions usually comes with trackers for their hand controllers.

**Figure 2.18:** A couple of trackers mounted on a persons legs.

Currently (2018) the most commonly used VR systems are HTC Vive and Oculus rift. They both use similar technology. Their hmd's contains two screens, one for each eye, shooting 90 - 120 images every second in 2160 x 1200 resolution and wall mounted sensors constantly tracking where everything is in 3d space within a volume of 5x5x5 meters. The computation is done on the computer and the hmd's are most commonly connected to a fairly powerful computer via hdmi and usb chords. The graphical computations are mostly done on the GPU by drawing each frame twice, once per eye. Though this can usually be optimized by tweaking and taking shortcuts like calculating objects that are far away once as these are less prone to be experienced as stereoscopic.

**Figure 2.19:** The Oculus Rift and the HTC Vive virtual reality goggles.

Recently Oculus released Oculus Go [41]. A headset developed to be more consumer friendly, its 200 to 300 dollars cheaper and comes without trackers and only works through tracking headset movement through built in gyros. HTC released their HTC Vive pro, aimed at business [42]. It has enhanced resolution 2880 x 1600 and audio, and higher tracking distance. Varjo is a new headset coming to market in the near years [43]. Varjo has overall a lower resolution but an area in the middle of the screen with increased resolution. This area is upon further release supposed to move depending on where the eyes are watching through eye tracking technology. There are also several solutions that use mobile smartphones and headsets they can attach to [44]. VR solutions using will mounted screens are called Cave systems and are considerably more expensive, around 1 to 2 million dollars [45]. Cave systems have higher resolutions 1260x512x4 (if 4 screens), gives the ability for several users to experience the same setup and completely lacks headsets or the feeling of loneliness that users of headsets can feel. Though at the cost of higher costs, harder installation process and usually demands special built hardware to run.

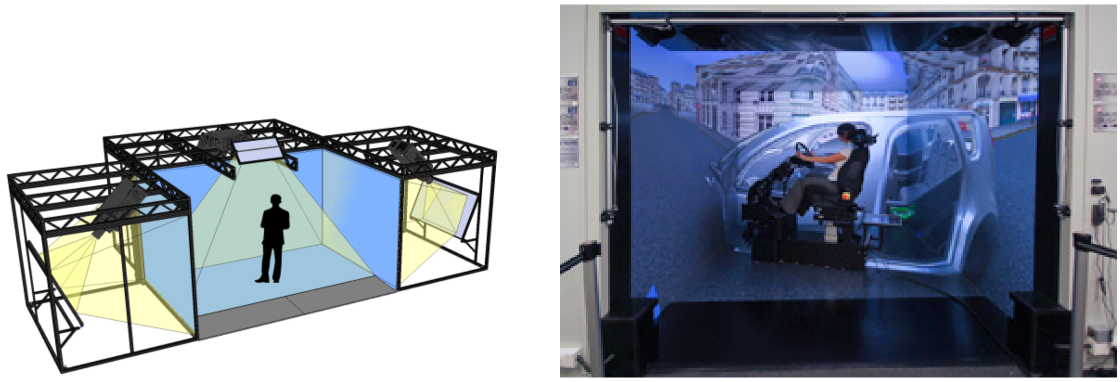


**Figure 2.20:** The Visbox, Inc. Virtual reality cave system and the Nvidia Cave system

## 2.6 Related Work

### 2.6.1 CAD modelling and visualization

Jacobssom and Syrén writes in their report about the needs of CAD visualization within the industry[46]. Their focus is on the needs of other departments than just the design engineer, who would otherwise be the main user and most commonly thought of. They performed 54 interviews with engineers from managers, mechanics, test, design, product coordinators, production and marketing to find the potential needs of visualizing CAD data within 5 different companies within engineering. They find that CATIA is the most used CAD software within the industry. Product development within the companies included in the study is often cross-functional with several different actors. They write about the Toyota manufacturing system [47] applied to the product development process, meaning there is waste in the development process also in form of waiting times, unnecessary or repeated work or loss of information thanks to incompatible software workflows. Waste within the product development process is discussed as something a potential visualization software has to eliminate. Furthermore they discuss the importance of knowing that the same technology can be used very differently depending on who the user is. Implementation of new software must be seamless and allow for users to personalize their experience to their needs, changing the software instead of changing the user. The new software has to support the process and not drive it. Meaning that the whole company can not change because a new software was introduced. Be specific and solution oriented, meaning that the software will not solve every problem within a company. It should also not be the most expensive and the biggest software, as long as it solves the problem. Robertson and Allen [48] early pointed towards the relation between CAD communication and engineer performance. They concluded that there was two reasons for this improvement in performance. It allows engineers to see different design solutions and it lets them see relationships between parts and how this assembly fit together. Furthermore, Jacobsson and Syrén also explains their conclusions about what makes a good visualization. A good visualization is when the model is not only looked at but also interacted with. It should be able to show a simplified overview of the visualized assembly and when needed a deeper more thorough view of the model should be possible to obtain. The performed interviews pointed toward a diverse need for CAD visualization. The managers felt they were not possible to properly review and be part of the feedback loop. It hindered there wish for earlier prototyping and testing. Though the security of exporting and importing models was also expressed as a concern. The Physical testing department expressed a visualization tool as a possibility to understand the surroundings of the test subject. The design engineers expressed their need to connect with stakeholders and discuss model information and design over cross-functional teams. They descried the problem that arose when doing test group studies. As much as 50 hours could be needed to explain the context of what was to be tested such as surroundings, usage and background. They also expressed needs to show assembly movement, such as springs bottoming. Similar problems with understanding arose when trying to explain design decisions to managers. If managers had access to model information beforehand, discussions would be more fruitful. The mechanics in the RnD divisions told the authors about the importance of the visual aid and understanding that a good software could bring through measuring tools and cross sectional views. The interviews also highlighted miscellaneous needs such as red mark or draw on models and ability to play film sequences of kinematics.

### 2.6.2 Choosing game engines

When choosing a game engine [49] Dickson et. al. writes about the important factors for the student when developing a course in game development with game engines. They are outlying 7 factors that should be thought about before deciding. They include a shallow learning curve, game engine popularity, Available programming languages, cross-platform development, the stability of the engine, the community using it and the documentation and finally affordability to consider. The learning curve should be short and allow for rapid prototyping. Game engine popularity is more relevant for the students interest, while in the scope of this report the importance of popularity should be for the additional aid the documentation and larger community would bring. The available programming languages is important as some languages are harder to use then others and some game engines dont need a programming language at all but can be programmed through the visual programming earlier described. The cross-platform development is important as most models and some textures are better developed outside the game engine and to make these transitions between softwares quick and painless is important for productivity. Falk Andersson et. Al. are also exploring the choice of game engines and the important factors through the development of a "white room benchmark" [50]. White room as in a sterile environment where only what is to be studied affects the environment, in this case a game level where features and and interactions can be studied independently of the game engine. Their research adds to the list och factors by discussing cost and affordability and suitable content creation tools. Content creation tools refers to the software used to create 3d models, materials and vfx. They also discuss the availability of an integrated development environment, meaning the level of complexity of working in different environments or softwares and the export between these softwares as a factor allowing for collaboration between different actors. The target audience as well as target platforms/ platforms are also discussed.

This is of course more relevant when discussing game development than engineering applications, but similarities exists as different engines have different possibilities and focus and this is important to consider whether the application is developed for an engineering board-meeting or millions of players. Their view is that these factors provide an incomplete evaluation and adds more. The openness or the interoperability between different softwares and the licensing restrictions of these are taken into account. Re-usability, the possibility to use the game engine in the long run with continued support and maintenance and ability to update the projects in newer versions. The "white-room" approach allowed the authors to demonstrate and test different game engines against the mentioned factors. They did this with an environment with features such as doors and other interactable objects, water that can be accessed by the user, stairs and steps. The technical features evaluated by this approach was the rendering of the scene with textures and light, testig shaders and representation as a 3d scene. The interior geometry representation, defining and directing the AI, collision detection between objects, physics, sound and scripting as means to control the engine through scripts.

### 2.6.3 Game design and game enhanced design

Kosmadoudi et. al. presents in [51] their research about using gamification to enhance the user experience when working in CAD. They find that as games have always been fun and engaging with a creative community, a good example is Valve Corp.s "Steam Workshop" where anyone can upload modified mechanics in games [52] and do also possess very realistic graphics. They argue that CAD software has become more than just accurate modelling, simulation and visualization tools. Today CAD software have added functionality to aid in manufacturing and analytic activities. This added functionality as argued to have made CAD software cumbersome and difficult to work with leaving drops in productivity as results. The article is focused on the mechanical CAD in the geometrical or modelling stage of the engineering design process. An empirical study reported about showed that users cannot remember possible alternative methods to solve a problem but relies instead of using the same menus or buttons or strategies as earlier. They do not understand why something is done in a certain way. They find that repeated use of mouse and keyboard is distracting users from creative design thinking. Several alternate ways of using CAD modelling is discussed using different hardware as pen and tablet, haptics or 3d-mice. During a study of 39 design students where the task was to spend a morning designing products on paper and then an afternoon of CAD modelling these drawings it was concluded that CAD has no or little value for the creative thinking. This proposition was strengthened by another author they referenced who carried out a survey of 200 CAD practitioners who found CAD modelling to continually hinder their creative process by premature fixation, circumscribed thinking and bounded ideation.

The use of game engines like Unreal Engine, Unity or Cryengine have been proven to aid the automotive, military simulations and VR development toolkit thanks to realistic graphics and simulation of Newtonian physics. This together with the inherent mechanic of interactive game play game engines serves as a powerful complement in the CAD engineering, architectural, construction or simulation process. The idea of using game mechanics to enhance the planning process within design and marketing is increasingly being studied. One of the bigger areas outside engineering that is reaping benefits is education. Where the engaging and self-reinforcing context that gamification can provide is used to motivate and increase students and their understanding. Kosmadoudi et. als. research further points towards the positive collaborative possibilities. As design is seldom done alone, an environment where understanding and explanatory activities is easier is beneficial. Much because the design process happens over several different stages. They also write that the future challenges will lay in realization of the importance of strategic game models in design studies. Another research project described in their report had performed a study on simulated clinical training. The game engines Unreal Engine, Id Tech engine and Source Engine was tested based on stability, availability, possibility of custom content creation and cooperative activities via network. It was concluded that game engines are good and inexpensive means to perform simulations. They also saw that game engines are continually getting better and new features are introduced as processing power gets better. Marketing also saw benefits when applying game mechanics to incentivize user behaviour. Another study Kosmadoudi et. al. describes points out that immersion can describe level och concentration achieved, hinting towards VR immersion and its benefits. The study continues with describing the optical experience and presence as the feeling of "being there" and the positive experience these factors create. These findings are later proven in a user experience study of four engineering students. Their brain waves and stress levels were measured vi bio medical devices as they were using CAD and a "game engine like" environment. The study indicated that the game engine like environment gave a more positive response with lower levels of stress and a positive brain functionality. Kosmadoudi et. al. continues with describing studies indicating positive responses in design and creative freedom when using hardware attachments like motion controllers tracking the body movements of the user. The last chapter deals with the use of game engines as simulation tools. The most used physics components for game engines is PhysX and Havok. These are what is handling object collisions, water flow or just objects falling to the ground. As Kosmadoudi et. al. writes "Game engines and their advanced "gaming physics” are capable of delivering physics capability for simulations other than in the entertainment area. Future CAD will reflect not only the current multi-functional engineering environment but also physics mathematics simulations based on the Newtonians game engines such as CryEngine", highlighting the importance of both softwares having components to handle modelling and simulation as well as creating an engaging environment for understanding and creativity.

I their study Dr. Uskov and Sekar writes about the trends and future tendencies of how companies will use game features in their business [53]. They finish their extensive study in four key conclusions. They point out that usage of game features is in its early stage but that it is a fast growing trend among business. Business that are prepared will have great benefits and game features will help solve the ever more complex engineering problems. These solutions will come thanks to the understanding and visualizing nature of game features. It will benefit teams and individuals alike through enjoyable training, deeper insights, health activities, harnessing of customer insights as well as innovation through the communities. And lastly, game features should be used for design of both prototyping as well as the final products or service.

### 2.6.4 Virtual reality

Georgiev et. al. Discuss in their report how to augment the creative design thinking through VR[54]. Their research conclude that virtual realities immersive nature help in the understanding of relationships and other spatial perception factors, as long as the user feel immersed in the experience. Abardeen Essentials, a business doing research and business analysis writes that virtual reality is the next step for manufacturing and engineering [55]. The immersive nature helps engineers take the step from 3d images to a 3d environment. Through laser scanners exact 3d copies can be created, helping rapid prototyping by making changes cheap and fast without the need for physical models. Johansson and Eriksson discuss the use of virtual reality as a communication tool within construction companies[56]. The summary was that all participants in their study thought VR helped in the understanding of the architectural and sun studies. Carlsson and Sonesson investigated the use of virtual reality in a user experience study [57]. As development times has become shorter and the importance of early prototyping has risen VR was used to find its potential effects on the development times and increase understanding of the technology. The study included building a physical test rig of a car interior with seat, screen control panel and a dashboard including steering wheel with just simple wood and no colors. They then used VR to layer a digital model over this interior to make it look real and make it interactive. They concluded that to not break the immersion when performing user studies a high level of congruence between virtual and real work objects is needed.

# 3
# Method

This chapter will present how the thesis was performed and in what order. It will lay out the structure of the following chapters and what they will contain.

## 3.1 Pre-Study

### 3.1.1 Literature study

To be able to present and evaluate different softwares and to explain the reasoning for using other software than traditional CAD software in engineering, there were concepts that needed to be explained. To understand the limitations on a high level the literature study focused on developing the understanding of CAD- and- Game engine architecture. Concepts like lightning and rendering techniques were studied to understand the state of the art and why CAD softwares and game engine softwares are used for different applications. Included in this study was research about 3d file formats and the loss and gains of information of some of the more common format of data exchange. Also similar cases were evaluated where authors used game engines in engineering applications to understand why they did so.

The literature study also explored the area of using CAD models in game engines for expanded understanding of how assemblies function in a game engine environment. This was more studied through the user studies since the area of game engines in engineering is a new field of study.

### 3.1.2 Market Analysis

The market analysis was performed to find and list the biggest actors within game engines and CAD softwares. The focus was put on deepening the understanding on more specifically what kind of environment the different software are moving in. This to be able to make a choice based on the needs of the user. The analysis was performed by critically reading material provided from the education Product development at Chalmers University of Technology as well as sources provided by the companies of the softwares. Market analysis done by larger analysis firms were also used as reference. A large part was focused on why the softwares are used differently.

#### 3.1.2.1 Software Study

The software study focused on the abilities of each software. This to be able to make an more informed estimate of the best tool to use for a specific case. Important aspects and pros and cons of the softwares was explained and what techniques is used to show the model on the users screen. If a software have some particular focus this was mentioned

### 3.1.3 User Study

The user study was focused on analyzing the user workflow of different softwares. This to see what the user identifies as key aspects of that software as well as the users experience in different softwares. An important aspect of the study was what the current pipeline to visualize CAD data in softwares other than traditional CAD softwares were. Why it was done this way and why the involved softwares was used. A qualitative interview structure were followed where lead users where interviewed. A lead user, or extreme user, is someone who is ahead of the market in using the product. Where their need have made them experience the inadequacies of the current solution and maybe have even invented their own solution or workflow to solve these issues [37]. Such a user can express the needs of a new solution on a better level than regular users.

### 3.1.4 Virtual Reality

This section handled what virtual reality is and how it can be used in engineering applications. It also explains the history of VR and why VR has not been utilized before. Since the technology has grown the last 10 years it still is developing rapidly concerning both the hardware and software used to producing images. This chapter mentions the current state of the art technology and some specific use cases. It also looks into trends and why VR is being used in engineering. The information was acquired though forums and the web, confirmed through papers and developers own homepages.

## 3.2 Design of Experiment

This chapter used the former sections of the methodology chapter to evaluate how the concepts mentioned was used in an actual smaller test case. The design of experiment included the choice of software, the use case best suited to fill a user need and how the case could be executed in real time rendering of an interactive environment in VR. The demands from the start was that the model should be a typical CAD model, either a .catpart or .catproduct , .step or .wrl file. These file types was chosen based on discussions and previous knowledge. The files should be around 10000 parts as such an assembly was, according to the Senior Product Manager for Unreal Studio at Epic Games, sure to stress test Datasmith. It should also preferably contain smaller assemblies to be able to scale the test as desired. The test should allow for exploring of the software features at start. When some feature was not understood it was explored through tests as well as discussed with relevant person, most often the mentor at Alten, the contact person at PiXYZ or the Senior Product Manager for Unreal Studio who were more then helpsome.

# 4

# Results

This chapter will present all the results put up in the method chapter. The experiments performed will be presented in more detail and there after all its results.

## 4.1 Pre-Study

The visualization market is big and a complete coverage of every available software and its pros and cons would take too long. But for a comparison, a few of the software more commonly mentioned during the research in the theoretical framework chapter will be described.

### 4.1.1 Market Analysis

According to a market analysis of the 3d CAD software performed by "Grand View Research" The market of CAD software was in august 2017 worth approximately 8 billion USD with investments in 3d printing contributing to expected further raise [59]. Cloud computing and rapid prototyping is leading drivers for the development of this segment and the automotive and aerospace industry is using CAD to a further extent. The development of design and illustrations is more outsourced to improve businesses design productivity. The competition is increasing in the segment and end-use industries are demanding new and better design products. This has led to solution providers venturing into complementary products and technologies to counter. The healthcare industry has surged with innovation as the 3d printing of tissue has increased demand of 3d CAD software with need for personalized models of body parts. A legislation making use of Building Information Management, BIM, mandatory in the UK has made countries in the middle east follow this legislation. BIM software is used instead of 2d drafting in the construction industry [18] and is a good example of the usage and potential of rendering technology for engineering. The key market players in CAD are Autodesk Inc., Dassault Systemes, PTC Inc., Siemens PLM Software Inc., and Oracle Corporation.

The market analysis by grand view research for the visualization and 3d rendering market segment describes a similar growing market and is currently valued to 950 million USD [60]. The increased demand for short just in time marketing and innovative solutions are described as significant drivers. The competition is big and user friendliness is used as a selling point. Lower cost and time management through photo realistic media, such as ray traced movies and pictures attracts business to use software solutions instead of traditional workflow. The possibility for real time feedback bring real time insights and the focus on innovation aids in the decision making process leading to maximized profits. While the increased competition leads to more user friendly interfaces making users more productive. The market is hampered by lack of infrastructure such as slow internet connections hurting the design, development and delivery of 3d content. Where infrastructure is solid and working visualization and 3d rendering software provide solutions for product design, modelling, animation, imaging, graphics, special effects like particles or fire and simulation. These solutions together is what contributes to the insights and increased understanding of the product. They also lead to faster decision making and better operational and marketing decisions.

Visualization and 3d rendering will grow significantly until 2025. As cited by the report: "Visualization and simulation allows enterprises to manage customer insights and provide customized services related to visual effects, animation, and graphics. Additionally, increasing competition is encouraging companies to adopt visualization and 3D rendering software for gaining a competitive advantage over their contemporaries". Business all over architecture, manufacturing, engineering, film and gaming are utilizing visualization and 3d rendering software to produce accurate design information. Adding to this is virtual reality further pushing the limits of developers as well as customers product understanding. The healthcare industry is a growing user of the technology and an example of how visualization can help understanding [61]. The authors of [62] states "There are several arguments in favor of use of virtual reality in medical education: it can be used for “difficult to simulate” scenarios and to standardize a scenario, for example, for use in exams. However, there are limitations to its usefulness because of the cost implications and the lack of evidence that it results in demonstrable behavior change". The lack of evidence is in turn discussed and confirmed to help in [63]. Grand View Analysis also writes in their report that healthcare applications include visualizing bodies for surgical planning and training as well as communicating with patients and families of patients. The leading actors in this market segment is Autodesk, Inc.; Dassault Systèmes; Trimble, Inc.; Corel Corporation; Adobe Systems Incorporated; The Foundry Visionmongers Ltd.; Chaos Software; and Luxion, Inc. The dominant actors have merged with and acquired smaller enterprises to cover more solutions. The leaders, for example Autodesk and Dassault Systémes that are both also mentioned as leading actors in the CAD market segment, are examples of the merging and acquisition trend.

CGArchitect.com presents itself as the leading community for architectural visualization professionals and have many times been presented together with Autodesk [64] and delivers "the largest annual awards event designed especially for the architectural visualization industry" [65]. It is regarded as a reliable source for artists to engage with its community and for information about the arch-viz industry. In February CGArchitect released their 2018 Architectural Visualization Rendering Engine Survey results [66], where the community of artists and enterprises where asked questions regarding their use of software and workflow for visualization purposes. The answers included software from the large actors Autodesk and Daussault that were mentioned earlier in this chapter. But included was some software that points to the earlier mentioned shift to real time media usage for maximizing profits, a shift in the visualization and rendering scene. 30% of respondents are today using non traditional media for rendering purposes. 21% of respondents are Using the game engine Unreal Engine made by Epic Games and 8% were using a similar game engine, Unity3D by Unity Tech. as a rendering tool. 53% were experimenting with other software and of those 41% were experimenting with Unreal Engine and 11% with Unity. 61% was somewhat or very likely to adopt the new rendering engine in their workflow. When voting on reasons of importance the reasons that were voted as "very important" was related to real time client explanations, design discussions and the possibility for Extra Reality, XR, (VR, AR, MR). The most used game engine and the game engine considered most important for the respondents workflow was Unreal Engine.

### 4.1.2 Software Study

#### 4.1.2.1 Unreal Studio

Unreal Studio is the property of Epic Games. A game developer created in 1991 [67]. They are behind several major game titles such as Fortnite, Gears of War, the Infinity Blade series and several others. Their games are ported to several systems including Windows PS4, XBOX, switch, Android devices, HTML5, IOS, VR and AR and Linux. The difference in graphics in their games range from the cartoon style of Fortnite to the realistic graphics of their now cancelled game Paragon, see 4.1. The range of graphics and the possibility to port a multitude of systems is possible thanks to their games being developed in their game engine Unreal Engine, making post-processing and compiling to different hardware easier. Unreal engine 4 is the latest engine version and it was released 2012 and is currently free to use below a 5000 USD in profits threshold.


**Figure 4.1:** Paragon to the left and Fortnite to the right are two completely different graphic styles.

Unreal Engine is built in the C++ programming language and it is also in C++ or the C++ based visual scripting system Blueprints that application development is scripted, example of blueprints in 4.2. Meaning interactive components like AI, interactive objects or player input can be made with either C++ or a visual scripting system like the one explained in 2.2.2.4. Epic Games also has released the source code for Unreal Engine. Meaning anyone is able to download the code and change the foundation of the Engine. Development of applications in Unreal engine is performed in the editor displayed in 4.2.

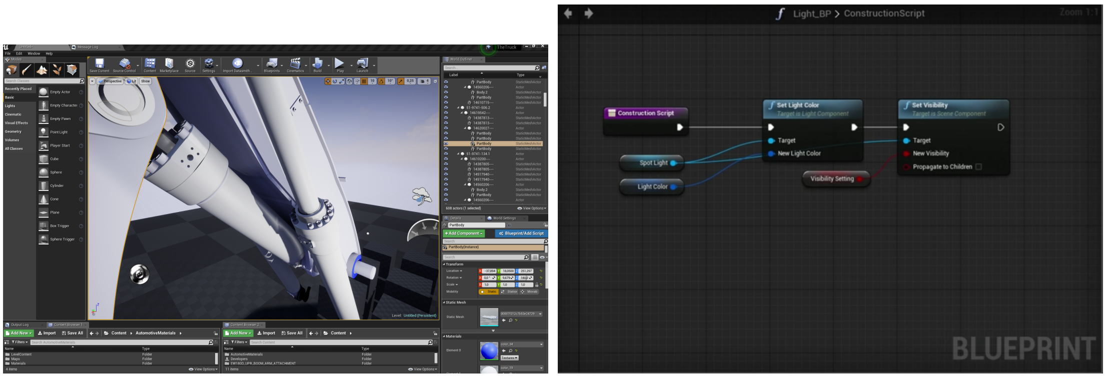



**Figure 4.2:** Unreals Editor to the left and its visual programming system blueprints to the right

When starting a project in Unreal Engine the user get to choose starting an empty scene or a template. A template is a scene with assets already present at start up. There are for example templates for VR or AR visualization, First Person View with a character able to be walked around with keyboard inputs or a template for airplane functionality, see 4.3 for more examples of templates. These templates contain scripts already implemented to get a fast start-up. Keyboard inputs or camera control are features implemented in a desired way in these scenes.

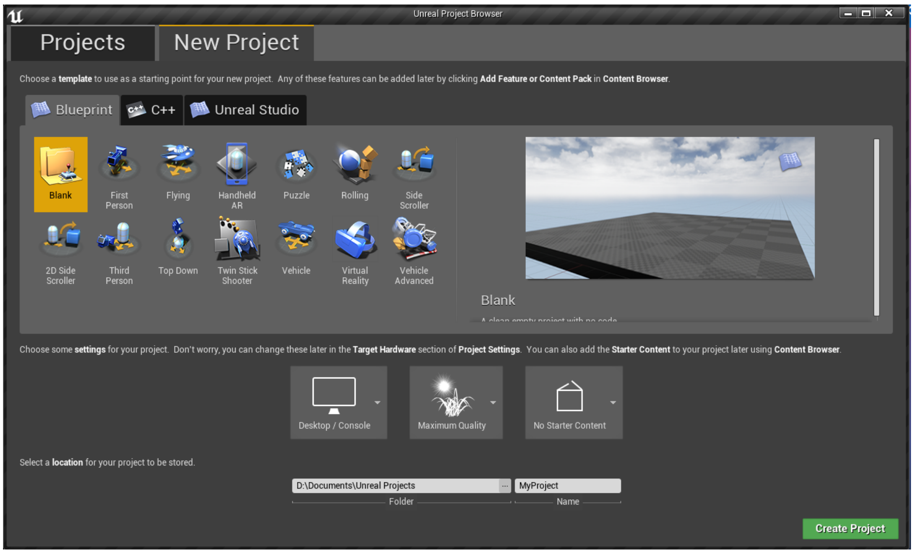



**Figure 4.3:** Example of the different template scenes available in Unreal Engine

A common theme within the editor is that every aspect of development is done in a visual manner. This includes ai development, material scripting and post processing effects which happens in a visual node based system similar to blueprints, see figure 4.4. This include the way models are created into the scene. Models are dragged and dropped from a content browser, instead of programmed in through code. Depending on the needs of the user Unreal can be programmed to display photo realistic graphics or a more stylized one. The engine comes with a marketplace where users upload scripts and plugins for assets and content. Examples include but are not limited to ai, environments or C++ scripts.

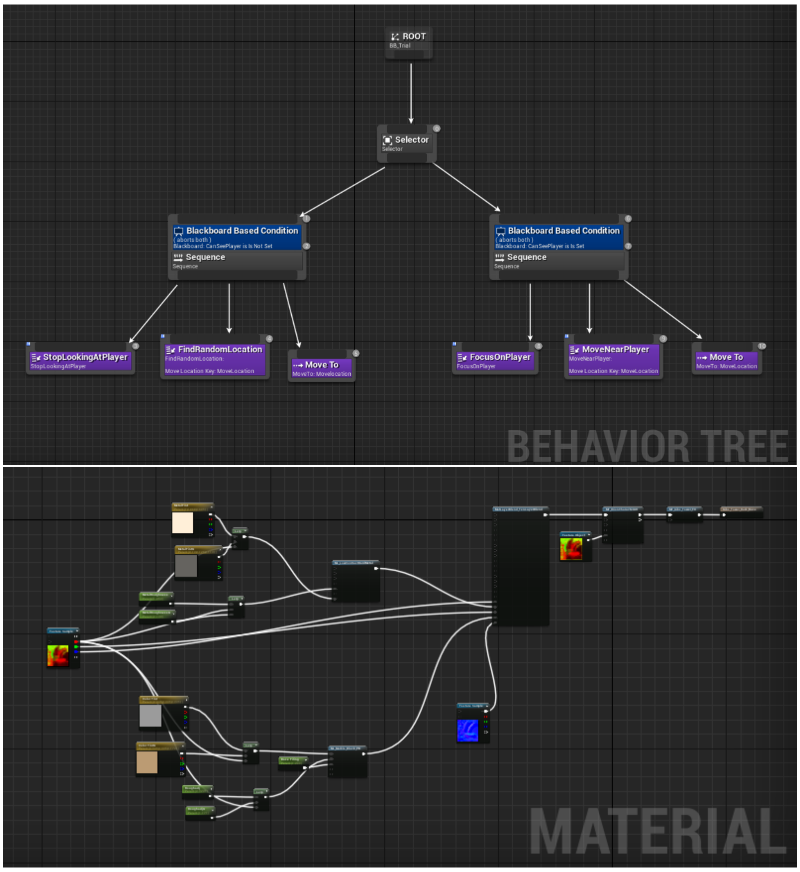



**Figure 4.4:** Examples of the visual programming system in Unreal Engine.

As a need have arisen from the engineering and architecture industry for real time visualization in both XR as well as on a normal computer screen, Epic was one company that expanded on features demanded by the industry such as a smooth workflow to import CAD data. They released Unreal Studio as a tool built up with the same editor as Unreal Engine but with higher focus on product visualization features and templates. The Datasmith plugin allows for fast import of raw CAD data. Datasmith tasselates the CAD model and apply colour or textures, depending on what was applied to the model before, to the polygons. This is done by datasmith automatically creating and applying UV layers and lightmaps to the tessellation. Before importing the data the user choose which Chord length is allowed, that is the distance between the newly created triangle and the original parametric surface, the normal angles, which will determine how exact the interpolated light model of the phong shading will be and max triangle side length as well as some setting for how material and colour will be applied. These settings are then followed and applied to all imported parts of a CAD assembly. The hierarchical tree structure is followed with parents and children of the original CAD software applied in the same way in Unreal. The other way datasmith works is by first manually importing the CAD data into 3ds Max, a 3d modelling software, and there manually determine which tasselation levels to use on individuall parts as well as determine UV maps as well as material, 3ds Max is explained further in chapter 4.1.2.6.

As a scene grows with more triangles, materials and particle effects the performance can suffer. Poor performance leads to less Frames Per Second, fps. When working in VR the fps has to be between 90 to 120 fps meaning how the scene is built up is important. For this Unreal has a performance graph and optimization tools where it is stated what the computer are working on at a specific frame, see 4.5 for example.

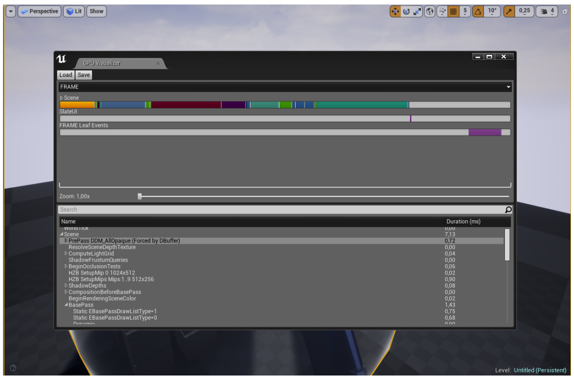


**Figure 4.5:** How the optimization tool looks in Unreal Engine.

Unreal Engine also revealed at a keynote presentation of the state of Unreal in 2018, together with graphics chip research and development company Nvidia, the beginning of support for ray tracing in real time. This technology combines the realism of ray tracing with the real time functionality of Unreal.

#### 4.1.2.2 Unity

Unity3d is developed by Unity technologies founded in 2004 [68]. It was founded as a tool to, as the founder David Helgason describe, "democratize game development". The founders made the Game Engine free and accessible to anyone. When the iphone launched in 2008, Unity was the among the first engines to support development for the Iphone. The Engine has one of the largest platform support and are able to port their games to Anything from Windows or Mac, to mobile platforms and game consoles like Xbox or Playstation, see 4.6 for a Complete Cover.

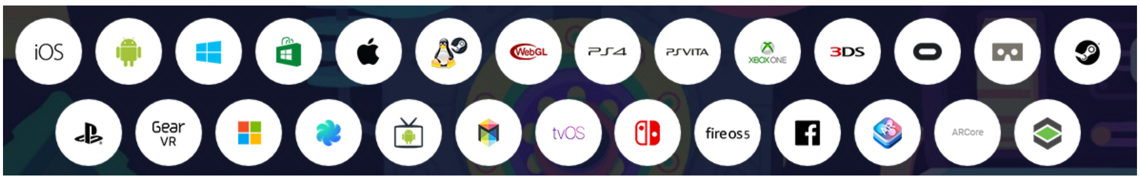
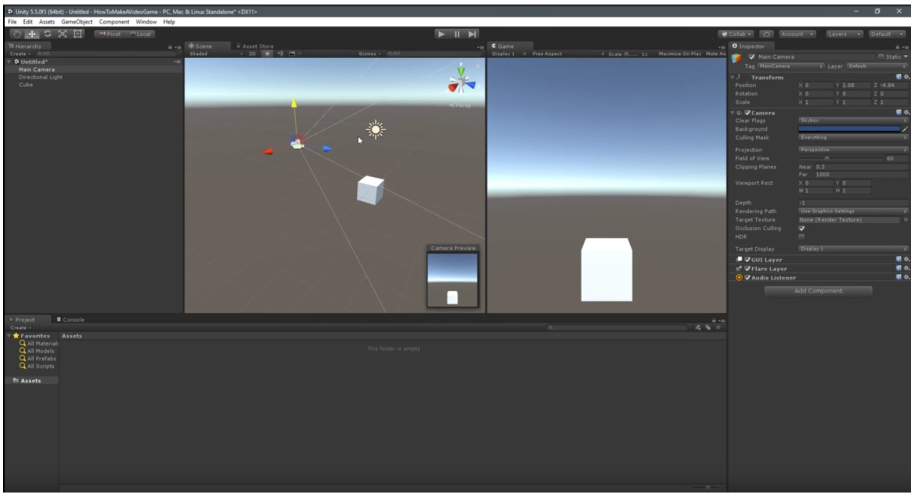



**Figure 4.6:** Unity supported platforms

Unity has a large community with a marketplace for everything from models to plugins with scripts, many for free [69]. Unity is to a large extent built for smaller applications and one-man development studios and suffer performance loss in larger applications when more objects are involved and more computations like AI or particle effects calculations happen at the same time. [70]. Graphics in Unity are on par with other contemporary game engines. Unity is currently offered in different pricing models with a free one built for personal use without income. Then there



**(a)**

**Figure 4.7:** a) The Unity editor

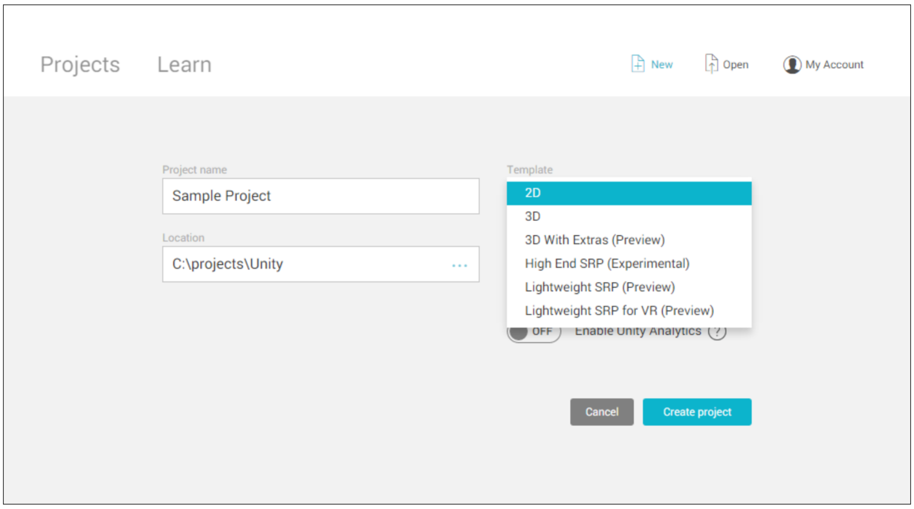
are rising prices for their business developer packages depending on the need of the developer or development studio.

Unity is developed using C++ and it is C++ that is the runtime language. But development of applications is done through the C-sharp programming language. Meaning everything from player input, Ai, materials and particle system is scripted in C-sharp. As Unity has a large community there are paid third party plugins for visual scripting as in Unreal Engine but at a cost. The Unity editor is similar to Unreal Engines and comes with drag and drop functionality of models into the scene, the editor is displayed in 4.7. Unity has some templates when starting a project and these are more focused on the render optimization and the size of the application instead of what kind of input and virtual cameras that are present in the scene see 4.8 [71]. These templates are also still in preview or experimental mode. Already implemented templates are templates for 2d or 3d development. Unity also have support for AR and VR development and have been argued to have better support for these kind of applications, but no test have confirmed this.

Unity also answered the growing need of real time rendered scenes for CAD visualization from the engineering and architecture industry. But unlike Epic Games, who chose to develop an in-house plugin from scratch, Unity Tech. chose to partner with CAD optimization, review, preparation and export software PiXYZ Software. PiXYZ is described more in detail in 4.1.2.3.

#### 4.1.2.3 PiXYZ

PiXYZ is developed by french company PiXYZ Software and is a CAD reviewing, optimization, preparation and exporting software. It is built for importing CAD data and optimize it for given polygon quality, polygon amount or performance optimization regarding removal of unnecessary features such as holes, screws or bolts (defeaturing). It is possible to use python scripts to optimize the parts or automate the process with built in features. See 4.9 to see the editor.




**(a)**

**Figure 4.8:** a) The different templates available in Unity

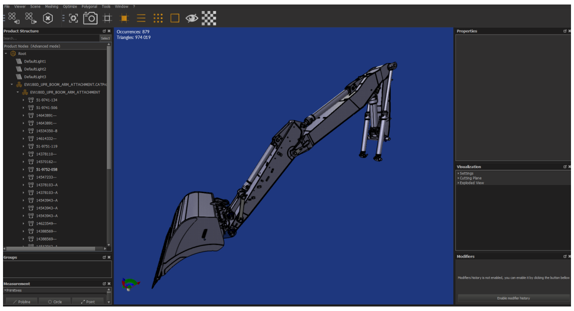


**Figure 4.9:** The PiXYZ editor.

PiXYZ is able to import several different common file formats including those from the two major Autodesk and Daussault Systéme software. File export supports FBX, 3DXML, OBJ, JT, WRLM and more. These file types are able to contain metadata such as material, color, texture and names of these and parts. PiXYZ Software contains three different preparation software and one for review. PiXYZ Pipeline is an automatic pipeline that, similar to Epics Datasmith, automatically optimize the CAD data and tessellates it and save this data in another format. PiXYZ Studio is a manual process for optimizing CAD data and last PiXYZ Plugin is plugins for Unreal Engine and Unity to be able to import the data prepared from PiXYZ Pipeline. PiXYZ Studio and Pipeline saves the CAD data in a desired format compatible with platforms such as Catia Composer, 3DExcite Deltagen, Vred, 3ds Max or Maya. Pipeline is built for smaller projects with around 100 parts and are not suited for larger CAD assemblies. PiXYZ Review is a software for optimizing the CAD data for looking at on the computer screen or in VR. PiXYZ Review is built for low memory usage meaning it can handle large CAD assemblies with the same hierarchical structure as the original file. Review also comes with features such as cutting plane, X-ray capability or product and manufacturing information directly visible in the software as well as support for VR view. The CAD data are imported into PiXYZ and shown in the viewport. Tessellation is needed to see the surfaces as the CAD data is only visualized with NURBS without patches, see 4.10.

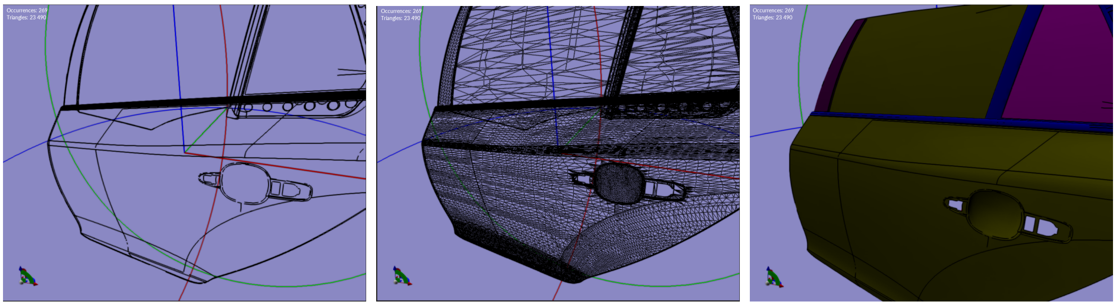


**Figure 4.10:** PiXYZ NURBS modell without tasselation and to the right PiXYZ tessellation options.

The options available when tessellating is similar to Datasmith, with options for chord length, max side length of the polygons as well as UV assignments, see 4.12. The optimization options provide capability for tree optimization meaning removal of empty information such as false nurbs or hidden parts, defeaturing meaning removal of repeated parts such as bolts and screws. Such parts are not needed for view in for example real time rendering engines as they slow down performance. There are also several options for normal alignment and repairing of broken surfaces with missing polygons, see 4.11. The CAD data are then exported to desired software.

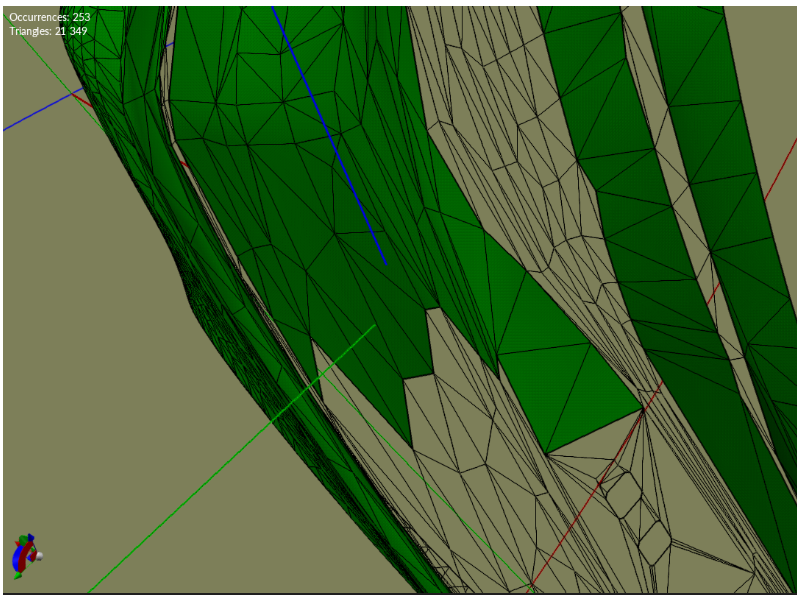



**Figure 4.11:** PiXYZ model where several polygons have normals aligned along the wrong axis, making them see-through.

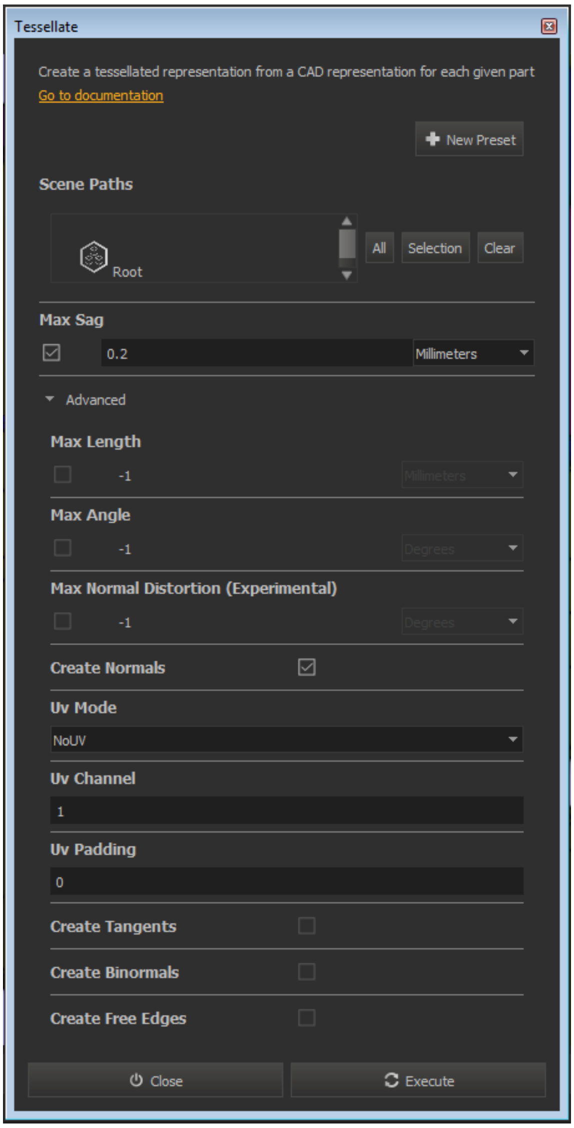



**Figure 4.12:** PiXYZ tessellation and optimization options.

#### 4.1.2.4 VRed

VRED is an Autodesk product within their manufacturing cluster [72]. It is a 3D visualization and virtual prototyping software with a focus towards the automotive industry. It is made for importing CAD models and applying materials, backgrounds and lights and and it is possible to prepare and optimize the models in the editor. VRED supports Ray-tracing and it is possible to render photo realistic pictures, see 4.13. VRED is built for viewing different combinations of models. With drag and drop functionality it is possible to change models like rims or doors on a car model, or the colour of said model. VRED also support Python scripting for more advanced features. For example virtual reality features like interacting with touch screens, see 4.13.

There are also animation features with key framing and certain camera post processing features such as bloom, motion blur and colour gradients. The animations can be applied to specific parts like doors or wheels, or to the whole model, for example making the car move. The optimization options available in VRED includes chang-

(a) (b)

**Figure 4.13:** a) Example of the photo realistic capability in VRED, courtesy of Yuji Fujimura [73] b) Virtual Reality application in Vred with touch screen functionality

ing normal alignment, erase geometry information such as translation or rotation or merging of surfaces.

#### 4.1.2.5 Catia

Catia is made by French software company Daussault Systéme and first saw the market in 1980 [17]. It is the most commonly used CAD software within the automotive and aerospace industry and it is currently in its 6th version but Catia V5 is still the most commonly used in the industry. Catia is built up by workbenches where a workbench contains relevant features for each engineering sector included in the workbenches, see 4.14.

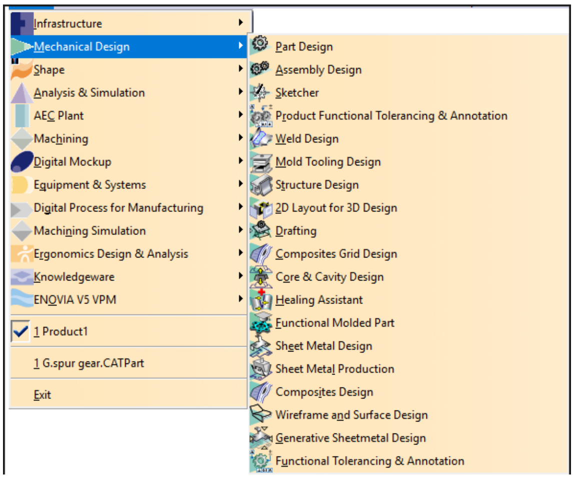
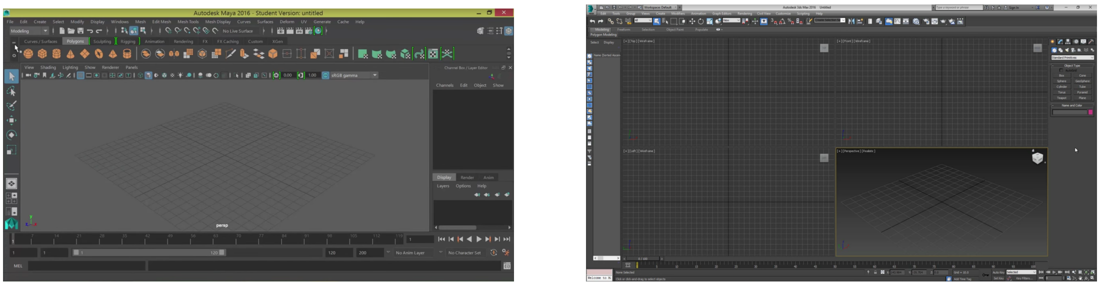



**Figure 4.14:** Different Workbenches in Catia with the mechanical engineering workbench detailed

There are specified tools in each workbench for surface design, electrical engineering or drafting, etc. The models in Catia are solid models with parametric surfaces. The parts can be collision tested and mass analysis is possible. Thanks to the surfaces being parametric, it is possible to decide inter-relationships between parts via scripts. It is also associative and all changes to a part is tracked and changed on every location the part is present.



(a) (b)

**Figure 4.15:** a) Mayas editor b) 3ds Max editor

#### 4.1.2.6 3ds Max and Maya

3ds Max (Max) and Maya are both 3d modelling and animation software used used within the animation and rendering industry for film, images, games and modelling. They are on the surface similar and for the sake of this report they will be explained together. But they are two different software with differences and to decide which to use a more in depth analysis is recommended. Maya and Max are the state of the art tools for modelling and animation within architecture, product visualization and film, their editors are visible in 4.15. The rendering tools shipping with Maya and Max allows for photo realistic images with virtual cameras, lights and post processing effects. Modelling in both softwares includes the possibility for scripting and several modelling toolsets that assist or automate some features. Both offer direct manipulation of polygons and rigging of characters, particles, key framing and material scripting and shaders.

### 4.1.3 User Study

#### 4.1.3.1 Design workflow at Alten Design And Visualization

The user study was performed with two lead users of 3d modelling and real time rendering software. As using real time software in engineering and product visualization is a new workflow an interview to get an idea of how this media is used was appropriate. The interviews were held with employees at the Alten Design and Visualization (dnv) department. Dnv consists of roughly 40 Computer Graphics (cg) artists, design engineers and currently one real time rendering developer working in Unreal Engine. The cg artists usually work in 3ds Max, Maya, Unreal engine or other animation and post processing software. The design engineers work in Catia, Solidworks or using normal product development workflows (matrices, discussions or relevant engineering workflows). The workflow for traditional still rendering, visualized in 4.16 is that customer contact is initiated either by customer contacting dnv with a problem or the opposite where dnv tries to find a problem that the customer have that dnv can solve.

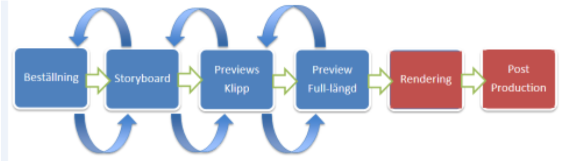



**Figure 4.16:** The workflow when working with still frame rendering

Earlier each artist has worked as independent "islands", as one employee described it, handling all customer contact and communication individually. But it is now mostly handled by a technical specialist and project leader dividing work appropriately within the department. After initial customer contact is made the contract is then evaluated and discussed to understand the problem and how the problem can be solved. In the case of the cg artists that problem is often the creation of a scene for a rendering of an image or an animation of that scene. The models used in the scene is often provided by the customer. The models are either delivered as solid CAD models or visualization or surface models. To be able to quickly iterate ideas with the customer a storyboard is made where a simple representation of the scene is set up, the storyboard can be sketches or reference photos or sometimes an actual simple scene in the software. This storyboard is used in discussions with the customer to get a better picture if how well the artist thoughts match the customers. When a common vision is set a simpler scene is set up in 3ds Max and a quick animation is done to further enhance the match between the customer and the artist thoughts. When the customer agrees on the design and animation flow, the full scene is built with character rigging, materials applied and finally the scene is rendered.

Real time rendering workflow differs some from the traditional one. As there are no rendering at the push of a button the work can be easier iterated, see 4.17. Changes are quicker to perform as lightning information and graphics stay more or less the same. Though this comes at the cost of less realistic graphics, the need for more powerful hardware and not being able to use the modeling and animation features of the 3d modelling software.

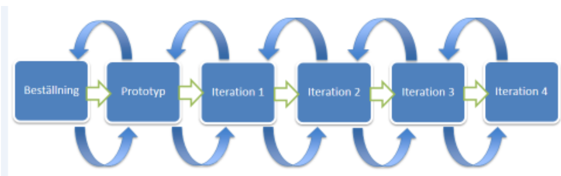



**Figure 4.17:** The more iterative flowchart of the real time rendering workflow

#### 4.1.3.2 Interviews

The interview questions can be found in B. The interviewee 1 is a cg artist and has been using 3ds Max, Photoshop and Aftereffects for more than 6 years. He is more involved in the animation and rigging part of the scene rendering than modelling. He also scripts and applies the materials on the models. The earlier workflow where every employee was like an isolated island working on their own projects meant for him a lot of administration duties and project leading. This took time from the animations. The bigger projects where a whole group are working together fits better as there is other artists to iterate ideas with. Larger projects also allows for each employee to have a role, one performs animations, another modelling and another post processing etc. The problem with working in these software are the graphics. Currently 3d modelling software could have better graphics in their editor. The preview graphics in these softwares are for modellers to see the scene objects, not to evaluate the final rendered image. But by tweaking values of the rendered image like resolution or lower lightning the artist can acquire a good enough perception of how the final scene will be but this requires rendering the image. The hardest part of the artists work is to find out what the customers wants. Feedback loops takes time and its hard to find out exactly what the customer wants. To show the customer the result the image also has to be rendered. Often the customer might not know what they want before they see it. Another problem is to actually solve what the customer wants. One example given by the artist was a customer that wanted a flow of liquid through a transparent pipe. The artist then had to figure out how to do a water simulation. Often the models delivered by the customer are not perfect meaning that when the water started to simulate in 3ds Max the model leaked. He then had to fake the water flow with other methods. "You just have to set up a list of solutions and work through them and see what works" he explained. Other times its the software that is seemingly the problem, where an example was given were a glass material suddenly started to act weird. "It was a problem i had never seen before, so we finally just shut of that function and it worked" the artist said. The problem with communication with customers is explained as necessary and the artists usually work on another problem in the mean time. The problem with models are the most common problem. "The models are never clean, its always something wrong that has to be cleaned up" was one reaction. Common problems is that the surfaces acts as their own geometries, meaning models can not be selected but only its individual surfaces. This also makes the tessellation not working properly. The reason is explained to be the modelers usually use these as visualization models. They are meant to look good, not work properly in simulations, it is also common with normals to be aligned along the wrong axis. "Things like these are our job to clean up and fix". Dnv often demands solid CAD models from their customers as these are built to be functional from the start. A model is considered best when it is as instantiated as possible. Meaning models are cloned as much as possible if possible, for example nuts and bolts or materials, so that changing one changes all clones. The artist explained that he has gathered a big personal scripting library to automate several of these tasks. The benefit of using real time rendering engines as Unreal Engine would be if it was integrated in 3ds Max. An example would be if the viewport of 3ds Max was directly switched to Unreal Engines. Would be beneficial if the graphics of real time rendering where present when iterating ideas with customers. "It would be very good for meetings with customers and also some for me if you could show the scene in real time rendering" the artist explained. When talking about the experience of Unreal Engines Datsmith the artist explained, "When using Datasmith for Unreal its good that you can send [export] the whole scene at the same time. Its also nice to not have to UV map the model yourself." But he also explained "... When using still rendering you get away with more "cheating". There are a lot of shortcuts you can take in still rendering that is not yet possible with real time rendering". The VR and AR technology was describes a good tools for visualizing design and improve understanding to customers.

Interviewee 2 is an Unreal Engine Developer and Maya cg artist and has used these software as well as Unity since 2008. He was teaching Unity to students but switched to Unreal Engine because of the blueprint system. The usual workflow as an Unreal Engine developer is getting a model from the customer that has to be cleaned. Cleaning is normally done with same process mentioned during the previous interview with changing of normals, fixing surfaces etc. This time the process is different as the model is supposed to be imported into Unreal Engine. UV mapping is very important this time while the material and custom mapping is skipped. The cleaning is done in Maya as there are certain features that maya can do that 3ds Max cannot when exporting. In unreal engine a blueprint is created to be able to control the model with blueprint scripts. "This way a model of a door can be opened or a fan can be made to spin when a certain button i presses for example" the developer explained. The customer knowledge is explained to be relatively low and the request is often to "make VR or AR" and a list of demands without knowing what they want to do with that. Other times the customers know exactly what it is they want and is happy to let the artist to the creative part. Often it is important to make a minimum viable product first and refine the scene afterwards. The team structure is often the Unreal Engine developer assisted by a cg artist that performs the clean up of the model. The hardest part in terms of the pipeline is the lightning and product visualization, as the developer explained "... to make the product pop and make it appealing". Interactive visualizations are the most appealing to the developer and seems to be the most appreciated by the customers. The developer believes that programming of interactive applications is going to be more and more important. The visualization and interactivity is converging. "A technical artist in Unreal is going to be more worth than a cg artist working in Unreal", referring to the cg artists ability to handle the visualization very good but no programming while a technical artist is not as good at visualization but can do programming. He believes that interactivity and user design is going to be key in future visualizations. When asked about the biggest time wasters in his work the developer pointed towards mesh clean up, normal switching, UV mapping and filling holes in the surfaces of the models. "Unreal cant do this currently, it can generate lightmap UVs but not normal UVs. I am now working on importing [an Assembly]. And one of the models had the UVs imported but it did now generate UV maps. And when it did they were all overlapping. so had to bring into Maya, generate second sets of Uvs. Stuff like that is taking time. One thing i would really like in unreal is face selection". An ideal scenario would for the developer would be Unreal Engine creating the UV mapping for him and the texel size to always be the same. When the developer is working with CAD data there is normally three different software involved. First the CAD data has to be prepared with connecting surfaces, renaming parts and putting them in a proper hierarchical tree structure. Then import the model into 3ds Max where all the optimization is done and finally import it into Unreal Engine.

The dream scenario for the developer would be to skip the 3d modelling software altogether and directly importing the models into Unreal. When asked about the possibilities of VR the developer is excited. Even though the technology is still young he thinks it is working today. "Visual inspection [of products] in VR is so much more useful than on a computer screen. I saw details on a whole other level. VR is so immersive that way. For one project i was looking more at materials in VR than anything else".He believes that when the possibilities for multi user experiences gets easier the technology will gain even more attraction. That will allow people will be able to work from anywhere in the world. It will transform business and business models, techniques and workflows alike. "Its just an interactive 3d chat scenario. You have all the tools available. One thing that is always lacking in projects is the connection with people. You want to be with the people and near the people you are working with". He also explains a possible future where workers are plugged into a virtual workspace with haptic suits and gesture commanded software. The future workspace is described more as "workshop" than a visualization software. He describes Unreal Engine and Unitys' relationship as Unreal catering more to the non-gameing community. Though Unity has a larger community with more third party plugins and scripts available.

### 4.1.4 Virtual Reality trends

The use for VR within the engineering industry is growing [74, 75, 76]. As VR technology have reached a stage of maturity where the technology is no longer a hype, as described in 2.5, business are exploring actual implementations. The emergence of gamifications "understanding and learning through interactive activities" has inspired design engineers to use VR in the product development process [77, 78, 79, 80]. The automotive industry has specifically seen the positive effects of VR on understanding and communication of highly complex assemblies [81, 82]. Automotive company CEVT has a department dedicated to VR visualization[83]. In a short interview with the virtual reality leader at the VR department at CEVT, he said "Management is beginning to understand the greater understanding VR and AR can have in the product development process. If we buy in VR equipment for a one time cost of 100000 that has the potential to save millions of kronor on virtual prototyping, why would we not take that chance?". With the ever increasing processing power of computers coupled with the gaming industries pushing of the limits with graphics VR is constantly improving. This has led to applications with collaboration features over VR such as Nvidia Holodeck [84], see 4.18. Nvidia Holodeck is a showcase of the available technology with features such as in-application drawing boards, measuring tools and cutting planes and cutting spheres. There is also possibility to manipulate parts such as moving doors and exploding views.

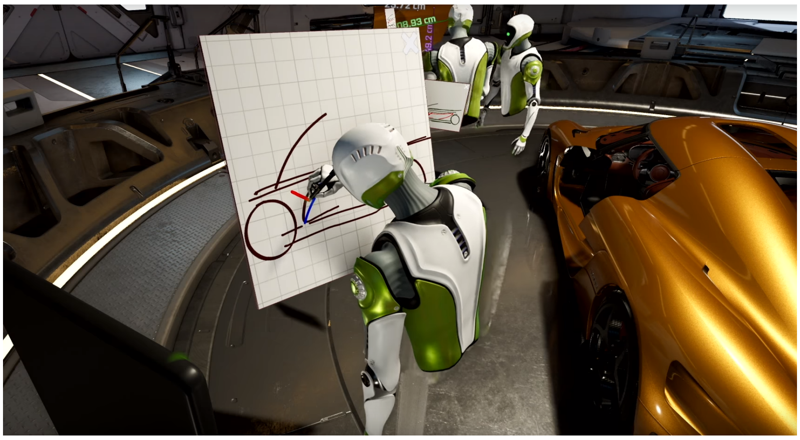


**Figure 4.18:** Collaborative VR in Nvidia Holodeck

Facebook is also investing in VR technology with its Facebook spaces [85]. Facebook spaces is supposed to work with other VR users as well as seamlessly with people not using VR hardware through in-application screens, see 4.19.

**Figure 4.19:** Collaborative VR in Facebook Spaces

The gaming industry is currently what is pushing the technology forward with games like Robo Recall [86] that has been known for its optimized graphics for VR. The use of VR as a training and understanding tool is as mentioned in 2.6.4 used in both military and healthcare simulations. The manufacturing industry can use VR applications for ergonomic studies by using trackers to simulate nut-runners and thereby skip the building of a test rig. VR is getting more mainstream and across the world VR experiences are built up as theme parks [87, 88, 89]. The mainstream:ing of VR means that the technology is getting cheaper and more resources provides the possibility to further enhance the technology.

# 4.2 Design of Experiment

## 4.2.1 Initial Experiment

All work and tests was performed on a computer with Windows 7 Enterprise, Intel core i7-5930k 3.50Ghz CPU, 32 gb RAM and a Geforce GTX 970 4MB GDDR5 graphics card. The initial experiment was all about learning the software, the settings and explore the possibilities with different file types. As the research showed Unreal Studio to be a more promising tool for engineering and product development purposes it was chosen as the real time rendering engine. PiXYZ was chosen as this would test the possibilities that Unity will offer when they become partners. Two licenses for PiXYZ was acquired one for the Studio editor and one for Review. The data was optimized and prepared in PiXYZ and then imported as FBX files into Unreal Studio. The first thing done was searching for a proper model. Through Wingquist laboratories [90] a car model was aquired in .wrl format. Unreal Studio could not import wrl files directly but PiXYZ could. So the file was imported into PiXYZ Stuio to be prepared and optimized, as these options existed in Pixyz. The normals was aligned along the wrong axis and was the first thing that had to be changed, see 4.20.

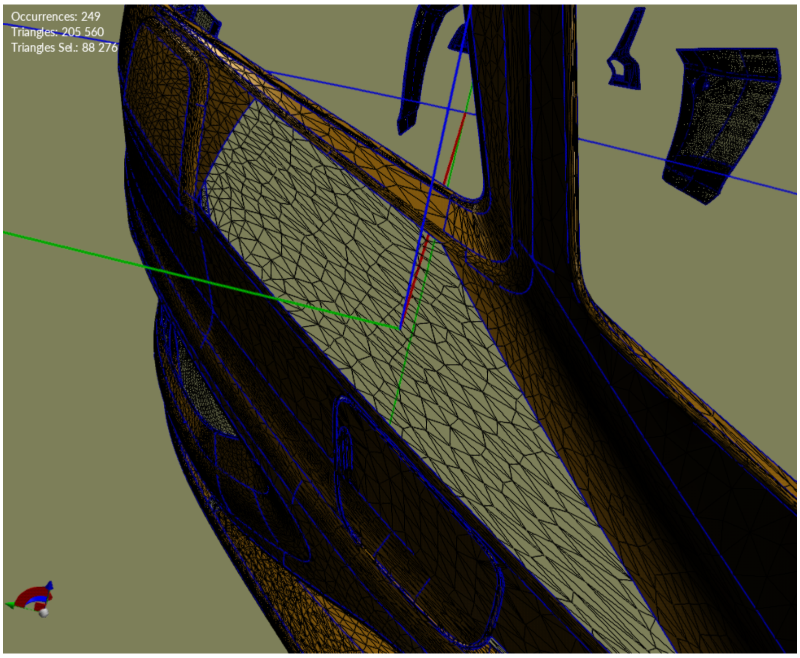



**Figure 4.20:** The triangles without colour are having their normals facing the wrong way

Tests were made with the built in tool for normal alignment in Pixyz, meaning executing an algorithm making the normals face the same directions as their neighbours and then another tool for flipping all the normals of a surface if these aligned the wrong way. The result was surfaces still with holes in them where normals stayed the wrong direction. The model was then tessellated to generate UV maps. The model was then exported as a FBX file to be further tested in Unreal. In Unreal the normals was first recalculated to see if Unreal would do something different which it did not. A material was then created in Unreal material editor that was set to use double sided lightning. In theory this should make the normal alignment irrelevant which is did, but the holes was now rendered on the screen with a different colour then the rest of the surface. When trying to build the scene and bake the lightning the surfaces turn black indicating missing UV maps. When checking for the UV maps in Unreal they are indeed missing. Without UV maps the lightning have no information to build the light on so the surface are rendered black. When doing the procedure through Pixyz again to see if there was some stage missed and doing the procedure again the UVs are still missing. Tests were also made with exporting from Pixyz as different fbx versions to see if it had to do with old or new versions but still no difference. Also tested identifying pathes, as in generating Nurbs on the surfaces, and then using a tool to repair the mesh. Repairing the mesh means generating new faces in incomplete surfaces. In this case the repair tool destroyed the model making it jagged and faulty. To test if the UV generation tool in Pixyz is not working with the current model a part of the unedited model is imported into 3ds Max where a UV map is generated. That part was then exported as an fbx again and imported into Unreal engine. It now worked indicating that the UV generation tool of Pixyz is not working with the current model.

When the test of Unreal Studio datasmith were to be made a new model was needed with a supported file type that fulfilled the demands set up. The hope was that the model would stress test as much aspects of Unreal and datasmith as possible. stress testing would mean having a lot of geometries. An excavator model from Catia V5 of file type .catpart and .cadproduct was chosen. The model consisted of 9406 parts which was considered a satisfactory amount that would stress test Unreal Studio. The first import into Unreal was made with just the boom, arm and bucket of the excavator, see 4.21, with standard values as import options, 2 cm in chord length and 15 degrees as max normal angle.

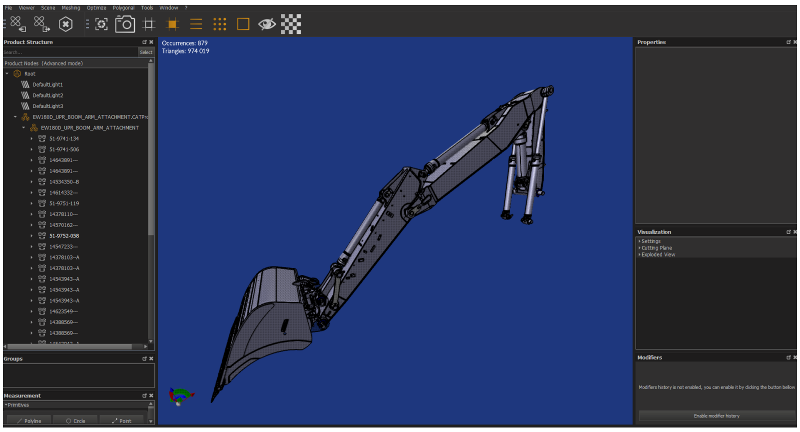


**Figure 4.21:** The model imported into PiXYZ

The import was successful after working 10 minutes, though the editor froze during the whole import. The important factor to be able to apply realistic looking materials is how well the UV mapping is performed. As mentioned is this usually performed in software like Maya or 3ds Max by hand and can take a long time on big models. With Unreal Studio being able to do this automatically it is an interesting feature to test. The model imported with proper UV maps, see 4.22. When applying materials to the geometry it refelected the light in a good way when buidling the scene.

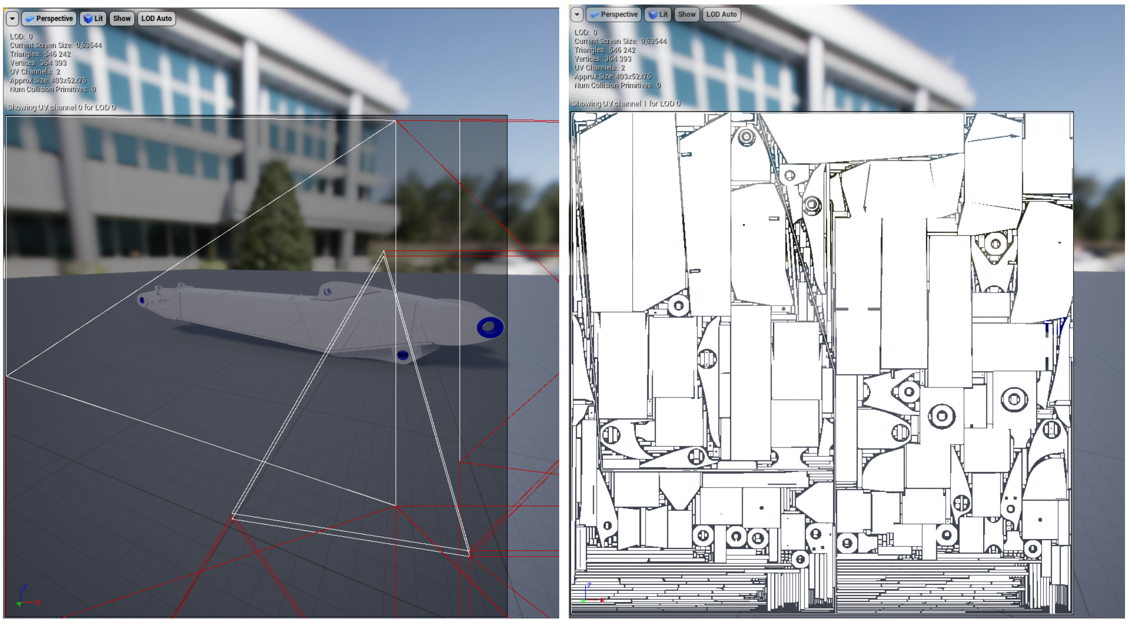


**Figure 4.22:** The model imported into Unreal with correct UV mapping

The model was also able to be interacted with blueprints programming and all parts were placed as their own geometries and put into a correct hierarchical tree structure, see 4.23.

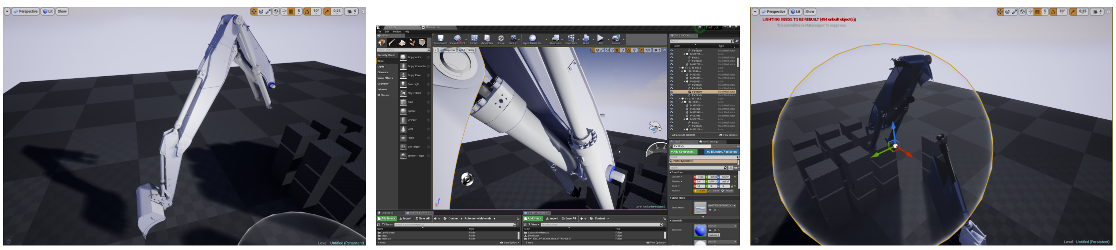


**Figure 4.23:** The model imported into Unreal to the left, with blueprint interaction and material applied and baked lightning

When looking at the optimization graph the green bar to the right and the maroon bar in the middle is the most processing heavy. The green bar is the post processing where there are slightly higher computation time for anti-aliasing. The maroon bar is for the basepass who is responsible for drawing objects on the screen.

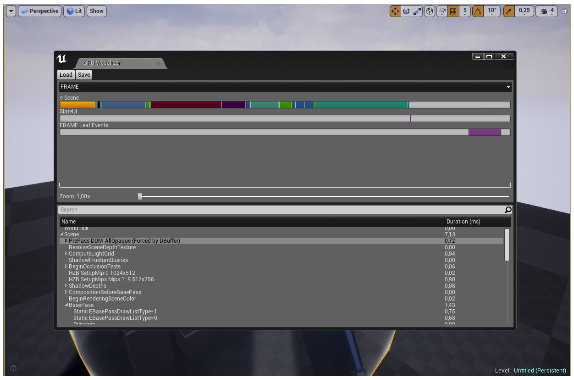



**Figure 4.24:** The GPU profiler

### 4.2.2 Main Experiment

The larger 9400 part assembly was more troublesome and constantly crashed Unreal Studio when directly importing raw CAD files, but worked when using 3ds Max to first export as a .udatasmith file. The import of the raw CAd data was made with different chord lengths, 0.1 - 10 cm and normal angles differing from 5-30 degrees, and also testing with and without importing materials and textures, in different combinations. Import was made with both catparts as well as with catproducts to see of the assemblies was the problem. Tests were also made with auto-save off to see if this affected the import somehow. This gave an improvement in that Unreal worked longer before crashing. When importing one part at the time Unreal succeeds in importing the data but it is impossible to say through performed tests what part is making Unreal crash or if it is the amount of parts that is the problem. When importing one catproduct file into Pixyz it takes 2 hours but succeeds. The model is only in Nurbs. When trying to tessellate this model it also fails. When opening the model in Max that had been saved earlier in 3ds Max and exporting this scene as a .udatasmith file and importing this into Unreal it works without problem with UV maps and materials applied. Indicating that the datasmith import works from Max to Unreal.

# 5
# Discussion

This report has explored the use of game engines to visualize CAD data and explored the possibilities with this. The test was performed by setting up demands for a fitting model that involved assembly size, file type and part amount. The test expllored not only Epics Datasmith but also PiXYZ that Unity now have partnered with to optimize CAD data. Datasmith and Unreal Studio was both in beta during these tests which means they can have bugs. There were three new versions during this thesis alone so Unreal Studio is rapidly improving and a new thesis exploring coming features would be interesting. The test did not fully test every feature of Unreal and PiXYZ but only features relevant for this thesis. Unreal Studio are so far focusing on getting their CAD converter component fully developed while PiXYZ is a full CAD optimizing and preparation suite. The test was performed on a Catia model of 9400 parts of an excavator model. The tests showed Unreal Studio working and a sub-assembly was fully imported, tessellated and UV wrapped. Materials could be applied and by using blueprints a sphere was programmed that made the model see-through. This proved that the CAD importer could make models that are fully interactive. Though the imported model had 420 parts and not the 9400 parts of the full assembly that where wished for. The full assembly did not work when importing and the reason could be two things. Unreal Studio is yet not capable of importing assemblies of such size, though where the limit is is not answered. Or the model is too old to be converted in a correct way. The model was according to part information latest changed in 2008, nearly 10 years ago from the time of writing this text. More than enough for Catia files to have undergone version changes that could brake the tessellation algorithms. The failure could also depend on the computer used. The computer was running windows 7 at the time of testing, though how this would affect is not researched. The hardware, 32gb ram, 970 gpu and i7 cpu could also be to weak. Though the specifications is well above what Epic Games recommends for the original Unreal Engine. These recommended specifications was set up for game development and not CAD conversion though, so they could be discussed if they should be higher as the CAD conversion during this test utilized 99% of the 32gb RAM during the conversion process. For future work on this subject the error of hardware and operating system should be eliminated. Using a more powerful computer with updated operating system. The CAD software version error should also be eliminated by using a fresh and new model or at least a model saved in a new version of the CAD software. The biggest challenge during the test of exporting and importing CAD data between software was the lack of earlier work to be inspired by. Usage of game engines for engineering purposes is new and innovative, using direct import of CAD data that automatically tessellates and UV wraps this data is even more so. Though the thought of using game mechanics in gamification and serious gaming has been around some years the implementation of those ideas has been more towards the actual learning and motivational part, while this thesis has been more on the actual export. The result is a lot of work that has to be guessed. Information like Chord length, max triangle length or how long an import should take is hard to come by. The lack of knowledge of the user in existing software is also a potential source of error. As some problems that arose could have been solved quicker if knowledge had been there, though when can that argument not be made for any mistake. UV mapping was such a thing that took unnecessary long time to come to a conclusion about. As engineering and computer graphics are different some concepts where unfamiliar and could probably have been solved quicker. That said the possibilities of this technology is exciting for the product development and marketing process. VRED is today used in ways similar to how the game engines would be used. Including VR development, interactive features such as changing tire models and colour with motion controllers or analyzing car features such as touch screens directly in the application. But VRED is focused on the automotive industry and their needs. It is used within both Volvo and CEVT to make great visualizations of their cars. But what game engines further brings is the same features, developed further and more optimized as the gaming community is developing rapidly and the ability to visualize more than just automotive models. The optimization features of PiXYZ is also an area that deserves further study. As the interviews showed, cleaning and preparation of models took a long time and felt unfulfilling for the developers and artist. PiXYZ have the potential to automate a lot of the tedious tasks described in cleaning of a model. And the question whu UV maps was not generated when using the software is still unanswered as the developers say that it should work, and for them to further help they would need the model to test. As VR and AR needs are expanding so will the needs, as more and more possibilities are opening up and explored. The real time engines, both game engines and software such as VRED, will not replace CAD or visualization software, yet, but is currently a strong complement for understanding and exploration of features that would be expensive to build and develop in real life. VR has potential to enhance the visualization of products and thereby expand on the understanding. The possibility of an exploding view of a model with all parts present in VR and realistic graphics would completely eliminate the need to produce cars just to pick them apart. Another area where VR can help is in variation simulation of non-nominal CAD assemblies. An area not covered in this report as there were not enough time. But to see split lines and details that variations in manufacturing creates without producing cars can of course save many resources for the environment and economically.

# 6 Conclusion

This rapport have investigated the use of game engines within VR in the product development process. It first showed the possibly positive influences VR can have in the product development process. If used right it can enhance understanding and communication by adding a dimension toe data review. But it is important that the VR experience happens as seamlessly as possible to not interrupt the current workflow, it should be used as a complement to the current process not replace it, yet. Business must learn to use the VR technology now or risk falling behind the ones that do. It has shown that Unity and Unreal Engine are the two big game engines that most 3d-visualizing business use or are experimenting with using today. They both produce near photo realistic real time rendering and both posses features to make the scenes interactive, Unity with the C-sharp programming language and third party plugins for visual programming and Unreal Engine with its blueprints system or C++ for low level programming. Unreal Engine developed Unreal Studio to answer a growing need from the engineering sector for VR and CAD visualization services. Unity partnered with PiXYZ software to do the same thing. PiXYZ are able to optimize and prepare CAD data in a way that both Unreal Studio and Unity can utilize. PiXYZ and Unreal Studio can both tessellate the CAD data for use in VR environments. They also produce ready to use UV maps and normal maps enabling material and textures to be applied on these models. The models where also easy to interact with when tested with Unreal Blueprints. Unreal Studio also comes with templates where VR motion controllers can grab the geometries from without need for programming, though more interactivity can be added though programming.

The Tests made are an introduction to the use of game engines in the engineering sector and for future work its recommended to export assemblies over 10000 parts as tests with this size was unsuccessful in this rapport thanks to version incompatibility. It would also be interesting to see how computer hardware affects performance and conversion times.

# Bibliography


[1] Alten Group *Alten Homepage* [Online] Available: http://www.Alten.se 2018

[2] Hilfert, T. & König, M. *Low-cost virtual reality environment for engineering and construction 2016*, Vis. in Eng. (2016) 4: 2. [Online] Available: https://doi.org/10.1186/s40327-015-0031-5

[3] Orland, Kyle. *Epic looks outside of gaming for new uses of Unreal Engine: Powerful real-time 3D is revolutionizing everything from film to architecture* New York : n/a. New York: Condé Nast Publications, Inc. (Mar 16, 2016) ArsTechnica.com

[4] Luis Cataldi *Making VR With Unreal Engine*, Proceeding SIGGRAPH '16 ACM SIGGRAPH 2016 Studio Article No. 10 [Online] Available: http://delivery.acm.org.proxy.lib.chalmers.se/10.1145/2980000/2970275/a10-cataldi.pdf?ip=129.16.69.49&id=2970275&acc=ACTIVE%20SERVICE&key=74F7687761D7AE37%2E3C5D6C4574200C81%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1517575063_2f932e2e75cf78d5485d599f6dae0e4e#URLTOKEN#

[5] Epic Games *What is Unreal Engine 4* [Online] Available: https://www.unrealengine.com/en-US/what-is-unreal-engine-4

[6] UnrealDocs *Documentation for Realistic Rendering* [Online] Available: https://docs.unrealengine.com/latest/INT/Resources/Showcases/RealisticRendering/

[7] PiXYZ Software *PiXYZ products* [Online] Available: https://www.pixyz-software.com/

[8] Crecente, Brian *McLaren now uses Epic's Unreal Engine to help design its high-end cars* [Online] Available: https://www.polygon.com/2016/3/16/11246586/mclaren-unreal-engine-design

[9] Lv, Zhihan et al. *Game On, Science - How Video Game Technology May Help Biologists Tackle Visualization Challenges* Available: PLoS One; San Francisco Vol. 8, Iss. 3, (Mar 2013): e57990.

[10] Pimentel, Ken *Datasmith and Unreal Engine Drive Real-time Architectural Design at Neoscape 2018* [Online] Available: https://www.unrealengine.com/en-US/enterprise/blog/datasmith-and-unreal-engine-drive-real-time-architectural-design-at-neoscape

[11] Daniel Berndtsson, Martin Dehmer Östebo, Fredrik Eliasson, Hanna Römer, Alexander Sandberg *Augmenting Board Games using Virtual Reality* Available: http://publications.lib.chalmers.se/records/fulltext/251036/251036.pdf 2017

[12] Gregory, Jason *Game Engine Architecture* 2:n edition Boca Raton, Florida, 2015

[13] Peddie, Jon *The History of Visual Magic in Computers* London, England, 2013

[14] Joey de Vries, CC BY 4.0 license as published by Creative Commons 2017 *Parallax Mapping* Available: `https://learnopengl.com/Advanced-Lighting/Parallax-Mapping`

[15] Hirz, Mario, Dietrich, Wilhelm, Gfrerrer, Anton, Lang, Johann *Integrated Computer-Aided Design in Automotive Development* Berlin, Germany, 2013

[16] Trip, Davi, CC BY 4.0 license as published by Creative Commons 2017 *Shading Models* Available: `https://upload.wikimedia.org/wikipedia/commons/0/0f/`

[17] Lindqvist, Lars *Lecture powwrpoints in course Advanced CAD given at Chalmers University of Technology* Gothenburg, Sweden, 2015

[18] Mikael Johansson *From BIM To VR -The design and development of BIMXplorer* Gothenburg, Sweden, 2016

[19] Computer Hope *Computer Hope Rasterize* Available: `https://www.computerhope.com/jargon/r/rasterize.htm` 2017

[20] Stackoverflow.com Forum thread *Line Rasterization / 4-connected Bresenham* Available: `https://stackoverflow.com/questions/13542925/line-rasterization-4-connected-bresenham` November 2012

[21] Kostas Anagnostou *Interplay of Light Blog; HOW UNREAL RENDERS A FRAME PART 3* Available: `https://interplayoflight.wordpress.com/2017/10/25/how-unreal-renders-a-frame-part-3/` 2017

[22] Marco Fratarcangeli, Valentina Tibaldo, Fabio Pellacini *Journal article; Vivace: a Practical Gauss-Seidel Method for Stable Soft Body Dynamics* Available: `http://www.cse.chalmers.se/~marcof/` 2016

[23] Autodesk Knowledge Network *Direct Polygon Manipulation* Available: `https://knowledge.autodesk.com` 2018

[24] Autodesk, Inc *Autodesk Maya* Available: `https://www.autodesk.se/products/maya/overview` 2018

[25] Autodesk, Inc *Autodesk 3ds Max* Available: `https://www.autodesk.se/products/3ds-max/overview` 2018

[26] Ernie Tretkoff *This Month in Physics History October 1958: Physicist Invents First Video Game* APS News, October 2008 (Volume 17, Number 9)

[27] Steve Theodore *Why Have Video Game Budgets Skyrocketed In Recent Years?* Available: `https://www.forbes.com/sites/quora/2016/10/31/why-have-video-game-budgets-skyrocketed-in-recent-years/#ecc63f33ea57` OCT 31, 2016

[28] Robert Ciesla *Mostly Codeless Game Development; New School Game Engines* Helsinki, Finland, 2017

[29] Mathworks *Mathworks Simulink* Available: `https://www.mathworks.com/products/simulink.html` 2018

[30] Epic Games *Unreal Engine Blueprints* Available: `https://docs.unrealengine.com/en-us/Engine/Blueprints`2018

[31] Unity Technologies *Unity Third party plugins for Visual Scripting* Available: `https://assetstore.unity.com/categories/tools/visual-scripting` 2018

[32] Lifelong Kindergarten Group at MIT Media Lab *Scrath programming language* Available: `https://scratch.mit.edu/` 2018

[33] ASTD Research *Playing to Win: Gamification and Serious Games in Organizational Learning* 2014

[34] Trevir Nath *Investing in Video Games: This Industry Pulls In More Revenue Than Movies, Music* Available: `https://www.nasdaq.com/article/investing-in-video-games-this-industry-pulls-in-more-revenue-than-movies-music` Nasdaq, 2016

[35] JEFF GRUBB, VentureBeat *Gaming advocacy group: The average gamer is 31, and most play on a console* Available: `https://venturebeat.com/2014/04/29/gaming-advocacy-group-the-average-gamer-is-31-and-most-play-on-a-console/` APRIL 29 2014

[36] Statista *Average time spent gaming weekly in Great Britain as of June 2014, by age (in hours)* Available: `https://www.statista.com/statistics/323943/average-time-spent-gaming-weekly-uk/` 2014

[37] Karl T. Ulrich, Steven D. Eppinger *Product design and Development* 5th edition, Avenue of the Americas, New York, 2012

[38] Virtual Reality Society *The History of Virtual Reality* Available: `https://www.vrs.org.uk/virtual-reality/history.html` 2017

[39] Luke Dormehl *8 virtual reality milestones that took it from sci-fi to your living room* Available: `https://www.digitaltrends.com/cool-tech/history-of-virtual-reality/` 2017

[40] Glyn Lawson Davide Salanitri Brian Waterfield *Applied Ergonomics Volume 53, Part B, March 2016, Pages 323-330, Future directions for the development of virtual reality within an automotive manufacturer* 2015

[41] Oculus VR, LLC. *Oculus Go homepage* Available: `https://www.oculus.com/go/` 2018

[42] HTC Corporation *HTC Vive Homepage* Available: `https://www.vive.com/us/product/vive-pro/` 2018

[43] Varjo Technologies *Varjo Technologies homepage* Available: `https://varjo.com/` 2018

[44] Google *Google cardboard homepage* Available: `https://vr.google.com/cardboard/` 2018

[45] Siddhesh Manjrekar et. al. *UKSim-AMSS 16th International Conference on Computer Modelling and Simulation CAVE: An Emerging Immersive Technology - A Review* 2014

[46] HENRIK JACOBSSON, JON-ANDERS SYRÉN *Requirements and needs for 3D-visualizations (a 3D-viewing platform) An in-depth study of instances of research and development at Scania CV AB* Gothenburg, Sweden 2013

[47] Liker, Jeffrey K; Ross, Karyn *The Toyota way to service excellence: lean transformation in service organizations* 1st edition McGraw-Hill Education 2017

[48] D. Robertson ; T.J. Allen *IEEE Transactions on Engineering Management; CAD system use and engineering performance* Volume: 40, Issue: 3, Aug 1993

[49] Dickson, Paul et. al. *An Experience-based Comparison of Unity and Unreal for a Stand-alone 3D Game Development Course* 2017

[50] Anderson, Eike Falk et. al. *5th International Conference on Games and Virtual Worlds for Serious Applications; Choosing the Infrastructure for Entertainment*

*and Serious Computer Games a Whiteroom Benchmark for Game Engine Selection* 2013

[51] Kosmadoudi, Zoe et. al. *Engineering design using game-enhanced CAD: The potential to augment the user experience with game elements* 03/2013, Volume 45, Issue 3

[52] Valve Coorporation *Valve Steam Workshop* Available: `https://steamcommunity.com/workshop/?l=swedish` 2018

[53] Dr. Alexander Uskov, Bhuvana Sekar *Serious Games, Gamification and Game Engines to Support Framework Activities in Engineering: Case Studies, Analysis, Classifications and Outcomes* 2014

[54] Georgi V. Georgiev et.al. *Augmenting Creative Design Thinking using Networks of Concepts* Los Angeles, CA, USA 2017

[55] Mark Barry *The Future of Manufacturing Design is Virtual Reality* Available: `http://www.aberdeenessentials.com/opspro-essentials/future-manufacturing-design-virtual-reality/` Dec 12, 2016

[56] JIMMY JOHANSSON, JOEL ERIKSSON *Digital 3D-modell och virtual reality som kommunikationsverktyg* Göteborg, Sverige 2015

[57] Magnus Carlsson, Tor Sonesson *Using Virtual Reality in an Automotive User Experience Development Process; An investigation of how to use head mounted display VR when evaluating digital interfaces* Gothenburg, Sweden 2017

[58] Available:

[59] Grand View Research *The 3d CAD software market analysis* Available: `https://www.grandviewresearch.com/industry-analysis/3d-cad-software-market` 2017

[60] Grand View Research *Visualization and 3d rendering Market Analysis* Available: `https://www.grandviewresearch.com/industry-analysis/visualization-3d-rendering-software-market` 2017

[61] Sik-Lanyi, Cecilia *Virtual reality healthcare system could be a potential future of health consultations, A Discussion of Virtual Reality As a New Tool for Training Healthcare Professionals* Budapest, Hungary, 2017

[62] Delacroix, Sylvie et. al. *Article in Frontiers in Public Health; A Discussion of Virtual Reality As a New Tool for Training Healthcare Professionals* Switzerland, 2018

[63] Akpan, Ikpe Justice; Shanker, Murali *The confirmed realities and myths about the benefits and costs of 3D visualization and virtual reality in discrete event modeling and simulation: A descriptive meta-analysis of evidence from research and practice* Kent State University, USA, 2017

[64] Noah Cole *Autodesk and CGarchitect Present First Autodesk Excellence in Infrastructure Awards* Available: `http://news.autodesk.com/2012-10-03-Autodesk-and-CGarchitect-Present-First-Autodesk-Excellence-in-Infra` 2012

[65] Twinmotion.com *2016 CGarchitect.com Architectural 3D Awards – Winners' Circle* Available: `https://twinmotion.abvent.com/en/2016-cgarchitect-com-architectural-3d-awards-winners-circle/` 2016

[66] Jeff Motle *2018 Architectural Visualization Rendering Engine Survey - RESULTS* Available: `http://www.cgarchitect.com/2018/02/2018-architectural-visualization-rendering-engine-survey` 2018

[67] Epic Games *Epic Games homepage* Available: `https://www.epicgames.com/site/en-US/about` 2018

[68] Unity Technologies *Unity Technologies Homepage* Available: `https://unity3d.com/` 2018

[69] Unity Technologies *Unity Asset store* Available: `https://assetstore.unity.com/` 2018

[70] Eike Falk Anderson et. al. *Choosing the Infrastructure for Entertainment and Serious Computer Games - a Whiteroom Benchmark for Game Engine Selection* Poole, UK, 2013

[71] Unity Technologies Documentation *Unity Project Templates* Available: `https://docs.unity3d.com/2018.1/Documentation/Manual/ProjectTemplates.html` 2018

[72] Autodesk Inc. *Autodesk VRED Homepage* Available: `https://www.autodesk.com/products/vred/overview` 2018

[73] Yuji Fujimura *Yuji Fujimura Portfolio* Available: `http://www.coroflot.com/YujiFujimura/portfolio` NewYork, NY, 2018

[74] Carlos Gonzalez *Does Virtual Reality Have a Future in Engineering?* Available: `http://www.machinedesign.com/blog/does-virtual-reality-have-future-engineering` 2016

[75] Pamela J. Waterman *Virtual Reality: A Powerful Engineering Tool* Available: `http://www.digitaleng.news/de/virtual-reality-powerful-engineering-tool/` 2014

[76] Laura Schina, Mariangela Lazoi, Roberto Lombardo, Angelo Corallo *Virtual Reality for Product Development in Manufacturing Industries* 2016

[77] Simon Jones *How Animech Designed a Custom Volkswagen VR Experience That Sells* Available: `https://www.unrealengine.com/en-US/blog/how-animech-designed-a-custom-volkswagen-vr-experience-that-sells` 2018

[78] Keef Sloan *BMW Brings Mixed Reality to Automotive Design with Unreal Engine* Available: `https://www.unrealengine.com/en-US/blog/bmw-brings-mixed-reality-to-automotive-design-with-unreal-engine` 2018

[79] Ken Pimentel *Theia Interactive's Harley Davidson AR Experience Showcases the Potential of Real-time* Available: `https://www.unrealengine.com/en-US/blog/theia-interactive-s-harley-davidson-ar-experience-showcases-the-potential-of-r` 2018

[80] Simon Jones *Unreal Engine Build: Paris'18 Showcases Top Talent for Real-time Design* Available: `https://www.unrealengine.com/en-US/blog/unreal-engine-build-paris-18-showcases-top-talent-for-real-time-design` 2018

[81] Erik Ridman *HiQ skapar VR-upplevelse åt Volvo Cars* Available: https://www.hiq.se/nyheter/2018/q1/hiq-skapar-vr-upplevelse-at-volvo-cars/ 2018

[82] FINWIRE *HiQ: Ny VR-lösning för Volvo* Available: https://www.di.se/nyheter/hiq-ny-vr-losning-for-volvo/ 23 february 2018

[83] CEVT Homepage *China Euro Vehicle Technology* Available: https://www.cevt.se/ 2018

[84] Sebastien Miglio *Real-Time Collaboration in VR is Revolutionizing Design* Available: https://www.unrealengine.com/en-US/blog/real-time-collaboration-in-vr-is-revolutionizing-design 2018

[85] Facebook *Facebook Spaces Homepage* Available: https://www.facebook.com/spaces 2017

[86] Steven Wong *Why 'Robo Recall' Is The Best Reason To Own An Oculus VR Setup* Available: http://www.alistdaily.com/digital/robo-recall-best-reason-oculus-vr-setup/ 2017

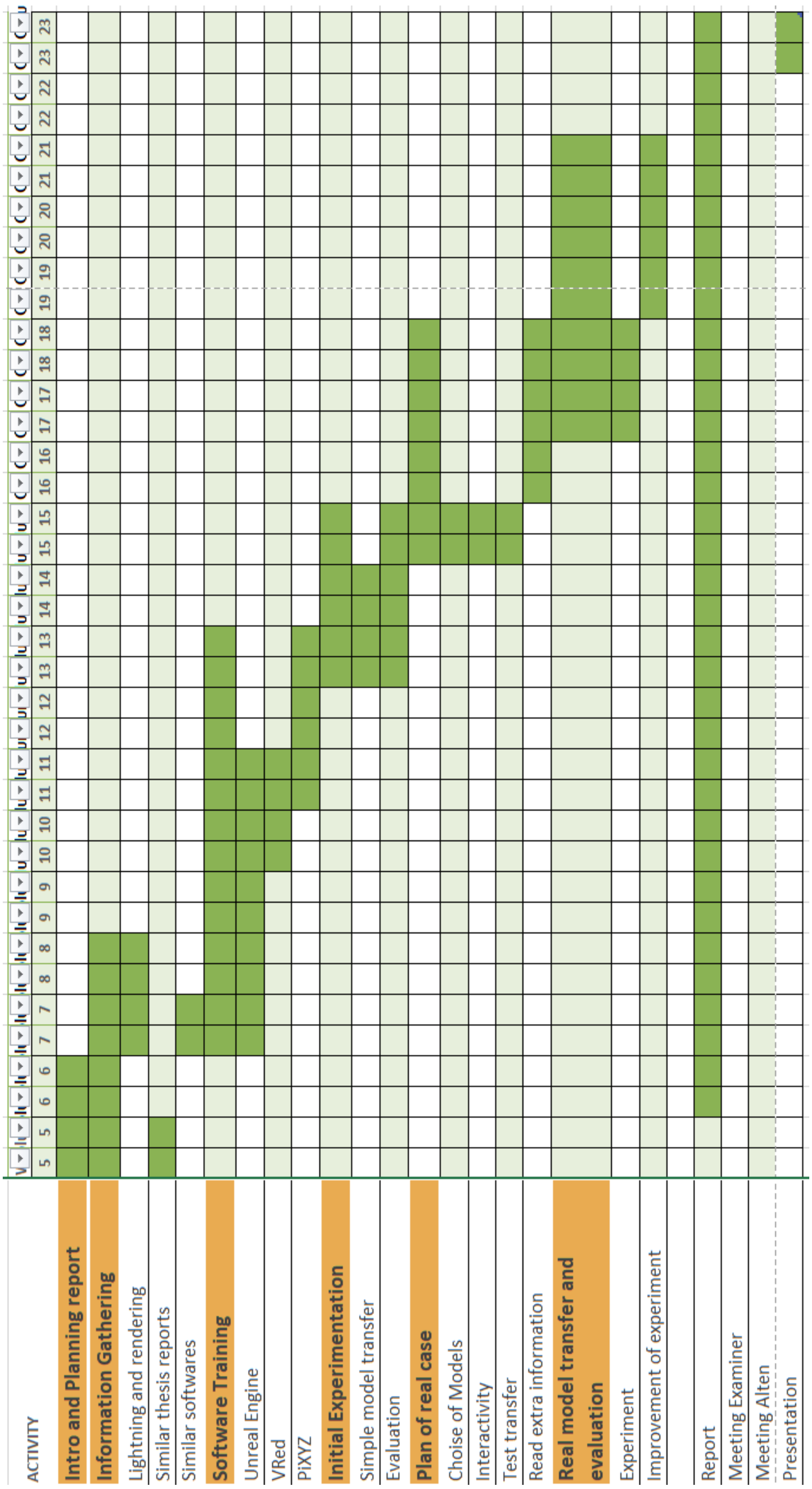
[87] Mission IX *Mission IX Homepage* Available: https://beta.missionix.com/ 2018

[88] Starbreeze Enterspace VR *Enterspace VR Homepage* Available: https://www.enterspacevr.com/sv/sweden/ 2018

[89] Jonas Leijonhufvud *Då öppnar Starbreeze sitt VR-center i Dubai* Available: https://digital.di.se/artikel/da-oppnar-starbreeze-sitt-vr-center-i-dubai 13 february 2018

[90] Wingquist Laboratory *Wingquist Laboratory* Available: http://www.chalmers.se/en/centres/wingquist/Pages/default.aspx 2018

# A
# Appendix 1


ACTIVITY
5 5 6 6 7 7 8 8 9 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 20 21 21 22 22 23 23
Intro and Planning report
Information Gathering
Lightning and rendering
Similar thesis reports
Similar softwares
Software Training
Unreal Engine
VRed
PiXYZ
Initial Experimentation
Simple model transfer
Evaluation
Plan of real case
Choise of Models
Interactivity
Test transfer
Read extra information
Real model transfer and evaluation
Experiment
Improvement of experiment
Report
Meeting Examiner
Meeting Alten
Presentation

[H]

# B

# Appendix 2

Jag kommer fråga lite frågor om ditt dagliga arbete i realtion till visualisering av produkter. Om du av vilken anledning som helst inte vill besvara någon fråga så är det bara att påpeka det så hoppar vi till nästa fråga. Jag kommer behöva spela in samtalet, och endast använda det för att analysera intervjun.

**Interview Details**

| | | | | | |
|---|---|---|---|---|---|
| Company Name: | : | Date: | Enter date | Time: | Enter time |
| Software of Choice | | Years in profession: | | | |
| Interviewee Title | | | | | |

**CAD transfer Questionnaire**

1. What softwares do you use in your daily work for visualization?
   a.
2. How do you use them?
   a.
3. What is the hardest part of your work?
   a.
4. What is the biggest waste of time, how do you deal with it?
   a.
5. Have you imported CAD data into a visualization platform and how did you do it? (BE MORE SPECIFIC?)
   a.
6. How often do you use it? How often would you like to use it?
   a.
7. What was the biggest obstacle? Why was this obstacle occurring?
   a.
8. Was there another way to visualize the data? Why would the data preferably been visualized this way?
   a.
9. What extra work or processes arose thanks to this workflow?
   a.
10. What would the dream scenario have been when visualizing data?
    a.
11. Have you used virtual reality or augmented reality in the visualization process? In what ways?
    a.
12. What way can Virtual reality or augmented reality help with in the visualization?
    a. **Understanding, connect the dots, communication**
    b.
13. Do you think interactivity can help in the design of a product? How?
    a. **Being able to move or interact with the scene, exploding views of products etc.**