TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MARKO LAITINEN

PRODUCTION OF VIRTUAL MODELS FOR CO-CREATIVE DESIGN IN CONSTRUCTION BUSINESS

Master of Science thesis

Examiner: prof. Kalle Kähkönen
Examiner and topic approved by the Faculty Council of the Faculty of Business and Built Environment
on 24th February 2017

ABSTRACT

MARKO LAITINEN: Production of Virtual Models for Co-Creative Design in Construction Business
Tampere University of Technology
Master of Science Thesis, 85 pages, 1 Appendix page
July 2017
Master's Degree Programme in Civil Engineering
Major: Structural Engineering
Examiner: Professor Kalle Kähkönen

Keywords: unity3d, game engine, virtual reality, augmented reality, mixed reality, virtual model, co-creation

Game engine based technologies are becoming a more common platform for immersive head mounted displays. Due to the experiences derived from immersive demonstrations, construction industry has adopted the game engines as platforms to produce virtual models for different uses. A virtual model created in game engine can serve as the base for immersive visualization or physic-based simulation of different natural phenomena. Both examples provide information that interests designers and clients in construction industry.

The research is divided into two parts. The first part is definition of a process for production of virtual models through empirical trials. The second part focuses how virtual models could be applied in co-creative design in construction business. The need for virtual models in co-creative design was researched by organizing themed interviews with different parties in construction business. The interviewees had either experience in co-creative design or modern technologies applied in construction and gaming industries. In addition, a case study for using a virtual model as a part of a facilitated workshop in co-creative design was done with help from University Properties of Finland Ltd.

As the result of this thesis, a process for production of virtual models from different data sources in construction project was defined. The process is presented in three parts: data preparation, data refinement and import to the game engine, Unity3D. Performance requirements defined in research outlines can be achieved through different optimization techniques explained in the thesis. To illustrate the versatility of virtual models, two case examples explain how virtual models can be evolved into immersive visualizations of the design data on different platforms. Implementation to co-creative design process can be done via an organized workshop or by using the virtual model as a platform to provoke conversation between different parties in a construction project.

PREFACE

This research is a master's thesis in Civil Engineering in Tampere University of Technology. The thesis was funded by A-Insinöörit Oy and it was examined by professor Kalle Kähkönen from the Faculty of Business and Built Environment in the TUT.

First of all, I would like extend my thanks to Risto for extensive guidance and reviews of my thesis during this whole process. His input on the thesis made me question my own theorems and strive to better results.

I want to thank A-Insinöörit Oy for providing an interesting topic for my thesis. Thanks to my thesis' examiner Kalle Kähkönen for providing time to ensure the academic quality of my research. Thank you Jussi and the University Properties of Finland for providing me such an inspirational environment to talk about co-creation and help to understand its different aspects.

I want to thank all my close friends during my journey at TUT, the folks at Tampereen Teekkarien PerinneSeura and the chairs of guilds 2015. You guys made my time at TUT unforgettable and gave me a lot of memories to cherish.

I am not sure how we ended up traveling in the middle of our research projects, but I do want to thank Anna for being my main academic support during my thesis and a close friend on the side. We will be hitting the cable again now that both our theses are done.

The one who I have had the most fruitful conversations about all aspects related to this thesis, and some beyond, is without a doubt Anssi. Thank you for challenging my views and providing insight from a whole different side of construction.

Finally, and the most of all, I want to thank my girlfriend Ella, for supporting me through the process. It's the small things during a long period of time that make the difference. She has been the driving force from time to time, reminding me there's more than the writing when the challenge has felt too big. Sometimes it's better to let it go, relax, and get back to it after a while.

Tampere, 30.6.2017

Marko Laitinen

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AR	Augmented Reality. Virtual objects are placed in the context of the real world and viewed through an AR device.
BIM	Building information model
CD	Collaborative design
CPU	Central processing unit
FOV	Field of view
GPU	Graphics processing unit
GUI	Graphical user interface
HVAC	Heat, ventilation, air conditioning
IFC	Industrial foundation classes. Standardized data exchange format in a construction project as defined in COBIM 2012.
LOD	Level of detail
MR	Mixed reality
NLS	National Land Survey of Finland
PD	Participatory design
SDK	Software development kit
TS	Total station
UVW mapping	A mathematical technique for coordinate mapping in computer graphics.
VR	Virtual Reality. The entire perceived real world is replaced with virtually created environment of 3D objects and the user is immersed in the model.
Virtual Model	Combination of design information produced in a construction project such as IFC, point clouds, native models from architectural design software and vector building information into a Unity3D project file.

1. INTRODUCTION

Call for co-creation and different types of visualizations has been increasing through the last decade in construction industry. Co-creation is a process that brings the designers, owners, and users of the building together in order to achieve a mutually acceptable result. Virtual reality (VR), augmented reality (AR), mixed reality (MR) and working in a Computer Assisted Virtual Environment (CAVE) are becoming increasingly common ways to present designs to the future users and stakeholders of the building in a co-creative design process. The different virtual models may serve as the basis for immersive VR visualization which takes users to a virtual construction site to examine the to-be-built construction. Within the virtual environment, users can explore, examine, feel, and base their decisions on more illustrative presentation of the project.

A construction project includes a variety of experts from different areas. In context of this thesis, a virtual model is a combination of design data produced in a construction project. This includes data such as different building information models (BIMs) and information derived from point clouds, combined into one platform. This research focuses on producing a virtual model in Unity3D which is a free-to-use game engine until a certain gross-revenue cap. Unity3D and its features are covered in Chapter 3.1.

1.1 Research background

A construction project consists of massive amount of data. Different designers (architectural, structural, HVAC, etc.) produce data for all aspects of a construction. The information must be presented to the client and other designers for the project to move forward. For a person who is not experienced in the construction industry may find it difficult to fully understand the impact and cost of design choices due to how the design data is normally presented in construction projects.

A combining trait for many engineering software is that their outcome is either a 2D drawing or a complex 3D model. To overcome this, different visualizations are rendered from the 3D models beforehand and presented to a client. Usually this is a very time consuming task since all images are rendered individually and the rendering process takes time. However, without a proper presentation of the design, the client may not be able to understand how the used materials impact the experience of living in the building or how a design plan has an impact on the cost efficiency of the building’s life cycle. A need for higher quality and a variety of visualizations can be seen.

A BIM has a large variety of meta-data incorporated to three-dimensional geometric object as further explained in Chapter 4. Virtual models can be used to create visualizations and simulations for different aspects of a construction project: city management, traffic, pedestrians, VR and AR views of a space and design, and such. Since BIMs incorporate a large amount of data in them, handling BIMs can be more complex for rendering software. Virtual models are not generally limited to software restrictions but rather to the abilities of the producer of the virtual model. Some custom functionalities, for example, in Unity3D require software development experience. However pre-customized functionality and easy-to-use add-on software are available in Unity Asset store for those who lack the mentioned software development capabilities.

The key motivator for this research is finding new ways to incorporate design data to co-creative design process along with defining how a virtual model could be produced efficiently. With the results of this thesis a new concept or fine-tuned concept of previously conjured co-creative design process may be created and implemented into a business model in the future.

1.2 Aim and objectives

The main goal of the research is to define a process for combining the data from different sources in a construction project and producing a virtual model based on the data sources described in Chapter 4. The produced virtual model is implemented to co-creative design process in construction projects. The virtual model must be able to visualize the design in different scenarios and it must be capable of simulating real-world physics. The virtual model can be utilized in different kind of visualizations, simulations, or analyses such as water flow analysis or sun light projection simulation for building facades, depending of the use case.

To achieve the research goals, the following research questions are to be answered:

RQ1: What is the most efficient way to construct a virtual model for Unity3D?

Different personnel in a construction project may use different software for their part of the design depending on their academic or professional background. An efficient way to construct the virtual model must be created for more streamlined and cost-efficient production process. The production of the model consists of gathering, preparing, modifying, and combining the data before importing it into a game engine for further customization. Production of a virtual model must be as automated as possible.

RQ2: What are the best practices to present a virtual model and design data to a client?

Determining what data is relevant for different scenarios is vital component of the production process of a virtual model. All irrelevant data should be either discarded from the virtual model to improve performance. Relevant data may vary with each client but the practices for different use scenarios remain the same. The data presentation platform should provide an intuitive user experience where the orienting for usage is done efficiently. The less time is used to teach the user of a virtual model's mechanics, the faster meaningful conversation of the virtual model's suitability and presented design solutions can be initiated.

RQ3: How can virtual models be applied to co-creative design in construction projects?

Co-creative design as a process brings the designers, owners, and users of the building together. The question that ought to be addressed is how co-creative design could utilize virtual models in a construction project and what kind of benefits it might reveal.

Upon production of a virtual model, example case studies focus on creating different kind of visualizations from the said virtual model, and how the virtual model based end-user products could be utilized as a part of co-creative design.

1.3 Research outline

Research outlining may be divided into two separate subsets: First subset, the production virtual model, focuses on researching different methods of producing virtual model from accumulated data and choosing the most efficient method or methods for the final process. Data production process is not included in the thesis. All the design data is presented as it is and is merely filtered or modified instead of modelled from scratch.

A virtual model is produced with a game engine. From different options, Unity3D was selected for the purposes of this research. The three main options available were Unity3D, Unreal Engine and WorldViz. Unity3D was selected due to its more implementation-ready content available in the Unity Asset Store (Unity Technologies, 2017). In addition, as a first-time developer, the Unity3D's graphical user interface (GUI) was found to be easier to use the Unreal Engine's GUI. The WorldViz was not selected due to its more demanding learning curve in the early phases of developing solutions for production of virtual models. For the process to be applied in construction projects, the initial experience with software must be as easily adopted by new users as possible.

A virtual model has different limitations and demands in complexity. Regardless of the use case, when presenting a virtual model, a sufficient performance is required for the immersion to be believable. For the performance to be acceptable with VR HMDs, a frame rate of 90 frames per second (FPS) is required. The mentioned FPS requirement is used as the target performance FPS for all virtual models used for visualization in this research. Depending on the hardware available, different levels of optimization must be done to achieve the targeted FPS threshold. However, the optimizations are case sensitive and depend on the provided content for a virtual model. Thus, empirical trial of the production of virtual models is presented in Chapter 5 with different methods for model optimization. Empirical trials of evolving a virtual model into end-product are presented in Chapters 6.1 and 6.2.

The second subset focuses on implementing the virtual model to co-creative design process. The need for virtual models in construction business in Finland will be studied by conducting qualitative interviews with the representatives of different parties in construction industry and by a case study of implementing a virtual model into a co-creative design process. The case study is conducted together with University Properties of Finland Ltd. and its specifications and results are covered in Chapter 6.3.

While there are multiple ways to create a virtual model, in the context of this thesis a single process for the model creation is documented. The process is properly presented in Chapter 5. The used software, methods and practices can vary a great deal due to multiple different software being available for different setups and production methods.

1.4 Research methods

In this chapter, research methods are defined for answering the research questions set in Chapter 1.3. The research methods can be selected based on the data needed to be gathered.

A research method dealing with numerical data is called the quantitative research method. Quantitative research is based on describing the researched subject and interpreting statistical and numerical data in larger quantities for detecting patterns and similarities in the data. Respectively, a research method dealing with different phenomena as the basis, is called the qualitative research method. The research is conducted by observing a phenomenon and combining data from previous studies into the findings. The conclusions are derived from results that are based on the gained data and previous studies in the same subject. (University of Jyväskylä, 2015)

Due the nature of this research, a qualitative research method was selected. The purpose of this research is to define a process for production of virtual models to be implemented into a co-creative design process by the designers in the future. This involves understanding different aspects of a construction project and how the current process works. By understanding the key problems in the processes, the virtual models can be applied to co-creative design to help solve these problems or provide more information to prevent the problems from occurring

The two subsets presented in Chapter 1.3 have different methods of gaining valid research data. RQ1 is part of first subset and the second subset consists of RQ2 and RQ3. As shown in Figure 1, the process for producing virtual models is created via empirical trials. The process of virtual model production consists of gathering relevant data, refining it for transferring to Unity3D and finally, importing the data to Unity3D. By testing how the data can be transformed to a Unity3D compliant data format and how can it be imported to Unity3D, a basis for answering to RQ1 is defined.

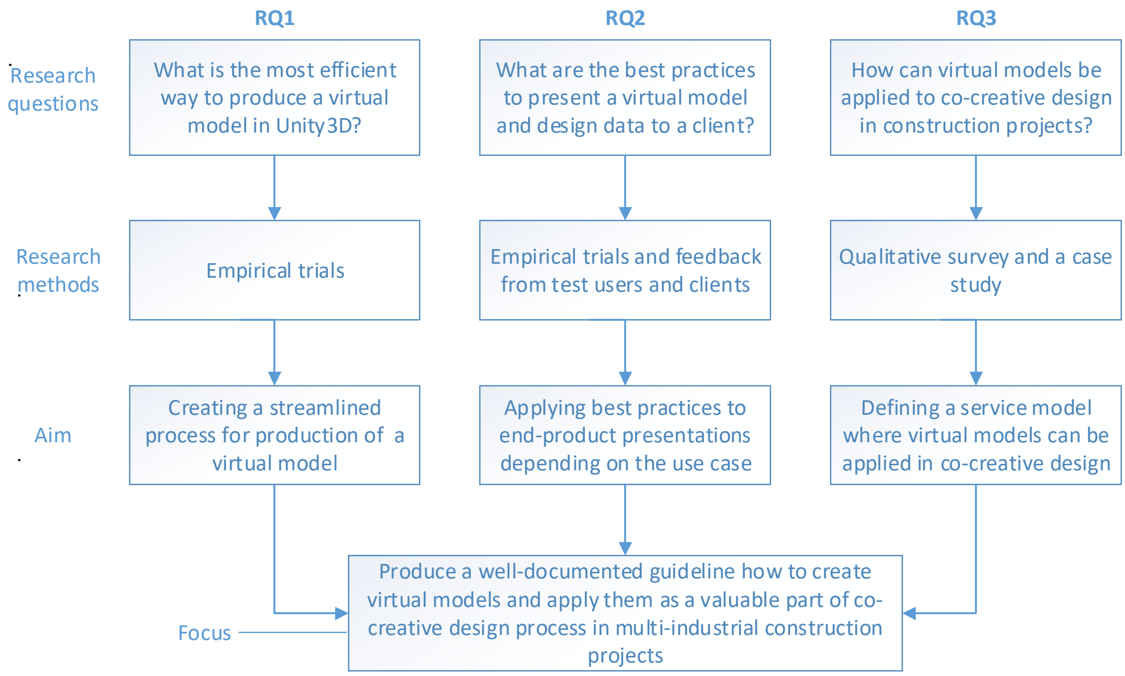

Figure 1: Flow chart describing how research questions will be answered

The production process of virtual models is created by conducting empirical trial of how different data can be converted to a Unity3D compliant data format. The trial is presented in Chapter 5. Additional empirical trials are performed for evolving a virtual model into an immersive visualization in Chapter 6. The empirical trials are executed in following fashion: The initial data is first imported to Unity3D with process defined in Chapter 5 without any optimization techniques covered in the same chapter. If the virtual model based end-product does not operate at set requirement level (90 FPS with VR), optimization techniques and further object exclusion are applied until the result meets the set requirements.

When the data can be presented in required performance framework, different virtual models can be applied to co-creative design. To answer the questions two and three, feedback is gathered from test users as virtual models are presented to them. For answering the RQ3, a case study of using a virtual model as part of a charrette is conducted. A charrette is a collaborative planning and negotiation process which engages different stakeholders and designers in producing different design solutions and implementing them into practice (Naaranoja, et al., 2015). In addition, qualitive interviews are conducted among professionals in different fields of expertise and backgrounds in construction business who work as different parties in construction projects. The purpose of the interviews is to discover how different parties of a construction project utilize co-creative design in their processes and how the parties view the importance of co-creation. Upon reviewing the answers and content of the interviews, virtual models' potential can be established.

Execution of the themed interviews started by mapping out personnel who could be interviewed. When estimating how many people should have been interviewed, it was considered that the suggested number for a qualitative research is approximately fifteen. In qualitative research, discretionary sampling can be applied to understand a phenomenon in a deeper fashion, or to get new theoretical views of a phenomenon. In such a scenario, a small group of interviewed personnel can be considered as sufficient to answer a research question. (Hirsijärvi & Hurme, 2011)

The purpose of interviews is to gain an understanding what kind technological solutions have been applied for co-creative design methods in Finland and what kind of practices different organizations use for co-creative design. For the need of the virtual models to be established, the problem areas of co-creative design process need to be discovered and considered when creating a new process. Due to the lack of proper documentation and existing research in this field, the number of interviewed personnel was decided to be kept at ten persons. This way, more time can be spent with different interviewees and a deeper understanding of the bigger picture can be gained.

The participants for the interviews were selected based on their background in construction business and their field of expertise in construction projects. Each of the interviewees had extensive experience in either co-creative design in practice or working with different technologies implemented in a co-creative design. Nine persons were asked to participate in the interviews, eight of them agreed to participate. Contacting of the interviewees was done via e-mails and telephone. Personnel participated in the interviews are presented in Table 1.

Table 1: Interviewed personnel

Interviewee	**Position**	**Main area of expertise for the study**
Jyri Tuori	VR Technology Manager Sweco Rakennetekniikka Oy	Technological
Marko Yli-Rantala	Construction Manager A-Insinöörit Rakennuttaminen Oy	Co-creative design
Olli Poutanen	Team Leader, Applied Games WSP Finland Oy	Technological

Suvi Nenonen	Specialist, Working and Learning Environments, University Properties of Finland Ltd.	Co-creative design
Jussi Savolainen	Managing Director Sumplia Workshop Oy	Co-creative design
Olli Niemi	Adjunct Professor University Properties of Finland Ltd.	Co-creative design
Jukka Vakula	Director, HMA Operations Technopolis Oyj	Co-creative design
Tero Vanhanen	Phenomena builder Fira Oy	Co-creative design

The interviews were conducted as face-to-face themed interviews. A predefined questionnaire served as basis for the interviews and was sent to the interviewees beforehand. The questionnaire was divided into three themes: the first part of the questionnaire, co-creative design in interviewee's organization, focused on how the organization has co-creation implemented in their own business model and how do they see co-creation as part of a construction project. Second part, co-creation practices and methods, surveyed how co-creation is perceived by organization's employees and affiliates and what kind of challenges are faced in the co-creative design process. The final part of the questionnaire, tools, and technologies for co-creative design, focused on how the interviewee's organization sees different technologies utilization possibilities for co-creative design. Via the third part, a spectrum is formed of what kind of technological solutions are already implemented for co-creative design to this date in the field of construction.

The interviews were documented and the results are used to examine the potential and possible need of virtual models in co-creative design process. The used questionnaire is presented in appendix A and the derived conclusions based on the interviews are presented in Chapter 8.1. As stated in "Conducting a research survey – the theory and practice of a themed survey" (Hirsijärvi & Hurme, 2011) the survey material can either be written as a manuscript of each interview, or alternatively the interviews are not written as manuscripts but conclusions can be drawn directly from the recordings of the surveys. Due to the nature and goal of the survey, the answers of individuals are not published in the thesis. Instead, the conclusions based on interviews are presented with an analysis of the potential need for virtual models in construction business.

2. RESEARCH FRAMEWORK

A research of related previous studies was conducted to gain understanding of different areas affecting this thesis' research. The main purpose of research framework is to map what has been studied so far and how it incorporates to this research. What can be applied to the processes created in this research and what needs to be discarded. Acknowledging the challenges, framework and different studies help to define a process which can be applied to a construction project.

2.1 Overview of related previous studies

Virtual and augmented realities or combined building information models are not new concepts in construction business. There are multiple studies that focus on the aspects of this study. However, none of the found research papers and articles had defined a process for combining building information model metadata or importing IFC files to Unity3D.

As the result of his doctorate thesis, Mikael Johanson (2016) created a new technical solution, BIMXplorer, for visualizing BIM content in a VR environment. Johanson had conducted comprehensive background research of how the datasets extracted from building information models are often too complex for a game engine to be directly used in 3D visualizations. Prior to his doctorate, Johanson had written scientific articles focusing on different aspects of working with BIM in VR; real-time rendering and visualization of BIM, how the hardware could have effect on VR visualizations through occlusion culling and hardware instancing and what kind of benefits immersive VR visualizations could have in a design process.

Johanson's doctorate thesis' results provided a solution for BIM visualization through efficient cells-and-portals culling system, efficient approach for integrating occlusion culling and hardware-accelerated geometry instancing and efficient single-pass stereo rendering technique based on hardware-accelerated geometry instancing. Even though Johanson's solution is working, it can be conducted through testing that it lacks the ability to add or modify the utilities provided by the software. BIMXplorer is remarkably fast in terms of processing IFC to real-time VR visualization, spending a few hours of developing a VR visualization with Unity3D is a more attractive way for a technology oriented person. The ability to fully customize the functionality of the software is a winning trait. While the BIMXplorer had only functionalities for measuring distance, and read IFC meta-data when BIMXplorer was released, Johanson has added new features after the initial launch, such as teleporting utility. Being able to teleport by using a pointer decreases the stress for user's neck when he does not need to gaze directly to where he wants to go. Instead he can use his peripheral vision to aim the teleporter beam.

As presented later in this thesis, currently there are multiple different platforms that provide different types of data for a virtual model. Tuomas Turppa researched how point clouds produced from laser scans could serve a purpose incorporated into game development (Turppa, 2014). Based on his research, laser scan material is too complicated for game development's purpose. The 3D polygon models generated from point clouds had to be heavily optimized in a 3D modelling software such as Autodesk 3ds Max. Another challenge rose upon conducting that the UVW mapping was insufficient for optimal implementation in game design.

UVW mapping is a mathematical technique for coordinate mapping in computer graphics. It is commonly used for projecting two-dimensional image (a texture) to a three-dimensional object of given topology. The abbreviation "UVW" represents the three dimensions like the standard Cartesian coordinate system. Instead of using "XYZ" which is the most common way of presenting Cartesian coordinate system, UVW is used to express different coordinate set. The third dimension in UVW mapping allows the texture to be wrapped on more complex and irregular objects than standard two-coordinate UV mapping. Each point in UVW system corresponds to a point on the given object's surface. The graphical designer or programmer generates specific mathematical function for implementing the map on the target surface. (Lo Turco & Sanna, 2009)

The UVW mappings in Turppa's thesis had to be hand made to suit the game engine rather than using what was created upon the creation of the 3D polygon models. The amount of time and costly equipment used for scanning individual objects to point clouds were not sustainable compared to modelling the 3D objects in 3ds Max. Laser scanned materials had their benefits when all the unsuitable or unoptimized traits were neglected. Whole sceneries taking place in real world locations could be modelled based on laser scanning material. If the process from point cloud data to low poly 3D models could be automated and optimized, laser scanning could be a viable and cost-effective way to generate content for game engines.

Stefan Boyekens wrote a book titled "Unity for Architectural Visualization" (2013). In his book Boyekens covered full process of customizing an architectural BIM model for Unity3D and how to import it into Unity3D. He instructed the readers how to construct a Unity3D project from scratch and how to add lightning, textures, and materials to all building objects in detail. The book provides a lot of helpful tips for everyone new to working with Unity3D but some of the information is outdated. Since 2013 Unity3D has released multiple updated versions and new assets have been published in the asset store to help people create their own virtual models. The book also focuses heavily on the process of importing data from Graphisoft ArchiCAD to Unity3D through various steps. Other file formats such as Trimble SketchUp and Autodesk FBX are mentioned but importing procedures for them are not as thoroughly illustrated as with the ArchiCAD.

2.2 Challenges of the research

Prior to the research, a well-documented procedure how BIMs should be handled before they can be used in a game engine or what should be considered while modelling a BIM if it is to be used as data in game development, could not be discovered. Without any prior knowledge, how the components should and could be optimized, a thorough study must be conducted in the according subject.

Besides BIMs, construction projects hold many types of data that can be used in a virtual model. Finding out the relevant data and which data can be used in a virtual model is a task in need of researching. Filtering out the unusable or too complex data and gathering latest revisions of wanted data should be as automated as possible. If outdated data is used, the benefits of virtual models decline due to the inaccuracies in the data.

For the production of virtual models to be applied to design processes in construction, the production of the models needs to be streamlined and automated as much as possible. Because the workflow with Unity3D and the process in question is not pre-defined in any way, the thesis writer needs to gain experience in using of multiple new software and have an understanding on how a construction design process usually works in terms of communication.

Virtual and augmented reality are based on the immersion they create. Without immersion, the human brain does not accept the content as believable which provokes user discomfort. The immersion can break from slightest inaccuracies or performance issues of a visualization. These performance issues consider for example dropped frames in rendering (frames skipped during rendering process) or tremoring geometry in the peripheral vision caused by unoptimized geometry. The virtual models must be created in such a fashion that performance is ensured with the hardware available and due to differences in hardware of different users, model optimization techniques must be covered.

In addition to the technological challenges, the construction industry as an old-fashioned industry, presents challenges of its own. Digitalization is a rising trend in construction (Agarwal, et al., 2016) but even so, the customs and processes of working in a design process are hard to modify. When introducing new technology or new ways to process data and introduce discussion to the design process, it may be hard to get individuals truly invested in the new way of thinking and doing.

When people are presented with information, the user interface should be self-guiding in terms of intuition: the less user must be taught how to use a virtual model, the easier it is to apply the model in active use and thinking process. If the user does not have prior skills to use a virtual model or has not seen one before, the first session might suffer from having a lot of time wasted in with the user being in awe of the model. Without focus, paying attention to the details and design solutions that should be addressed can be difficult.

2.3 Co-creative design in construction

Co-creation can be described as a management initiative, or form of economic strategy, that brings different parties together, to jointly produce a mutually valued outcome. (Prahalad & Ramaswamy, 2004). As for co-creative design in construction, the process is an iterative design facilitative in which future users of designed project are brought into the process of designing the future facilities they will have in their disposal. As a concept, co-creative design offers the clients to have an input on the design choices and increase the information being transferred between designers and the users which is seen as a decrease of cost and higher customer satisfaction as well as better response in the requirements of the space than without co-creative design process. (Lundström, et al., 2016)

Lundström, et al. (2016), did a case study on developing campus spaces through co-creation. In their study, they split co-creative design into three different categories, as shown in Figure 2, each of them having different interest and focus group: participatory design (PD), integrated design (ID) and concurrent engineering (CE). The figure illustrates how different forms of parts of co-creative design complement one another. Per their definition, PD aims to provide construction professionals with a comprehensive picture of user utilization so that they can better understand their mission. Thus, the primary objective of PD could be seen as maximizing the value-in-use. (Lundström, et al., 2016) Integrated design is viewed as a process aimed to improve the overall design quality rather than integrating the designers in collaborative discussion. Virtual models can be used as a part of PD or collaborative design (CD) which consist of true collaboration between PD and ID. An example of using a virtual model as part of CD is documented in Chapter 6.3 as a case study done in collaboration with University Properties of Finland Ltd.

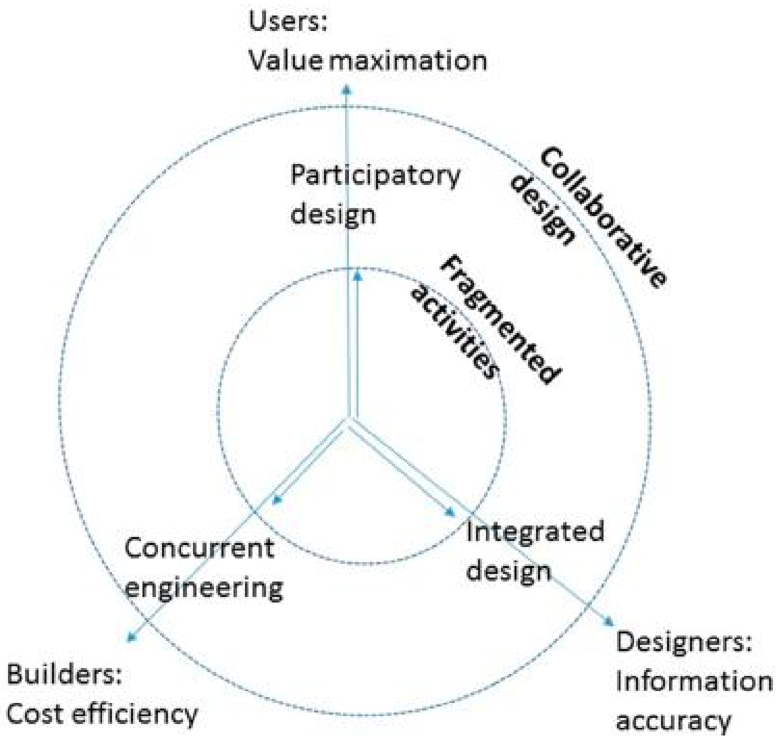

Figure 2: Different forms of co-creative design (Lundström, et al., 2016)

In their study, Lundström, et al. (2016) studied a construction project of turning a lunch restaurant in a university to new learning space using charrette in PD. The study consisted of following the project from concept design phase to time when the premises had been in use for six months. As the result of a well-organized co-creative design process, the new learning space increased and diversified the usage of the premises. Besides the increased utilization of the space, users had a larger emotional investment in the process due to feeling that they were creating a space for themselves and their ideas were incorporated to the design rather than feeling neglected and not incorporated in the design process.

Co-creative design does not only lead to better utilization of the space in question but it increases trust between different parties in a design project (Savolainen, et al., 2015). Through trust and open conversation, Savolainen, et al. (2015) observed in a case study that cost estimations were prioritized reflecting the future users of a space and cost saving suggestions were implemented without risking the upcoming utilization of the space. In addition to cost savings, the users had a better understanding regarding the usage of the new space they had access to. Thus, the space is utilized better than in a traditional construction project.

3. THE USED TECHNOLOGY

There is multiple different software for handling BIMs. The software can be divided into two categories: BIM authoring tools and BIM validating tools. BIM authoring tools include software that are used to create BIM data. Trimble Tekla Structures, Autodesk Revit and ArchiCAD can be used as an example of BIM authoring tools. The BIM validating tools are used to validate and combine different design plans into a merged BIM. Validation can be done via clash checking or using automated checklists for model information or meta-data. For example, all doors need to have IfcDoor property included in them. Solibri Model Checker is a good example of this type of software. BIM validation tools can also be used in project management for a construction site. Virtual models benefit from data from both BIM authoring and BIM validating tools. The data they both provide can be imported to Unity3D and used as pleased.

3.1 Unity3D

Unity3D is a cross-platform game engine with capabilities to create virtual models. Since all data is compiled to a single executable file that benefits from different gaming aspects for optimization such as occlusion culling, frustum culling and baked lighting, virtual models can be considered as 3D computer games from a point of view. However, the use of the model might deviate from standard game design. Usually games have a predefined object or they are goal-directed but a virtual model is created to present specific data. The license fee of Unity3D depends on the user's organization's usage of the software and gross revenue of the company or individual using it as defined in the Unity3D's end user license agreement's additional terms (Unity Technologies, 2017).

Depending on the skill level of the person producing a virtual model, a high-end performance desktop computer may be required for using Unity3D in an effective way. When working with IFC, SketchUp, Revit or ArchiCAD files, the models are not optimized for a game engine. Usually models have an unnecessarily large number of vertices per object which increases performance requirements. The virtual model producer should always optimize the polygon count of the used models or pre-parse the data as much as possible without losing vital detail before importing them to a game engine.

3.2 Virtual reality

Virtual reality (VR) is an artificial three-dimensional, computer generated environment which is presented to the user in a way that the user suspends belief and accepts it as a real or real-simulated environment. (Tötösy de Zepetnek & Sywenky, 1997) Immersion to a VR space can be created using multisensory experience through vision and sound.

The user of VR can be fully immersed in the design through head-mounted displays (HMD) so that he cannot observe the world outside of the VR space. Interaction with simulated objects or living things happens in real time.

There are multiple choices for VR hardware for consumers as shown by couple of examples in the Figure 3. Often consumers hear talk of Oculus Rift, HTC Vive, Google Cardboard or Samsung Gear. All mentioned technologies have their traits in terms of VR. So far only consumer friendly, or out-of-the-box ready solution, VR equipment with separate controllers are Oculus Rift and HTC Vive. The basic design for each set is quite similar: the HMD and controllers are tracked via photo sensors by the lighthouses and the data of movement is transferred to the software creating the immersion of user.

Figure 3: HTC Vive and Oculus Rift HMD VR systems (tomshardware.com, 2016; roadtovr.com, 2017)

Virtual reality as a concept is not a new idea. It has been studied for years and some research papers have been written about implementing computer aided design (CAD) ideas to VR (Whyte, et al., 1999). Even though their concept of VR and end-product was not an immersive VR space but a 3D visualization of buildings, they researched different effects of level of detail (LOD) models in VR environment.

VR can be used broadly speaking in two different use cases: simulating a real-world environment without being present in the mentioned environment for training and education purposes or for development and simulation of an imagined environment for a game or interactive story. Both use cases have a variety of different aspects to consider but in the scope of this thesis it is not relevant.

3.3 Augmented reality

Augmented reality (AR) is a real-time direct or indirect view of a physical, real-world environment which is supplemented by computer-generated elements (Azuma, 1997). Computer-generated content is overlaid on the real-world environment to compose AR rather than replacing the real-world environment as VR does. AR can be described as non-real-time view of world depending on the definition of AR, but in the scope of this thesis, AR is always referenced in a real-world, real-time environment which is changed depending on the overlaid content and user actions. Depending on the used technology (hardware and software), the user may or may not have any influence on the AR content. The simplest version of AR is simply laying informative content on physical world whereas more complex AR solution would incorporate different menu-items and interactive objects to the AR solution.

The methods of how AR is achieved may be broadly divided into two categories: marker-based and markerless system. Marker-based system uses symbols placed in real-world as reference points for overlaying AR content on top of the physical environment. The computer interprets the symbol to overlay on-screen graphic as if the overlaid content was directly on top of the reference point in physical world as illustrated in Figure 4. Usually the markers are physical objects but there is nothing that prevents the symbols to be displayed electronically, such as on a laptop screen or on a tablet. In addition to markers being able to be created digitally, the markers can be movable: when the marker moves, AR objects moves with it in tandem. (Craig, 2013)

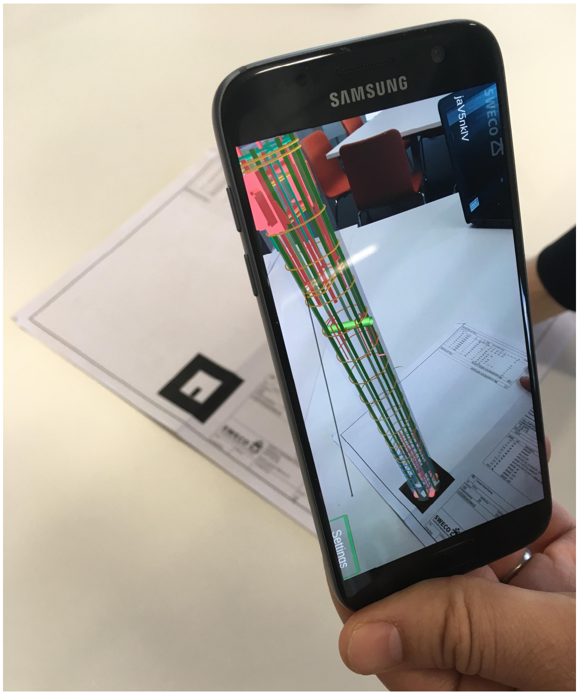

Figure 4: Example of marker-based AR. The rebars of a column are visualized via mobile device. The QR-Code acts as a marker for the AR software

Markerless system is better described as a system that does not utilize ficidual markers but benefits from the surrounding world. Natural feature tracking (NFT) works as the basis for mobile augmented reality, which provides the base for AR HMDs and the use of mobile phones and tablets as AR platforms. The NFT uses the device's location data, camera, digital compass, gyroscope, and accelerometer to determine the device's position and orientation in physical world and recognizes natural features such as building facades or skylines. It can also be used to utilize surfaces such as building facades and real-world objects. The acquired location data and compass is compared to the device's database to determine what the device is looking at and thus allowing the AR data to be displayed on-screen in correct position and location. (Craig, 2013)

AR devices can be anything from handheld phones and tablets to AR HMDs. The quality and the use of different AR devices may vary but usually the use case defines what kind of device is used.

3.4 Mixed reality

Mixed reality, or hybrid reality, can be differentiated from augmented reality by the user experience and utilization of AR content. AR brings virtual content to a real-world environment without obscuring the user’s vision. Mixed reality application does the same but in addition, it can utilize the real-world content in the virtual model and the virtual content can interact with real-world objects and vice versa. (Milgram & Kishino, 1994) Mixed reality applications could be used for both educative purposes and entertainment purposes. As for construction business, the mixed reality features could be used to guide workers in construction sites. Through MR glasses or lenses the workers could be provided with information about safe routes on the site, installation instructions for pre-cast elements, status of the building phase or a visualization of rebar plans combined which completion progress updates as the worker places more rebars.

Usually MR hardware includes at least a device with one or multiple cameras. Depending on the MR solution, the software to understand surrounding environment, Global Positioning System (GPS) and digital compass can be used to understand location and orientation in world-scale environment. With spatial mapping and GPS data, MR device can pin point how virtual content should be layered on top of the real-world environment. The algorithms and technology in different software may vary between developers but main functionality remains the same.

Spatial mapping is a topological modeling technique which analyses an object’s geometric properties. Topology expresses the spatial relationships between connecting or adjacent vector features (points, polylines, and polygons) in a geographical information system (GIS). In MR application, the adjoining and enclosing spaces or objects and their approximate distance relative to viewer’s position is mapped and incorporated to a virtual experience. A mixed reality application merges two different coordinate systems, a virtual coordinate system and a physical coordinate system, into one.

During the production of this thesis, only viable solution for a AR HMD is the Microsoft HoloLens which is an optical see-through device, as shown in Figure 5, that can render virtual models to user’s field of view without obstructing the visual connection to the real-world. The HoloLens takes advantage of Microsoft’s state-of-art hardware and software solutions that provide key functionalities for MR experience such as spatial mapping and the use of voice commands.

Figure 5: Microsoft HoloLens is a wireless AR HMD with two displays and a display protecting visor (Microsoft, 2016)

The HoloLens utilizes spatial mapping to perform realistic MR experiences. Through spatial mapping, virtual content can interact with real-world content, for example models can be placed on real tables. The primary functionalities in HoloLens that benefit from spatial mapping are hologram placement, occlusion, physics, and navigation. As HoloLens scans the environment, it can use existing surfaces to place objects on them. For example, a scale model of a house can be placed on a table and observed from different angles.

When the surrounding environment is mapped by HoloLens, the real-world objects are scanned if they occlude the placed holograms. The spatial surfaces are represented as triangulated meshes. By being able to hide holograms behind real-world objects, the immersion of MR experience is significantly increased as it can be perceived as if the object truly was in the same space as the user. The occlusion can also be used to prime expectations for a natural user interface based upon familiar physical interactions. For example, if a hologram can be hidden by a real-world object, the natural expectation is that the object could physically interact with the real-world objects as well. So, the user expects that the hologram collides with a table instead of going through it. (Microsoft, 2016)

4. INITIAL DATA SETS

A construction project has multiple different formats for data. Depending on the software, it can handle multiple data formats simultaneously, combining the design options. Sharing design solutions with all participants in a construction project can sometimes be overwhelming. Even though Common BIM Requirements 2012 (COBIM) (buildingSMART Finland, 2012) set a standard for sharing information via Industrial Foundation Class (IFC) file format in Finland, different countries have their own national regulations concerning information transfer. United Kingdom and the United States of America have their own national standards supporting the use of IFC in cross-industry BIM transfer. (buildingSMART International, 2016) However, every nation does not have a standardized protocol for data transfer in construction projects, but they often end up using data formats supported by the used software. When considering interoperability between different software in construction industry, IFC is often the most suitable data exchange format. In addition, the information level can vary between different designers. Depending on the discipline of the project there may be many different formats for different kind of data. The main problem for the receiver of information is optimizing the available data for the company's own use.

While IFC is a standard format for sharing information in construction project, Unity3D does not natively support it. This brings challenges for handling and optimizing the data for utilization. The IFC is not the only possible data file format for sharing information in construction project; it is a commonly agreed format and required by COBIM but there are other file formats that can be used in production of virtual models.

4.1 Data types

Each designer in construction project generates their own design plans with software of their preference or as dictated by employer's policies. In addition, open source data is available in different formats. The following chapters focus on what kind of data is available instead of focusing on the specific data formats in each category. All the data types can be used directly, or as a basis for refined data, in a virtual model. The process of filtering, optimizing and importing the data to Unity3D will be covered in Chapter 5.

4.1.1 Industry Foundation Classes

Industry Foundation Classes, or IFC, is a commonly used file format for transferring BIM data between different disciplines of designers in Finland (Nbs, The Building Information Foundation RTS, 2013). It is a neutral data format to describe, exchange and share information typically used within the building and facility management industry sector and it is constantly being developed further to answer to the needs of industrial data exchange. The current release of IFC available during the research of this thesis is IFC4 Addendum 2. (buildingSMART International, 2016)

The IFC file format is developed and maintained by buildingSMART International, which is an international non-for-profit industry alliance defining IFC as its data standard (buildingSMART International, 2016). buildingSMART International's core purpose is to standardize the procedures, workflows and processes for BIM and the company has multiple standards concerning the IFC file format. The base standard for IFC is ISO 16739:2013 standard which specifies the use of IFC for data sharing in the construction and facility management industries (International Organization for Standardization, 2013). In detail, the standard specifies BIM exchange format definitions for the building's life cycle, sharing information between different disciplines and BIM projects' structure and object attributes and definitions.

buildingSMART International has certification for the IFC compliant software. Most major software providers in construction industry have certified their software to support IFC 2x3 Coordination View 2.0. The certified companies include software providers such as Autodesk, Graphisoft, Nemetschek and Trimble (buildingSMART International, 2017). By the amount of standardization and supporting companies in both developing and using the IFC file format, it can be deduced that IFC is internationally supported file format for transferring BIM data between different parties in a construction project. While it may not be supported in its newest form in every software, the buildingSMART is determined to develop it further for more efficient information transfer process.

Depending on the usage in virtual models, IFC may contain much more information than is needed. All the metadata incorporated in IFC properties may be irrelevant when for example the usage of the data is a 3D shell model of a building. In such situation, operator of the virtual model has no need for defining what kind of room bounding the selected walls might incorporate to the design. All irrelevant data and objects should be excluded from the model before importing it to Unity3D for further modifications since Unity3D lacks the tools for more complex 3D modeling, and thus the modeling ought to be done in Blender or 3ds Max before importing the models to Unity3D.

While IFC provides a large amount of data that should be excluded due to data's unnecessary detailed information, it is always case sensitive. IFC metadata can and should be used in scripting and/or for filtering of the data or performing batch operations for multiple objects to save time in performing simple modifications to multiple buildings.

4.1.2 Native models

Revit and ArchiCAD are the two most used architectural modelling software in construction business in Finland. (Nbs, The Building Information Foundation RTS, 2013) Since they both have a large market share, processes for both software are covered later in this research.

Architectural models are BIMs but they deviate from a BIM produced by structural designer. Requirements for architectural modelling in Finland are defined in COBIM 2012 series three. The third series of COBIM 2012 defines modelling principles for both new buildings and use of BIMs in renovation projects. The series covers all architectural design phases in construction project from project planning to maintenance. When comparing the architectural model to structural designer's model, the architectural model has more data in terms of furniture, texture and layer information for walls, floors, ceilings, and other visible objects. In a structural model, only components that are relevant for the structural functionality of a building are modelled. (buildingSMART Finland, 2012)

The detail level is different with each model. A wall serves as a good example: in a structural model, the wall objects contain the load bearing components, connections between different parts, embedments and possible rebars. In an architectural model the wall has correct geometrical information and material information down to the detail for instance what kind of finish the outer and inner layers have. Since both BIMs have a different use they are not the same but complement each other. (buildingSMART Finland, 2012)

As for interoperability with Unity3D, both ArchiCAD and Revit models must be exported to different file formats to be used as a part of a virtual model. The process to import architectural models to Unity3D is covered in Chapter 5.

Structural BIMs in Finland are often created using Tekla Structures which is a structural design software developed in Finland. (Nbs, The Building Information Foundation RTS, 2013) Due to the complexity of Tekla Structures' native data format, it cannot be used directly as an initial data set for virtual models. The Tekla Structures models must be exported to IFC file for them to be used as a part of a virtual model.

4.1.3 Point clouds

Point clouds are a representation of information that consist of multiple data points, each having unique values in three-dimensional coordinate system. Individual points may have more data than just coordinates such as color information and classification. The points of a point cloud can have coordinates specified in a local coordinate system which is a coordinate system where the building's origin is near the coordinate system's origin. The local coordinate system is placed to a global coordinate system via data point which is referenced in global coordinate system. The data may be georeferenced which means it is tied to real-world coordinates and represent data points in physical world.

Point clouds can be created in different ways, including point cloud creation via laser scanning, tacheometry, photogrammetry or manually inputting each data point. The latter is by far the slowest and most time-consuming way to produce a point cloud because in all the other methods, the point cloud generation is an automated process. (buildingSMART Finland, 2012) (Savisaari, 2017)

In the context of this thesis, laser scanner means a scanner utilizing light detection and ranging (LiDAR) which is a non-contact remote sensing technique measuring reflected light from emitted energy pulse (a laser beam). (GIS Geography, 2017) A total station (TS) is an electrical or optical instrument that uses electronic transit theodolite in conjunction with electronic data collection and storage system. (Laurila, 2012) It measures the sloping distance data points, distance, vertical and horizontal angles to measuring point. In comparison to TS, point clouds created via photogrammetry utilize complex mathematical functions for creating a point cloud from photographs.

Photogrammetry can be defined as the process of deriving metric information about an object through measurements made on photographs of the object. The relationship between images of the object and the object itself can be established by different means that can be divided into two categories: analog, using the optical, mechanical, and electronic components; or analytical, where the modeling is mathematical and the processing is digital. Usually the image processing is done digitally due to its superior speed compared to analog tasks. The information derived from photographs is used as a basis for creating a representation of photographed object in collaboration with computer vision which is an artificial intelligence application used for identifying objects from images. There are different software solutions available for creating a point cloud, utilizing different sources of information (images, videos, point clouds) as the basis for computing the point cloud or mesh model. (Mikhail, et al., 2001)

The computing of photographic measurements to produce the final XYZ coordinates of individual points is done via triangulation and resection. By mathematically intersecting converging lines in space, the precise location of an object can be determined. Resection is the procedure to determine final position and orientation of the used camera when picture is taken. Figure 6 illustrates this basic principle of photogrammetry. (Geodetic Systems, 2017)

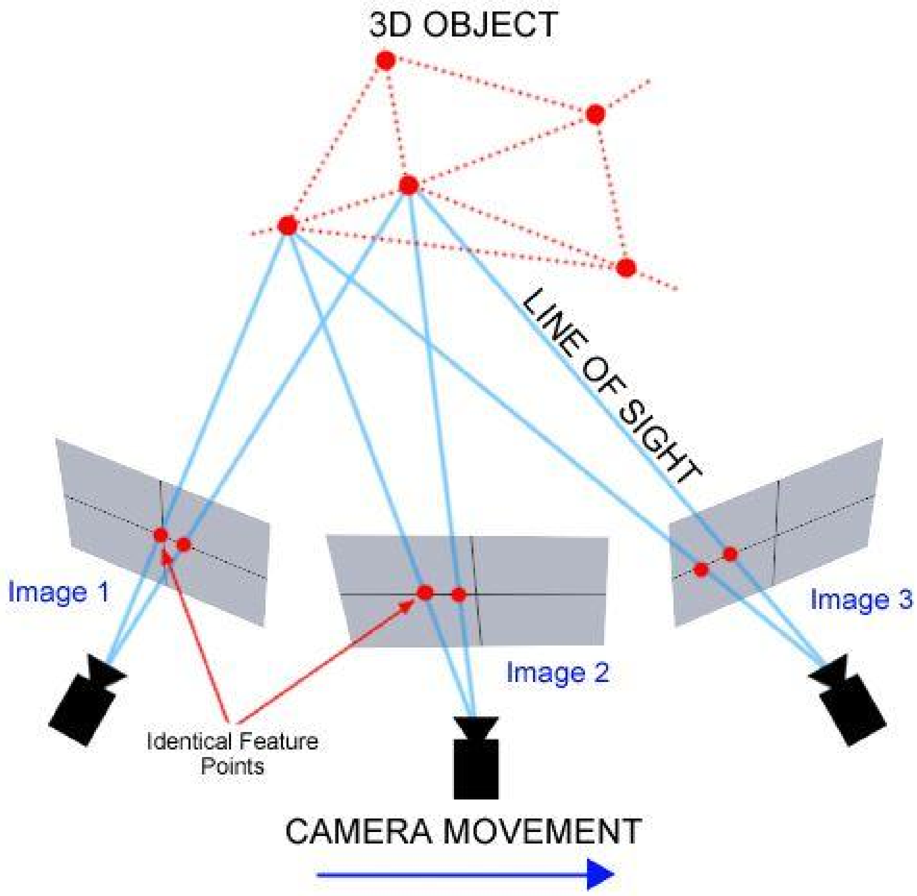

Figure 6: The basic principle of photogrammetry (Mason, 2016)

As a point cloud is a representation of data, the same data can be visualized and presented in different ways. Triangulated meshes or texturized three-dimensional models are generated by combining the points with vectors and estimating the forms generated by combined vectors of different points. Point cloud data should always be optimized, unified and classified before creating a three-dimensional mesh model. Due to noise in point cloud data, which is an inaccuracy caused by external factors such as lightning conditions or internal inaccuracy of the used equipment, the mesh model may be inaccurate. Although, when optimized it can serve as a unified three-dimensional model that represents more the real-world object as a solid instead of a point cloud. The data itself does not change between different representations but the information it provides does. (Savisaari, 2017)

Point cloud data can be utilized in large variety of software. A point cloud can be used as a base for BIM modelling and a triangulated mesh can serve as a ready-to-go geometry model, and as IFC when combined with meta-data. Without the meta-data incorporated to the component, a triangulated mesh is simply a geometric object. Some use it as a basis for modelling BIMs but it can be incorporated as is, if the point cloud's purpose serves as a reference for design plans in a scanned real-world space. When using point clouds with

BIMs it is essential that both the BIM and the point cloud are specified in the same coordinate system. Without orienting the design plans and point clouds, faulty geometry may be modelled and it affects the collaborative design phase since the clash check of combined BIM becomes inaccurate due to false location data.

4.1.4 Open source data

The National Land Survey of Finland (NLS) has a file service portal for all open source information provided by the nation of Finland for free implementation. It provides data such as aerial scan based point clouds of Finnish landscape, general maps, and topographic map raster images and multiple databases. The databases are constantly updated, though the frequency depends on the data type. Update intervals differentiate from weekly updates to yearly updates. (National Land Survey of Finland, 2017)

National Landscape Database (NLD) is a project done in collaboration with different companies, cities and NLS. The project aims to provide essential landscape data, such as survey data and data from cities own database, in a digital format for free utilization. The data will be unified to be as interoperable as possible and the database will contain 3D information of key locations in Finland. Each of the locations will have unique ID and full history background of the site. The NLD focuses to unify the practices using and storing location data in Finland. It is based on voluntarily providing the data for use of everyone in a centralized location, which is the NLD.

Finnish Transport Agency (FTA) offers three different ways to use their information in their open data program: a file service portal where users can download what they want and two application programming interfaces (API) that can be used to query information directly from their applications. The two available APIs are Web Map Service (WMS) and Web Feature Service (WFS). Both APIs are open and can be accessed whenever wanted, free of charge. The FTA database offers users information about different trafficking aspects of Finland. The main categories are road traffic, public transportation, sea traffic and transport networks. All the categories have different layers of information that can be queried and filtered on demand. (Finnish Transport Agency, 2017)

In addition to mentioned databases, many cities in Finland provide their own material for utilization such as an open API for WFS and WMS databases. The databases and map services offer up-to-date data of the city's maps, environmental planning charts, public transportation, laser scans of the city and some cities offer a model of the whole city.

4.2 Open source data vs. self-produced data

Data origin should always be validated and the data quality assessed before using it in a virtual model. Depending on the use scenario, open source data may be too inaccurate.

For example, when modelling a city block, the point clouds provided by NLS's file service portal may be outdated. In such case the possible missing buildings or roads should be scanned with a drone for more up-to-date information of the area if it is necessary for the success of a project.

When using self-produced data, the expertise of the person or organization should be validated in same fashion as using measured data in design solutions. Accuracy of the data is a key factor. Unnecessary work done due to inaccurate data in the beginning of a design phase can always be accounted as lost time and money. By having standardized working methods and using file formats that are commonly used by variety of experts, the quality of measured data can be ensured.

The upsides in NLS and FTA materials are the quality and reliability. Even though they may not always be up-to-date, they are validated and reviewed by experts. The user can be sure that the material he has at hand is correct and may be used in his projects. The ready-to-go data needs less effort to be utilized. Using point clouds as an example: NLS point clouds are cleaned up from excessive noise in laser scan data, the point clouds are classified and the information of accuracy is available at NLS website. When creating a point cloud from scratch, the target needs to be scanned with a laser scanner. The produced point cloud must be cleaned up from noise and minor inaccuracies and the points need to be classified with Terrasolid TerraScan or similar software for further utilization.

4.3 Utilization of the data

With different technological solutions, different amount of data can be used as the basis for creating a virtual model or embedded into the model. The data can be imported to Unity3D or it can be requested from external service through various APIs at runtime. As emphasized multiple times in this thesis, the use case dictates what data and in what fashion it should be included in the model itself.

The data can be utilized to either provide sufficient grounds for giving feedback about the design plans or to illustrate how the design plans would affect the functionalities of a space or area. The transform of information can be done in different ways. Web-based user interfaces can serve as a platform to collect feedback from a large user group. By distributing a domain address to public information channels, larger group of users can be drawn to explore the design plans and provide their insight on how the plans work or how they should be modified.

A web-based interface can be executed as WebGL build from Unity3D. The end-product must be uploaded to a server and a domain must be connected to it. Typically, Unity3D gives a standardized HTML front-end file which contains the basic functionality to run a website and a Unity web-player component embedment which serves as the program. Figure 7 illustrates how the architecture of a Unity3D WebGL build was formed in Shen, et al. case example (2012). The build can be divided into three sections: Proprietary BIM models and the Unity3D game engine containing the project file where all interactivity and game behavior is adjusted, the server side which serves as the program's back-end containing the build files and web server components and the client side which a front-end, run from the user's web browser.

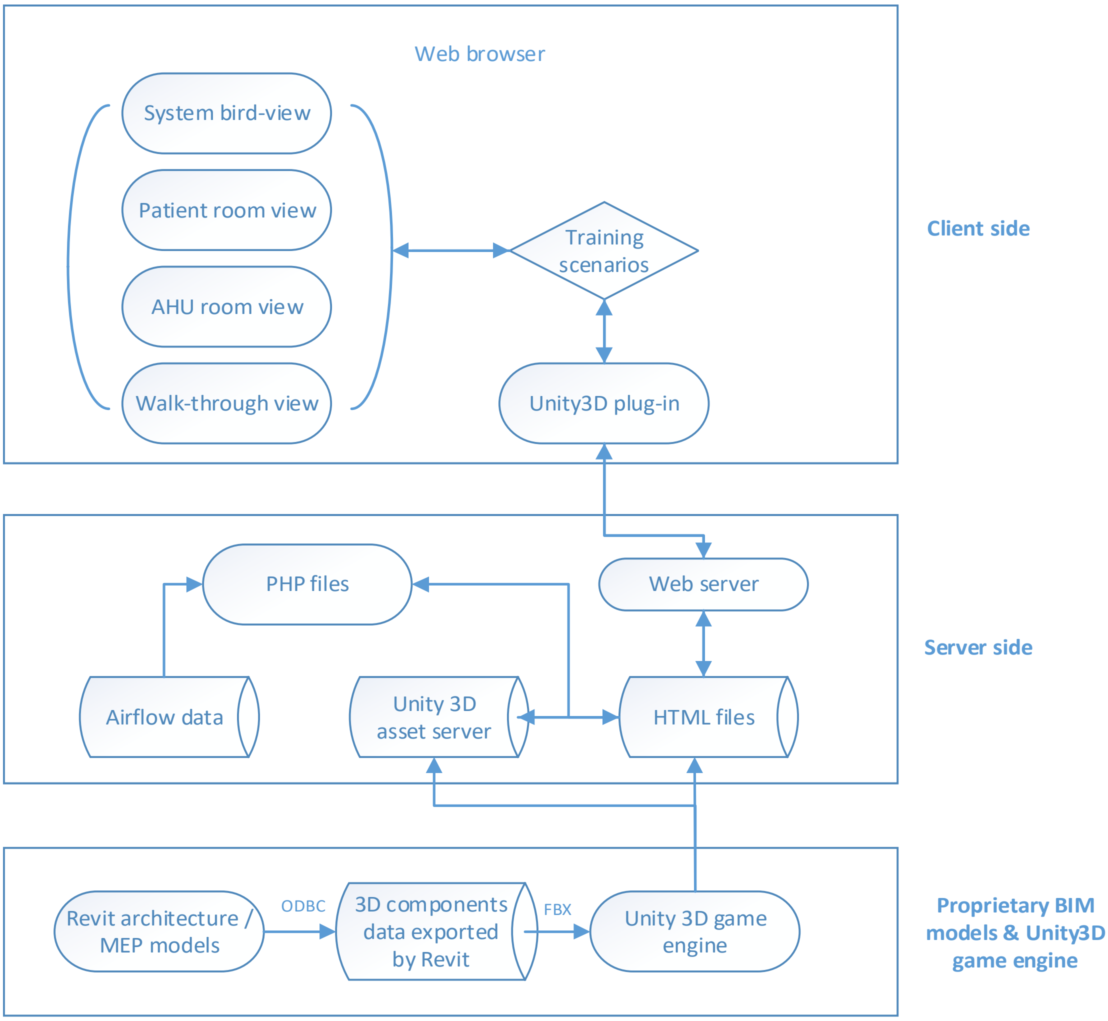

Figure 7: An example of architecture of a web-based game solution (Shen, et al., 2012)

In addition to a web-based solution, the applications can be built to mobile platforms or more commonly to a windows platform to be run from a computer from one executable file.

The ideal situation would be when the applied data could be used as it is and does not need any modification or filtering. However, this seldom is the case. When planning for the data allocation, it should be decided if the data is imported at runtime or directly to the Unity3D project. Depending on the use, each possibility must be considered separately. When the data is imported from a database by demand, the best solution would be creating a query at runtime and updating it as needed. This way the project file does not grow to be too large and the system is more dynamic in case of a need of modification. When importing data directly to Unity3D, it needs to be considered that updating the data is not as streamlined as at runtime when executing an application. The upside of importing the data directly to Unity3D, is that the data is ready for utilization. If all data is requested at runtime, application load times will increase significantly and user experience is compromised.

5. PRODUCING A VIRTUAL MODEL

This chapter describes how the production process of virtual models is achieved through an empirical process in Chapters 5.1 to 5.3. The last sub-chapter presents different techniques for optimizing the virtual models.

The first thing that should always be addressed when creating a virtual model is answering the question "What do I want to achieve with this model?". The use case of a virtual model defines what data should be incorporated into it, how should the data be presented and what kind of LOD should be used.

The process defined in this thesis is not absolute. There are numerous ways of producing a virtual model, however due to the research outlining, only one complete process is defined to avoid a too large scope.

The production of a virtual model, the process is divided into three phases:

- 1st phase is data preparation. The needed data is gathered and exported to IFC or 3ds Max compliant data format
- 2nd phase is data refinement and exclusion of all unnecessary objects and information from the data
- 3rd phase is importing the data to Unity3D and checking for possible problems in orienting all the models if not done beforehand

For converting and modifying different types of data, multiple different software is used in this thesis. 3ds Max is used as the main modeling software for modification and optimization of all file formats containing 3D geometry. 3ds Max is a commercial software with license costs but it can be replaced by an open source software, called Blender (Blender Foundation, 2017). Datacubist Simplebim is used for excluding unneeded data out of IFC models.

Most of the repetitive tasks for modifying the geometric models are done with modifiers which are 3ds Max's built-in tools, each designed for specific task or tasks. Since 3ds Max and Blender do not support IFC file format natively, as listed the supported file formats (Blender Foundation, 2017) (Autodesk, 2016), IfcOpenShell, an open source toolkit and geometry engine, provides IFC support for both mentioned software (IfcOpenShell, 2015). The only downside is that IfcOpenShell is supported by older versions of 3ds Max and Blender, it being available for 3ds Max 2015 and Blender 2.73.

Regardless of the initial data format, the process to transfer the data from initial file format to Unity3D requires exclusion of irrelevant data and possibly optimization of the data in a modeling software as presented in Chapters 5.1 and 5.2. The following chapters summarize the processes of data transfer from a software to another without losing vital information. The flow charts first illustrate different file format conversions to 3ds Max and finally the process to transfer the models from 3ds Max to Unity3D. All the data formats are normally present in a construction project.

5.1 Data preparation

Gathering and preparing the data for a virtual model can save time from refinement phase when done properly. Otherwise excess time in refinement phase will be used on excluding objects from the model that should have been excluded in the data preparation phase. Depending on the data format and discipline of the designer, the process can vary depending on the software.

5.1.1 Architectural models

Virtual model often acts as representation of information. The outer shells of components are the main interest in exporting due to them being only visible surfaces of the object if cross-sections are not created with a dedicated shader. A shader is the actual element which holds the code how objects are drawn and the materials in Unity3D are representations of shaders. A Revit file can be directly imported in 3ds Max which decreases the amount of work needed to prepare the model for exporting. All objects needed can be included in a single 3D view. All materials and textures are compliant with 3ds Max due to both software being produced by Autodesk and therefore they use the same renderer. The same principle applies to ArchiCAD: the easiest way to gather the objects for a virtual model is to include them in a single 3D view. As per Revit, all modeling needs to be done with 3D objects avoiding use of 2D objects as much as possible.

While transferring data from architectural software to Unity3D is relatively simple, the task can take hours if not done properly. The iteration between different number of objects and export settings was a slow process. Through testing it was discovered that the number of objects have effect on both loading time of the model and the time it takes to optimize the produced virtual model. With more game objects in one model, the occlusion culling (explained in Chapter 5.5.1) parameters must be defined with care for reaching required performance level.

ArchiCAD and Revit can be used in same fashion when transferring information to 3ds Max. The difference in operational viewpoint comes from the file formats. Due to Revit being a part of Autodesk's software suites, the native file format of Revit (.rvt) is natively supported by 3ds Max as import format. The ArchiCAD project files (.pln) are not supported by 3ds Max so the information has to be exported in different file format.

The covered two architectural software, Revit and ArchiCAD, have different approach on how to deliver the information in 3ds Max compliant data format. Revit is part of Autodesk's software suites as mentioned before, therefore it has a native interoperability file format for 3ds Max. The Revit native files (.rvt) can be linked or imported to 3ds Max. The difference between linking and importing a Revit file is that when linked, modifications done to Revit file are updated in the 3ds Max project. If the Revit file is imported to 3ds Max, the modifications done in Revit will not update to 3ds Max.

For ArchiCAD the exporting process is slightly different. After gathering all objects ready for export in a single 3D view, the view needs to be saved as (exported) as a 3D Studio file (.3ds). The 3D Studio file format is a legacy file format for 3ds Max and is the best suitable format available for data transfer between ArchiCAD and 3ds Max. Upon exporting, the user can define settings for the export with on-screen GUI. Depending on the size of the model, it is recommended to use a few minutes of consideration for the export settings. The exported 3D Studio file can construct objects with four different setups:

- ArchiCAD objects: each object is exported as is and no grouping is applied
- Element types – surfaces: Groups objects by ArchiCAD construction elements and creates internal groups for each surface in objects
- Layers - surfaces: creates groups for each ArchiCAD layer, and within those groups, creates internal groups for each surface
- Surfaces: creates a group for each surface

When exporting, the drawing unit scale must be matched with the used scale in 3ds Max. If the 3ds Max operates in metric world with one meter as the base unit, the export drawing unit scale needs to be set for 1000 millimeters. If the scale was not set with matching properties, the scaling of models in 3ds Max was not in real-world scale which produces problems when combining different geometrical data. When available, the colors should be matched to 3D Studio color palette and the texture information should be included in the 3D Studio file. If not, it was found that all of the objects were not assigned their designated material and color set in ArchiCAD and thus, the color and texture information would not be transferred into Unity3D in the end. The Figure 8 summarizes the process of data transfer from architectural software to 3ds Max.

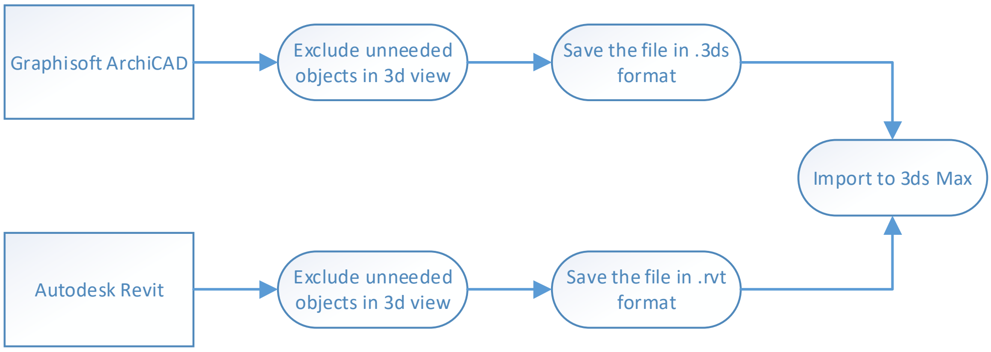

Figure 8: Process of transforming data from Architectural software to Unity3D

As seen from the flow chart Figure 8, the exporting processes for ArchiCAD and Revit are near identical, differentiating only in the used file format for exporting. However, the file format transformation in ArchiCAD consumes somewhat more time than Revit and may result in unwanted results when importing to 3ds Max if done carelessly. Different file formats could be used in the data transition between architectural software and 3ds Max. The used formats were selected due to Revit being part of the same Autodesk's design suite and thus, the interoperability is already optimized by Autodesk. The 3DS Studio file format was deemed appropriate due to it being a legacy format for 3ds Max, resulting in less problems in the data transformation.

5.1.2 IFC models

For the IFC models to be used in a virtual model, the detail level is often too great as was found through inspection of the models. As an example, a rectangular column that could be modeled with twelve polygons (two per a side of rectangle) was sometimes over thirty polygons. When this occurs with every column in a six-story office building, as the results, the excess polygon amount increases. The use case of the IFC model dictates the needed objects per exporting scheme. When exporting from a software to IFC, if the sole purpose of the IFC file in question is to be used as a base for a virtual model, the initial grouping should be done accordingly. All unnecessary objects for the virtual model should be excluded from the export.

However, usually IFC is used as transferring information between different designers in a construction project as presented in Chapter 4.1.1. It was found that the producer of a virtual model may not have much control over the content of the received IFC. Instead he needs to filter out the unnecessary components according to his specification and the use of the virtual model.

When the project size increases, so does the number of different files. Therefore, all IFC files intended to be used in the production of a virtual model should be gathered into a centralized location for the preparation phase. This reduces amount of time used into locating the relevant models. The actual data exclusion process of IFC models with Simplebim will be presented in Chapter 5.2.

5.1.3 Point clouds

Point clouds may be acquired through different methods as presented in Chapter 4.1.3. Whether the point clouds are generated via photogrammetry or laser scanning, the point cloud data should always be classified and unified if not done by the producer of the point cloud.

Point clouds are not natively supported by Unity3D. Even though there are assets available for Unity3D that can view point clouds in Unity3D editor, for point clouds to be viable, they need to be turned into triangulated meshes for importation to Unity3D. A free-to-use asset available for point cloud viewing in Unity3D was tested in this research. However, the amount of customization needed for displaying the point clouds was excessive. Laser scanned point clouds were not able to be presented in the Unity3D editor due to errors in scripts provided with the asset. Thus, the free asset was discarded from the research. In addition, with a price of seventy-five dollars, a working point cloud viewer could have been purchased.

While point cloud information is precise, the point clouds can be too dense for triangulating a reasonably dense mesh. As a point cloud is triangulated by combining the points with vectors, a denser point cloud produces a mesh with more polygon than a less dense point cloud. In turn, this increases the cost on performance. COBIM 2012 suggest a five-millimeter density for point clouds. According to Savisaari (2017), the COBIM requirement can cause problems resulting in large point cloud files which in turn causes problems to interoperability. Therefore, the point clouds need to be segmented with reasonable file sizes to have a better overall interoperability with different software.

With a classified point cloud, different data can be filtered out fast. For the purposes of this research, the methods for point cloud classification are not covered. The point clouds' point classification should follow the American Society for Photogrammetry and Remote Sensing's (ASPRS) LiDAR point class specification, which is presented in Table 2 below.

Table 2: LiDAR point class definitions (The American Society for Photogrammetry and Remote Sensing, 2009)

Classification value	Meaning
0	Created, never classified
1	Unclassified

2	Ground
3	Low Vegetation
4	Medium Vegetation
5	High Vegetation
6	Building
7	Low Point (noise)
8	Model Key-point (mass point)
9	Water
10	*Reserved for ASPRS definition*
11	*Reserved for ASPRS definition*
12	Overlap Points
13-31	*Reserved for ASPRS definition*

The triangulation can be done with TerraScan which is a dedicated software for modification and classification of point clouds. By importing the point clouds to TerraScan, filtering out all irrelevant points per the point classification definitions and vectorizing either buildings or terrain surface from the point cloud, the result can be exported to 3ds Max for further customization. The process can be automated with macros; however, the macros were found to be suitable for point clouds with the same density and definitions for consistent results. The vector presentation of buildings and ground surface do not have texture information but it can be applied via Terrasolid TerraPhoto if orthogonal and oblique images are provided.

The Terrasolid applications use Bentley Microstation as their workspace. The supported export file formats are therefore defined by Microstation's capabilities. The triangulated meshes can be exported as OBJ or FBX files for customization in 3ds Max.

5.2 Data refinement

When working with different file formats, the data needs to be combined either in Unity3D or with Simplebim to form a complete scene. Since 3ds Max is a modelling software, it was found not to be the best suitable option for combining different data but it is better used for modifying the data as needed. The Simplebim is used for trimming the IFC models before transition to 3ds Max.

One construction project may have multiple data sources for the needed information. The geometrical properties can be derived from IFC files, architectural models, mesh models and point clouds. All mentioned can be imported to Unity3D via a multistep process.

5.2.1 Architectural models

As defined in Chapter 5.1.1, the export from Architectural models should be done to 3D Studio format. The 3D Studio file format is supported as import format for 3ds Max. When importing a 3D Studio file, the 3ds Max automatically sets the material properties for objects in scene to use standard materials. It was discovered that if an object has been defined textures or multiple different materials (for example a wall with two types of texturing), 3ds Max applies a Multi/sub-object material for those objects. This keeps the ability to define textures for all different segments of an object in Unity3D.

The intial grouping of objects before importing to Unity3D has an impact on the occlusion of said objects. If a group of objects is formed to be an instance from the perspective of Unity3D (walls combined as a single object), the objects cannot be culled when a portion is visible. This sets requirements for the data preparation and refinement phases. During testing, it was found that if the grouping is done carelessly, it will have a cost on the performance of virtual models and the overall usability of the virtual models due to ineffective occlusion culling. Therefore, the grouping needs to be done so that an individual instance of an object cannot contain too large number of polygons which in turn translates to grouping of objects to be solely as a responsible of the model producer. However, while the objects may be grouped when delivered to the producer of the virtual model, the excess time consumed to divide the grouping for suitable size of objects is paid in full with improved performance of the virtual model. More of occlusion culling is presented in Chapter 5.5.1.

The furniture imported to 3ds Max from architectural software was found to be often in need of polygon count optimization. Several different 3ds Max script were coded in the MaxScript format to speed up the process including repetitive tasks relating to optimizing large quantities of objects. MaxScript is 3ds Max's native scripting language. First the objects in need of reducing the polygon count were selected and ProOptimizer modifier. Via testing different amount of optimization, it was found that a fifty percent optimization is generally usable. When the objects' polygons are reduced over a certain threshold, the geometry becomes severely warped. With a fifty percent optimization, the geometrical properties of objects do not disform too much for it to stand out in a virtual model. The results were then reviewed and adjustments to the rate of polygon reduction of individual objects were made if necessary. MaxScript for polygon reduction is shown in Script 1. After the optimization, modifiers were collapsed to the base modifiers (editable mesh /

editable poly) for the changes to become permanent. If ProOptimizer is not collapsed to base modifier, the optimization calculation needed to be done each time the 3ds Max file was opened.

```
(
	max modify mode

	/* selection array */
	selObjsArr = selection as array

	/* loop for going through the array */
	for o in selObjsArr do
	(
		/* select object, add modifier “ProOptimizer”, calculate number
		of polygons and apply reduction of 50% */
		select o
		modPanel.addModToSelection (ProOptimizer ()) ui:on
		$.modifiers[#ProOptimizer].Calculate = on
		$.modifiers[#ProOptimizer].VertexPercent = 50.0
	)
)
```

Script 1: Applying ProOptimizer modifier to selection in 3ds Max

The Script 2 was executed for all objects in a 3ds Max scene. It goes through an array of objects in the same fashion as the first script but instead of applying ProOptimizer modifier, it applies Unwrap UVW and UVW Map modifiers to the objects in an array. First it applies the Unwrap UVW modifier and executes command to flatten the object’s UVW map to be used as the basis for lightmaps. The created UVW unwarp can be used in general light simulations but for more realistic results the UVW unwrapping should be done by hand for each object to ensure the production of a quality UVW map. After unwrapping the UVW map, a box UVW map for textures is applied and set to match real-world dimensions.

```
(
	max modify mode
	/* selection array */
	selObjsArr = selection as array

	/* go through the array one by one */
	for o in selObjsArr do
	(
		select o

		/* apply Unwrap UVW */
		uvMod = (Unwrap_uvw())
		addModifier o uvMod ui:true

		/* Flatten map with rotation angle of 45 degrees, allow
		rotate of the sections, apply spacing of 2mm between
		each section */
		uvMod.flattenMap 45.0 #([1,0,0], [-1,0,0], [0,1,0],
		[0,-1,0], [0,0,1], [0,0,-1]) 0.02 true 0 true true

		/* set UVW Map to channel 2 (used for lightmapping) */
		$.modifiers[#Unwrap_UVW]
		$.modifiers[#unwrap_uvw].unwrap.setMapChannel 2

		/* apply UVW Map modifier */
		modPanel.addModToSelection (Uvwmap ()) ui:on

		/* Select box type map and apply Real-world map size */
		$.modifiers[#UVW_Map].maptype = 4
		$.modifiers[#UVW_Map].utile = 1
		$.modifiers[#UVW_Map].vtile = 1
		$.modifiers[#UVW_Map].length = 1
		$.modifiers[#UVW_Map].width = 1
		$.modifiers[#UVW_Map].height = 1
		$.modifiers[#UVW_Map].realWorldMapSize = on
	)
)
```

Script 2: Applying Unwrap UVW and UVW Map modifiers to a selection array

5.2.2 IFC models

IFC models should be imported to Simplebim for trimming the IFC files before transition to 3ds Max. The use case dictates what should be left in the model, but the process applies to all use cases with IFC models.

Simplebim provides several implement ready template files for pre-defined rules of object exclusion. Each template is defined for a specific discipline and software. The user has the option to create custom templates for handling the initial exclusion. A custom template should be created for repetitive tasks to suit the needs of an organization's business model. If a company provides visualization services for a finished building, it would be cost-effective to create a template which filters out all embedded objects that are not visible due to them being hidden in the final product. The template would reduce the time spent on the IFC file considerably and provide consistent data for exporting to 3ds Max, which in turn provides grounds for creating ready-to-use scripts for 3ds Max, thus speeding up the overall process.

Figure 9 illustrates how the amount of transferred data can be reduced by filtering out the unneeded parts. In the example, one floor of a building was visualized in VR. By excluding everything below the floor's slab and the ceiling above, a general filtration was done in matter of minutes. Further filtration can be done by using the IfcObjectClass do identify object classes that are not required for the visualization.

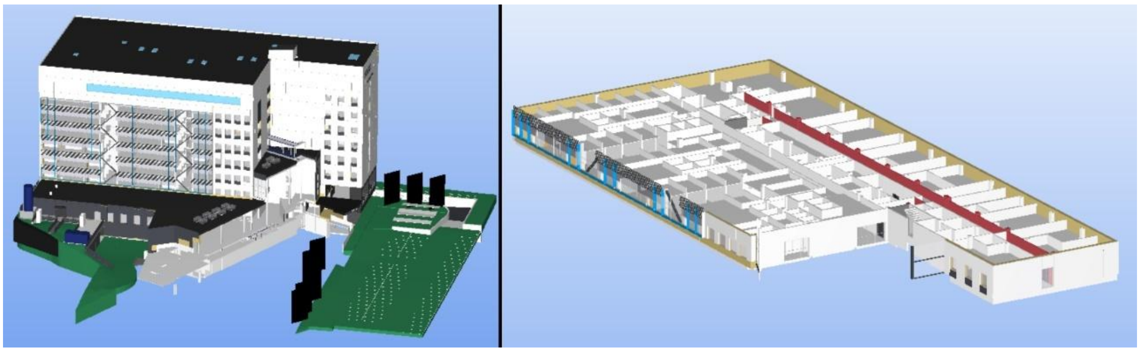

Figure 9: Example of data exclusion in Simplebim. The initial model (left), included objects for data transfer (right)

From Simplebim, the IFC needs to be exported for the filtration to be applied. For implementation in a virtual model, the Simplebim export settings needed to be adjusted. Colors of 3D objects and IfcOpening objects should be included in export settings for the visual appearance of the model to be as defined. The export from Simplebim needs to be done by object class to reduce the file size of individual files and to provide a preliminary grouping of objects by including them in separate files. The data transfer process is shown in Figure 10.

Figure 10: Process of transferring data from IFC to 3ds Max

When transferring to work in 3ds Max, IfcOpenShell handles the interpretation of IFC files to be imported to 3ds Max 2015. Upon importation, the file should be saved as 3ds Max native file (.max). When IFC file is imported to 3ds Max, individual material is generated for each object. Placing a same type of material for matching objects needs to be handled via a 3ds Max script. By reducing the number of materials in a Unity3D scene, the performance is improved.

The transfer process was not completed with problems. The IfcOpenShell plugin creates a transparent object for all holes with IfcOpening classification. The objects needed to be deleted by hand in 3ds Max. This was accomplished by grouping all visible parts of the model and then deleting all objects that were left out of the created group. If the deletion was not done, each of the hole objects created a rectangular transparent game object in Unity3D resulting in multiple unneeded game objects that needed to be either deleted or turned off in the editor.

5.2.3 Triangulated meshes

Comparing triangulated meshes generated from point clouds, in an essence, they are not so different from the geometrical models imported from BIMs: both have often an excess number of polygons and need to be optimized. However, the optimization process for a triangulated mesh often has and extra step for efficient optimization.

The triangulated surface of a generated object is a precise representation of a surface. However, with when working with virtual models, a smoothened result was found to be a better option for utilization in a virtual model. A quad-mesh can be generated from the original triangulated mesh in 3ds Max by applying a conform modifier to it. This provides both optimization in polygon count and a smoother, more suitable surface for use in a virtual model.

As an example, a mesh was optimized using the conform modifier. By creating a plane on the level of the lowest point of the mesh, the generated quad-mesh does not droop over the edges. Another plane was generated above the mesh to be modified with the conform modifier. As result, the quad-mesh drapes over the triangulated mesh, generating a smoother surface to be used in a game engine. The polygon of the area of two hundred square meters was dropped from 27 633 to 2 500. The differences in mesh topology are presented in two illustrations of the same two meshes. Figure 11 presents a perspective view of the meshes and Figure 12 a top view.

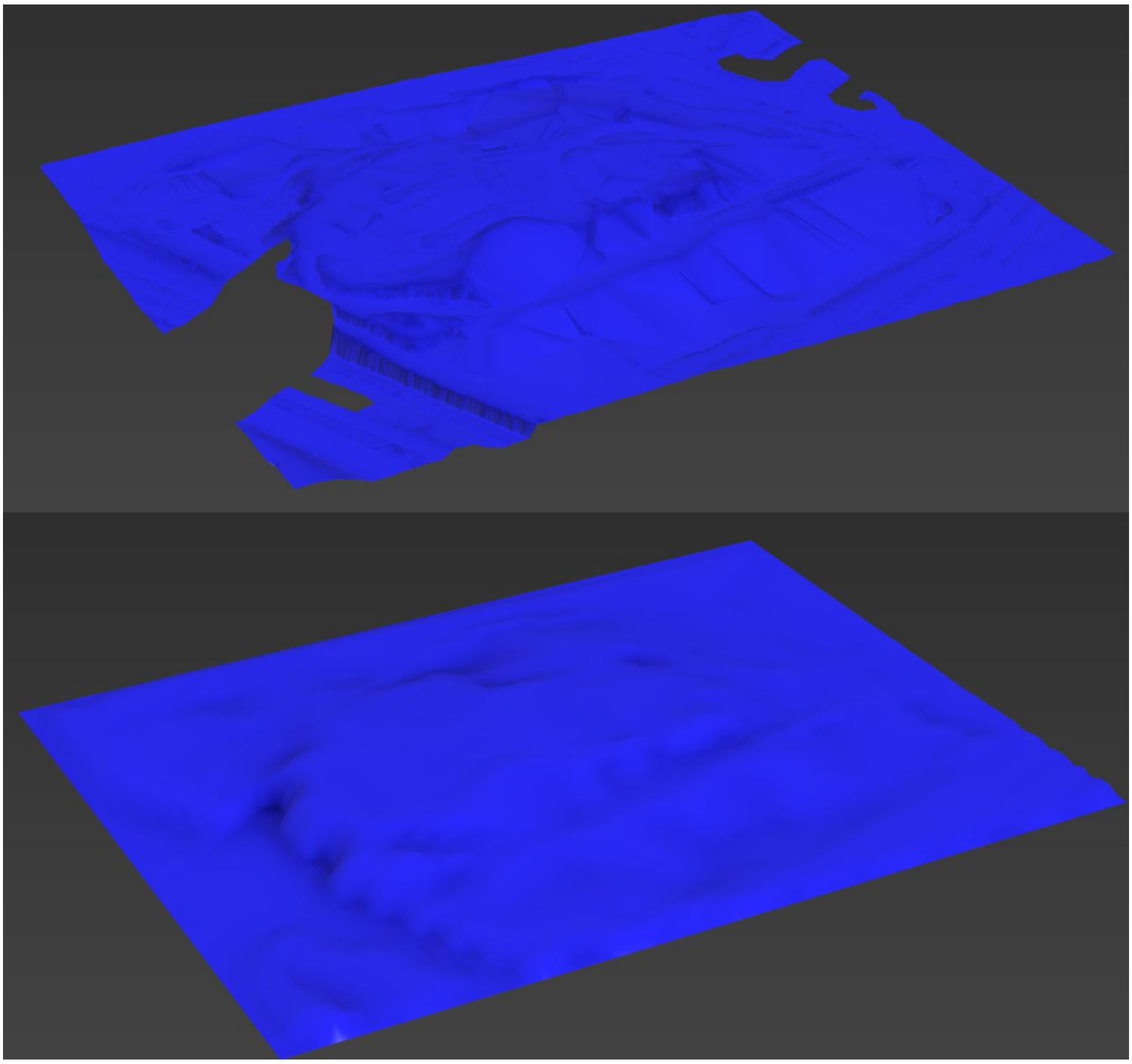

Figure 11: Perspective view of the initial mesh (top) and the quad-mesh generated using it as the basis (bottom)

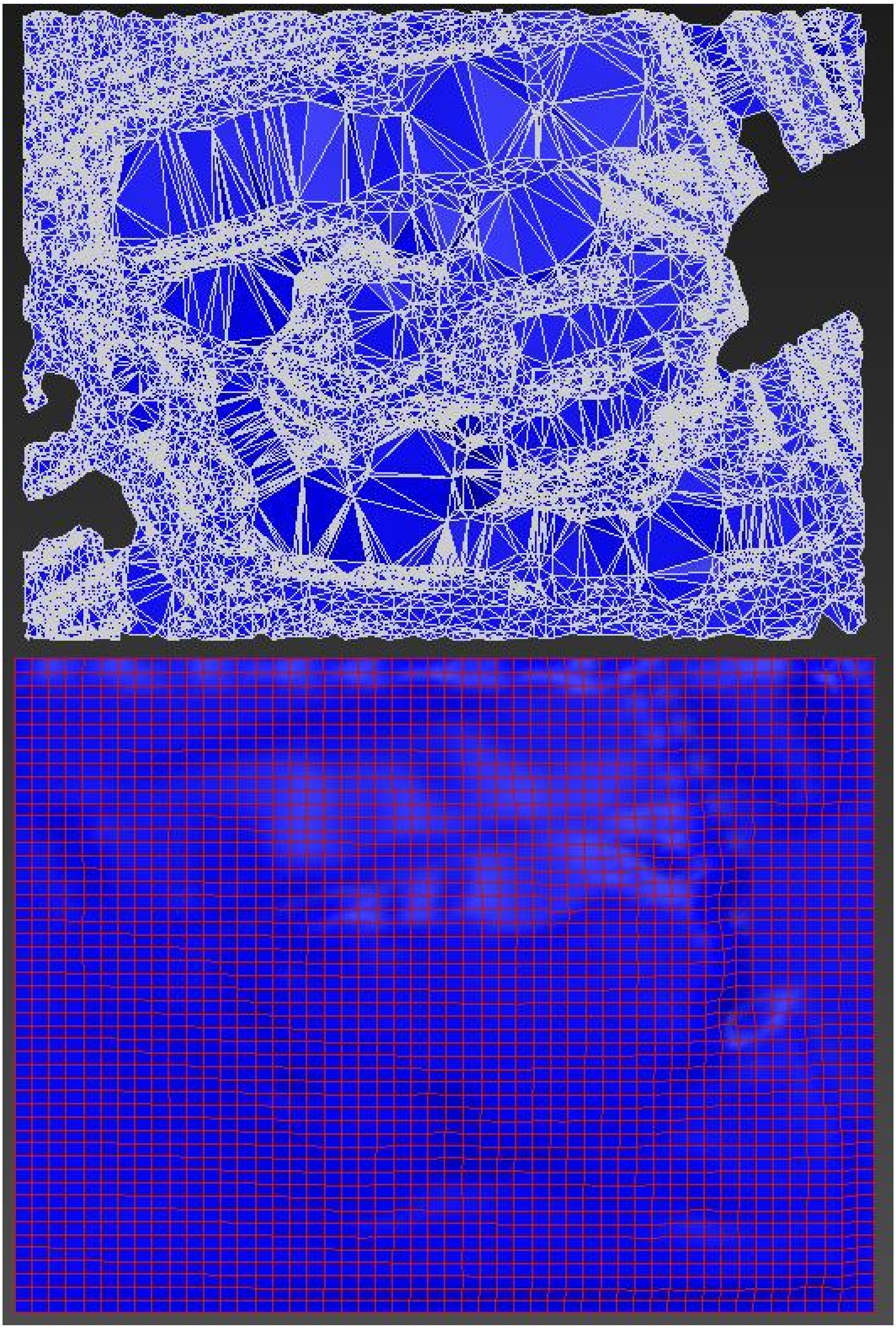

Figure 12: Top view of the initial triangulated mesh (top) and quad-mesh generated using it as the basis (bottom).

5.3 Import to Unity3D

Importing to Unity3D is a straightforward process. However, problems may occur when the data sizes are too large or the data is inconsistent. Unity3D supports 3ds Max's native file format (.max) as a working format but it does not import them straight to the editor. When 3ds Max files are imported to a Unity3D project file, Unity3D converts them to usable data via background FBX conversion. Even if the editor shows the 3ds Max file in the project folder, the actual data is in FBX format. The process for data transfer from 3ds Max to Unity3D is illustrated in the flow chart in Figure 13.

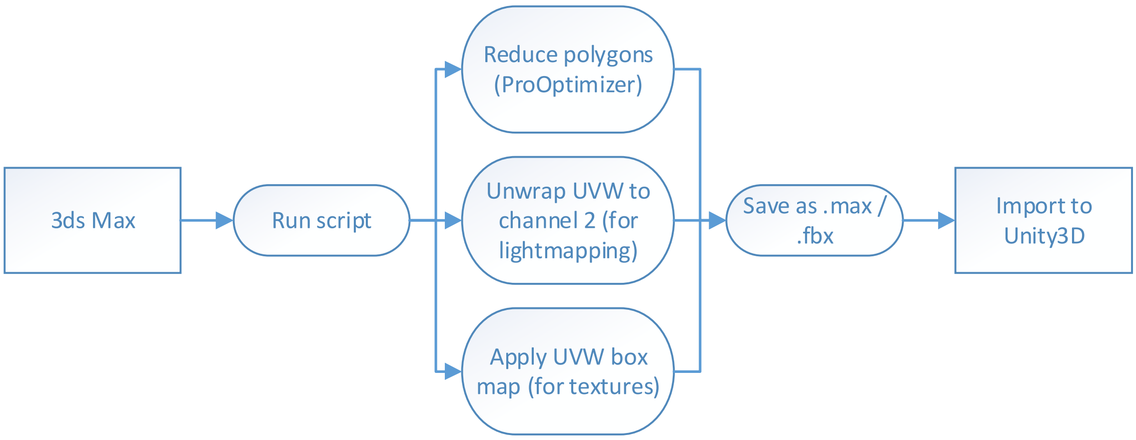

Figure 13: Process of transferring data from 3ds Max to Unity3D

Working with either 3ds Max files or FBX is a double-edged blade. If 3ds Max files are imported to a Unity3D project, the files can be directly opened and edited from the Unity3D editor. However, the file sizes may be larger if the 3ds Max files contain a lot of modifiers that are not collapsed per object. In addition, 3ds Max has to be installed to the client computer for them to be able to work with the 3ds Max files. Unity3D has a three-minute timeout for import of 3ds Max files. It was found via testing that if the file size or the number of individual objects is too large, the import fails due to import timeout. In such scenario, the objects needed to be divided into multiple files or the 3ds Max file needed be exported as an FBX file and imported to Unity3D.

Working with FBX brings advantages in handling the file data since Unity3D does not have timeout for the import. That enables handling of larger files. Editing FBX files is not as straightforward as with 3ds Max. The FBX needs to be manually imported to 3ds Max and exported to the Unity3D project asset folder. If the FBX is overwritten in the asset file instead of reimporting, the set modifications in the Unity3D project persist. This is useful in situations when multiple models are used to produce a virtual model. If the models origins do not match, they need to be oriented manually. If a model is reimported, the orientation needs to be done again.

Materials with texture information for objects in a FBX or a 3ds Max file can be imported to Unity3D without losing the association between the two. Before importing the files to

Unity3D, a folder named "Textures" must be created to the project folder and the texture files imported there prior to FBX or 3ds Max file importation. As the model files are imported to Unity3D, the existence of the materials folder is checked and textured are applied to materials if they are present.

5.4 Demands of virtual models for different scenarios

The usability and GUI of a virtual model are key factors for a user friendly experience. A basic controller setup or operating instructions should always be provided for people using the virtual model in question for the first time. Figure 14 illustrates how different scenarios have effect on the used data and the production of a virtual model. The example scenarios are picked due to their diversity in used data and demands of the model for it to be usable in its respective scenario.

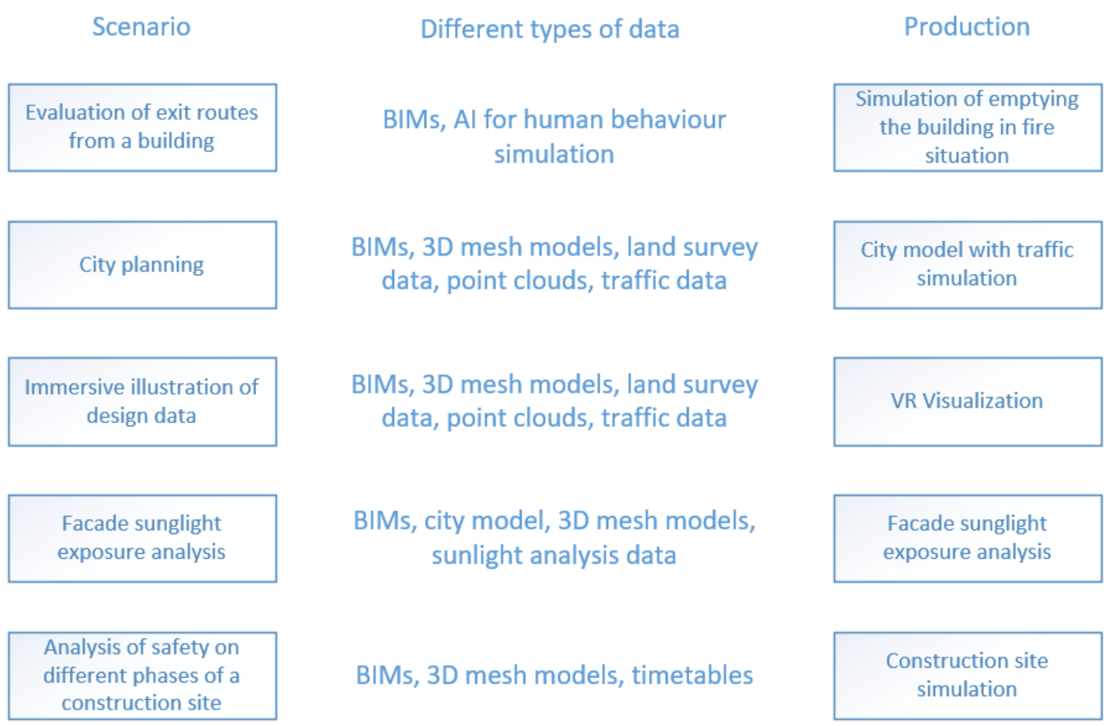

Figure 14: Different use scenarios require different data for a virtual model

In addition to the needed data, the different final products have their own process for development due to different types of data sets they require. Using external databases or calculations as the basis for a virtual model defines more requirements for implementation of the data than presented in this thesis. To prevent the thesis scope from expanding too much, only two case examples of creating an end-product are included.

5.5 Model optimization

Due to excess number of polygons and vertices in BIMs, the virtual models often need to be optimized for a suitable performance to be achieved. Unity3D has different type of performance optimization techniques that can be implemented in the scene design. Some of the optimization can be done beforehand, such as reduction of polygons per object. The following sub-chapters focus on optimization techniques applicable for the Unity3D game engine. Most of them are implemented by hand but the gained advantage in used time compared to enhanced performance is worth the trouble.

5.5.1 Visibility culling

Visibility culling, or occlusion culling, is a technique which detects objects hidden from a virtual camera's FOV, either by lining them outside of the FOV or by occluding them with another object, and rejects the occluded objects from rendering. There are multiple different techniques to apply occlusion culling to a rendering platform through hidden surface removal, all of them resulting in reduced amount of resources needed for rendering a scene. Some of the first occlusion culling techniques is view-frustum culling where the game environment is hierarchically divided and every node outside of the camera's FOV view is rejected along with its descendants. However, even though view-frustum culling and the z-buffer algorithm, which sorts objects per their distance from viewer and z-buffer, are considered to be relatively old, they still act as the basis for different methods of occlusion culling. (Pantazopoulos & Tzafestas, 2002)

The simplest technique for culling objects is the back-face culling, used of then in polygonal environments, such as games or virtual models. If object's polygon normal points away from the viewer, they are not rendered. There is not much need for complex calculations when only polygons' normal directions are monitored and those not visible will be left out, making back-face culling a very light technique. (Pantazopoulos & Tzafestas, 2002)

View-frustum culling leaves objects un-rendered when they are placed outside of the user's view-frustum. Usually, a four-sided pyramid with its apex on the eye position is taken as a view-frustum, with two additional clipping planes in the near side (cutting the apex) and far side (cutting the infinite pyramid). Typically, the operation is performed per object instead of per polygon as per in back-face culling. View-frustum culling is often used with a spatial subdivision scheme such as an octree. Octree is a grid based spatial subdivision structure where an axis-aligned cell binds all scene's objects. The cell is subdivided giving eight sub-cells where the included and intersected objects are stored. The subdivision continues until no object is included in the created sub-cells. Each cell stores a list of included objects. (Pantazopoulos & Tzafestas, 2002) In Unity3D, the pyramid's depth is defined with a clipping plane, making the pyramid finite. Figure 15 shows how frustum and occlusion culling differentiate from each other. View-frustum culling hides object outside of the camera's FOV, occlusion culling hides objects culled by another object in the foreground of the FOV.

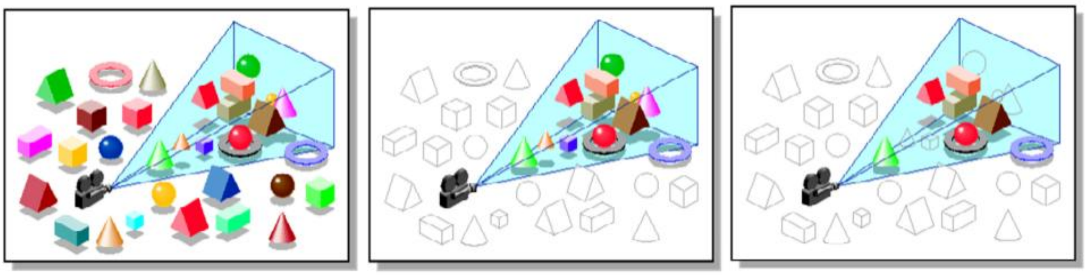

Figure 15: Illustration of culling. No culling (left), frustum culling (middle), occlusion culling (right) (Johanson, 2016)

Comparing view-frustum culling and occlusion culling, more computation needs to be done for effective occlusion culling due to occlusion culling being much more complex process as it requires computing how objects affect each other in a 3D scene. However, through occlusion queries, the actual detection of objects can be done in cohesion between CPU and GPU. The occlusion queries allow an application to render some of the geometry, then query from the GPU if an object is rendered or not. (Johanson, 2016) (Bartz, et al., 1998)

Indoor environments can take advantage on cells-and-portals culling system. It divides the environment to cells (rooms) that are connected with portals (doorways). By determining which objects are in the same cell as the camera or which portals can be seen through, the computation time for deducting which objects are rendered can be streamlined. Figure 16 illustrates the working principle of cells-and-portals culling system. The left picture highlights in green which rooms can be seen to from the user's perspective. On the right, the users FOV is highlighted as green, illustrating how doorways act as portals. (Lowe & Datt, 2005) (Johanson, 2016)

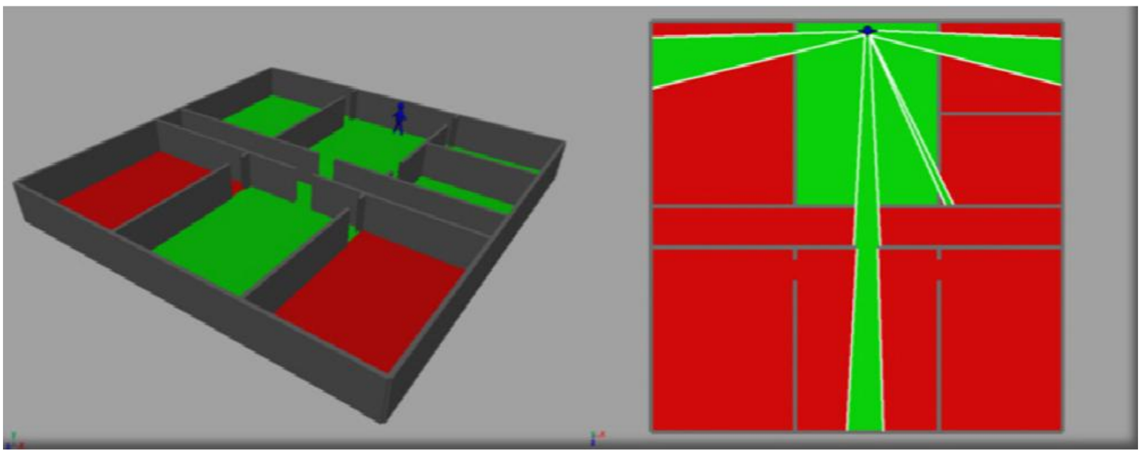

Figure 16: The base principle of cells-and-portals culling (Johanson, 2016)

Cells-and-portals culling system could be effectively utilized in rendering BIMs. By utilizing BIMs meta-data such as IfcSpace (cell) and IfcOpening (portal) properties, a ready-to-go cells-and-portals system can be established. The culling system could be formed while loading the model for presentation without any additionatial calculations or preprocessing. When combined with hardware accelerated occlusion systems (view-frustrum and occlusion queries), visibility of portals can be taken in account when deducting what objects should be occluded. Thus, the amount of computing occlusion queries can be reduced and a more effective way of presenting BIMs is created. (Johansson & Roupé, 2009)

For a virtual model, all of the mentioned culling systems can be applied. View-frustum culling is automatically applied in Unity3D occlusion calculations. Based on object definitions, objects can be set as occludee or occluder static objects. When set to occludee static, objects can be occluded by those set to occluder static. If object is not to either occludee or occlude static, they will be always drawn in the background regardless of being visible, increasing the amount of drawcalls. Only exception to this is using occlusion areas which represent cell divisions for view volumes. The occlusion culling in Unity3D utilizes the cells-and-portals culling system by processing the scene into target volume cells and creating portals for them.

The view volumes can be defined by hand in Unity3D. The view areas are designed for occlusion of moving objects due to them not being able to be defined as static. The areas have a bounding volume that can be accessed through portals defined in occlusion baking. By dividing the scene into smaller cells, the occlusion becomes more precise resulting in better overall performance. The sice of automacially created cells and portals can be defined by giving the Unity3D's occlusion bake parameters of the smallest occluder and smallest hole in the scene.

5.5.2 Level of detail

The general idea behind level of detail optimization is to reduce an object's polygon count as possible in relation to the viewing distance in such fashion that it does not affect the visual quality of the object. This generally means that multiple instances of the same object must be modeled, each containing different number of polygons. The further the object is viewed, the less detailed model is presented which speeds up the rendering process. Figure 17 illustrates how the LOD system works. In the top picture, different level of detail models of a rabbit are presented with different amount of polygons. The lower picture shows how lower detailed model of the same object can be used when viewed further away.

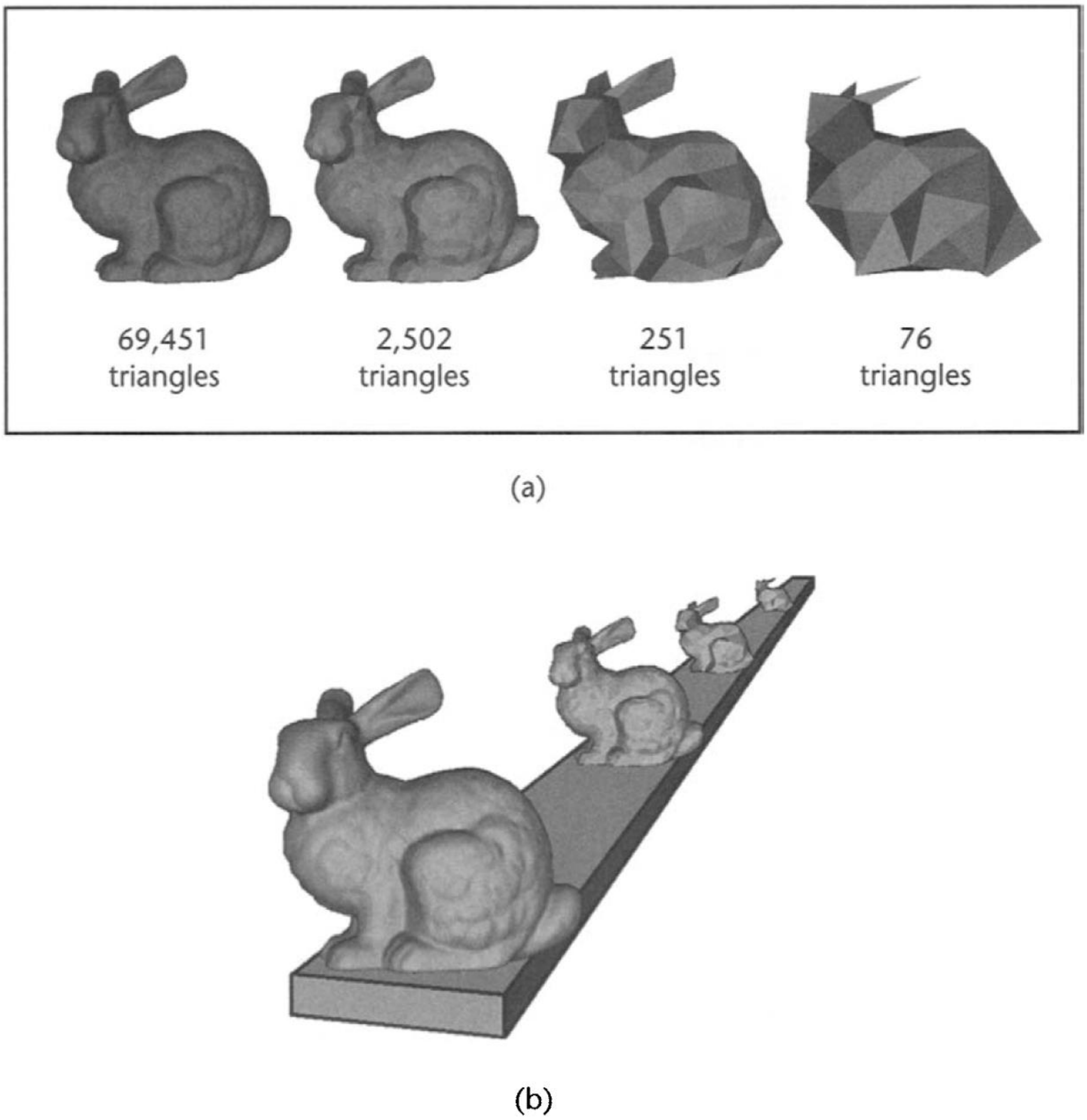

Figure 17: Fundamental concept of LOD. Different versions of the same object (a). The further the object is viewed, less detailed model can be shown (b). (Luebke, et al., 2002)

The approach to LOD can be divided into different versions. A traditional approach, discrete LOD, creates multiple versions of every object with different level of detail during an offline preprocess. At run-time, the appropriate LOD model is chosen to represent the object. In comparison, continuous LOD creates a data structure at run-time instead of the preprocessing stage. A major advantage of this approach is better granularity: due to the level of detail for each object being specified exactly, rather than selected from a few pre-created options, no more polygons than necessary are rendered. Better granularity leads to better use of available resources. Continuous LOD also supports streaming of polygonal models, in which a low-polygon version of a base model is refined dynamically, thus providing progressive rendering and interruptible loading. (Luebke, et al., 2002)

Continuous LOD can be extended to view-dependent LOD by suing view-dependent simplification criteria to dynamically select the most appropriate level of detail for the current view. Nearby portions of an object can be shown at higher resolution than distant portions, or silhouette regions of the object can be shown at higher resolution than interior regions as shown in Figure 18. Polygons are allocated where they are needed within the rendered objects. Difficult visualizations such as scientific visualizations can be interactive rendered with view-depended LOD without manual or extra processing for segmentation. (Luebke, et al., 2002)

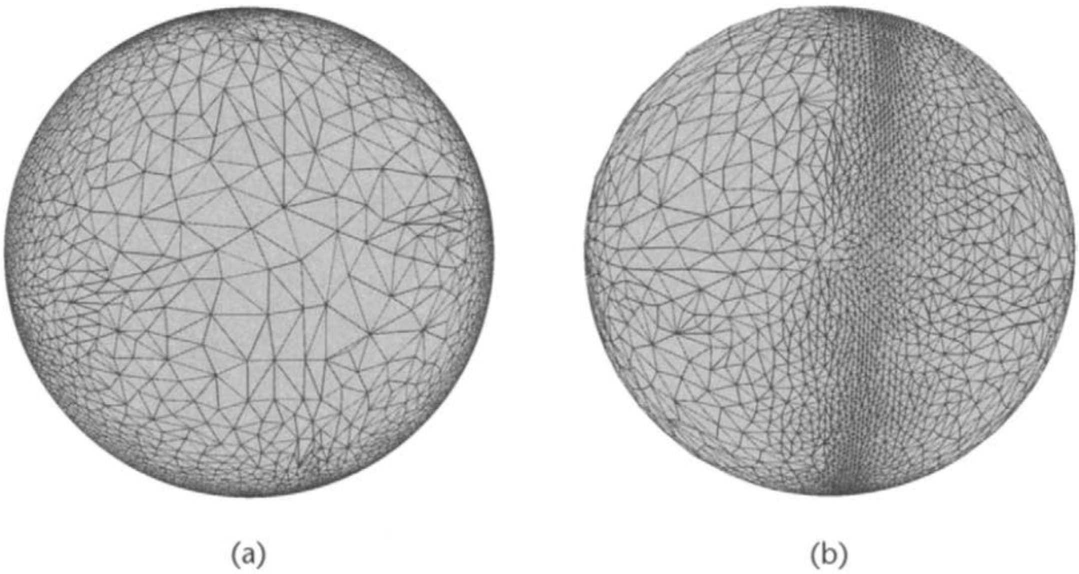

Figure 18: A sphere simplified with silhouette preservation. A head-on view (a) and a side-view (b). (Luebke, et al., 2002)

Although view-depended LOD provide considerable advantages, it also has drawbacks. The extra processing required for evaluating, simplifying, and refining the model at run-time with extra memory required to construct the view-dependent structures has an impact on performance. Often game engines try not to use view-depended rendering unless it is required for large complex objects, such as terrains. While discrete LOD has a time-consuming preprocess, it is simple and works best with low-end graphics hardware. Continuous LOD imposes a cost in processing and memory usage, but in turn provides better granularity which can lead to better fidelity. (Luebke, et al., 2002)

Game engines such as Unity3D provides implementation of discrete LOD systems. While Unity3D lacks the ability to create continuous LODs from one single model, it handles the transition between different discrete LODs smoothly. Handling the transfers between different LODs can be done via the LOD groups in Unity3D editor. Each group consists of one object's different LODs. The LODs are set by hand along with the rule parameters for each LOD's activation. The blending of different LODs can be set to use the fade mode which blends the different LODs seamlessly to ensure a smooth transaction from LOD to another. However, the modeler has to implement the blending of objects by a blend factor, calculated from the object's screen size.

There are several existing LOD schemes for buildings and cities. One of them is defined by CityGML which is an open source data model and exchange format for storing digital 3D models of cities and landscapes (Open Geospatial Consortium, 2017). The CityGML LOD system can be directly implemented as Unity3D LOD due to the LOD gradual improvement can be applied to viewing distance in the same fashion as illustrated in Figure 18. The LODs are defined with a scale from zero to four as follows:

- LOD 0: Regional, landscape
- LOD 1: City, region
- LOD 2: City districts, projects
- LOD 3: Architectural models (outside), landmarks
- LOD 4: Architectural models (interior)

An illustration of different LODs effect on buildings is illustrated in Figure 19. The LOD 0 is left out because it outlines structural 3D geometry.

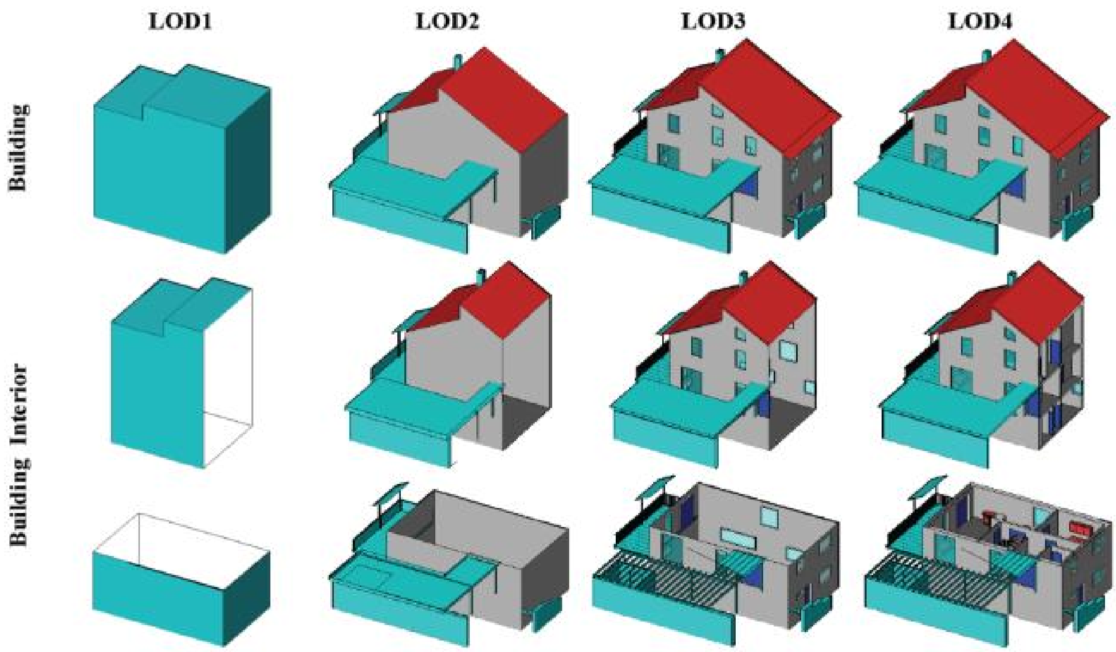

Figure 19: Illustration of CityGML LOD for buildings (Karlsruhe Institute of Technology)

American Institute of Architects (AIA) has defined LODs for individual buildings that could be used with Unity3D. Even though the AIA definition is described as the level of development, the concept is the same as level of detail. The AIA LOD structure consist of five different LODs as follows:

- LOD 100: Overall building massing indicative of area, height, volume, and orientation may be modelled in three dimensions or represented by other data.
- LOD 200: Model elements are modeled as generalized systems or assemblies with approximate quantities, size, shape, location, and orientation.

- LOD 300: Model elements are modeled as specific assemblies accurate in terms of quantity, size, shape, location, and orientation.
- LOD 400: Model elements are modeled as specific assemblies that are accurate in terms of size, shape, location, quantity and orientation with complete fabrication, assembly, and detailing information.
- LOD 500: Model elements are modeled as constructed assemblies actual and accurate in terms of size, shape, location, quantity, and orientation. (American Institute of Architects, 2013) (BIMFreak, 2014)

An illustration of different LODs per the AIA system is shown in Figure 20.

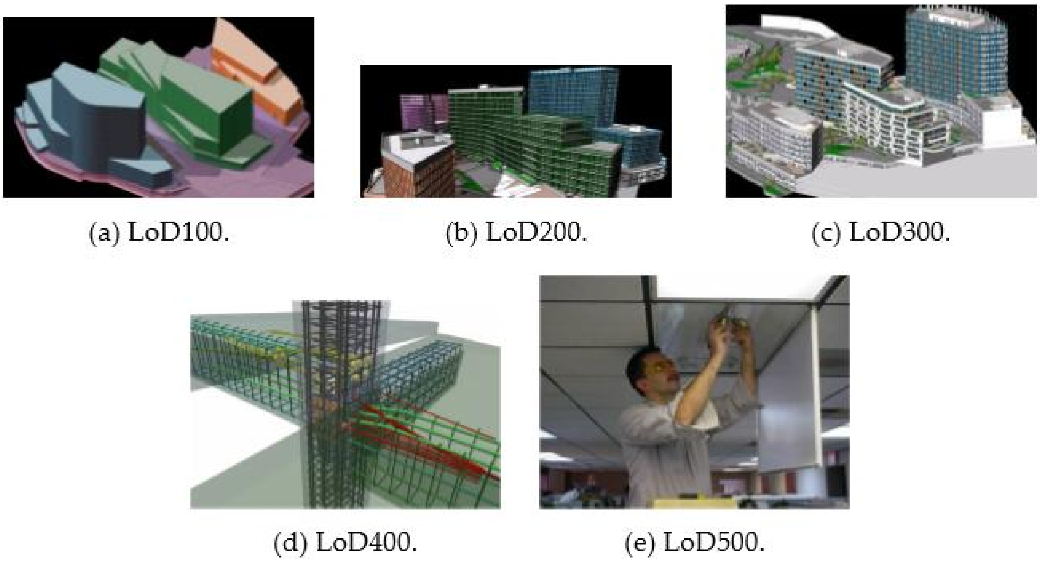

Figure 20: Illustration of the AIA defined LODs (Biljecki, 2013)

5.5.3 Lights

Scene lightning can be modelled in several ways. Unity3D has four types of light: directional light, spot light, point light, and area light. Directional light can be used to simulate the sun, projecting light globally in an angle from an infinite distance. Spot light emits light in an constrained angle, resulting in a cone-shaped region of illumination. It can be used to simulate a flashlight or a similar light source. Point light sends light in all direction from a point in space. Point lights can be used to simulate lamps or other local light sources. Area lights are defined as a rectangle in space. Light is emitted in all directions uniformly across their surface area from one side of the rectangle. (Unity Technologies, 2017)

Unity3D has three types of light calculations: real-time lights, pre-baked lights, and baked lights. Real-time light effects are calculated at runtime. Real-time lightning is generally used for moving objects as the effects of the light they emit needs to be constantly recalculated. The baked lights are used with static light sources that do not move in the scene, such as lamps. By baking the lights, a lightmap is laid on top of the texture map of objects, creating a pre-lit environment which causes less stress for the GPU rendering. Pre-baked lights are a middle ground of real-time and baked lights. Some of the light analysis is precomputed to be called upon at runtime.

The less real-time lights a scene has, the less optimization is needed for the light simulation. When incorporating lightning effects such as global illumination or ambient lightning, the amount of data increases constantly. Global illumination is a system that models how light bounces off from surfaces onto other surfaces, creating indirect light. By utilizing global illumination, the scenes become more realistic as the light acts in a more natural fashion. If the performance of a model does not become compromised, real-time lights are used due to their ease of iteration between different settings. The iteration of lights' settings takes up more time when using baked lights due to the time of calculating the new environment after modifying the light settings.

5.5.4 Texture mapping

As presented in Chapter 2.1, texture mapping is projecting two-dimensional image (a texture) to a three-dimensional object of given topology. Texture mapping adds color, texture or other surface detail like glossiness, reflectivity, or transparency to the object (Slick, 2016). In computer graphics and game development, texture mapping is used to change the appearance of a surface without additional geometrical modeling in example, a straight, smooth wall's surface is made to look as it is laid from bricks and mortar with natural deviation and irregularities in its surface. The end-result is more realistic improving the graphical appearance of an application. (Xu & Chen, 2012)

The number of materials in a scene increases the stress on the GPU. Therefore, the number of materials should be kept as low as possible for best performance. A texture atlas is a constructed material where different objects using different type of textures are incorporated to a single material. By unwrapping the object's UVW map for textures and adjusting the meshes or polygons to use a certain area of the UV coordinate grid, the object's texture can be defined to be incorporated to the same space occupied in a map. Multiple different objects' textures can be inserted in one texture map with matching UVW coordinates to create a texture atlas which holds the information of each texture in the same material.

6. UTILIZATION OF A VIRTUAL MODEL IN CONSTRUCTION PROJECTS

The following chapter contains the empirical trials of virtual model utilization by illustrating how two virtual models were evolved into VR and MR visualizations and how virtual models can be applied in co-creative design. The virtual models that act as the base for visualizations have been produced via the process defined in Chapter 5. The case study in Chapter 6.3 is an example of how a VR visualization can be used as part of co-creative design in practice.

A virtual model is not necessarily a standard in a project and such should not be created without an actual need for it. Innovative culture proceeds the capabilities and versatility of creating virtual models in different companies. However, when projects are considered as a part of a business where excess time consumes money, there might not be time to fine-tune the process of model production. Researching a process may not be a part of project's life cycle. In such case, the model utilization should be evaluated in relation of the spent time and gained utility.

Virtual models can be evolved into various end-user products. The end-user products can be divided into three different main categories: visualizations, simulations, and different type of analysis. Visualizations are representations of geometric information with or without textured surfaces and final products. Simulations are used to estimate different scenarios that deviate from normal use and do not necessarily produce numeric results. Analysis are considered as specific test scenarios that produce numeric results that can be measured and used as a basis for further calculations.

In co-creative design, the presentation of information is a key element to pass information. Information itself is valuable, but a bad presentation could produce misinterpretations of information which would provoke unsatisfactory conversations and a dramatic loss of information. The design of a virtual model's presentation is related directly to its use case. The case studies covered later in this chapter have specific methods and ideologies to present information in a fashion that suits the intended use of them. Understanding what needs to be shown to pass on information is vital for the production of a usable virtual model.

A key element in all virtual models is presentation of information. 3D models that can be used to inspect the design options can be shown as VR, AR or MR applications or normal desktop applications. Simulations can be built on top of the visualizations of different spaces. By providing behavior and physics models to the game engine, facades lightning analysis can be performed by creating a surface matrix on building's facades and capturing how many ray traces are projected on the said matrix. By gathering this information and processing it to a simplified UI, virtual models can provide essential information.

However, the use of virtual models in construction projects can be limited by the innovative capabilities of the participants in the design process. Different type of use scenarios for the model can be created and the same model can serve the project in different ways. The virtual model in the third case example could be altered for different use scenario by adding new and modifying existing functionalities. The building can serve as a visualization platform for further interior design or be used as an iteration tool between different floor plan design options. It is also a room-bounding model of portion of the building. With its current content, it could be used to simulating fire situations in selected rooms when combined with behavior for human or smoke behavior via an AI.

6.1 Trial 1: Producing a VR Visualization for HTC Vive

When creating a virtual model for HTC Vive, a key factor that should be considered is the capability of a desktop computer or a laptop in use. The hardware of used computer defines the amount of optimization required due to the need of a constant ninety frames-per-second frame rate. In this case study the specifications of used computer are as follows:

- GPU: nVidia GeForce GTX 1080
- CPU: Intel Xeon E5-1630 v3
- RAM: 16 GB
- Chipset: Intel C612
- Hard drive: Kingston 120 GB SSD

The visualized project was an office building in Helsinki with rentable facilities for multiple organizations. Initial data was acquired in IFC format from the project's architect, HVAC designer and electrical designer. The data was delivered in three files: first containing the structural and interior design components, second containing all electricity components and the third containing all HVAC components. The initial models are shown in Figures 21, 22 and 23.

Figure 21: Architectural IFC model of the building

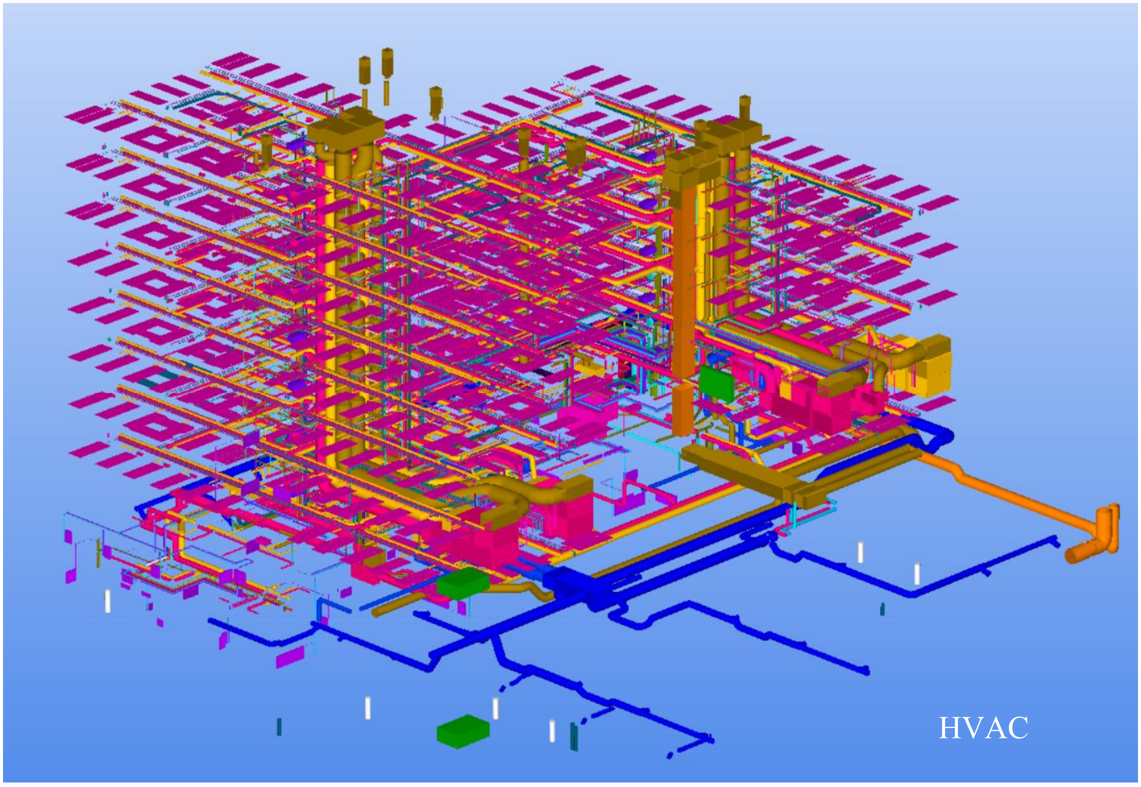

Figure 22: HVAC IFC model of the building

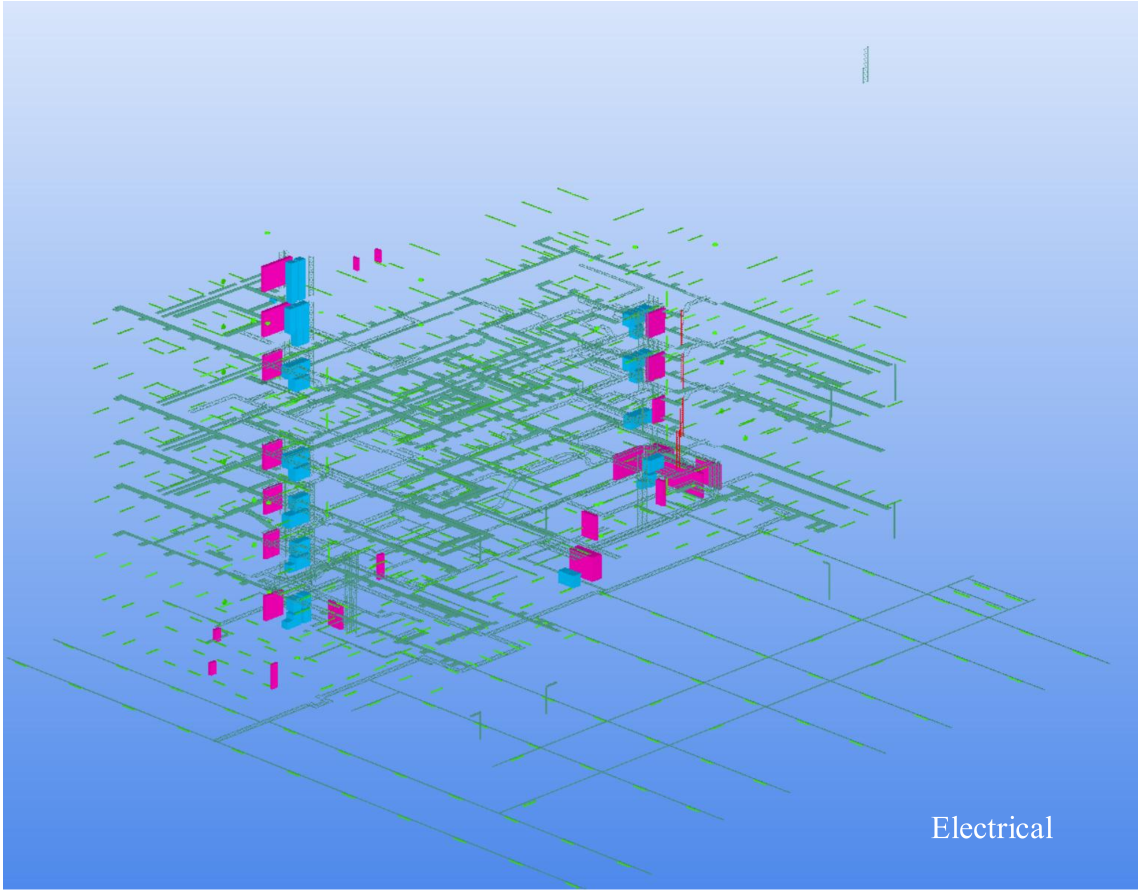

Figure 23: Electrical IFC model of the building

The IFC files were processed in Simplebim 6. After importing the IFC files separately to Simplebim, unnecessary objects were excluded from the model with a predefined template. Exclusion was done by first excluding all objects not belonging to first or second floor and then by excluding the unnecessary objects of the remaining floors such as space objects. The second floor was excluded on every part except the slab which serves as the ceiling of first floor and a few walls that were visible from the lobby of the first floor. In addition, all the objects outside of the building on the ground floor were excluded since the model served as illustration of how the designed floor plan would work. Additional excluding of the model was done by discarding all the objects belonging to spaces or rooms that are not going to be presented in the model.

The remaining objects were exported in IFC format from Simplebim. The files were divided by IfcObjectClass to multiple different IFC files to create easy-to-manage subsets for 3ds Max. Each of the subsets were imported to 3ds Max 2015 with the IfcOpenShell plugin and saved as 3ds Max native files. Because all the models were created using same coordinate system and orientation, models did not need to be oriented in Unity3D.

Most of the objects were used as is and no optimization had to be done to them. However, the furniture of the model was the direst need of modification. Because they were exported form architectural design software, the objects were not suitable for game engine use. The normals of the objects' polygons were inconsistent in their orientation and the objects had an excess number of polygons. Figure 24 illustrates the phases done in optimizing a lamp in the scene. The first image shows the lamp as it was imported from IFC to 3ds Max. In the first image (top left), the dark polygons are having their normal pointing to the lamp's center and the bright polygons' normal are pointing away from the lamp's surface. Second image (top right) shows unified normals where all the normals are oriented similarly. Third image (bottom left) is the lamp after applying polygon reduction. The lamp's polygon count decreased from 3739 to 1218. The number of lamps in the scene was twenty-one, so the total polygon number decrease was 52941. The last image (bottom right) shows the lamp in the virtual model. While the geometry slightly alternates from the start, the lamps are viewed from ground level and usually some distance away. In that aspect, the altered geometry does not have any effect on user's perception of the space.

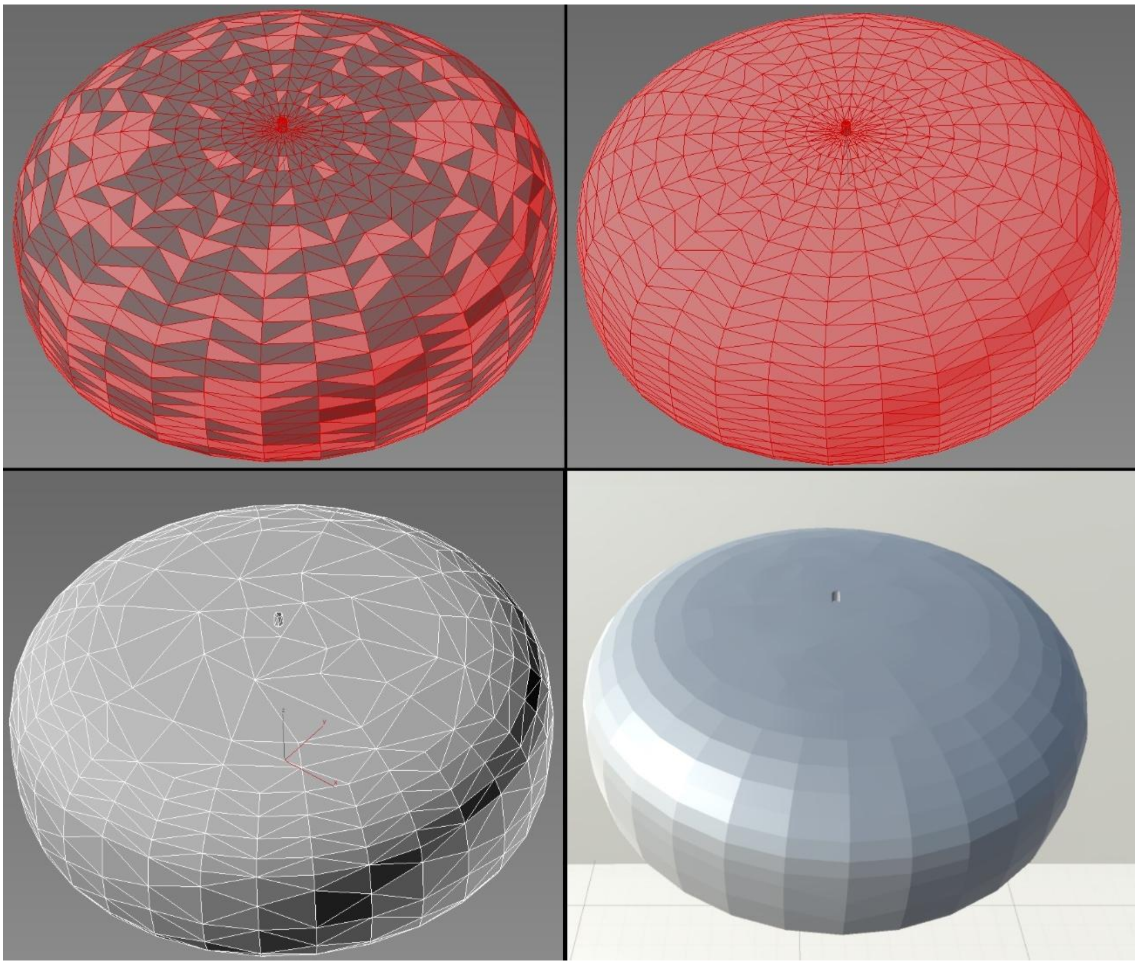

Figure 24: A lamp's polygons' normals before (top left) and after (top right) unifying and the final, optimized lamp model in editing software (bottom left) and in a virtual model (bottom right)

The optimization was done for multiple different object types. It was executed with a single script to a selection group as needed. The used script transformed the objects to editable mesh, unified their normals and applied ProOptimizer with sixty-five percent polygon reduction. After optimizing the objects, UVW Unwrapping was applied and the unwrapped UVW map was saved in channel two for use of lightning calculation in Unity3D.

6.1.1 Setting up the scene in Unity3D

The models were saved as a 3ds Max file and imported to Unity3D. Upon importing, the objects were inspected for reversed polygon normals and fixed when necessary. Since the orientation of the models were matched, combining of the models was accomplished in a matter of minutes. Point lights were assigned to number of locations to produce a more realistic scene where lights and shadows could be presented

The LabRenderer is a shader pack published by Valve. It is available in the Unity Asset Store and it provides the same rendering shaders that were used in Valve's VR game "The Lab". It is easy to implement into a Unity3D project due to implementation menu that comes with the asset. It increases the performance of real-time lights in a scene. However, it limits the maximum number of real-time lights to eighteen, which can produce problems if the scene is large and all lights are calculated in real-time. This can be averted by dividing the structure to multiple scenes in example by dividing all the floors to independent scenes.

After importing, all scene materials were converted to use LabRenderer shaders with the editor extension provided by the asset. All lights needed to be assigned "Valve Realtime Light" script for them to be utilized in rendering. "Valve camera" script was assigned to scene's main camera for the Valve materials to be registered in the main camera's FOV.

The interiors of the scenes were adjusted to match the architect's vision of the spaces. A room card was distributed with the model containing specifications for surface materials and colors as demonstrated in Figure 25. The color scheme was matched with the architect's plans, but textures were not applied.

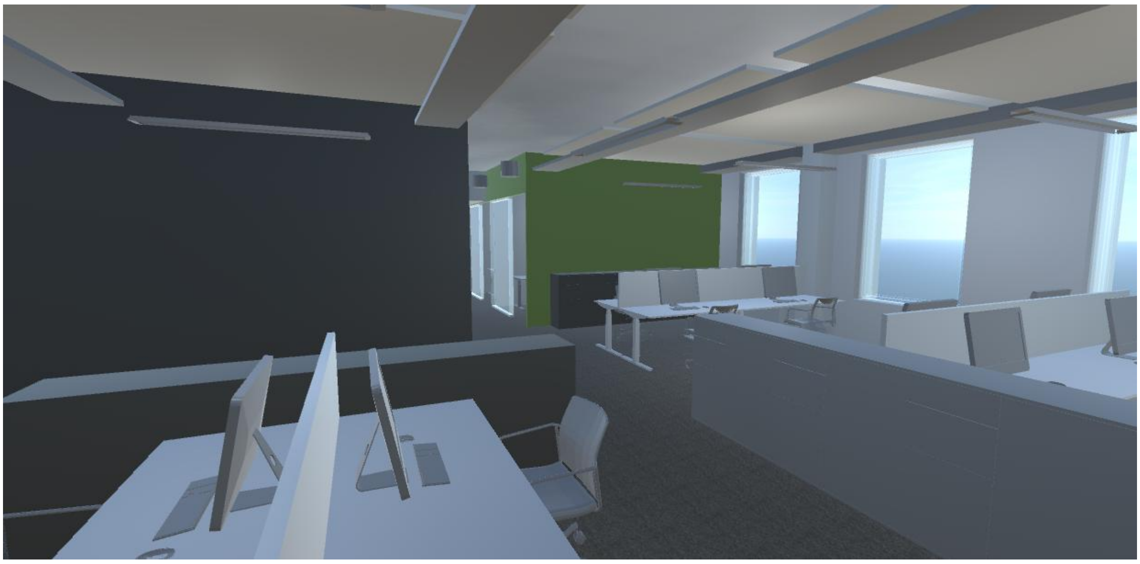

Figure 25: A screenshot from the virtual model

Due to excessive amount of furniture, HVAC and IT components in the virtual model, occlusion culling was a key aspect in optimizing the model. By reducing the number of draw calls by occluding objects, the performance requirements are decreased. Every object in the scene was set to occludee static so they would be occluded by objects set to occluder static. All walls and slabs were set as occluder static. The occlusion parameters were set to that all the walls occlude objects behind them. Figure 26 illustrates how objects outside the camera's FOV, or behind occluder objects, are not drawn. The camera is positioned identically in both pictures.

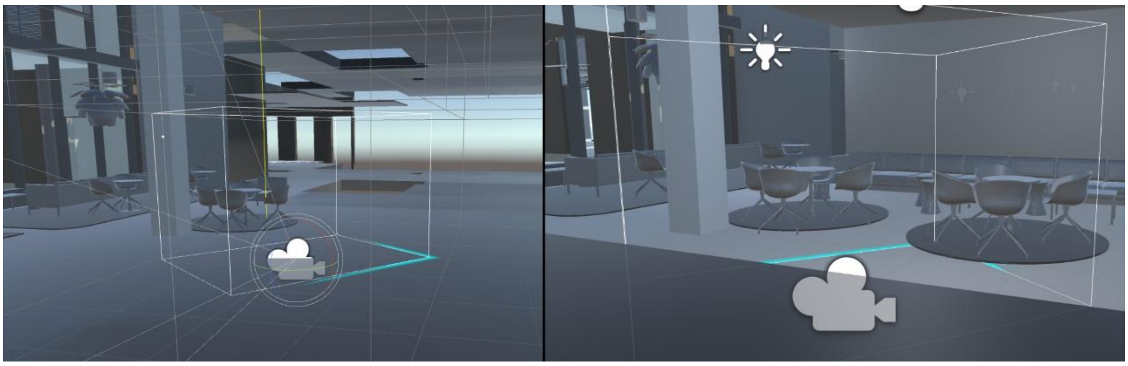

Figure 26: Illustration of occlusion culling's effect

6.1.2 Functionality in VR

The used assets for creating functionality in the scene were SteamVR and Virtual Reality ToolKit (VRTK). Both are available for use in the Unity Asset Store. SteamVR handles the communication of the HMD and controllers with Unity3D. VRTK is a free-to-use toolkit containing multiple scripts and prefabs for implementation of different interactivity for VR projects. The SteamVR application connection to virtual model is done by adding a CameraRig prefab to the scene. The CameraRig contains scripts and models to incorporate HTC Vive's headset and controller tracking to the virtual model. After the release of Unity3D 5.6, a Steam VR_Update_Poses script needs to be added to the scene's main camera for the controllers to be active in the model.

To enable VRTK, a VRTK software development kit (SDK) manager was added to the scene. This was done by creating an empty game object and attaching the script to it. The script was configured for HTC Vive and SteamVR, which can be done by a selection menu and auto populating the objects with a predefined procedure in VRTK documentation. Under the VRTK SDK Manager, three independent game objects were created to handle the model's interactive ensembles: objects for left controller's interactivity, right controller's interactivity, and the needed scripts for the HTC Vive PlayArea, which is defined when the HTC Vive is setup for the first time.

The interactivity was designed so that user had to focus on using only one of controllers to be able to explore the virtual world. Both controllers were assigned similar scripts so that either one of them could be used. The controllers had four different functionalities in the model regardless of the scene: locomotion utility (teleportation or walking), resetting the scene in case of user getting lost or into an area not designed to be accessed, switching scenes, and activating a laser pointer for catching the attention of other viewers by pointing to some specific object in the model. The Vive controller buttons are illustrated in the Figure 27. The controller actions were applied to the buttons as following:

- Trackpad: Press to activate teleport Bezier beam, release to teleport to cursor location
- Trigger: Activate laser pointer
- Menu button: Reset the scene
- Grip button: Switch scene

Tooltips were attached to the corresponding buttons in the virtual model so that users can check the controls without removing the HMD which would interrupt the VR experience.

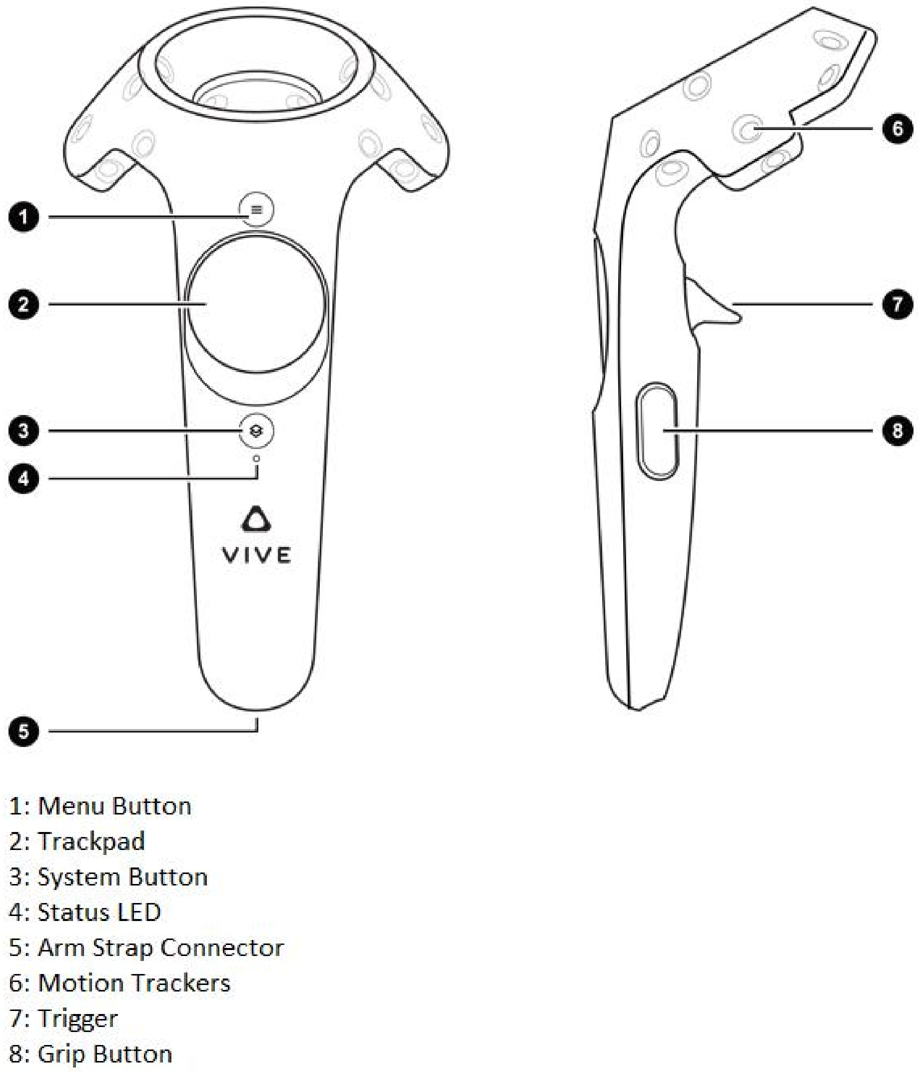

Figure 27: HTC Vive's controller (Unity Technologies, 2017)

The scene utilized two different pointers to be used for either teleportation or a straight pointer. Both used VRTK_Pointer script as their base and a renderer script was added to create intended type of pointer. The pointer script could be assigned to any of the buttons in the Vive controller. Two instances of the same script were assigned to controller, each having their own button for activation. A Bezier renderer was added to the trackpad's pointer script to create a Bezier curve for pointing the location to be teleported to. A simple pointer renderer was attached to trigger button for creating a forward laser beam. Teleportation script was attached to the Vive's play area for enabling the teleportation functionality. Both controllers were assigned scripts to capture different events from pressing and releasing different buttons in controllers.

6.1.3 Challenges during the project

The project itself was a streamlined process but some aspects of the project took longer than estimated. Filtering out the unneeded components and deciding what information is relevant for the model, was one of the challenging tasks. To be able to discard objects in the model, it had to be made sure that they would not be visible from any angle where the user could move. After deciding the layout of the movement area in the model, component filtration was done in Simplebim with ease.

Due to ArchiCAD not being designed for producing objects to be imported to game engines, the number of polygons per object was usually too big. Reductions were made, but it had to be iterated by trial and error, what percentage of the polygons would be removable without compromising the object's geometry. After multiple iterations, a constant of forty percent was applied in the process with satisfactory results. For the lamps in the scene, sixty-five percent was deemed satisfactory. The number of polygons per object was reduced greatly, but the visual appearance of the models did not change too much for the models to look disfigured.

Creating believable lightning to complement the design solution is a great undertaking. A good lighting setup does not attract any attention but when done poorly, it distracts the user and drives focus away from assessing the design. To be able to create suitable lightning conditions and reduce the time of baking the lights, TheLabRenderer's real-time lightning script was exemplary. It allowed the use of real-time lightning, which is faster to setup than baked lightning, to be used without decreasing the FPS rate.

6.2 Trial 2: MR Visualization for Microsoft HoloLens

Building a visualization for HoloLens differs from a VR application significantly. The hardware requirements for model optimization are higher, and even though HoloLens packs strong hardware, the model can easily be too complex for the visualization to run smoothly.

A HoloLens visualization is an AR application or MR application, depending how well it incorporates spatial mapping and interactions with the real world. Medisiina-D, a section of the Turku University Hospital (TUH) was used as the project for HoloLens visualization. The project has two scenes: an indoor scene with four rooms and a corridor and an outdoor scene with a general plan of the surrounding environment as shown in Figure 28. The outdoor scene contains a quad-mesh terrain generated from point cloud and the façade of the target building and surrounding buildings. Neither of the scenes are texturized. Lightmaps are applied to objects in the indoor scene and the lightning is baked.

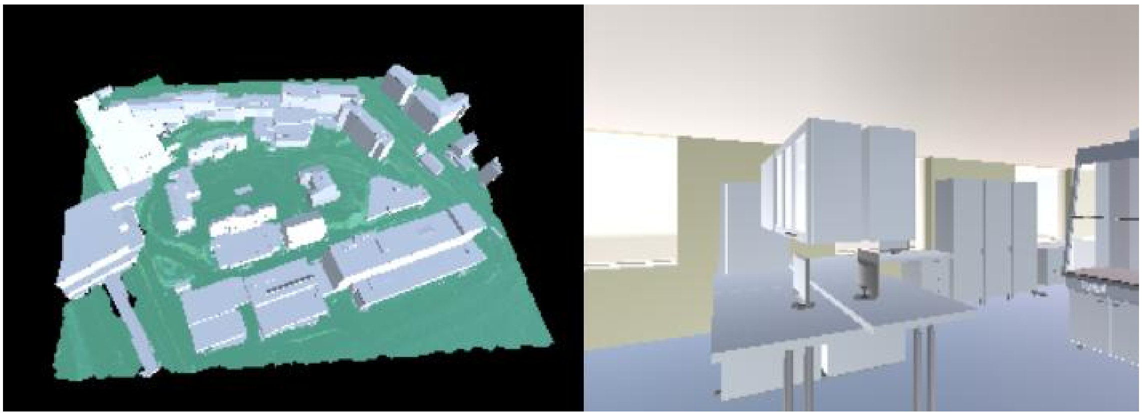

Figure 28: Screen captures of the two scenes in the visualization. On the left, outdoor scene of TUH area and on the right indoor scene of Medisiina-D

Transition between scenes was executed with a panel menu which is movable and can be docked to real-world objects, such as walls. In the outdoor scene, the panel menu contains functions for switching scenes, moving, resizing, and rotating the model scene. In the indoor scene, the toolset contained the options of switching back to outdoor scene and measuring distance between objects. The graphical interface of the menu is simplified as shown in Figure 29 providing an effortless user experience. The menu could be more polished but it was not relevant to the case study due to the case being an example of creating a simple interactive demonstration.

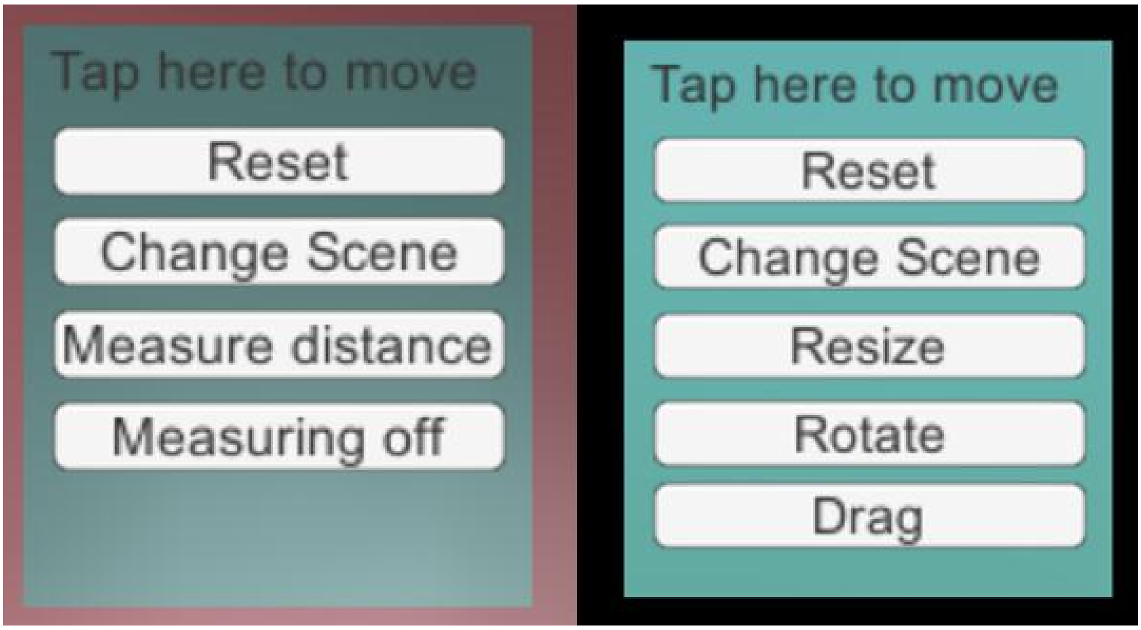

Figure 29: Illustration of used panel menu in both scenes. On the left, indoor scene panel menu, on the right, the outdoor scene panel menu

The outdoor scene's model of Medisiina-D which is more detailed than the surrounding buildings was optimized to reduce the number of polygons to draw in the model to achieve a better FPS rate. At the start, the building consisted of 410 909 polygons and 190 900 vertices. The number was reduced by excluding irrelevant building parts (parts inside the façade) and optimizing the geometry on the parts left. At the end of optimizing the model, polygon count was 150 200 and the number of vertices was 69 157. The terrain was optimized by applying a quad-mesh over the triangulated surface. The quad-mesh smoothens the hard edges of a triangulated mesh giving a more realistic appearance for it and it reduces the number of polygons in the model. The polygon count of the terrain was reduced from 27 633 to 2 600 polygons. The optimization of the scale model of the building could have been taken further but this was sufficient to experience a suitable holographic experience.

6.2.1 Applying interactivity to the project

After creating the 3D model, HoloLens functionalities need to be applied and configured in the Unity3D project. HoloToolkit is a free-to-use GitHub repository, which provides tools for basic HoloLens applications and usage for Unity3D based projects. Typical HoloLens gestures gaze, bloom, select and voice commands can be applied to the virtual model with HoloToolkit. Within the toolkit, there is a shader pack containing shaders for optimizing HoloLens' performance and different scripts for applying interactions and event handling.

The HoloLens has two ways to receive input commands from the user: gestures and vocal commands. Voice input is self-explanatory; user gives commands by speaking. However, the physical gestures are the main method of interacting with HoloLens content. There are three basic gestures: ready, select and bloom. Ready command is given by holding a hand in HoloLens' field of view (FOV), fist clenched and with the back of user's hand, raising index finger to point upwards. The reception of ready command is indicated with a round cursor. Selection or air-tap is done by first performing the ready command, then in a rapid motion, tapping the air. Bloom gesture is done by turning the user's palm to face upwards, all fingers brought to tight nit and then expanding the fingers, as in a fashion of a blooming flower. Selection and bloom gestures are illustrated in Figure 30.

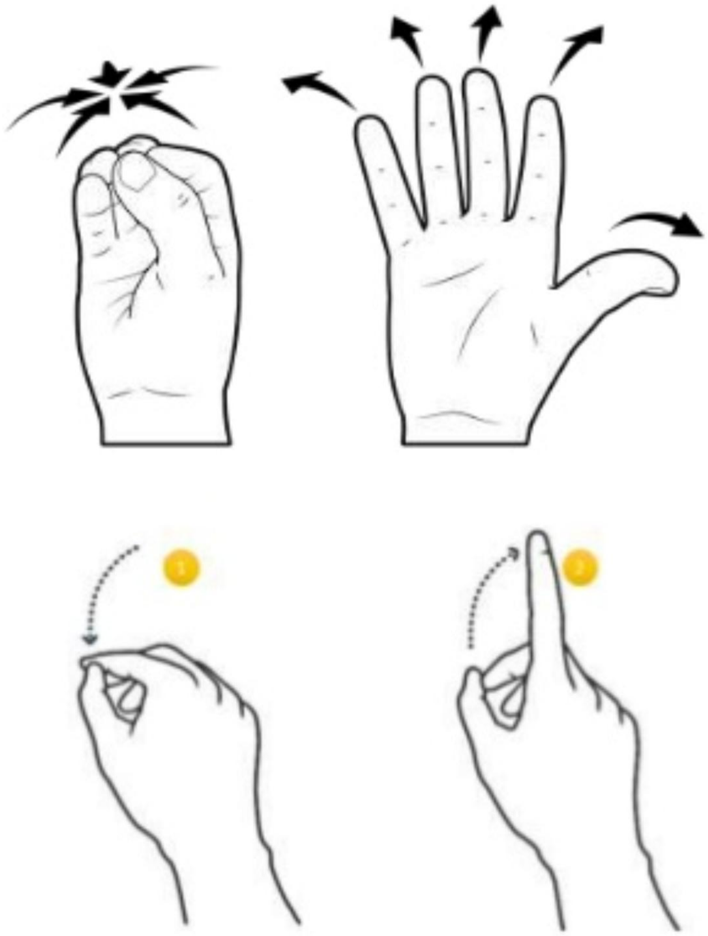

Figure 30: Illustration of bloom and select gestures (Meijers, 2017)

The first step in building a visualization for HoloLens is setting up the scene. The scene's main camera needs to be set at origin (0, 0, 0,) with clear flags set to solid color, background as black (RGBA 0,0,0,0) which renders as transparent in HoloLens. Culling mask set for the layers of the objects which are to be visualized in the MR application. Camera projection should be set to perspective and near clipping plane to 0.85 as shown in Figure 31. All the settings are taken from Microsoft's suggestions for HoloLens scenes in Unity3D. While the suggested values are proven to be efficient, through testing it was found that a clipping plane of 0.5 is more suitable for a better user experience. (Microsoft, 2016)

Figure 31: Recommended camera settings in Unity3D editor

The basic functionalities with HoloLens consists of two components: gazing an object, receiving the event of gazing an object and the actual interaction with the object. For instance, moving an object is done by gazing to it and performing selection gesture. The application registers the object which is gazed and then performs the selected interaction, for example dragging and moving an object across the room. Each of the interactive objects needs to be assigned with scripts to invoke the wanted behavior. All materials are set to use HoloToolkit/StandardFast shader which is included in the toolkit.

Different managers, game objects that manage specific scripts or functionalities, were implemented to the design. Input methods are set to the project with InputManager component which contains the functionalities for gazing, identification of gestures and script architecture for detection of the events of mentioned functionalities. SpatialMapping manager allows the project to visualize and access spatial mapping data on the HoloLens. SceneManager handles transitions between different scenes by loading a new scene on command.

The HoloToolkit has different methods for moving objects in a scene. “Hand draggable” script enables user to perform an air tap on an object and drag it across the scene. Another placement method is using a script named TapToPlace which when enabled, draws the spatial mesh from HoloLens and uses its information for object placement. For this project, another TapToPlace script was created for moving the menu panel. Script 3 shows the changes made in the script’s code.

```
Code before:

gameObject.transform.position = hitInfo.point;
gameObject.transform.rotation = toQuat;

Code after:

gameObject.transform.position = hitInfo.point;
this.transform.rotation = Quaternion.FromToRotation(Vector3.back, hitInfo.nor-
mal);
```

Script 3: Changes in the TapToPlace script's code to have proper functionality for panel menus spatial recognition

The small alteration changes the behavior of the code to rotate the menu to match the orientation of the surface it is placed to while keeping its buttons facing outward of the surface. In this way, the menus will not be embedded accidentally to a wall or table in real-world space.

6.2.2 Building the final solution

The application was built with the settings illustrated in Figure 32. In addition, different HoloLens capabilities had to be enabled in the player settings under Windows Store. The required capabilities for this project were Location and SpatialPerception. The virtual reality SDK, Windows Holographic had to be enabled. Used version of the Universal Windows Phone SDK (UWP SDK) was 10.0.15063.0

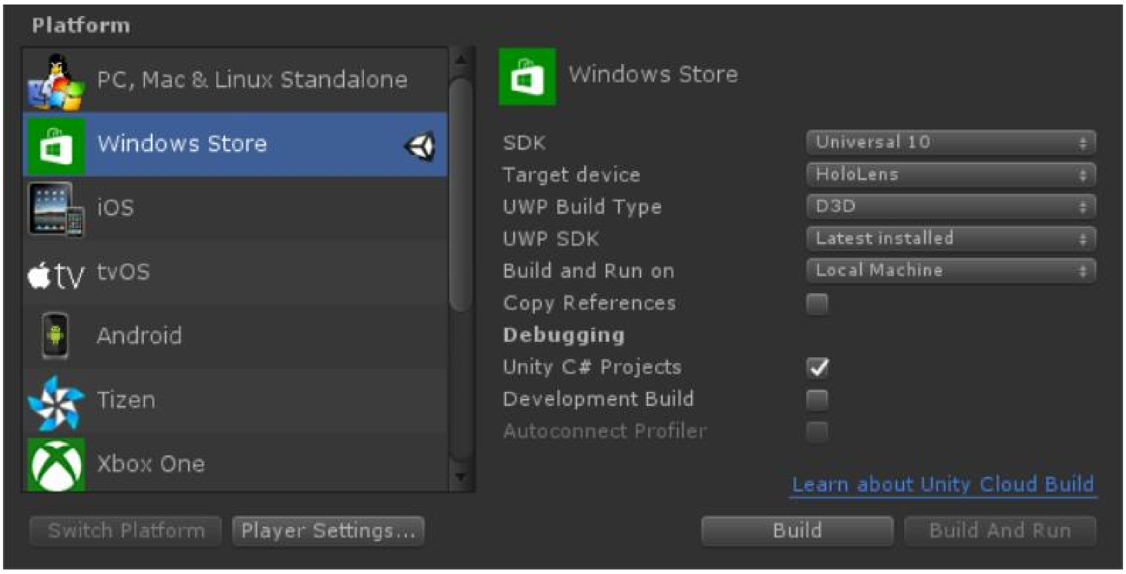

Figure 32: Unity3D editor's build settings for HoloLens application

After building the solution in UWP SDK format, it had to be imported to Microsoft Visual Studio for deployment to the device. Used HoloLens needs to be paired with the computer to enable software deployment to the device storage. The pairing is done by enabling developer mode in both Windows 10 and the HoloLens and deploying the application either via USB cable or remote accessing while being in the same local area network (LAN). In Visual Studio, the deployment was done with settings for a 32-bit operating system (HoloLens has x86 architecture) and the build option set as a release build. It is possible to start the application in debug mode when USB cable is connected, but it is not necessary.

When the application has been deployed to HoloLens, it can be started by opening the menu with bloom gesture and selecting the application. After docking the application window to somewhere in the surrounding environment, HoloLens loads the application and the immersive experience can start.

6.2.3 Challenges during the project

While being rewarding, this project had multiple different problems that had to be solved accordingly. First was the issue of frame rate. The model had to be optimized thoroughly to achieve a reasonable frame rate for the MR experience. If the frame rate drops too low, it negatively affects the user experience of the application and does not serve its purpose of being immersive.

The design process with HoloLens would be have been more fluent if Holographic Remoting Player (HRP) was applied to the process. With HRP enabled, different iterations of the MR application could be tested with the Unity3D editor’s play mode. Unity3D 5.6 can stream an application via wi-fi to HoloLens for faster testing process. Unfortunately, at the time of writing this thesis, a suitable closed wi-fi for using HRP was not available at the location so the testing was done by building an application and importing it manually to HoloLens. The HoloToolkit provides InputManager object with functionality to simulate HoloLens actions in the Unity3D editor’s play mode. This was used to roughly estimate the functionalities before building a solution. The user experience in simulated environment does not imitate a real experience of using HoloLens but it is suitable for estimating success of the functionalities input

Adding different kinds of interactivity proved to be a challenging task. Some minor adjustments had to be made to different C# scripts in order for the out-of-box functionalities to work as intended in a custom project. The TapToPlace script which incorporates spatial mapping information to object placement set some requirements for modeling coordinates. While it performed as it was supposed to, when the tap event was initiated, it often rotated the models out of place and out of the gazing point. In the script, rotation was set to objects by the rotation of HoloLensCamera game object, which holds the main camera in the scene. So, for the objects to be rotated to user, their rotation had to be relative to the HoloLensCamera. With HoloLensCamera’s all rotations set to zero, all the game objects had to match those accordingly. Moreover, the gaze distance of the script was reduced for a smoother user experience

Because thesis writer did not have any prior experience with creating menu items for MR applications, the modeling and combining of different tasks to menu buttons was challenging. After grasping the basic concept, it was easier to implement different versatile functions to buttons and, in general to modify the menu design.

6.3 Case study: Virtual model as a part of co-creative design

Virtual models can be applied in co-creative design process in number of ways. It can act as supportive part of the decision making, as a collaboration tool or as an illustrative tool for design plans. Different types of analysis can be performed based on a virtual model with visual feedback from the model or actual data gathered with project defined algorithms in simulations. The most important part of using a virtual model in co-creative design process is the virtual model's versatility. While it can provide the user direct feedback from a visualization, it can also give indirect feedback to the model operator spectating how different users can grasp the concept of using the model as the operator or designer had defined.

When utilized properly, a virtual model can serve as an essential part of participatory design which is described in Chapter 2.3. By immersing the user in a virtual model through VR HMD, the user can gather information from the model by experiencing the future space or site in design. It gives all the parties valuable information: for the user, a more straightforward way to understand the design options, for the designer, a feedback channel to gain actual user experiences how their decisions have effect on the space. Construction managers, architects and future users can use virtual models as a platform for sparring conversation and providing focus to issues on hand.

As a case study for using a virtual model, VR visualization was created of soon-to-be renovated part of University of Eastern Finland's (UEF) facilities for teaching students in biomedical program. The virtual model was used as a part of a workshop which was organized by a professional facilitator whom had extensive practical and research experience in co-creative design. The workshop incorporated different parties of the construction project: students, teaching staff and representatives of the organization owning the building.

Figure 33 illustrates the floor plan of the area. Four different environments are marked with colors pink, light yellow, blue, and light lime. The corridor connecting three of the spaces is marked as green. Blue area is a quiet space for independent studying. The light yellow is a combined lounge type break room and a group study space. The two tables on the left side of the space are designed for two small groups to explore their ideas in a lively environment and the right side of the space is the mentioned break room. Pink area consists of two individual, multifunctional study spaces, each serving a group the size of seventy students. The study spaces are divided with full-size showcases creating a fillable glass wall between the spaces. The light lime area in the bottom left corner of the floor plan consist of two twenty-person group teaching rooms that provide more privacy for example anatomic study sessions.

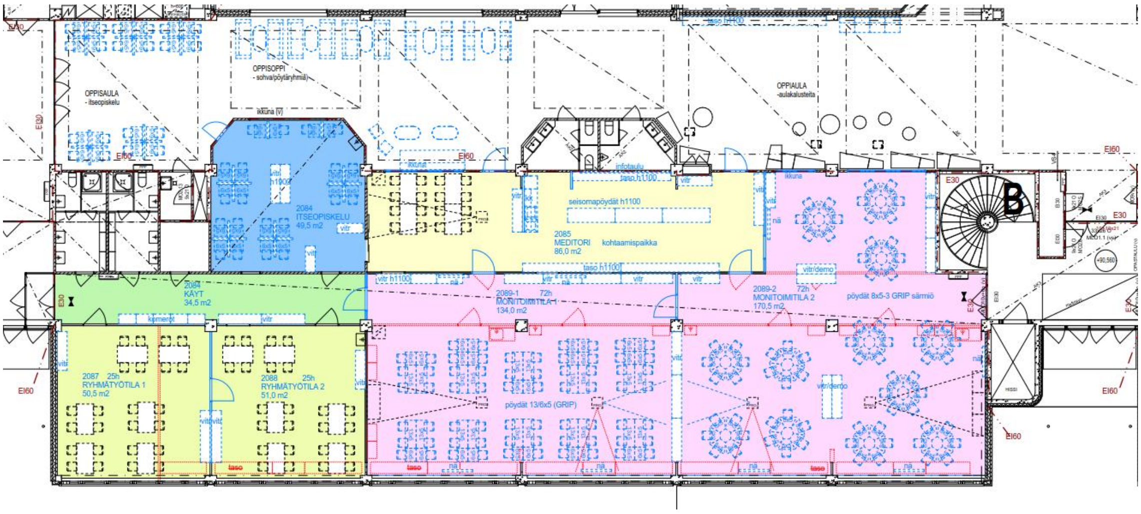

Figure 33: Floor plan of the area used in the virtual model (QVIM Architects, 2017)

The workshop was executed as charrette and its purpose was to examine the upcoming changes in the functionality of four different spaces in three different use scenarios. The goal was to modify the spaces to serve a better purpose as teaching or self-study environments. The charrette began with an orientation session which activated the participants to stir their minds for the following designing phase. First the consensus of last charrette was reviewed and the new planar design was presented. The design solution also acted as a basis for the charrette.

After orientating participants to the charrette, they were divided into three different groups and each group was given a use scenario. The groups were instructed to keep in mind how the spaces would serve their given use scenario and what kind of benefits or problems the design solution would have. The use scenarios were following: studying independently or in a small group of students, a teacher-guided group study and a traditional class room study such as lecturing.

Each group took part in three different control points, each having a unique way of presenting information for the participants as well as unique aspects to consider. The control points were:

- architect's tour around the space where he showed what parts will be modified and described how they will be modified
- iterating different floor plan solutions and their effect on cost estimation with a 2D floor plan with movable cardboard walls
- a VR visualization for HTC Vive, created from the virtual model, where the usability, sight lines and feel in different areas, space sizes and overall functionalities of the post renovated situation were assessed

The groups had fifteen minutes in each of the control points. Each control point's organizer stirred the conversation between participants to get them engaged in bringing their ideas to the design and bring forth the concerns if there were any. The control points had few pre-design topics to guide the conversation. The VR control point had following topics to discuss:

- What are the different routes to arrive in the spaces? Are some of the spaces too far to access?
- Are the sizes of the spaces suitable for the functionalities of each space? Are the spaces too large or too small?
- How are the sight lines and views in different spaces? Where should the available monitors and projectors be situated to serve a purpose? What kind of effect does columns and different structural blockages have on the space?
- What spaces benefit from big windows and natural light? In what kind of scenarios would the windows and see-through doors have uncomfortable effects?

The virtual model based VR visualization was used as a part of the third control point mentioned above. Figure 34 presents a screenshot from the virtual model. While one of the participants examined the space in VR, others saw what he saw through additional screen. The virtual model acted as a support for reviewing the design solution and the participants felt it was a great way to examine all the different areas and their access routes. The participants learned the navigation in the virtual model within a minute, even though they did not have any prior experience of using a HTC Vive. Due to the virtual model's usability and combined information from 2D floor plan, conversation soon arose between participants and they used the model as a visual aid to examine different parts of the spaces.

Figure 34: Two adjoining classrooms in the virtual model

During the session, feedback was gathered from the participants by observing the participants and by engaging them with questions of the usability of the virtual model. In addition to the verbal feedback, each group wrote down their answers to the questions presented earlier. The participants felt that it was a great way to illustrate the design solution and to fully understand the scale and visual lines of different spaces. For the group study spaces the virtual model was an essential help to observe different problems in given scenarios and it made the participants to challenge ideas and the design solution.

After all the participants visited all control points, they were given task to gather the daily routines of each of the spaces, what supports the functions of the spaces and what needs to be done to for them to get the needed functionality. Each group presented their suggestions and concerns to others and a consensus was formed after discussion. The consensus was brought to attention of the architect to act as the basis for refining the design solution further.

The virtual model was based on architect's design and was modelled in one-to-one scale of the area. Model was lightly textured (only walls had texturing) and the color scheme was applied per the architect's design. The model was optimized for a laptop used commonly in structural design. Optimization was done via baked lightning, occlusion culling, and simplified 3D objects. The virtual model was created from an ArchiCAD model with the process as described in Chapter 5. The greatest challenge in production of the virtual model and VR-visualization was optimizing it to the given laptop so that the VR visualization would run with a steady ninety frames-per-second frame rate. Without acceptable frame rate, the VR visualization would induce motion sickness to unexperienced users and they would view the use of the visualization uncomfortable.

It was found that a charrette is great way to use a virtual model in co-creative design. In this use-case it would not have been great value if used without the charrette, since the charrette had the participants in right state of mind to assess the design plan. The virtual model acted as a support for understanding the design solution and its impact on the different spaces and as a tool to provoke conversation. However, the architect's design did not incorporate multiple different design options that could have been shown to the users. Having multiple different design options would have provoked more conversation and drive the focus better to discuss how different changes would have effect on the usability of the faculties in question.

Depending on the design phase, a multi-optional virtual model could serve a different purpose, depending on what changes in the design. For example, a virtual model could present different floor plan and interior design options. Via different options, the participants could get more stimuli to encourage bringing different ideas to the conversation.

7. DISCUSSION

Based on this research, the production of a virtual models is a complex, multiphase process. Software used in construction industry has been developed to serve the needs of different parties in construction projects. Due to that, the software interoperability with software outside of the industry has not been optimized. Therefore, the amount of work that needs to be put to create a process for transferring data from structural orientation to a game engine, exceeded expectations.

The production process of virtual models is relatively the same for infrastructural and building construction sites. However, they differentiate in the process of evolving the model into an end-product. Due to infrastructural models consisting of wider and more open areas than a building, the model cannot gain advantage from occlusion culling in the same fashion as a virtual model of a building. The optimization can be done more efficiently via different LODs of the game objects in an infrastructural virtual model.

However, the construction industry has awoken during the last year. Different plug-ins from independent developers or software distributors have come to the market. Often the plug-in focuses on porting a design directly into a visualization that is based on game engine technology. So far, no software distributors have provided a solution that could export the design material directly to a game engine allowing the producer of the virtual model to decide what kind of purpose the virtual model would then serve.

Virtual models can be applied into co-creative design as a basis for collaboration as shown via the case study in Chapter 6.3. While the virtual models can contribute to PD, it should be remembered they can be used for another purpose as well. The one who develops a business model for using Unity3D based virtual models as part of performance based fire design will, in my opinion, disrupt the way fire design can be implemented. This happens due to Unity3D capabilities of accurate physics simulations.

While the virtual models can be produced, and presented for audience, there is still much to be developed in terms of the production process, utilized technology and the detail level of the models. So far, the solutions described in this thesis are based on the writer's own experience and background research. So, in that essence, the spectrum is quite narrow. When gazing to the future, technology will evolve to provide better visualizations. To outline the possible use scenarios of new technology solutions in the future, the field of use is limited to construction industry.

7.1 Future of HMD applications

During the research, feedback was gathered from variety of users whom are not familiar with VR. Based on the feedback, it could be derived that LeapMotion would have been a great addition to the VR visualizations. LeapMotion is an add-on for VR HMDs that can be used to track the user's hand in the user's field of view as shown in Figure 35. (LeapMotion, 2016) By incorporating the LeapMotion, user's hands can be modelled either with rigid skeletons or rendered hands that track the users fingers and palms in the VR. When viewing a VR without LeapMotion, inexperienced users often felt that it was hard to grasp how the controller is positioned in their hand due to the lack of seeing the hand. LeapMotion can also be used to create interaction events to render the controllers unnecessary. Different menus and interactions can be triggered by turning the palm upwards or pointing with one finger to a direction.

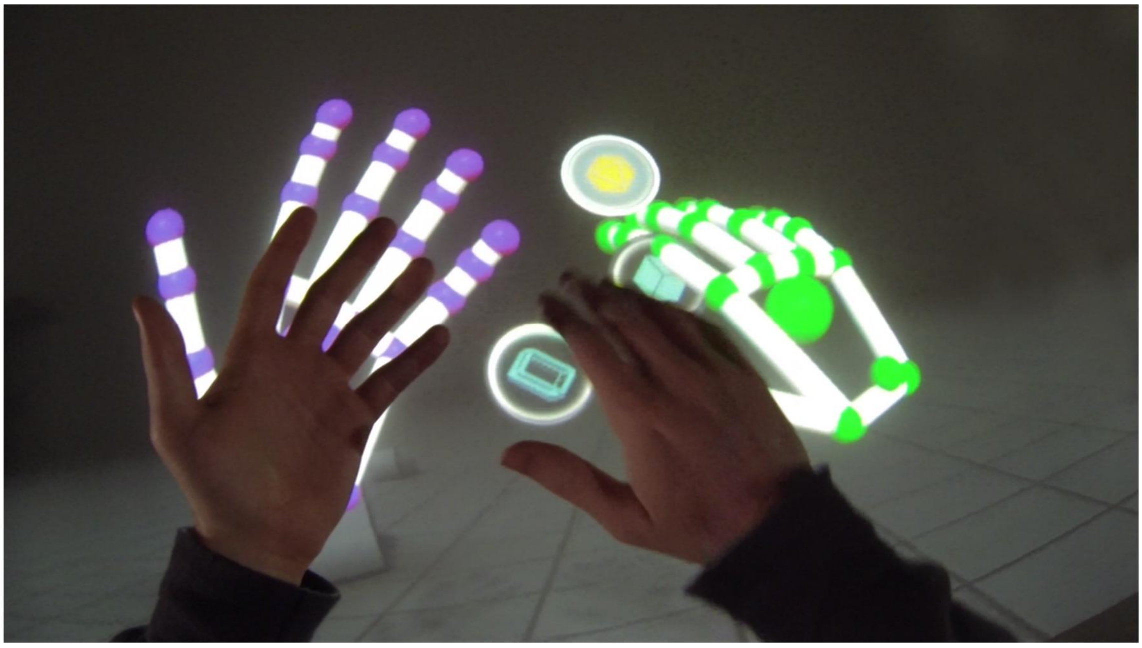

Figure 35: LeapMotion brings user's hands to VR (LeapMotion, 2016)

Daqri has provided their own concept of a MR helmet. It incorporates thermal camera, Intel RealSense chipset and multiple cameras to be used to understand the spatial surfaces around the helmet. In the final product, Daqri Smart Helmet, the user is fitted with a helmet that is approved by regulations for use in construction sites and the helmet has a built-in MR visor for presenting different design or installation plans on site. This concept provides a massive amount of different use scenarios for utilizing the heads-up display (HUD) in FOV of the user. (Daqri, 2016)

By combining location tracking and spatial understanding, instructions for fitting plumbing could be displayed directly on the real parts. The plumber could be guided how the pipes should be attached so that all HVAC equipment fits as planned. The instructions are not limited to a specific industry. Basically, all instructions that require assembly, connections or movement of parts can be visualized in the user's FOV with a see-through lens that understands three-dimensional space and is combined with location data either in world-scale or relative to a nearby known point.

In emergency situations, all users wearing a DAQRI Smart Helmet or a similar MR/AR hardware could be guided to the nearest exit route by tracking the location of the helmets. The emergency routes could be provided directly to each helmet with predefined parameters. With a centralized data structure, a construction site manager could predefine the timetable of needed work in a specified room. In same fashion, as in an emergency, the workers could be guided with on-screen instructions to the work site instead of how to exit the site safely as fast as possible.

To date, all VR applications are restricted with the wiring of the HMDs. TPCast is a wireless solution for VR with HTC Vive that would solve this problem and give the immersed user more freedom to move around the play area. TPCast is a transmitter that is fitted on top of the Vive HMD. It has an external battery that can be clipped on the users belt or on top of the HMD. It uses a local wireless network to pass the information between the headset and a receiver. TPCast provides the user with real-time visualization with less than two millisecond latency per different articles provided by multiple online magazines regarding VR such as Digital Trends, UploadVR and RoadToVR. While these medias cannot be defined as reliable sources for confirming information, TPCast can be preordered from HTC Vive's China website and the users are expected to receive their devices during the third quarter of 2017.

The technology needs to be improved for the described scenarios, but it also needs to be shielded and user attitudes changed in the construction industry. When imagining the possibilities, it often incorporates very delicate technology. The devices would need maintenance periodically and they would need to be handled with care even if they would be shielded with casings or such.

As the HMD technology is developed further, it sets a challenge for CAVEs. The advantage of a CAVE is bringing personnel together to evaluate the design in near-immersive fashion. The gained advantage of a CAVE is based on the larger number of users being able to inspect the design at the same time, as well as the operator having the ability to focus the users into a specific area of a design instead of each of them wandering around the design. When the HMDs evolve enough for multiple users to be able to inspect the same design, the CAVE will likely to be less used method for visualizing the plans. The CAVE could be replaced with multiple MR HMDs unless the CAVE system prices come down and it becomes more portable solution.

7.2 Virtual working environments

Different types of MR solutions are expected to be brought to market, such as Google Lens or new version of the HoloLens. Despite of the used hardware, MR is expected to bring new type of virtual meetings were all participants are presented as avatars in each users' HMD. Shared information, such as scale models of a building, can be distributed through a file server and wirelessly downloaded to each user's HMD. The model could be synchronized with each of the participants in a fashion that the modifications done to the model are updated to all participants in real-time. The solution would same time and money when necessarily all participants do not need to be physically present in the meeting and they would not lose the advantages of interacting and communicating with people in the same room.

Even though the technology will evolve, it sets a challenge for developing better processes to transform information to be available for importing them to the hardware platform. During the year 2017, Autodesk has launched its service, Autodesk LIVE, which exports Autodesk files to a cloud-based service and then imports them to Autodesk Stingray. Stingray is Autodesk's authored game engine which is mostly used as platform to handle project data created in Autodesk software. In addition to Autodesk, Trimble has acquired SketchUp as a part of their software selection in the year 2016, which now has a direct link with HoloLens. It is likely that increasingly more software developers offer their own solution for deploying design plans to VR or AR with a direct link or as a one-button service

In addition to wireless technology for HMDs, more precise and larger tracking methods may be created in the following years. Full-body tracking of a person is mainly used in entertainment industry, but it could serve a purpose in co-creative design if the tracking is done on-the-fly without a great number of different sensors. If the user could be fully immersed in the VR environment, a greater perception of depth in relation of users own physical measurements would be achieved. This would provide valuable data for all industries concerning a user in an enclosed environment, such as a car or a machine's operating seat. For construction industry, this could provide information for operational design of hospitals: how many persons can treat a patient in the designed environment with all the equipment and personnel needed. By optimizing the needed space without compromising the patient's health, cost efficiency of a designed hospital would improve.

The same concept could be brought to construction sites as well. When confronted with a problem on the site, a construction worker usually phones his supervisor or the designer of the structure. Instead of explaining on the phone what is the situation if the person is not standing with him on the site, by utilizing an MR helmet's camera, the designer or supervisor could see what is going on the site. Adjustment of the plans or providing additional information to help the worker, the updates could be sent to construction worker's

MR visor. This would speed up the process in problematic situations where distances between different parties prevents visits on site.

While the design of structures has stayed relatively the same, new concept of parametric design is a rising trend in construction industry. Parametric design is designing concurrent structural components, such as the joists in roof strusses, can be designed with algorithm that optimizes the structure in terms of mass and durability. When combined with MR, the iteration of different structures could be done on-site, assuming the hardware could perform such calculations on the fly.

When the transform from normal desktop working to different virtual solutions can be done in matter of seconds, rather than minutes or hours, it will disrupt the way different parties collaborate in construction projects. By enabling remote services with virtual, augmented or mixed reality, a new collaboration platform may be created for one or multiple devices. Even though the devices might not use any common platforms, it would be possible that one rises above others to be used in a larger scale. It would require a fluent and intuitive process to create content for the platform and easily accessible technology.

8. CONCLUSIONS

8.1 The need for virtual models

The need of virtual models was studied by conducting interviews with experts from different companies as defined in Chapter 1.4. The selected companies have expertise in their specific areas of construction project and all have different approach on how information should be handled. As described in Chapter 1.4, the interviews were conducted as themed interviews with three main themes: co-creative design in interviewee's organization, co-creation practices and methods and tools and technologies for co-creative design.

At this moment, there are no standardized protocols or working methods to implement co-creative design in construction projects. Consultants and customers have different ideas of what co-creative design should be in practice and what kind of results it should yield. The concept of co-creation is familiar across the interviewed personnel but usually the practices how the design processes should incorporate co-creative design are agreed upon in each project separately. Some companies have their own service model to offer, however even though there is not an industry-wide standard practice to implement co-creation to a construction project, the co-creation itself is seen as a valuable and desirable aspect.

By implementing co-creation, the construction project adds value in terms of streamlining the design solution and prioritizing options when there is need for it. All the interviewed personnel agreed that co-creative design can produce savings by either scaling down the needed space or maximizing the use-value of a space. Without a co-creative process, it was felt that it is not as easy to achieve proper communication and collaboration between designers, users, and builders.

The lack of common terminology and understanding of technical terms between construction and non-construction oriented personnel is seen as the most difficult part of co-creation and communication in a construction project. When discussing in a multi-oriented project organization, the understanding of each party's view in design phases and solutions is a key component in the success of a project.

Depending on the company's business model, co-creation can be defined as a feedback system from the future users and owners of the space or as a workshop platform used for development and evaluation of design plans. At the time, there are numerous applications for visualizations in marketing, designing and constructing buildings and environments, each having their own use methodology.

Based on the interviews, modern technologies such as CAVEs, VR, AR, and MR applications can serve a purpose in co-creative design. Depending on the intended use and communication method, a proper tool is applied to the design process. None of the above are constants in a construction project, but they can be utilized when needed. The design process does not focus on which technologies should be implemented but rather what technologies are needed to pass information between different parties of a project. The modern VR, AR and MR solutions are viewed as a huge benefit for visualizing and discussing design solutions. So far, the technologies are not used on a large scale in construction industry in Finland but they are becoming increasingly more common. Many companies explore different ways to use new technologies in a construction project and co-creative design. The response form new users have been positive and it encourages the developers to come up with higher quality products and streamlined processes.

The downside in developing technological solutions in the construction field in Finland, is the isolated nature of competitive business. The progress is not considered on a large scale. Developers and companies focus on how they could improve the gross revenue of the represented company by getting a competitive edge compared to others. This mentality does not progress the whole construction industry in Finland which could convert to world-wide competitive edge and acknowledgment of expertise as a nation. A cross-organizational work group should be convened to work together in solving common problems and developing new kind of business models for VR and MR technologies. The solutions would be distributed freely to organizations. While this would not benefit individual organizations in their competition with others in Finland directly, it would drive the whole construction industry forward by utilizing new technological solutions and co-creative design possibilities.

Based on the interviews, there is a clear need for virtual models in the construction field. The best way to utilize them is to visualize information in a variety of ways and use it as a tool for user feedback in a co-creative design process. Different kind of ways to present information in understandable format can consist of various types of visualizations depending on what is to be visualized. The visualizations would be directly connected to collaborative and participatory design as illustrated in Lundström, et. al. study (2016) and in the case study in Chapter 6.3. The models could serve as a basis for the designers and users to have a mutual understanding of the design plans and what needs to be done for the plans and spaces to serve their purpose in the best possible way.

8.2 Fulfilment of research goals

The research outline and main research questions were defined in the first chapter of this thesis. The research questions were set as following:

- RQ1: What is the most efficient way to produce a virtual model in Unity3D

- RQ2: What are the best practices to present a virtual model and design data to a client
- RQ3: How can virtual models be applied to co-creative design in construction process

A process for producing a virtual model from information created in a construction project was defined. Through iteration, the process was streamlined by applying different automation, such as scripts, to the process. However, the original research question was to answer what is the most efficient way to produce a virtual model. From the amount of research needed to develop a working process to produce a virtual model at all, it cannot be said that the process is absolute or definitively the most efficient. Thus, in the original setting of the research question, it can be said that the question remains unanswered. The research yields usable results but not in the sense that was anticipated in the beginning. The research question was set to provide an absolute answer to a question which is very difficult to answer definitively due to the amount of different approaches to the same research problem, and thus different perspectives on the perception of being the most efficient.

The best practices of presenting information to clients was studied by providing virtual model presentations to different group of users. Direct oral feedback was gathered from first experiences with the virtual model and by observing the behavior of users in the model. For presenting design information to a client, a 3D representation of the is considered an easier way to gain understanding of the design than 2D drawings. However, to understanding of the depth of objects was considered much easier in an immersive visualization of the design. Based on the feedback, an immersive visualization can be considered as one of the best practices to present information to a client in construction projects. Even though some of the users may experience motion sickness from the VR, AR or MR HMDs, majority of the users found the experience more pleasant than uncomfortable. In addition, the motion sickness can be reduced by well-produced virtual models that run with steady FPS to create an effective immersion. Therefore, the second question is answered.

Different ways to implement virtual models to co-creative design in construction was studied via a case study of using a virtual model as part of a charrette and by conducting themed interviews. Based on the conclusions of the interviews in Chapter 8.1 and the findings in the case study in Chapter 6.3, the third research question is answered.

8.3 Future work

Due to the research outlining some of arisen problems or notations could not be addressed in the scope of this research. However, they can be recognized as potential areas that can yield more results and form the practice of co-creation in construction industry and thus should be researched.

Process optimization and variation: First step could be researching the defined process further: could be more optimized, could some of the defined steps be done in different fashion or with future updates of software, can some steps be completely left out? A different approach to transform data could also be implemented. By writing a parser in Python programming language (or similar), the vertex information could possibly be extracted from IFC files and then be reconstructed into more optimized mesh in same fashion as producing triangulated meshes from a point cloud. The process could be automated to the point when the models are generated with a press of a button, resulting in a less time consuming and thus, more cost-effective process.

User experience: What kind of user experience virtual models provide in a larger scale? A pattern may arise by presenting virtual models with the same type of data visualization to larger group of users and gathering feedback regarding the experience provided by the visualization. The user experience study could dictate what kind of data is best presented in what kind of fashion.

Automatic geometrical model optimization: As described in this research, geometrical models created in design software are not generally suited to be used in a virtual model due to excess number of polygons. It could be researched if and automation could be implemented to decimate the structured faces of an object and reconstructed in more efficient way for rendering.

Incorporating BIM meta-data to virtual models: BIMs contain a massive amount of data in addition to the geometrical properties of objects. Due to the needed data conversions for the geometrical objects to be imported into Unity3D, the meta-data is lost during the process. However, it could be extracted to a separate file before being destroyed by file conversions. With scripts the model geometry and its meta-data could be recombined in Unity3D for usage in virtual models. This could be applied to city models where information could be filtered through the meta-data. In addition, the meta-data could be used as a basis for further scripting like identifying different spaces in the model or calculating mass of the structural frames or elements and iterate the cost efficiency of different design options.

Locomotion in real-world scale immersive models: For some visualizations, a mini-map was created for the user to be able to orient himself in the environment and prevent getting lost. However, the mini-map often proved to cause more distraction than actual help in navigation. While it provided information to the user of his location in the model, it would have served a better purpose as a teleportation canvas. By pointing to the map and activating teleportation, users would have wanted to teleport to the pointed location on the map. This would be an effective way of locomotion in multi-story building models or larger landscape models.

9. REFERENCES

Agarwal, R., Chandrasekaran, S. & Sridhar, M., 2016. *Imagining construction's digital future.* [Online]
Available at: http://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/imagining-constructions-digital-future
[Accessed 10 April 2017].

American Institute of Architects, 2013. *Guide, Instructions and Commentary to the 2013 AIA Digital Practice Documents,* s.l.: American Institute of Architects.

Autodesk, 2016. *3ds Max supported file formats.* [Online]
Available at: http://help.autodesk.com/view/3DSMAX/2016/ENU/?guid=GUID-566E59EE-8221-4AC6-824B-5062C5AE0B32
[Accessed 15 January 2017].

Azuma, R. T., 1997. A Survey of Augmented Reality. *Presence,* 6(4), pp. 335-385.

Bartz, D., Meißner, M. & Hüttner, T., 1998. *Extending graphics hardware for occlusion queries in OpenGL.* Tübingen, s.n.

Biljecki, F., 2013. *PhD research proposal: The concept of leve lof detail in 3D city models,* Delft: Delft University of Techology.

Blender Foundation, 2017. *Blender.* [Online]
Available at: http://www.blender.org
[Accessed 15 January 2017].

Blender Foundation, 2017. *Blender features.* [Online]
Available at: https://www.blender.org/features/
[Accessed 15 January 2017].

Boyekens, S., 2013. *Unity for Achitectural Visualization.* 1st ed. s.l.:Packt Publishing.

buildingSMART Finland, 2012. *Common BIM requirements,* s.l.: buildingSMART Finland.

buildingSMART International, 2016. *IFC Wiki, IFC definition.* [Online]
Available at: http://www.ifcwiki.org/index.php?title=IFC_Wiki
[Accessed 4 January 2017].

buildingSMART International, 2017. *IFC Certified Software.* [Online]
Available at: http://buildingsmart.org/compliance/certified-software/
[Accessed 7 June 2017].

BuildingSMART, 2014. *IFC Overview summary.* [Online]
Available at: http://www.buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary
[Accessed 26 December 2016].

Craig, A. B., 2013. *Understanding Augmented Reality: Concepts and Applications.* US: Morgan Kaufmann Publishers Inc.

Daqri, 2016. *Smart Helmet.* [Online]
Available at: http://daqri.com/products/smart-helmet/
[Accessed 6 June 2017].

Finnish Transport Agency, 2017. *Open data.* [Online]
Available at: http://www.liikennevirasto.fi/web/en/open-data#.WL6gI4VOJ9A
[Accessed 7 March 2017].

Geodetic Systems, 2017. *What Is Photogrammetry?.* [Online]
Available at: https://www.geodetic.com/v-stars/what-is-photogrammetry/
[Accessed 11 May 2017].

GIS Geography, 2017. [Online]
Available at: http://gisgeography.com/lidar-light-detection-and-ranging/
[Accessed 21 March 2017].

Hirsijärvi, S. & Hurme, H., 2011. *Tutkimushaastattelu : Teemahaastattelun teoria ja käytäntö.* Helsinki: Gaudeamus Helsinki University Press.

IfcOpenShell, 2015. [Online]
Available at: http://www.ifcopenshell.org/
[Accessed 15 January 2017].

International Organization for Standardization, 2013. *ISO 16739:2013 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.* [Online]
Available at: https://www.iso.org/standard/51622.html
[Accessed 7 June 2017].

Johanson, M., 2016. *From BIM to VR,* Gothenburg: Department of Civil and Environmental Engineering.

Johansson, M. & Roupé, M., 2009. *Efficient real-time rendering of building information models.* Las Vegas, Chalmers University of Technology, Göteborg, Sweden, pp. 97-103.

Laurila, P., 2012. *Mittaus- ja kartoitustekniikan perusteet.* 4th ed. Rovaniemi: R AMK University of Applied Sciences.

LeapMotion, 2016. *LeapMotion Technology.* [Online]
Available at: https://www.leapmotion.com/product/vr#113
[Accessed 8 June 2017].

Lo Turco, M. & Sanna, M., 2009. *Digital modelling for architectural reconstruction. The case strudy of the chiesa confraternita della misericordia in Turin,* Kyoto: s.n.

Lowe, N. & Datt, A., 2005. A new technique for rendering complex portals. *IEEE Transactions on visualization and computer graphics,* 11(1), pp. 81-90.

Luebke, D. et al., 2002. *Level of detail for 3D graphics.* s.l.:Elsevier Inc..

Lundström, A., Savolainen, J. & Kostiainen, E., 2016. Case study: developing campus spaces through co-creation. *Architechtural engineering and design management,* 27 July.

Marxentlabs.com, 2013. *USPS Augmented Reality direct mail promotion; How to save on AR campaigns with the U.S. Postal Service.* [Online]
Available at: http://www.marxentlabs.com/usps-augmented-reality-direct-mail-promotion-how-to-save-on-ar-campaigns-with-the-u-s-postal-service/
[Accessed 20 February 2017].

Mason, A., 2016. *Making 3D Models with Photogrammetry.* [Online]
Available at: https://thehaskinssociety.wildapricot.org/photogrammetry
[Accessed 7 July 2017].

Meijers, A., 2017. *Achmea technight - HoloLens developement.* [Online]
Available at: https://www.slideshare.net/ameijers/achmea-technight-hololens-development
[Accessed 12 May 2017].

Microsoft, 2016. *HoloLens hardware.* [Online]
Available at: https://www.microsoft.com/en-us/hololens/hardware
[Accessed 11 May 2017].

Microsoft, 2016. *Recommended settings for Unity.* [Online]
Available at: https://developer.microsoft.com/en-us/windows/mixed-

reality/recommended_settings_for_unity
[Accessed 12 May 2017].

Microsoft, 2016. *Spatial mapping.* [Online]
Available at: https://developer.microsoft.com/en-us/windows/mixed-reality/spatial_mapping#conceptual_overview
[Accessed 11 May 2017].

Mikhail, E. M., Bethel, J. S. & McGlone, J. C., 2001. *Introduction to Modern Photogrammetry.* s.l.:John Wiley & Sons Inc..

Milgram, P. & Kishino, F., 1994. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems,* Volume 12, p. 1321–1329.

Naaranoja, M., Ketola, P. & Niemi, O., 2015. Charrette supports facility developement - Case Musica. In: *How to Co-Create campus?.* Tampere: University Properties of Finland, pp. 180-195.

National Land Survey of Finland, 2017. *File service portal.* [Online]
Available at: https://tiedostopalvelu.maanmittauslaitos.fi/tp/kartta?lang=en
[Accessed 2 February 2017].

Nbs, The Building Information Foundation RTS, 2013. *Finnish BIM Survey,* s.l.: The Building Information Foundation.

Open Geospatial Consortium, 2017. *What is CityGML?.* [Online]
Available at: https://www.citygml.org/about/
[Accessed 7 July 2017].

Pantazopoulos, I. & Tzafestas, S., 2002. Occlusion Culling Algorithms: A Comprehensive Survey. *Journal of Intelligent and Robotic Systems,* Issue 35, pp. 123-156.

Peñaflorida, R., 2016. *Tom's Hardware.* [Online]
Available at: http://www.tomshardware.com/news/htc-vive-store-vr-demo,31566.html
[Accessed 15 February 2017].

Prahalad, C. & Ramaswamy, V., 2004. Co-Creation Experiences: The Next Practice in Value Creation. *Journal of Interactive Marketing,* Issue 3, pp. 5-14.

Savisaari, A., 2017. *Utilization of point cloud data in architectural design of repair and renovation projects,* Tampere: s.n.

Savolainen, J. et al., 2015. *Stirring the construction project management with co-creation and continuous improvement.* [Online]

Available at: http://www.sciencedirect.com/science/article/pii/S2212567115001513
[Accessed 5 March 2017].

Shen, Z., Jiang, L., Grosskopf, K. & Berryman, C., 2012. *Creating 3D web-based game environment using BIM models for virtual on-site visiting.* Lincoln, Construction Research Congress.

Slick, J., 2016. *Surfacing 101 - The basics of texture mapping.* [Online]
Available at: https://www.lifewire.com/texture-mapping-1956
[Accessed 13 June 2017].

The American Society for Photogrammetry and Remote Sensing, 2009. *LAS Specification version 1.3 - R11,* Maryland: The American Society for Photogrammetry and Remote Sensing.

Turppa, T., 2014. *Laserscanning as a part of game developement,* Helsinki: Metropolia University of Applied Sciences.

Tötösy de Zepetnek, S. & Sywenky, I., 1997. The Systemic and Empirical Approach to Literature and Culture as Theory and Application. *Library Series of CLCWeb: Comparative Literature and Culture.*

Unity Technologies, 2017. *Input for OpenVR controllers.* [Online]
Available at: https://docs.unity3d.com/Manual/OpenVRControllers.html
[Accessed 30 June 2017].

Unity Technologies, 2017. *Unity Asset Store.* [Online]
Available at: https://www.assetstore.unity3d.com/
[Accessed 20 February 2017].

Unity Technologies, 2017. *Unity manual: Types of light.* [Online]
Available at: https://docs.unity3d.com/Manual/Lighting.html
[Accessed 21 June 2017].

Unity Technologies, 2017. *Unity Pro, Unity Plus and Unity Personal Software Additional Terms.* [Online]
Available at: https://unity3d.com/legal/terms-of-service/software
[Accessed 30 January 2017].

University of Jyväskylä, 2015. *Laadullinen tutkimus.* [Online]
Available at:
https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tutkimusstrategiat/laadullinen-tutkimus
[Accessed 4 July 2017].

Whyte, J., Bouchlaghem, N., Thorpe, A. & R., M., 1999. From CAD to Virtual Reality: Modelling Approaches, Data Exchange and Interactive 3D Building Design Tool. *Automation in Construction,* 2000(10), pp. 43-55.

Xu, J. & Chen, X., 2012. Study and analysis of texture mapping technology based on OpenGL. *Applied mechanics and materials*, 29 November, pp. 1990-1993.

APPENDIX A: INTERVIEW QUESTIONNAIRE

Interviewee

Time

Place

1 Co-creative design in your organization

1.1 Is a co-creative design process a standard protocol in your organizations? Do you demand it?

1.2 Do you think co-creative design is a vital part of a construction project?

1.3 How big an impact do you think co-creative design has on construction project?

2 Practices for co-creative design

2.1 Are there any challenges in the co-creative design process? What kind?

2.2 How do you motivate your employees and affiliates to take part in co-creative design?

2.3 Are any of your interest groups interested in co-creative design and the use of AR/VR technologies?

2.4 Do you strive to promote and develop the existing practices for co-creative design?

3 Tools and technologies for co-creative design

3.1 What kind of technologies do you favor in co-creative design?

3.2 What kind of tools does your organization have for co-creative design?

3.3 Do you follow the use of new technologies for co-creative design actively?

3.4 Do you think working in a CAVE environment is a good way to collaborate?

3.5 Have you implemented VR/AR technologies into co-creative design? What kind of experiences have you had?

3.6 Do you feel there is a need for VR/AR technology in construction project? In what kind of use cases?