



Cultural Heritage Reconstruction of the Cinema Flamman Using VR

Achieving cultural heritage preservation in virtual reality by reconstructing the cinema Flamman

Bachelor's thesis in Computer science and engineering

Joakim Burman
Jonathan Eksberg
Mirai Ibrahim

Tommi-Roy Karlsson
Johan Pettersson
Adam Rohdell

BACHELOR'S THESIS 2021

Cultural Heritage Reconstruction of the Cinema Flamman Using VR

Achieving cultural heritage preservation in virtual reality by
reconstructing the cinema Flamman

Joakim Burman

burmanj@student.chalmers.se

Jonathan Eksberg

eksberg@student.chalmers.se

Mirai Ibrahim

gusibrmi@student.gu.se

Tommi-Roy Karlsson

tommiroy@student.chalmers.se

Johan Pettersson

guspetjobb@student.gu.se

Adam Rohdell

rohdell@student.chalmers.se



UNIVERSITY OF
GOTHENBURG



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Cultural Heritage Reconstruction of the Cinema Flamman Using VR
Achieving cultural heritage preservation in virtual reality by reconstructing the cinema Flamman
Joakim Burman Jonathan Eksberg Mirai Ibrahim Tommi-Roy Karlsson Johan Pettersson Adam Rohdell

© Joakim Burman, Jonathan Eksberg, Mirai Ibrahim, Tommi-Roy Karlsson, Johan Pettersson, Adam Rohdell 2021.

Supervisor: Thommy Eriksson, Department of Computer Science and Engineering
Examiner: Michael Heron, Department of Computer Science and Engineering

Bachelor's Thesis 2021
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The placeholder model created in SketchUp. It was used during the development of the application to visualize the implemented functionality before the final 3D model was delivered.

Typeset in L^AT_EX
Gothenburg, Sweden 2021

Cultural Heritage Reconstruction of the Cinema Flamman Using VR
Achieving cultural heritage preservation in virtual reality by reconstructing the cinema Flamman

Joakim Burman, Jonathan Eksberg, Mirai Ibrahim, Tommi-Roy Karlsson, Johan Pettersson, Adam Rohdell

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Virtual reality is adopted in many areas such as the military, education, health care, and preservation of cultural heritage. This thesis will further explore the possibility of cultural heritage preservation in virtual reality by reconstructing the environment of the old and defunct cinema in Gothenburg named Flamman. The aim is to provide a proof-of-concept experience, in which a user can get the feeling that they are located in the actual cinema. To achieve this, a focus on good interaction design and a pleasant user experience is of high priority, as not to introduce common issues in virtual reality applications such as motion sickness. Tools such as Blender and Unity are utilized to implement these goals. To further ensure that the created application fulfills these main goals, user tests are conducted in which the test participants can express their experience. The results from these tests indicate that the application offers a satisfying experience, and the feeling of being in a cinema is present. Ultimately, all of the goals were met by the implemented features. Not only is it possible to watch a movie, but users can also share the experience together by using the multiplayer functionality.

Keywords: Virtual Reality, User Experience, Cultural Heritage Preservation, Software Development, Game Design

Sammandrag

Virtual reality har flertalet användningsområden där det till exempel används inom det militära, utbildning, sjukvård och även för att bevara kulturarv. Denna uppsats kommer fortsätta att undersöka de möjligheter för kulturarv som virtual reality erbjuder. Detta kommer att genomföras med en rekonstruktion av den gamla och nedstängda biografen Flamman i Göteborg. Målet är att skapa en prototypapplikation som kan återskapa känslan av att befinna sig i denna förgångna biograf. För att uppnå detta krävs ett stort fokus på god interaktionsdesign samt behaglig användarupplevelse, annars finns risken att applikationen kan åstadkomma åksjuka för användaren. Verktyg såsom Blender och Unity används för att implementera dessa huvudmål. För att ytterligare försäkra sig om att den skapade applikationen uppfyller dessa mål kommer användartester att genomföras, där deltagarna kan uttrycka åsikter om sin upplevelse. Resultat från dessa tester indikerar att applikationen levererar en bra upplevelse och att känslan av att vara i en biograf återskapas. Genom de implementerade funktionerna uppfylls alla mål. Förutom möjligheten att se film kan även användare dela denna upplevelse tillsammans, genom att använda multiplayerfunktionen.

Acknowledgements

We would first like to thank our supervisor, Thommy Eriksson, for providing us with helpful information and insights throughout the project and being easily accessible when questions arose. Furthermore, we would also like to thank all of the participants in our user tests. The feedback we got from those sessions was incredibly useful. Finally, we would like to thank our examiner, Michael Heron.

Joakim Burman, Jonathan Eksberg, Mirai Ibrahim, Tommi-Roy Karlsson,
Johan Pettersson, Adam Rohdell

Gothenburg, June 2021

Contents

List of Figures	xiii
1 Introduction	1
1.1 Purpose	1
1.2 Delimitations	1
1.3 Goals	2
1.4 Thesis outline	2
2 Theory	3
2.1 Virtual Reality	3
2.1.1 What is a Good VR Experience	4
2.1.2 Motion Sickness and Other VR Problems	5
2.1.3 Interaction Design and UX in VR	5
2.2 Cultural Heritage Preservation	6
2.2.1 Cultural Heritage Preservation in VR	6
2.3 Ethical and Social Aspects	6
2.4 Hardware	7
2.4.1 HTC VIVE	7
2.4.2 Oculus Rift S	7
2.5 Software	7
2.5.1 Unity	8
2.5.2 Unity Technologies Used	9
2.5.3 3D Modeling	10
2.5.4 Blender Technologies Used	11
2.5.5 Assets	11
2.5.6 SteamVR	11
2.5.7 Mirror	12
2.5.8 Version Control	12
2.6 Agile Development	13
2.7 User Tests	13
2.7.1 Avoiding Ethical Issues	14
3 Methods	15
3.1 Agile Development	15
3.1.1 Scrum Board	15
3.2 Hardware	16

3.3	Software	16
3.3.1	Unity	16
3.3.2	Mirror	16
3.3.3	SteamVR	16
3.3.4	Version Control	16
3.3.5	3D Modeling	17
3.4	User Tests	18
4	User Tests	19
4.1	First Evaluation	19
4.2	Second Evaluation	19
4.3	Changes Made and Conclusions from the User Tests	20
5	Results	21
5.1	Final 3D Model of Flamman	21
5.2	How to Start the Application	22
5.3	How to Start a Movie	23
5.4	Other Gameplay Mechanics	25
5.5	Virtual Reality Experience	25
5.6	Multiplayer Experience	27
5.7	Performance	27
6	Discussion	28
6.1	Choosing Unity	28
6.2	Performance	28
6.2.1	Lighting in Unity	28
6.2.2	Cinema Chairs	29
6.2.3	Occlusion Culling	30
6.3	Source Material	30
6.4	Size of The User in VR	31
6.5	Creation of the 3D Model	31
6.6	Limitations Discovered	32
6.6.1	User Usability Versus Original Design	32
6.6.2	Cinema Chair	33
6.6.3	VR Implementation	34
6.7	User Experience	34
6.7.1	User Tests	35
6.7.2	Technology	35
6.7.3	Lighting in Flamman	35
6.7.4	Network Functionality	36
6.7.5	Ethics	36
6.8	COVID-19	37
7	Conclusion	38
7.1	Recommendations	39
8	Further Development	40

Contents

8.1	Dynamic Light From the Projector Screen	40
8.2	Dynamic Curtain	40
8.3	Further Improvements to Performance	40
8.4	Choosing a Movie Through a Movie Roll	41
8.5	Foldable Seats	41
8.6	Game Like Features	41
8.7	Play Other Movies	41
8.8	Chat or Voice Feature	41
8.9	Possibility to Host Public Movie Rooms	42
8.10	Better Historical Recreation	42
Bibliography		43
A Appendix 1		I
A.1	Evaluation of User Tests	I
B Appendix 2		IV
B.1	Images of Flamman	IV
B.1.1	Side by side comparison of the current, renovated Flamman and the application	IV
B.1.2	Objects with no direct reference to Flamman	VII
B.1.3	More pictures of Flamman	IX
C Appendix 3		XII
C.1	Epics	XIII
C.2	Scrum Board	XIV

List of Figures

2.1	Occlusion culling in Unity. What the player see is to the left, and what the Unity engine renders to the right	10
3.1	The model of Flamman done through SketchUp	17
5.1	Side by side comparison of the lobby	21
5.2	Side by side comparison of the front of the cinema saloon	22
5.3	Side by side comparison of the back of the cinema saloon	22
5.4	Side by side comparison of the corridor	22
5.5	The starting screen of the application	23
5.6	Keybind and tutorial boards in the lobby	23
5.7	Projector door to teleport to projector room	24
5.8	The projector room	24
5.9	The menu to choose a movie	25
5.10	Red band to block the exit	26
5.11	One of the eight posters in the corridor	26
5.12	Two users that are waving to each other	27
6.1	Result of baking the lights	29
6.2	Side by side comparison of the corridor	30
6.3	Walkway to the projector room according to the original blueprint . .	32
6.4	The three options for the discussed new walkway from the top	33
B.1	Side by side comparison of the lobby	IV
B.2	Side by side comparison of the railing	IV
B.3	Side by side comparison of the projector door	V
B.4	Side by side comparison of the access to the saloon	V
B.5	Side by side comparison of the corridor	V
B.6	Side by side comparison of the saloon chairs	VI
B.7	Side by side comparison of the saloon chairs	VI
B.8	Side by side comparison of the saloon scene	VI
B.9	Side by side comparison of the saloon side lights	VII
B.10	Board with the old photos of Flamman in the lobby	VII
B.11	Ticketbooth in the lobby	VIII
B.12	One of the two trees in the lobby	VIII
B.13	Table and shelf for movie rolls in the projector room	IX
B.14	The projector in the projector room	IX

List of Figures

B.15	Corridor with posters	IX
B.16	The view from the projector room out to the saloon	X
B.17	The corridor on the left side of the saloon	X
B.18	The entrance doors to Flamman	X
B.19	From the lobby to the stairs	XI
C.1	Epics created to achieve the project goals	XIII
C.2	The Scrum Board hosted at Trello	XIV
C.3	Part of the sprint archive	XIV

1

Introduction

The concept of *Virtual Reality* (VR) dates back to 1935 and was introduced to the world in a science fiction story written by the author Stanley G. Weinbaum[1]. In the story, Weinbaum wrote about a pair of glasses that, when used, enables a person's senses to transport to different worlds while the body remains grounded on earth. While this story only foreshadowed VR, it acted as an inspiration for scientists to keep working towards making this a possibility. The progression of VR technology has led to it being used in many fields such as education, military, health care, and as this paper will demonstrate cultural heritage preservation [2].

The building Flamman opened in 1935 in central Gothenburg and was used as a cinema until 1982 [3]. In 2006, Flamman was listed as a historical building and has since been used as a community center. For this project, Flamman was chosen because of two reasons. The first reason is that the building is still intact, making it easier to get any references needed, such as pictures and blueprints. The second reason is Flamman's architectural value, as it is considered "one of the first modernist buildings in Gothenburg" [4]. This thesis will present a reconstruction of Flamman in VR and preserve its cultural heritage. By doing so, both present and future generations will be able to experience Flamman as it once was, a cinema.

1.1 Purpose

This project was created with two main purposes in mind. Firstly, to create a proof-of-concept application of the building Flamman in VR and preserving its cultural heritage. Secondly, to ensure a pleasant *User Experience* (UX) by creating a believable and immersive experience. Cultural heritage and the preservation of old and forgotten environments is important, especially as cultural aspects and tradition are at risk of disappearing [5]. By giving people the opportunity to visit a place of historical value, they can feel connected to the specific time and take part in an enriching experience.

1.2 Delimitations

The exterior and the surrounding area around Flamman will not be implemented, as the work is solely focused on the experience inside the building itself. Therefore,

only the interior of Flamman is created. However, only the most relevant parts of the model are available through the experience, meaning it is not possible to go into the bathroom and storage facilities of Flamman.

The graphical fidelity of the experience was decided not to be photo-realistic to allow high frame rates when the user is inside the virtual application but realistic enough to deliver an immersive experience.

1.3 Goals

To achieve the purpose of cultural heritage, the project aimed to reach the following goals, which are based on the requirements in the project description [4]:

- To be able to watch a movie within the virtual cinema.
- To be able to interact with other users in the virtual cinema.
- To be able to experience the reconstructed environment in VR.

To achieve the stated goals, epics were created that can be seen in Appendix C.

Cultural heritage and the preservation of old and forgotten environments is an important part of understanding cultural values, and this project aims to create an experience that closely resembles Flamman's original cinema experience. There were various reasons why the project members chose this project. Having interests in game development and movies were the main part, but also for acquiring new knowledge about the different tools used for creating this application.

1.4 Thesis outline

This report is divided into eight sections: Introduction, Theory, Methods, Results, Discussion, Conclusion, Further Developments, and Related Work. In the Introduction, the background, purpose, and scope of this project will be presented, along with the goals. Theory will include essential information about VR and cultural heritage preservation, together with a discussion of the ethical and social aspects of these two concepts. Theory will also cover the various tools, practices, and processes used in this project to give the reader the knowledge needed to understand how these methods were applied in the project, which is mentioned in the following section, Methods. After Methods, the project's result is presented, followed by a discussion of this result in Discussion. Conclusion will contain a summary of how the result and discussion is connected to the goals for this project. Further Development will include suggestions for how the project could be developed further.

2

Theory

This part of the thesis elaborates on the theory that is applied throughout the whole project. For clarity and structural purposes, this chapter is divided into different sections.

2.1 Virtual Reality

Virtual reality is a simulated experience created by a computer to allow a person to explore and interact with a 3D environment [6]. To achieve the simulated experience, you need two components: a head-mounted display and a form of input. With these two components, the person will be isolated from the real world and immersed within the 3D environment, perform actions and become a part of the virtual world. Throughout the 1900s, scientists experimented with sense-stimulation inventions and head-mounted displays. Among those inventions are there two that stand out: the Sensorama [7], and The Sword of Damocles [1].

Sensorama, created by the American cinematographer Morton Heilig in 1962, was an immersive and multi-sensory device that simulated ride-based experiences such as riding a motorbike and driving go-karts. To make the experience immersive, Heilig used special sounds, winds, and vibrations. In 1968, computer scientist Ivan Sutherland invented the first head-mounted display and called it the Sword of Damocles. Though progress during the 1900s was made, it was not until the 1970s when VR became a reality as technology had advanced enough to produce devices to support the virtual settings [8].

VR was getting increasingly popular during the 1980s as VR headsets were developed and appeared in popular culture, such as movies and books [9]. However, at the beginning of the 1990s, VR decreased in popularity as VR headsets were too expensive and the technology was not advanced enough to produce immersive experiences. It was not until early 2010 when Oculus released the first commercial headsets that VR's popularity once again increased since the technology became obtainable [10]. Both tethered and standalone headsets exist. VR technologies have developed enough to be utilized in more than just entertainment applications and have expanded to areas like military, education, architecture, health care, and heritage preservation.

2.1.1 What is a Good VR Experience

To understand what makes a good VR experience, two key aspects of virtual reality must be understood, *immersion* and *presence*.

According to Mel Slater, author of the paper "Immersion and the illusion of presence in virtual reality", immersion is an objective measurement of what the technology delivers [11]. An example of how to increase the immersion of a movie is to increase the size of the screen or the number of surrounding speakers.

Presence is a phenomenon that describes the state of feeling present inside a virtual world [11]. The larger the presence, the barrier between the real world and the virtual world is more abstract. Presence is the subjective measurement of the human reaction to the experience.

The quality of the virtual experience rises with immersion and presence and is thereby essential to VR [11]. Therefore, achieving both of them is important for a good VR experience. As the technology of the VR headsets is set by the manufacturers, little can be done to improve immersion, such as increasing the screen size mentioned earlier. Presence, however, the subjective reaction to the experience, is possible to be improved.

There exists a list of requirements for VR headset displays to achieve a pleasant user experience, of which every point has to be met [12]. The most important points of the list are a wide field of view of 110 degrees diagonally or more, a high resolution of the display, low latency, precise tracking, and a high frame. The list is mainly targeted towards the hardware developers of VR headsets and cannot be changed by software such as the display resolution. However, it is important that the software developers design experiences that allow the hardware to utilize its full potential.

Frame rate is the number of new individual rendered frames of a time span, which is measured in frames per second (FPS) [13]. A high frame rate will make the experience feel smoother, more responsive, and more immersive. A game running at a lower FPS than 75 is likely to induce disorientation, nausea, and motion sickness, with the risk and effect increasing with lower FPS.

The resolution is correlated to the frame rate since an increasing resolution increases rendering time, therefore impacting the frame rate [13]. Since a high frame rate and resolution are crucial for immersion and presence, the graphical fidelity has to be adjusted with lower quality textures. Furthermore, simpler lighting and shadows and lower polygon models also help reduce graphical fidelity.

Another way to improve a VR experience is to make the user feel comfortable and safe while in a virtual world. To have a safe experience, it is important to not expose the user to distress. For example, it has been shown that VR can induce vertigo by exposing the users to a virtual scene high up in the air [14]. Similarly, too small spaces should be avoided to not affect people with claustrophobia. Furthermore, having a large or complex layout is not ideal either, as the user can get lost. This can be mitigated with simple map layouts and through audio and visual aids such

as noises to direct the user's gaze or lamps showing the direction of the exit [15]. These aids are not only to help the user navigate but also to magnify the user's immersion.

2.1.2 Motion Sickness and Other VR Problems

Motion sickness is an issue that occurs to some users when using VR headsets[16]. To minimize the problem, some things have to be considered when developing a VR application. The problem of motion sickness stems from the fact that one can move around in a scene and experience a difference between body and sight. This is comparable with the dizziness some humans experience when traveling by vehicle. The eyes see one thing while the body feels another, and these mixed signals are confusing the brain. As we briefly mentioned in subsection 2.1.1, motion sickness is bad for the overall VR experience. However, a low FPS is not the only way to induce such a feeling, as well as a high amount of FPS is not the only way to combat that feeling. In the paper *How to Deal with Motion Sickness in Virtual Reality* written by Bruno et al. they provide a list of things to do to avoid motion sickness, as well as a list of things to do to induce motion sickness [16]. The recommendations always include corresponding responses to user input, including a frame of reference and treating the camera as eyes rather than a camera. This means that the possible movement and interaction should be as realistic as possible and maintain the immersion from start to end. Things to avoid include motion acceleration of the character and momentarily taking away control from the user's view, which is disorienting and induce motion sickness.

Avoidance of motion sickness is closely tied together with what is considered to be a good VR experience, as previously written about in subsection 2.1.1.

2.1.3 Interaction Design and UX in VR

When using VR applications, the user will be required to be more actively involved than they would be using a conventional application [17]. In the VR environment, users will also have a great deal of freedom to control their movements and interactions. Therefore, ensuring the user's safety and making them feel comfortable will lead to good interaction, which is essential for having a good VR experience.

In a VR application there are often menus involved and therefore, all menus should be created with a good user experience in mind, and the interaction in the application should be as seamless as possible [18]. Therefore the best practice is to make menus in a VR game in 3D for the best usability. The focus does not have to switch between a 2D menu and the 3D environment. For example, a 2D HUD permanently attached to the screen may cause eyestrain and compromise the user experience through a loss of presence while also making the user more prone to motion sickness.

2.2 Cultural Heritage Preservation

Cultural heritage can be described as the legacy left by previous generations and is a collection of sites, objects, and traditions that brings together the history and the values of different cultures worldwide [19]. Cultural heritage consists of both tangible artifacts and intangible attributes. Tangible heritage can include physical items such as archaeological sites, buildings, books, games, clothing, and art. The pyramids in Egypt, the Colosseum in Rome, and Da Vinci's painting Mona Lisa are some examples [20][21][22]. Intangible heritage refers to abstract attributes such as languages, traditions, rituals, music, knowledge, and skills. Examples of intangible heritage are the practice of Chinese acupuncture or the Mexican festivity, day of the dead [23][24].

This is where virtual reality can play a role in not only preserving the culture but also immersing the user in the culture to interact with it [25]. Although the use of VR is more recognized in the entertainment field, in recent years, it has been used for cultural purposes and has become a tool for reconstructing historical environments that have either been destroyed or to preserve them without any effects on the original object [26][27]. This way of utilizing VR makes cultural heritage digitally accessible for educational and entertainment purposes [28].

2.2.1 Cultural Heritage Preservation in VR

VR brings other aspects to cultural heritage than just reconstruction, which an ordinary physical 3D model cannot recreate [25]. For example, VR allows for hypothetical virtual reconstructions of the past, leading to a deeper understanding of the object's history and why it has its appearance today.

What makes VR such a good candidate for cultural heritage compared to a digitized 3D model is its ability to provide the primary visible object and provide the user with the environment and the atmosphere that is desired for the object [25]. Since VR can create a connection between concrete data such as archaeological remains, and the atmosphere, it can more realistically reproduce the wanted scene in an immersive and more elaborate experience for the user.

2.3 Ethical and Social Aspects

As previously stated in Section 1, the use of VR extends to many fields. For example, VR applications can be helpful tools in fields outside of the entertainment industry [2]. The use of VR for educational purposes has shown to be a helpful tool for students learning, and engagement [2]. Some further examples include high-tech training in the medical and military fields, where people can be exposed to stressful and hostile situations without compromising their safety or be a danger to others [29]. It has also shown an impact on cultural heritage preservation by not only preserving historical sites digitally but also making them more accessible [30]. This, in turn, provides the knowledge needed to understand cultural values.

Like any technology, VR applications raise questions regarding the social impact they can have on individuals [31]. Though there is research suggesting that VR can help with social isolation by letting users explore different real-world simulations, other scientists suggest otherwise. An overuse of VR could lead to addiction, self-isolation, and dissociation from the real world. Such an overuse could cause a strain on the individual's mental health and the relationships they have around them. Due to the high immersion of VR, experiencing intense emotions could have lasting effects on the user, especially if the content is deemed violent [31].

2.4 Hardware

For developing and testing a VR application, adequate VR equipment is needed. Furthermore, to ensure that the application is usable by a wide variety of users, it needs to be developed with the consideration of different brands of VR hardware. Specific implementations are sometimes needed due to the brands varying in functionality. Two of the most common types of equipment brands are HTC and Oculus. Both brands of equipment are commonly used by users on popular platforms like *Steam* [32].

2.4.1 HTC VIVE

The HTC Vive is a virtual reality headset created by HTC and Valve as their partner [33]. The headset was initially announced in 2015 and released in 2016. In order to be able to walk around freely, two base stations are needed, placed on opposite sides of the room tracking the positioning of the user. By combining the headset with two wireless handheld controllers, the user can interact within the VR environment.

2.4.2 Oculus Rift S

A direct competitor to the HTC Vive is the Oculus Rift S. The Rift headset, which is currently developed and manufactured by Facebook inc [34], is unlike the HTC Vive. The Oculus Rift S does not require base stations and is only tracked from the headset. The Oculus Rift uses five cameras on the head-mounted display to track the surroundings and an accelerometer, combined with advanced software to predict the path most likely taken. The Oculus Rift also can be combined with two wireless handheld controllers, which are tracked similarly to the headset. Instead of having cameras, the controllers have infrared LEDs.

2.5 Software

Many different kinds of software were used throughout the project. This section aims to provide the underlying theory behind each software.

2.5.1 Unity

Unity Real-time development platform is a game engine that can be used to develop games and graphical applications across multiple platforms [35]. Unity has, among a lot of other things, built-in support for developing applications in VR. Furthermore, Unity provides an extensive library of features that simplify the design of various assets that are included in the application, such as adding video players and menus, and also VR libraries such as SteamVR, Section 2.5.6. In addition to this, the Unity engine makes it possible for the developer to create their own features using the *C#* programming language. Generally, a well-known engine like Unity has extensive, easy-to-understand documentation, and its community provides multiple sorts of guides and examples that might resolve issues that occur during the development phase.

Unity differs from other real-time development platforms such as *Unreal Engine*, in the sense that it is considered to be more beginner-friendly. Unreal Engine is typically used when better graphical performance is an important feature of an application [36]. Another difference is that Unity scripting is done in *C#*, while scripting for Unreal Engine is done in *C++* [37]. This further elaborates on Unity's beginner-friendly environment since *C#* is considered to be an easier language to learn and use [38]. However, both these development platforms are capable of producing high-end graphics, and they both provide support for animation, terrain editing, VR, and more. The choice of the game engine for this project is argued for in 3.3.1.

In Unity, every object is called a *GameObject* [39]. A *GameObject* can not do anything by itself until properties are added to it. Different properties could include a script, a collider, or a material. These properties add to the functionality of the *GameObject*. A *GameObject* that is supposed to be a character, for example, needs to have different components in order for it to function as a character. By itself, it is only an empty container. However, when the different properties have been added, the *GameObject* is a character.

Scenes are the main way of working with the project in Unity [40]. Scenes are the workspace that contains all of the objects as well as how they interact with each other. It is possible to switch between scenes to categorize which *GameObjects* to load when, within the application. For instance, a menu scene could be used to show the user the main menu with different settings and the choice to play or exit the application. Following this scene, the main scene could be loaded when the user chooses to play from the main menu. The main scene, in this case, would contain the heart of the application and can focus on gameplay and mechanics, while the menu scene would contain all of the logic for the menu system.

The tools and features that Unity already houses are commonly used when developing an application. They are often used to implement physics, customize interaction between objects, and in the case of VR, improve immersion, presence, and performance.

2.5.2 Unity Technologies Used

In Unity, there exist different light properties that can be utilized. Some examples of properties are Point lights, Spot lights, and Directional lights [41].

In Unity's manual, where they describe types of light [41], they describe that a Point light sends out light equally in all directions, where the light diminishes the further away from the source it gets. Furthermore, they describe that a Spot light works similarly to a Point light, but it sends out light at an angle, in a cone shape. The Directional light is described as sending out light in a direction only as if it was coming from a sun infinitely far away.

"Shadows add a degree of depth and realism to a Scene because they bring out the scale and position of objects that might otherwise look flat" [42]. Shadows is a feature that can be turned on for specific light sources, enabling the light source to cast shadows. If a light has shadows turned off, the light will pass through material and walls, henceforth being called as *light bleed*. Similarly, a GameObject has to have the option to cast shadows enabled to create shadows, and the GameObject that will receive the shadows has to have the option to receive them enabled as well. A GameObject can cast shadows onto itself, however for this to work, both the option to cast and receive shadows need to be enabled [42]. For shadows to exist, there must be a light source with shadows enabled, a GameObject to cast shadows, and a GameObject to receive them. If any of the three are omitted, a shadow will not show.

All of the calculations for lights in Unity are done in real-time, utilizing hardware performance [43]. Baking lights into a lightmap is an option for the Unity engine to perform the calculations for the lights in the editor, meaning the calculations are done in advance [43]. Baked Lights are useful for lighting things that will not change at runtime, and reduce the shading cost at runtime as well as reduce the rendering cost of shadows.

The limitations of baking a lightmap of the Unity scene is that the property of the light cannot be changed at runtime and that dynamic game objects do not receive lights [43]. A third limitation is that a baked lightmap does not contribute to specular lighting, meaning there are no bright spots on shiny objects.

"Occlusion culling is a process that prevents Unity from performing rendering calculations for GameObjects that are completely hidden from view (occluded) by other GameObjects" [44]. This means that objects set to occludees are not rendered as they are hidden from view behind an object set to occluder, improving performance. An example of how occlusion culling works can be seen in Figure 2.1.

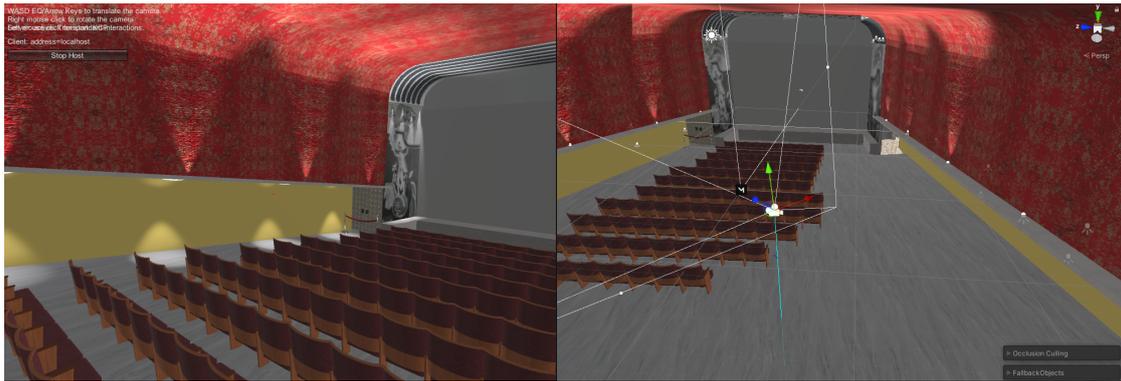


Figure 2.1: Occlusion culling in Unity. What the player see is to the left, and what the Unity engine renders to the right

2.5.3 3D Modeling

When modeling the different objects for the VR application, there are some things to consider. Since there is a great focus on performance, the meshes and textures should have as little impact on the performance as possible, without conflicting with the quality and the immersion of the application.

One thing to consider is the scale of the assets that are being modeled [45]. It is often useful for designers and artists to use realistic measurements when modeling assets for an application, but in VR, that becomes even more important to preserve immersion and a positive user experience. A VR experience does not always have to be realistic, but it is important that it is at least believable and that textures, meshes, and objects that the user can interact with are sized in a reasonable manner. The scaling and positioning are perhaps the most important for objects that users can grab, such as door handles and weapons. Objects that represent things that users interact within everyday life, such as light switches, cutlery, chairs, doors, also need to be realistic in regards to size and positioning in the application space [45].

In the case of 3D modeling for VR, there are two major things used to add detail to a scene: geometry and textures [45]. While adding more geometry always will look more realistic, it is not the most optimized way. The polycount in each frame has to be lower for a VR application compared to a regular desktop application. This means that at some point in the modeling process, geometry has to be substituted by textures to approximate the assets. This might, in the end, appear less realistic, but it is a necessary step in order to achieve a good user experience [45].

Sketchup is a 3D modeling software for a wide range of drawing applications such as architectural, interior design, landscape architecture, civil and mechanical engineering, film and video game design [46]. Despite its many applications, Sketchup is mostly used within the area of architectural design. Sketchup is not designed to be used with animations, and there is no support for this in the software [47]. Thus, modeling video game characters and other interactable objects might prove to be easier done in another 3D modeling software.

When modeling assets for games, a commonly used software is *Blender*. Blender is a free and open-source 3D-modeling software that includes tools appropriate for many different aspects of 3D modeling, such as rigging, modeling, animation, rendering, and so on [48]. The fact that Blender is free is partly the reason why Blender is a popular option when choosing software for 3D modeling, and such, Blender is a popular software amongst freelancers and smaller studios. Since Blender is an established program and widespread, there are many tutorials on the Internet, which makes Blender a software that is easy to learn and to use [49].

2.5.4 Blender Technologies Used

Duplicate Linked is a tool included in Blender, which allows the use of several duplicates of the same object, but only needing to load one [50]. Through duplicate linked objects, every duplicate object has linked all its data to the original object, meaning only data for one object has to load.

2.5.5 Assets

In Unity, there is built-in support for using assets, either by importing them manually or by downloading them from the Unity asset store [51]. An asset is something that someone has created and is sharing with the public, either for free or for a fee. An asset can be anything from a model, a texture, or modules that solve different things, like a framework for setting up a server for the application or ways to communicate.

SteamVR is a plugin that makes it possible for the application to communicate with the SteamVR application [52]. Furthermore, it contains prefabs and scripts that make it possible to set up and interact with a VR environment.

UniGLTF is a plugin that makes it possible to import and export GLTF 2.0 files in Unity, to get the correct materials [53].

The *Sketchfab* asset makes it possible to import and export models directly from the Sketchfab website, as well as importing GLTF files in Unity [54]. This is, in essence, similar to the UniGLTF plugin.

Github for Unity is an extension that integrates GitHub directly into Unity, making it possible to communicate with GitHub without leaving Unity [55].

2.5.6 SteamVR

SteamVR is a software product developed by Valve [56]. It supports a variety of VR headsets and is needed in order for some headsets to work. SteamVR works as a bridge between the VR hardware and the applications that use it. This allows standardized interaction with different VR applications. To develop with the SteamVR application, the SteamVR framework needs to be used. By using SteamVR, any game or application could be run on any headset that is supported, rather than

separately having to develop for each headset. Thus, making it easier for both the consumers as any supported headset could be used, as well as for the developer.

SteamVR also has a feature that allows for the developer to assign actions to different buttons of the controller, giving control over how the user can interact with different things within the application [57]. The way that it works is that actions defined are mapped within the SteamVR application, and then these can make a call to the code.

2.5.7 Mirror

Mirror is an API that handles networking, and it makes it possible to connect different players to the same session, making it possible for the players to watch the same video in the same room [58]. Mirror was created by a few Unity users when Unity's native networking functionality was discontinued. Therefore, Mirror's core is almost identical to the deprecated Unity networking core.

To establish a connection, a user starts the application containing Mirror as the host, thereby opening a connection on their machine [58]. Other users can then connect to the host without running any different code or application. The reason behind keeping the host and client functionality within one project is to achieve a significant gain in productivity for the users.

2.5.8 Version Control

Git is a widely used and very popular type of version control system [59]. Git is designed as a linear workflow in the form of a timeline with different branches. The timeline consists of different commits of work made by users and can be branched out of at any point, effectively copying the project's current state while allowing for further changes without changing the original timeline. At any point, the different branches can be merged back with the original timeline or any other branch to synchronize the work that has been done on the different timelines.

Many different development platforms support the git version control system, the most popular one being GitHub [60]. Github allows entire teams to work together on projects by hosting workspaces on their cloud servers. This allows teams to use git to collaborate on different projects or code bases while uploading the project to Github. Github also features different tools for teams to customize their workflows to better fit their specific needs, such as pull requests. Pull requests allow team members to submit a request with their recent changes to a branch which other team members can comment, approve or request changes on. This allows for smoother review of the code within the project and better collaboration between team members.

To ensure that there is always a consistent, working version of a project, teams tend to never directly commit their work to a master branch [61]. By doing this, there is always a stable and working build of the project ready to be deployed at any time.

2.6 Agile Development

Agile software development is a modern and popular collection of methods for project management [62]. It stems from a group of software methodologies that revolve around iterative development. This means that projects are developed continuously and after release instead of the contrary, focusing on one big release date to work towards.

The development phases in an agile workflow are referred to as *sprints* [63]. A sprint defines a time frame in which the development takes place. A sprint may vary in length from project to project but is commonly between one to four weeks and is usually thereafter fixed for the rest of the project. The first day of a sprint is where *sprint planning* should take place, in which work is moved from a product backlog to a *sprint backlog*. This work is often expressed as tasks formed after a *user story*, which follows a specific format to ensure that all parties involved in the project have the same vision. These user stories are supposed to be created or at least revised by the stakeholders to quality assure the project [63].

When the tasks for a sprint have been laid out, the tasks are supposed to be estimated by the team members to quantify their respective effort needed to be completed. There are several methods to accomplish this, and one method to help with estimations is to estimate each task with points following the Fibonacci sequence [64]. This sequence fits great since the numbers are more easily distinguishable than just following a sequence of a subset of the natural numbers.

To organize the user stories and tasks for a project, a scrum board is often used [65]. The scrum board allows for a good structure as well as an overview of the current state of the project. A typical example of a scrum board would be Trello [66]. Trello allows its users to structure their projects in the form of a board that contains several lists of cards. Each user story and task is represented by an individual card. When a task changes its status from one state to another, this can be represented by moving the corresponding card between different lists.

2.7 User Tests

User tests allow developers to get feedback from users who have never previously used or seen the application and therefore provide insight into problems that the developers have become blind to [67]. The users participating in the test should preferably be a part of the target audience for the application. This allows the test results to result in changes in the application that would be of great benefit for the end-users [68].

A technique that could be applied to user tests is *Think-Aloud*, where the test participant expresses their thoughts out loud while conducting the tests [69]. There are several benefits of this method. It is cheap and easy to learn, meaning that no extra equipment is needed, nor does the user have to adapt to new technology that might disrupt the testing. It is also a flexible technique that can be applied to

all stages in the design process, from early paper prototypes to fully implemented systems. However, it might be an unnatural situation for the test user since most people do not talk loudly to themselves during the day. There is also a risk of filtered statements, where the user does not communicate some experiences because they want to appear smart or do not want to annoy the moderators with too many comments on what they experience as faulty behavior in the application [69].

Before or during the test, the test user is usually told to perform specific actions within the application [68]. The test's purpose is often to see if the test user gets confused or lost in the application, if they can complete the tasks they are asked to complete, or to detect potential bugs. These test sessions can also be moderated, where the test user or the moderator can ask questions during the test to further detect software/design issues.

2.7.1 Avoiding Ethical Issues

To avoid ethical issues, a project should have a user-focused approach from the beginning and throughout the project [70]. By doing thorough research with stakeholders, fewer mistakes can occur and thus, minimize ethical issues. When a design has certain issues, the users would be the most affected by them. Therefore, it is also important to conduct research studies with expert users, end users, and have users participate with the designed product. When involving users, the standard protocol is to get their informed consent since their data is being gathered and used [70].

Additional ways to avoid ethical issues are discussed in sections 3.1 and 3.4, developers can make use of agile software development and conduct user tests to ensure a good user experience [70]. Conducting user tests towards the end of the design process informs the developers if they succeeded with avoiding certain issues, but the essential part to avoid the issues is doing good research beforehand. By getting the user's consent and involving the users in the design stages, the process becomes more interactive, which increases the chances of making a better and usable product [70].

3

Methods

This chapter will present the methods used in this project. It includes how the user tests were performed, which hardware, software, assets, and tools were utilized. The practices applied during the development are also presented here.

3.1 Agile Development

In this project, an agile approach was used, and the project was developed in sprints. The length of each sprint was decided to be exactly one week long, and each Tuesday, a new sprint was started with a sprint planning session. The work was followed up on each Friday, where a mid-sprint meeting took place. In these meetings, the sprint was discussed amongst the team and the project supervisor, who acted as the stakeholder.

3.1.1 Scrum Board

To support the agile workflow, a Scrum Board was used. The Scrum Board was hosted on Trello and contained all of the user stories and tasks in the current sprint, as well as the archived tasks of the already completed sprints. This is shown in Appendix C.2.

Epics had a designated number ranging from zero to four. User stories were defined with the corresponding epic's number, followed by another number to uniquely identify it. Thereafter, tasks were formulated based on the user stories, and they yet again received an additional number.

Each task was given a number in the Fibonacci sequence to determine the estimated effort, which corresponded to a color in the Scrum Board. A reasonable total effort for each sprint was concluded to be around 60, meaning an effort of 10 per person in general. Depending on this total number, the week's tasks were set, as well as which task the group members were assigned to.

3.2 Hardware

The group had the opportunity to borrow two VR headsets for testing, namely an Oculus Rift S and an HTC Vive. The borrowed headsets were passed around the group members each sprint to allow different members to work on VR tasks that required a headset to test. Additionally, this allowed different members to be testers for the pull requests in the GitHub repository that required VR headsets to be fully tested.

3.3 Software

This section will present thorough explanations on the utilization of the different software mentioned in the chapter Theory.

3.3.1 Unity

Unity was the chosen game engine for this project. Unity allowed the project members to simultaneously work on the project within the same scene but in different git branches.

3.3.2 Mirror

The Mirror asset within Unity allowed the implementation of network functionality into the project. When implementing Mirror, we exposed a simple network-object representing the player to the other people connected to the network while maintaining the more advanced mechanics and logic locally.

3.3.3 SteamVR

The implementation of SteamVR in Unity proved to be more difficult than expected, as it was found that getting both headsets that were used for testing to work was difficult. A beta version of SteamVR was released during development which solved the problems that existed, and the Unity version that was used was updated. With the added features from the plugin, the development with SteamVR became a more positive experience. For example, the SteamVR feature for key bindings was used to implement tailored customization for the different types of VR equipment.

3.3.4 Version Control

Git was used as version control for this project with the help of GitHub. The git structure of the project revolved around two main branches, Master and Dev. The Master branch was intended as a kind of production branch, where no active development would be made, and instead, it would always be the latest stable build. The Dev branch, on the other hand, was the main branch for development during the sprints. For each task in the sprint, a new branch was created from Dev. When the task was completed and everything was committed to the specific branch for

the task, the task would be merged with the Dev branch after it was tested by the designated testers. At the end of the sprint, when all of the task-specific branches were merged with the Dev branch, the Dev branch would be merged with the Master branch to update the branch with the latest stable build.

3.3.5 3D Modeling

The group began with modeling smaller objects such as specific furniture and objects to interact with. To then implement the work on an actual model of the building, a basic, placeholder model of Flamman was created with the software SketchUp. This software was chosen due to its ease of use and ability to quickly create models. This placeholder model of Flamman can be seen in Figure 3.1 below.



Figure 3.1: The model of Flamman done through SketchUp

Due to the SketchUp model only being a temporary placeholder, a new model had to be created. This new model was decided to be created in Blender, and the workload was divided amongst three group members. The first person would make the lobby and stairs, and the second person would make the cinema saloon and the hallway next to it. The third person would create all of the furniture, intractable objects, and other smaller models to be placed inside Flamman.

The cinema chairs in the new model were placed utilizing the tool Duplicate Linked. This tool made it possible to place all of the chairs in the saloon with the data linking back to a single chair. Since only two types of chairs were needed, one to start a row and one to end it, 458 chairs could be placed while only having the need to load data for two.

At first, when the members modeled different things, there was no need for file collaboration. When all three members were finished, the files were sent as a .zip file to one user, who then had to import the files and combine them into one project file. Unfortunately, when the models were put together as one model, the file size became too large to send between the members through GitHub. The solution for

the large file size was accomplished by dividing the textures and the Blender file, making it several files rather than one. Furthermore, some of the texture files were quite large, and they had to be compressed to comply with the GitHub file size limit.

After importing the new 3D model into Unity, all that was needed was to move the features from the placeholder model to the new model, and move the location of GameObjects into their new position.

Everything was created by the group except one thing, the model of the tree in the lobby. The model of the tree was instead downloaded for free through the online 3D model trading website cgtrader [71].

3.4 User Tests

The user tests in this project were outlined to test the entire application in two different states. The first state was a prototype of what the final application would look like, and tests performed on this state were intended to give rough feedback on the usability and intuition of the application. The second state was closer to a final product, and the tests here were intended to give feedback that allowed for final fixes for any minor issues in the implemented features.

The user tests were conducted by two group members, where one was the acting supervisor for the test, and the other member was documenting the test. The group contacted persons with access to their own VR equipment. The user then conducted the test on their computer with screen sharing on for the test conductors to study their behavior. These tests were designed to test the main functionality of the application, as well as testing the overall intuition and UX. All of the user tests were performed with the think-aloud method while performing the tasks given.

After each state of user testing, the results were compiled into a document detailing the problematic tasks, what the problem was, and the solution for it. To each task, the number of people who had problems with it was also added. These problems were divided into three categories, functionality, interaction design, and aesthetics.

4

User Tests

To get input from the end users and to minimize bias, user tests were conducted. This was the basis for the last sprints as there were details important to the user experience of the application. The results from the user tests provided constructive information to develop the application further. The tests were conducted in two different sessions, with one week apart, allowing for a full sprint of tasks to be completed in between each session. The sprints connected to the user tests contained tasks completely based upon the user feedback received from the tests, as this pointed out many parts of the project that needed to be improved.

The results from the evaluations can be seen in their entirety in Appendix A.

4.1 First Evaluation

In the first series of user tests, the results were unanimous and could be summarized into four main points. The points that most people had problems with were converted into tasks and had the highest priority in the next sprint. The points were mostly focused on functionality and user experience. The points with the highest priority were the following:

- *Start a movie* - There were unclear instructions, as well as a lack of feedback if the movie was chosen or not.
- *Movements* - How to move, as well as changing between different movement types, were unclear. Also, moving around seemed to be clunkier than expected.
- *Interacting with objects* - Not being able to interact with door handles or move the movie rolls around.
- *Navigation* - Not clear where the door to the projector room was, and there were no instructions on how to use the application.

4.2 Second Evaluation

In the second series of user tests, the results were not as unanimous as in the first series of user tests but were generally more focused on aesthetics. The points can

be summarized into three main points. The points that most people had problems with were again converted into tasks and had the highest priority in the next sprint. The points with the highest priority were the following:

- *Interacting with objects* - There were some bugs regarding occlusion culling that made some interactable object to disappear when moving them, as well as the door in the cinema room not being interactable.
- *Manual* - It was hard to read the text in the picture describing the key bindings.
- *Lights* - The lighting in Flamman was not realistic.

4.3 Changes Made and Conclusions from the User Tests

The first evaluation resulted in some drastic changes to the usability of the project. The main focus was on the instructions that the user receives when the application starts, since this is what a lot of the feedback was focused on. To respond to this, a tutorial-board was added in the lobby of Flamman to show the user how the application is supposed to be used. The second aspect of usability was the ability to interact with objects in the environment. To address this, functionality to lift particular objects was added, specifically to the movie rolls in the projector room and the plants in the lobby. Furthermore, the functionality to reach and grab the door handles was added instead of pressing a button on the controller. Another aspect that was added was the possibility to move in any direction as opposed to only in the direction of that the user is looking in.

The second evaluation resulted in changes that were centered around the instructions of the controls, bugs, and the lighting in Flamman. The scale of the player model was changed to be more realistic, and the door model was changed so that the door handles were placed at a lower level. The instructions for the controller mapping were moved to the wall to be easier for the users to read. Barriers were also added in front of the doors that are not interactable, as not to confuse the users.

In total, there were four different testers that could give feedback on what should be done or changed. By asking the testers to do basic tasks, the observers were able to note what was hard to understand. Furthermore, by asking the tester what they thought about the environment and the experience, changes could be made to lighting, shading, and modeling, among others. Thus having the ability to finalize the application in a more user-friendly way.

5

Results

All of the three goals set in Section 1.3 have been accomplished. It is possible to watch a movie within the virtual cinema, alongside having the ability to have other users join a session to interact with. The environment of Flamman has also been reconstructed according to the bachelor thesis proposal.

The entirety of the project can be seen on the Github repository for the project [72].

5.1 Final 3D Model of Flamman

The final 3D model of Flamman has a close resemblance to the pictures provided by the project supervisor. Figures 5.1, 5.2, 5.3 and 5.4 shows comparisons of the final model to four original pictures. In Appendix B, there is a comparison to the newer pictures of Flamman today. The Appendix also contains pictures of the separate objects existing in Flamman with no direct comparison to Flamman. Furthermore, there are more pictures of the 3D model. However, these pictures do not have a corresponding picture for comparison.



(a) Picture from the original Flamman

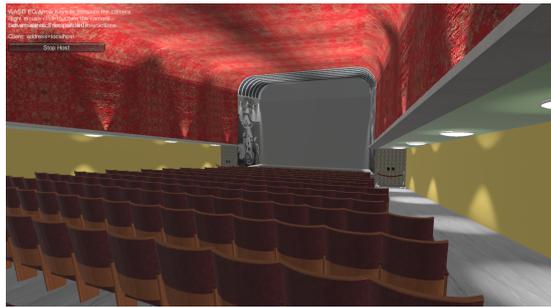


(b) Picture from the application

Figure 5.1: Side by side comparison of the lobby



(a) Picture from the original Flamman

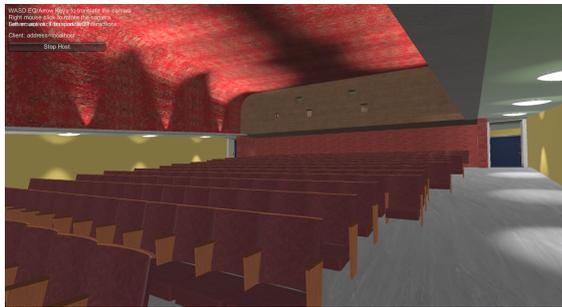


(b) Picture from the application

Figure 5.2: Side by side comparison of the front of the cinema salon



(a) Picture from the original Flamman

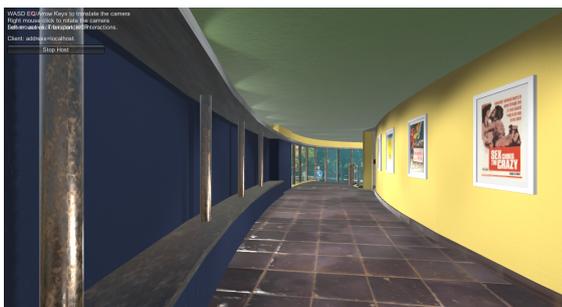


(b) Picture from the application

Figure 5.3: Side by side comparison of the back of the cinema salon



(a) Picture from the original Flamman



(b) Picture from the application

Figure 5.4: Side by side comparison of the corridor

5.2 How to Start the Application

As the user starts the application, the user sees a black screen with three alternatives in the top left corner, *Host (Server + Client)*, *Client* and *Server Only* seen in Figure 5.5. Host is the option to choose if the user wants to start a server. The user is

now hosting the server and is also a client of the server. If the application should be used for multiplayer, either Host or Server Only can be picked. However, if choosing Server Only, the user can not join the server as a client. If the user is joining an already existing multiplayer session, the option Client should be used, and the IP address of the host should be the input in the text field.

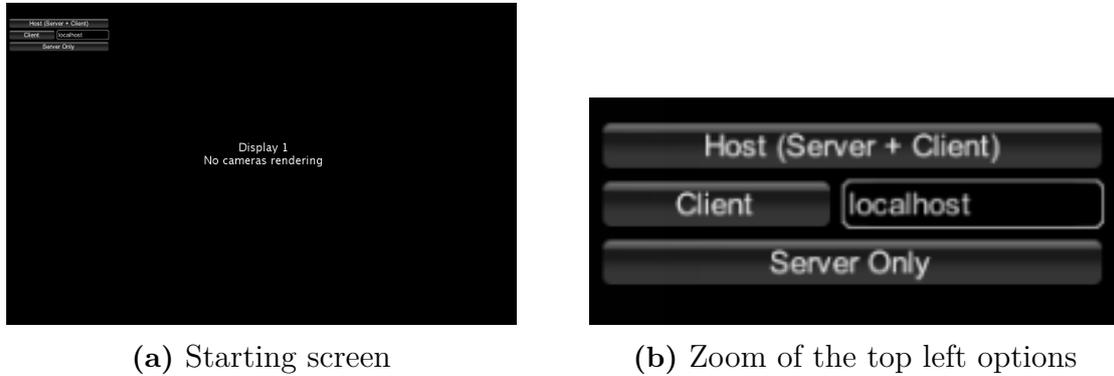


Figure 5.5: The starting screen of the application

5.3 How to Start a Movie

After choosing between the options of either starting or joining a session, the user is placed in the lobby facing two large boards, seen in Figure 5.6. The user is now free to explore Flamman. To understand how to start a movie, it is necessary to read the tutorial board. The left board contains a picture with information about the controller keybindings that change depending on the user's headset. The right board contains a tutorial on how to play a movie.

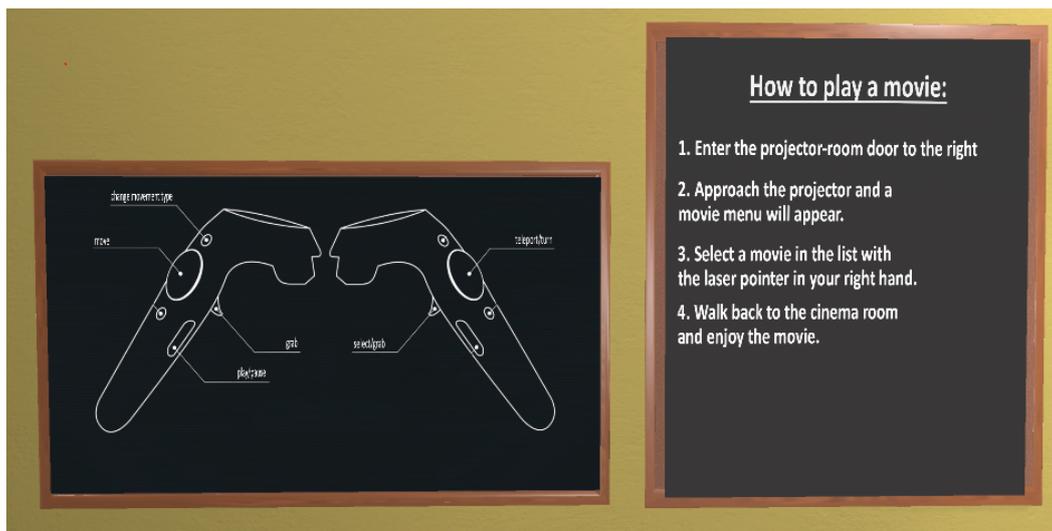


Figure 5.6: Keybind and tutorial boards in the lobby

The first step is to enter the door to the projector room with the text *Projektorum*,

placed under the stairs to the right of the board. This door can be seen in Figure 5.7. To enter the door, the user has to grip the door's handle with the grab buttons of the controllers. The user is then instantly teleported up to the projector room, as shown in Figure 5.8.



Figure 5.7: Projector door to teleport to projector room



Figure 5.8: The projector room

Step two of the tutorial board tells the user to approach the projector for a movie menu to emerge in front of the projector, showing the available movies. Step three describes to the user that a laser pointer to select a movie will appear when being close to the menu. Aiming the laser pointer at a movie in the list and pressing the select button will start the movie instantly. A small pink pop-up indicates that the movie has started, seen in Figure 5.9.

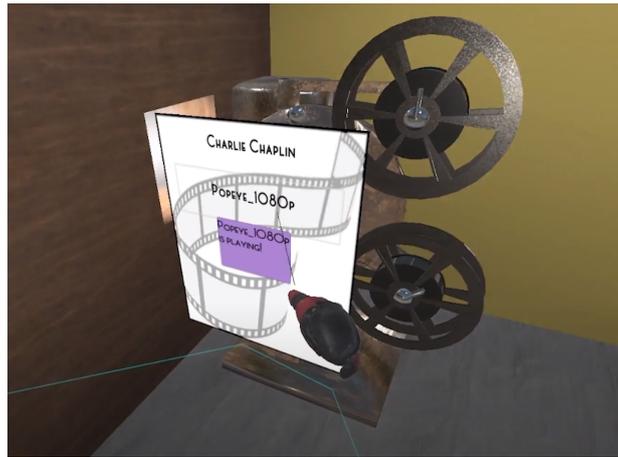


Figure 5.9: The menu to choose a movie

Another indication that the movie has started is that a small point light will shine bright in front of the projector, and the lighting in Flamman is dimmed. When the movie is playing, sound can be heard from the projector screen up to the projector room, which also acts as an indication.

Step four advises the user to walk to the cinema saloon to better view the projector screen. This is done by teleporting back to the lobby by grabbing the door's handle in the projector room with the text *Utgång*. Now that the user is back in the lobby, the user has to get to the cinema saloon by walking up the stairs and then enter the saloon. When inside the saloon, the user can choose to sit in any of the chairs available. It is possible to pause and play the movie with the pause button on the controller, as not to require the user to walk back to the projector room.

5.4 Other Gameplay Mechanics

The user can move around in different ways in this application, namely teleporting or walking. The type of movement is changed by a specific button, visible in the keybindings board in Figure 5.6. When teleportation is activated, the user can freely teleport by aiming with the controller towards the ground and moving the controller joystick upwards. Moving the joystick to the sides activates snap rotation, and the user rotates in that direction. If the second movement type is selected, the user can instead walk around with the joystick in all directions.

To interact with objects in the scene, the user has to reach for them with their hand and press the grab button, similarly to how a user opens the doors.

5.5 Virtual Reality Experience

The user can always freely explore Flamman. To stop the user from walking outside the model, a red band with a sign saying *Ej utgång* is placed in front of each of the exit doors. This red band can be seen in Figure 5.10.



Figure 5.10: Red band to block the exit

Before starting a movie, the entirety of Flamman is well lit by bright spotlights. When walking in the corridor outside of the cinema saloon, the user can see eight movie posters encased in a white wooden frame and a glass top, seen in Figure 5.11.



Figure 5.11: One of the eight posters in the corridor

When inside the projector room, as seen in Figure 5.8, the user can walk and look around freely. When looking to the right, the user can see and interact with the movie rolls placed on both the table and the shelf. If the user is looking to the left, the user can see the cinema saloon through the windows.

The lighting in Flamman changes when the cinema is playing a movie, dimming all of the lights to a lower intensity. In the saloon, every light is turned off, making it almost entirely dark. To light up the projector screen, a single light is shining on the projector screen, from which a small amount of light reflects onto the cinema chairs

for illumination. This reflection is static and does not dynamically reflect what is happening on the screen. The sound from the movie is coming directly from the movie screen, making the user feel as if the speakers are placed there.

Since the chairs in the saloon are folded down permanently, it is not possible to interact with the seats. To give the user the illusion of sitting down in VR, the user is advised to sit down in a chair.

5.6 Multiplayer Experience

Several users can connect as clients to the same server using the IP address provided by the host. All players have an avatar that consists of a head and two hands seen in Figure 5.12. The users can see each other, however, they can not talk to each other through the application. All users will see the same movie being played in the cinema, provided that they are using the same video files.



Figure 5.12: Two users that are waving to each other

5.7 Performance

High performance and a high refresh rate are essential, as previously mentioned in Section 2.1.1. The FPS when playing a movie is steadily 80, with a refresh rate of 80-90 Hz with no stuttering. However, when no movie is playing, both the FPS and refresh rate dips down on some occasions. The worst drop is when a player stands by the movie screen and is looking in the direction of all the chairs. In that instance, the FPS drops down to about 25-26, and the refresh rate is about 31 Hz. When being in the lobby, the FPS is steadily 80, with the refresh rate being about 80-90 Hz, just as when no movie is playing.

6

Discussion

This chapter will discuss and analyze the the project's results, and will go further into detail of the thought process of the decision making.

6.1 Choosing Unity

Unity was chosen as the preferred game engine for this project. This was partly due to the beginner-friendly environment and scripting language it offers compared to Unreal Engine. The project members also had an interest in learning more about Unity. Some group members had previous experience in Unity, which made Unity the best choice for the group.

6.2 Performance

As mentioned in Section 2.1.1, high performance is important for a good user experience since low FPS numbers are likely to induce disorientation, nausea, and motion sickness. Having to balance high performance while also ensuring high visual fidelity was a complex task.

The result of the performance test made in Section 5.7 was done with a computer containing an Intel i5 10400F processor, an Nvidia RTX 2060 Super graphics card and 16GB of RAM. As such, the performance of the application may be lower on a computer with less powerful components.

6.2.1 Lighting in Unity

Every light in the scene needs computational power to calculate. It was therefore important to optimize how the lighting was used in the application to improve performance. It was found out that the point lights used are more difficult to process than spotlights, as a point light can be described as six spotlights aimed outward as a sphere. The point lights were thereby replaced by spotlights to achieve the same effect but lowering the amount of light to calculate, greatly improving the performance.

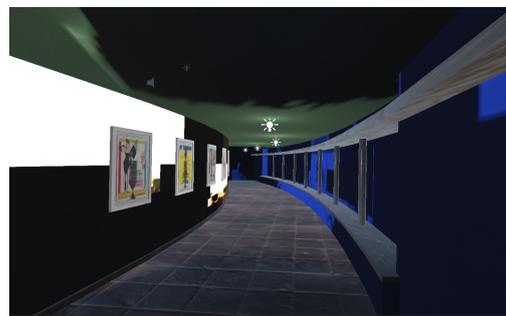
In order to make the experience more realistic, shadows were implemented, as this

was something that the participants in the user test wanted. The implementation of shadows solved an earlier problem of light bleed that was apparent when the cinema was dark, Section 2.5.2. Some performance loss occurred by implementing these shadows, but it was a necessary solution and to resolve the user feedback.

The process of baking lights could have solved the majority of issues regarding the experienced light performance, Section 2.5.2. However, this produced poor results, showing irregular black and white spots on the walls, as seen in Figure 6.1. Baking the lights took about three hours for a computer to render, and the result was consistently disappointing, even with different settings. The reason as to why this problem persists is unknown and could be due to many factors, such as interference between the many light sources or modeling issues in the 3D model.



(a) Baked lights in the saloon



(b) Baked lights in the corridor

Figure 6.1: Result of baking the lights

6.2.2 Cinema Chairs

A problem of poor performance was presented when the cinema saloon chairs were added to the 3D model. After checking the system diagnostics, it showed that the computer was not utilized to the full potential, indicating a bottleneck in the system. Investigation showed that the cause of low performance was a difficulty to load the data of all the chairs. Using the tool called Duplicate Linked in Blender, it was possible to make all of the chairs link its data back to a single chair. This removed the need to load data for every chair, immediately fixing the problem of the bottleneck.

To further increase performance, a new, more basic model of the chair was created with a more simple design, lowering the number of polygons of each chair by a considerable amount. Moreover, the textures of the chair were changed, making it less detailed which also improved the performance.

The total number of chairs was decreased as well, from 684 to 458, removing some rows and columns of chairs. To compensate for the chairs missing, some were enlarged, and the distance between rows widened. The reasoning for the removal of chairs was because, when inside the VR application, the user would not notice the difference. In addition, improving the performance was prioritized rather than the accuracy of the original design.

6.2.3 Occlusion Culling

The final solution for a higher performance was a process called occlusion culling, described in section 2.5.2. This was implemented by setting the walls to occluders and setting all of the chairs, lights, and other small objects to occludees. This enabled these occludee objects to not be rendered when hidden from view behind the walls. The most considerable performance gain is the lobby and corridor, offering great improvement by not rendering any of the chairs. In the saloon, where the chairs are seen, the effect of occlusion culling is less significant but still made a noticeable improvement.

6.3 Source Material

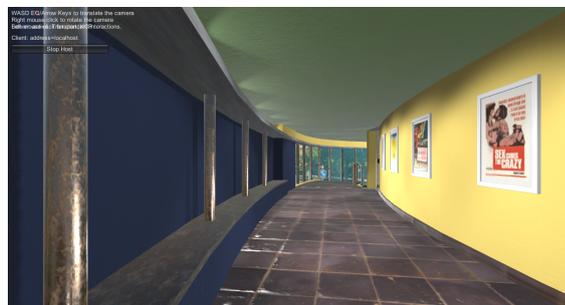
An issue that was discovered was the lack of documentation of Flamman. Due to it being such an old building, a few number photos of Flamman existed in its original state. In a collection of photos, given to the group by the project supervisor, containing 138 photos of Flamman, only 15 were of its original blueprints and 6 black and white photos in its original state. The remaining photos were from 2019, taken by the supervisor of what was a repainted and renovated Flamman. No information could be found about colors and materials at first, leading to most walls and objects being in white, as in the newly renovated photos.

The supervisor later compiled a document containing information about the colors of the walls and roof. This document contained gathered information from two blogs [73][74]. The additional information was still insufficient, with walls being described as dark blue, and the roof as green, allowing interpretation of different shades. The walls in the cinema saloon were described as being covered in red and yellow velvet and the roof as having speckles of gold.

To solve the problem of the uncertain colors, the supervisor created a color guide to follow. This color scheme made from a collection of photos and color schemes from the interiors of similar modernistic buildings of the 1930s. This color guide can be seen in Figure 6.2.



(a) Color scheme constructed from collection



(b) Color scheme from the application

Figure 6.2: Side by side comparison of the corridor

6.4 Size of The User in VR

An unforeseen problem that emerged was the size of the player avatar, in relation to the size of the 3D model. After importing the model into Unity and testing with the player avatar, it was discovered that the height was too high, as door handles and railings were at eye level. At first, it seemed only necessary to increase the height of the avatar in Unity, but this led to problems with stretching the hands in disproportional ways, and the tracking of the user's laser pointer was off.

It was instead required to increase the width, length, and height of the character proportionally, which introduced a new problem. The player's avatar became too wide and was not able to walk between the rows of chairs in the cinema saloon. The solution to this was to remove the collision of the chairs and the avatar, by placing an invisible wall at the backrest. This prohibits the user go through them. It was not originally intended but it improved the experience, as the user could walk closer to the backrest when the character no longer collides with the chair seat, meaning a more realistic experience when sitting.

6.5 Creation of the 3D Model

An encountered issue during the process of the project was the creation of the 3D model, considering how it was unknown whether it would be provided to or created by the group. Instead of waiting, it was decided to create the 3D model ourselves, a decision that was detrimental to the project's established plan, which had to be adjusted.

Blender was chosen for the creation of the 3D model as it is a free 3D modeling program and one of the group members already had experience with it. Another reason is the existing resources with Blender tutorials which makes the software is easy to learn and use.

With the help of blueprints, it was simple to 3D model the saloon and the lobby with the correct dimensions. However, the height of the building difficult to interpret from the blueprints. Therefore some educated guesses had to be made with the help of the pictures, and adjustments made with the feedback from the user tests. Moreover, the textures were difficult to interpret in regards to the 3D modeling and additional estimations had to be made.

By creating a new 3D model ourselves, it implemented additional features more conveniently as the model could be custom-built for said features. Changes and fixes could also be applied immediately, allowing a faster workflow without the need to communicate with a third party. This meant that the feedback from the user tests could quickly be changed between testing states. As the group was now the only developer, quality control could be managed.

The objects within the cinema, like the projector, posters, and bookshelf through estimations of how it could have looked like. There were no documentation describing

the projector room, and we took a guess of how it would have looked like, making models based on objects which was from the same period of time.

6.6 Limitations Discovered

During the progression of the project, several design choices were discovered and needed to be addressed. In this section, these choices and limitations will be discussed.

6.6.1 User Usability Versus Original Design

When creating the VR experience, it was understood that there exists a large difference between what a good game design is, and what is accurate for the recreation of Flamman's design. After a thorough discussion with the group and the supervisor, it was decided that in conflicts such as these, designing towards a good user experience is preferable.

One discussed limitation was how the user would navigate to the projector room and start a movie. It was discovered that the way to the projector room was very long, which can be seen in Figure 6.3. Implementing a walkway in the model would mean that a user has to walk unnecessarily long to reach the projector just to start a movie, and then walk the same way back to watch the movie. This implementation would result in a tedious experience. It was decided that time spent on designing this walkway could be used more sensibly. Whilst designing the walkway would be true to the recreation of Flamman, it was deemed a bad design choice for the user experience.

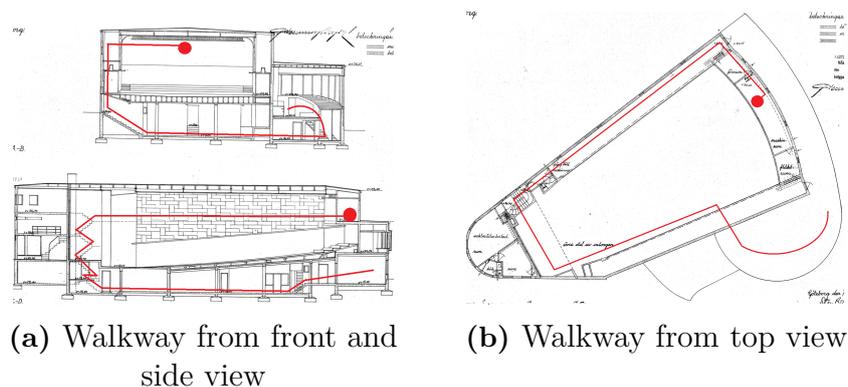


Figure 6.3: Walkway to the projector room according to the original blueprint

Another option was to create a small stairway leading directly from the cinema to the projector room. Although this solution would be very quick and intuitive for the user, it meant adding a walkway that did not exist in the original design, as seen in Figure 6.4.

The resulting design choice was to keep the door in the original place shown in the

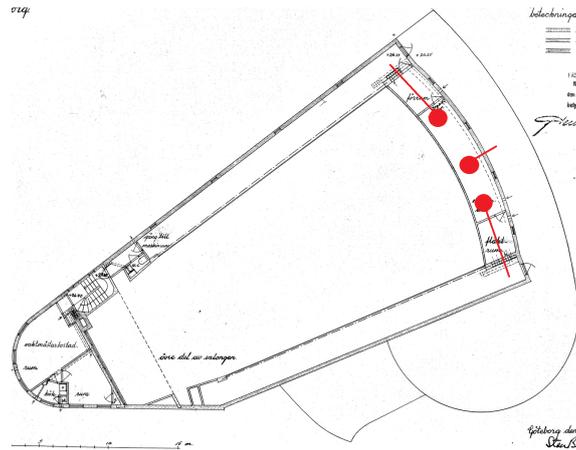


Figure 6.4: The three options for the discussed new walkway from the top

blueprints but removing the long walkway. Instead of the having the user walk to the projector room, the user is teleported there instantly.

In the end-product, choosing a movie is done by walking close to the projector where a pop-up menu will show the available movies to choose from, as discussed in 5.3. It was discussed whether or not choosing a movie would be better done by placing a movie roll containing the chosen movie inside the projector. However, this was never implemented due to other priorities. The original design of Flamman contained two projectors inside the projector room [73]. This was changed to only one projector in the finished application because having two projectors would be confusing for the end-user.

Another solution to the problem of reaching the projector room and choosing a movie would be to simply ignore it, by choosing the movie in the lobby by buying tickets. This was discussed during the start of the project, but it was decided that a projector room was something that was wanted.

6.6.2 Cinema Chair

It was stated early in the process of the project that it would be a nice feature to have the chairs react to a person sitting, meaning the seat would fold down. However, it was discovered that when implementing the duplicate link on the chairs to increase the performance, every chair would fold down, as they all would act as the same seat due to the data being the same. It was then chosen that every seat should be folded down as standard, and no such feature could be implemented.

The row of chairs in the final project is placed in a grid with no curve, not as the original design. To achieve the curve, each chair on a row had to be manually placed and rotated, being difficult to be done correctly. As the design of the chair only supported the alignment of chairs on a line, the chair had to be reworked for this curve. Thus, this idea was discarded, as it was thought of as too time-consuming for little gain, especially as the project's deadline was approaching.

6.6.3 VR Implementation

As previously mentioned in Section 3.3.3, SteamVR was the framework for the VR implementation. This was easy to use since most of the things needed to make a VR application are included. An example is the laser pointer that was used to choose a movie. However, by utilizing this prepared tool, other options were not considered. As discussed in Section 6.7.1, it was not obvious for the users to use the laser pointer to choose a movie. This raises the question, what other implementations could have been done to create a better VR environment. Not relying on a framework for the VR implementation would not have been an option, since the time needed for such a project would be extreme.

SteamVR was harder to implement than initially thought as it took more time and effort than a widely used framework would normally take. Furthermore, there even existed a known workaround since it is a well-known problem, called Tony's hack [75], which added several hours to the implementation of SteamVR.

During the project, Valve released a beta version of the SteamVR application, which could be used together with a newer version of Unity. The beta version solved the problems experienced in the project but ended up being more time-consuming than initially estimated

Other options than SteamVR could have been explored, which could have made a difference to the project, in terms of user experience and the actual development.

6.7 User Experience

Experiencing the cinema of the VR application is different from being in the real cinema of Flamman since it is difficult to get a completely correct vision of how it was. The textures were difficult to recreate; although the source material was followed, there is still a possibility of the end result being influenced by our own preconceptions of shades.

Another part of the experience of going to a cinema is to meet and see other people. This is something that was implemented into the application to get the authentic cinema feeling. Even if the application supports multiplayer, it is hard to replicate real physical people with virtual ones and get the same feeling as one would have gotten in a real cinema. Similarly, recreating the feeling of visiting a cinema in VR and implement it to feel the same as visiting a place physically is difficult. However, the application users still seemed to be able to envision how it actually was to visit Flamman.

Additionally, there have been changes to today's cinemas since Flamman opened, changes that we find essential to the cinema experience today. An example of these changes is the development of the surround sound, which did not exist at the time when Flamman was built. This makes us have a preconceptions of how things should work and feel in a cinema experience, which could have made an impact on the development of the application. This could also be true for the users testing the

application as well and to those who in the might use it in the future.

There is also a balancing of good user experience and historical accuracy in recreating something like Flamman in VR. This balance is not always as clear as one might think, and continuous discussion and reevaluations are always needed. Testing also helps with the regard of maintaining this balance.

6.7.1 User Tests

Many of the participants wanted to have the option of interacting more with the environment than expected. Instead of pressing different buttons on the controller to see what they did, they tried to press and move different objects. An expected example we did not think about was moving the player from the lobby to the machine room. Instead of just pressing a button, the users tried to reach for the door handle to open it like a regular door. The testers also wanted to choose a movie by touching it on the movie list or picking up the list.

Feedback from the testers also noted the materials used and how they could be improved to improve the VR experience. However, some of the input on materials was based on their own conception of how the materials should be. While the input overall was valuable, the users had preconceptions of how it should look like which conflicted with the historical documentation of the building, and was therefore not applicable.

With the help of user testing, it was possible to add additional features or fix already existing features. It was possible to see what progression was made by having two instances of user test and some testers being the same tester.

6.7.2 Technology

As discussed, it is difficult to completely recreate the experience of Flamman due to a lack of source material; however, the difficulty also comes from the fact that technology has evolved immensely since the creation of Flamman in 1935. An example of this is that the invention of stereo sound, meaning two different speaker channels for movies, was invented in 1935, the year Flamman was built. Knowing which exact technology was implemented in Flamman and when, requires more extensive research than what was done in the project. However, it is safe to assume that both stereo sound and some sort of multichannel sound technology were used throughout the operation of Flamman; the problem is which of the technologies to recreate in the application. It was decided to recreate the stereo sound in the application, which means the left and right audio source will come from the projector screen. This was because it had a better resemblance to older cinematic sound systems.

6.7.3 Lighting in Flamman

It is difficult to recreate the lighting inside Flamman exactly as how it was in its original state as there only existed six black and white photos that show little detail

of the lighting. Much time would be needed to be spent going through historical documents to find proper documentation about it, which the group could not spend. Much of the lighting was created with inspiration from the photos of the newer, renovated Flamman. This means that the lights in the VR experience might not be as accurate to the original building as what was wanted.

6.7.4 Network Functionality

The multiplayer functionality of the application resulted in being more difficult than anticipated, and some refactoring of this functionality had to be made. During development, the movie stream itself did not seem to be transmitted over the network; hence no movie was played for anyone except the hosting player. This led to restructuring the code behind the network functionality to instead keep a local video player for each player connected to the server. This meant that the only thing being controlled over the network was the pause/play function and the function for loading a specific movie.

We chose to have an abstract model to move freely and not have the 3D representation of the user being bent in unnatural ways. The model representing a user consists only of a head and two hands. By having this abstract 3D model, other users can still see the model move around, but the immersion does not break by having the model clip through the floor or bending in unnatural ways.

Communication between users is currently limited, can not be done besides the user's hand movements. Voice chat was deemed too complicated and time-consuming to implement, and text chat was considered too clumsy in a VR application.

6.7.5 Ethics

The main focus is to enrich the experience by making a good and fair representation of Flamman's architecture and cultural heritage. Though some creative liberties were taken when developing a 3D reconstruction, most architecture and interior remained the same for authenticity reasons. However, some changes were made to the 3D model in regards to the actual Flamman. Those changes covered the actual building, decoration, and objects, e.g., the movie rolls, projector, and chairs. As discussed in section 6.6.1, these changes were made for a better user experience and to improve the application's performance. This can cause an ethical dilemma as we are not staying true to the original design. Thus becoming a confusing recreation of Flamman, i.e., someone might think that these changes existed in reality. We also need to cover the interactive aspects. This refers to the situations users could experience while interacting with the application. As stated before in section 2.3, high immersion can cause the user to have lasting intense emotions. Hence, this had to be taken into account during development and not design any content that could be deemed disturbing. This was done by creating a harm-free environment, i.e., not having tumultuous sounds or dimming lights to limit eye strain.

6.8 COVID-19

In order to comply with the restrictions and recommendations in place due to the global pandemic COVID-19, we had to adapt our work procedures in compares to how they normally would have been. An example is how we worked with VR headsets as we had to borrow equipment to work with from home, rather than on-site. Furthermore, the user tests have been conducted remotely, with the test participants own equipment, to not risk any infection-spreading through the headsets.

The work has been done remotely, which means that we have had to rely completely on tools that enable productive workflows while working from home. As we would have used most of the tools regardless, the work was not affected as severely as it could have been. Due to social distancing, the group has not met in person but has only had virtual meetings. This also applies to the meetings with the supervisor.

7

Conclusion

When planning for the project, the most important parts of a cinema visit were envisioned, and the goals in section 1.3 were set thereafter. To combine the goals with VR, considerations had to be made about the amount of reality and how usability was needed to be implemented for the application's balance. In this conclusion, explanations of how the goals were achieved will be presented.

Various concepts were considered for the user to start a movie. The steps described in section 5.3 were mostly navigational. Since the placement of the door to get to the machine room was based on the original Flamman. There were few compromises to be made. Teleportation was another implementation based on usability as described in section 6.6.1.

A network feature was important to the project; to enjoy the cinematic experience, there had to be an option where users could share that experience. The goal of having multiple users is achieved with the implementation of Mirror, 2.5.7, an asset that helps with network functionality. This ensures that users who want to meet and see each other within the application can do so because what would a true cinema experience be without others there.

Having the ability to explore Flamman was a key goal that needed to be implemented. When creating the 3D model, the user had to be able to walk around and also have a character model themselves. The movement of the user as described in section 5.4, where it was mentioned that multiple different movement types were discussed and tried. However, with feedback from the user tests, it was possible to apply easily to understand and well-refined movement mechanics where the user was able to move in multiple directions. Moreover, it was also an incentive to have interaction with some objects, like throwing and grabbing. Since some testers seemed to expect to be able to interact with some objects, as explained in section 6.7.1.

User testing was an important part of being able to collect data on how the application was perceived. While observing and listening to the user's feedback, the flaws of the application became apparent. This made it possible for further development towards more usability and also reality aspects that our applications lacked. Having four users to test the application made it possible to have multiple different perspectives as well as meeting the goal since the requirement was to have at least three or more testers.

Ultimately, all of the goals were met by the implemented features. With some help from our testers, parts of the goals could be more oriented toward usability, while others more towards reality. Not only is it possible to watch a movie, but users can also greet others and explore Flamman at its fullest glory.

7.1 Recommendations

As a proof-of-concept application, some of the features could be either improved upon or differently implemented. Some limitations had to be made due to time constraints, and sometimes early ideas might later seem redundant. Some recommendations for a similar project would be as followed:

1. Implementing network features early. For things to work in a multiplayer aspect is very different from single-player, the server needs information that then needs to be sent around, and permissions are given. For others to know what's going on, what can and cannot be done is something the server needs to moderate. Therefore it is important to implement multiplayer features early.
2. Having dependable libraries and being aware of what specific libraries do. When developing an application, having libraries can save a tremendous amount of time. Therefore, having such libraries ready might take some time, but in the end, it might alleviate more time. This is especially true if the application is based on something like VR, which requires extra hardware and special implementation to work. This project's use of software was showcased in section 2.5.
3. An important part of developing a project like this is time management. With the help of an agile workflow, the developers can work towards goals systematically and in a more structured way. This provides a way of achieving the goals with a more precise workflow and grants a better understanding of the development needed. Through this, the process of what needs to be done at the time and in the future is recognized. It becomes clear what has been done with an agile workflow and what tasks are being worked on. It alleviates disorientation in the development team and makes sure that work does not get overlapped.
4. Understandably, many of the problems that arrive in development are not always easy to predict. Thus, making it difficult to prepare for. However, having an open mind and not being afraid to ask for help ensures that development does not get stalled. When a feature might be too complex, and a different valuation is needed, compromises may be necessary and lead to alternative solutions.

8

Further Development

As with many projects, there are always improvements that can be done, which is the case for this project. The following features were discussed to be implemented but were discarded since some of them required an extensive amount of time or conflicted with the project's final goal.

8.1 Dynamic Light From the Projector Screen

A dynamic light reflecting off the projector screen, lighting up the cinema saloon, was discussed early in the project. It was even a point brought up by the project supervisor to improve immersion and realism. Instead, a static white light was shining on the chairs in the saloon, which was seen as a temporary implementation until the project had been mostly completed. However, this static light was never replaced with a dynamic light due to other priorities.

8.2 Dynamic Curtain

Dynamic curtains, meaning that a curtain would obstruct the projector screen when it was not in use, and fold when the movie is playing, was discussed to be implemented. The 3D model of the curtain was created, but the implementation of the needed scripts was not prioritized as more immediate matters had to be done. The feature to instantly pause and play a movie exists with a press of a button on the controllers; it was also thought to be a hindrance as the animation to move the curtain back and forth would block the movie.

8.3 Further Improvements to Performance

As discussed in Section 6.2, a lot of time was spent on increasing the application's performance. However, there still exists a small problem of lower FPS when inside the cinema saloon, and no more solutions were found on how to improve it. Occlusion culling could be utilized further, but for this to work, a large rework would be necessary of the 3D model, splitting everything up into several smaller sizes. Currently, the roof is in one piece, which means that when a user is looking at it, the roof is rendered, even if some parts are occluded. Smaller sizes would allow the

engine to only render the pieces in view, lowering the computational power needed to render objects.

8.4 Choosing a Movie Through a Movie Roll

It was discussed whether or not choosing a movie would be better done by placing a movie roll containing the chosen movie inside the projector. This was never developed because it introduced complications with VR that were not yet implemented, and there were concerns that it would take too much time.

8.5 Foldable Seats

As discussed in Section 6.6.2, all of the chair seats are folded down permanently, which was done to improve performance. If the application was further developed, time would be spent to find a solution to implement foldable chairs while still have high performance.

8.6 Game Like Features

Game-like features were one of the earliest features that were discussed. These features would include but are not limited to an interactable character in the ticket booth and the ability to buy tickets and popcorn. Since the project's goal was to construct a cultural heritage reconstruction, concerns were raised that taking time to implement interactions would divert the focus from the main goal. As such, game-like features were lower prioritized and thus never implemented.

8.7 Play Other Movies

Currently, there is only the possibility to play movies included in a static folder in the project. Furthermore, if several users are connected, everyone needs to have the same movie file as the host. In future work, stream a movie with other users could be implemented, or even stream from a third-party source like Youtube.

8.8 Chat or Voice Feature

Another suggestion is adding communication between users. In the final application, there is not a option to communicate between different users, but has use third-party software. A voice chat could be implementation where users can talk with each other inside the application, since having a text chat in VR was cumbersome and felt out of place.

8.9 Possibility to Host Public Movie Rooms

Another aspect that could be explored is the possibility to host public sessions where anyone with the application could join. Meaning that instead of having to connect directly through having access to the specific IP address of the host, and instead implement the possibility to host publicly.

8.10 Better Historical Recreation

As discussed in Section 6.3, it was difficult to recreate Flamman as it was originally. If information was more accessible, a better historical recreation could have been made. In consideration of the covid-19 pandemic, doing on-site research for documentation containing descriptions of Flamman could not be done.

Bibliography

- [1] Virtual Reality Society, “History of virtual reality,” 2017. URL: <https://www.vrs.org.uk/virtual-reality/history.html/> (visited on 2021/02/04).
- [2] Virtual Reality Society, “Applications of virtual reality,” 2017. URL: <https://www.vrs.org.uk/virtual-reality-applications/> (Visited on 2021/04/28).
- [3] Wikipedia, “Flamman (göteborg),” 2021. URL: [https://sv.wikipedia.org/wiki/Flamman_\(Göteborg\)](https://sv.wikipedia.org/wiki/Flamman_(Göteborg)) (visited on 2021/05/14).
- [4] T. Eriksson, “Cinema reconstruction in vr,” 2021. URL: https://www.chalmers.se/SiteCollectionDocuments/CSE/Kandidatprojekt2021/Datx02-21-43_Cinema_reconstruction_VR.pdf (visited on 2021/02/04).
- [5] Writers Tools, “Cultural traditions in danger of disappearing,” 2020. URL: <https://writer.tools/subjects/s/sociology/cultural-traditions-in-danger-of-disappearing> (visited on 2021/05/11).
- [6] J. Bardi, “What is virtual reality,” 2019. URL: <https://www.marxentlabs.com/what-is-virtual-reality/> (visited on 2021/05/12).
- [7] History of Information, “The sensorama: One of the first functioning efforts in virtual reality,” 2021. URL: <https://www.historyofinformation.com/detail.php?id=2785> (visited on 2021/04/28).
- [8] S. Fisher, E. Wenzel, C. Coler, and M. McGreevy, “Virtual interface environment workstations,” *Proceedings of the Human Factors Society Annual Meeting*, vol. 32, 02 1988. URL: https://www.researchgate.net/publication/4709345_Virtual_Interface_Environment_Workstations.
- [9] S. Charara, “Virtual reality: Then and now - why it won’t fail this time,” 2015. URL: <https://www.wearable.com/vr/virtual-reality-then-now-why-it-wont-fail-this-time> (visited on 2021/05/12).
- [10] T. Eriksson, *A poetics of Virtuality*. Chalmers University of Technology, 2016. URL: <https://publications.lib.chalmers.se/records/fulltext/231283/231283.pdf>.

- [11] M. Slater, “Immersion and the illusion of presence in virtual reality,” *British Journal of Psychology*, vol. 109, no. 3, pp. 431–433, 2018. URL: <http://publicationslist.org/data/melslater/ref-344/bjop.12305.pdf>.
- [12] P. Kourtesis, S. Collina, L. A. A. Doulas, and S. E. MacPherson, “Technological competence is a pre-condition for effective implementation of virtual reality head mounted displays in human neuroscience: A technological review and meta-analysis,” *Frontiers in Human Neuroscience*, vol. 13, p. 342, 2019. URL: <https://www.frontiersin.org/article/10.3389/fnhum.2019.00342>.
- [13] J. Lee, M. Kim, and J. Kim, “A study on immersion and vr sickness in walking interaction for immersive virtual reality applications,” *Symmetry*, vol. 9, no. 5, 2017. URL: <https://www.mdpi.com/2073-8994/9/5/78>.
- [14] M. Bergeron, C. L. Lortie, and M. J. Guitton, “Use of virtual reality tools for vestibular disorders rehabilitation: a comprehensive analysis,” *Advances in medicine*, vol. 2015, 2015. URL: <https://doi.org/10.1155/2015/916735>.
- [15] D. Rumiński, “An experimental study of spatial sound usefulness in searching and navigating through ar environments,” *Virtual Reality*, vol. 19, 11 2015. URL: <https://doi.org/10.1007/s10055-015-0274-4>.
- [16] B. Patrão, S. Pedro, and P. Menezes, “How to Deal with Motion Sickness in Virtual Reality,” in *22o Encontro Português de Computação Gráfica e Interação 2015* (P. Dias and P. Menezes, eds.), The Eurographics Association, 2020.
- [17] B. Kenwright, “Virtual reality: Ethical challenges and danger,” 2019. URL: <https://technologyandsociety.org/virtual-reality-ethical-challenges-and-dangers/> (visited on 2021/04/21).
- [18] Skarredghost, “How a developer can reduce motion sickness in vr games,” 2016. URL: <https://skarredghost.com/2016/11/28/how-a-developer-can-reduce-motion-sickness/> (visited on 2021/02/05).
- [19] C. Brumann, “Cultural heritage,” in *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)* (J. D. Wright, ed.), pp. 414–419, Oxford: Elsevier, second edition ed., 2015.
- [20] UNESCO, “Memphis and its necropolis – the pyramid fields from giza to dahshur.” URL: <https://whc.unesco.org/en/list/86/> (visited on 2021/05/12).
- [21] UNESCO, “The colosseum, the great amphitheatre: Historic centre of rome,” 1980. URL: <http://www.unesco.org/archives/multimedia/document-1058> (visited on 2021/05/12).
- [22] E. Franchi, “What is cultural heritage?,” 2015. URL: <https://www.khanacademy.org/humanities/special-topics-art-history/arches-at-risk-cultural-heritage-education-series/>

- arches-beginners-guide/a/what-is-cultural-heritage (visited on 2021/05/12).
- [23] UNESCO, “Acupuncture and moxibustion of traditional chinese medicine,” 2010. URL: <https://ich.unesco.org/en/RL/acupuncture-and-moxibustion-of-traditional-chinese-medicine-00425> (visited on 2021/05/12).
- [24] UNESCO, “Indigenous festivity devoted to the dead,” 2002. URL: <http://www.unesco.org/archives/multimedia/document-3743> (visited on 2021/05/12).
- [25] F. Fassi, A. Mandelli, S. Teruggi, F. Rechichi, F. Fiorillo, and C. Achille, “Vr for cultural heritage,” in *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pp. 139–157, Springer, 2016. URL: <https://core.ac.uk/download/pdf/55259721.pdf>.
- [26] K. Nodjimbadem, “The heroic effort to digitally reconstruct lost monuments,” 2016. URL: <https://www.smithsonianmag.com/history/heroic-effort-digitally-reconstruct-lost-monuments-180958098/> (visited on 2021/03/28).
- [27] A. Gruen, F. Remondino, and L. Zhang, “Reconstruction of the great buddha of bamiyan, afghanistan,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, pp. 363–368, 2002. URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.6.2965&rep=rep1&type=pdf>.
- [28] S. Dow, J. Lee, C. Oezbek, B. MacIntyre, J. D. Bolter, and M. Gandy, “Exploring spatial narratives and mixed reality experiences in oakland cemetery,” in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pp. 51–60, 2005. URL: https://www.researchgate.net/publication/220982547_Exploring_spatial_narratives_and_mixed_reality_experiences_in_Oakland_Cemetery.
- [29] N. Babich, “How vr in education will change how we learn and teach,” 2019. URL: <https://xd.adobe.com/ideas/principles/emerging-technology/virtual-reality-will-change-learn-teach/> (visited on 2021/04/21).
- [30] E. Selmanović, S. Rizvic, C. Harvey, D. Boskovic, V. Hulusic, M. Chahin, and S. Sljivo, “Improving accessibility to intangible cultural heritage preservation using virtual reality,” *J. Comput. Cult. Herit.*, vol. 13, May 2020. URL: <https://dl.acm.org/doi/fullHtml/10.1145/3377143>.
- [31] M. Madary and T. K. Metzinger, “Real virtuality: A code of ethical conduct. recommendations for good scientific practice and the consumers of vr technology,” *Frontiers in Robotics and AI*, vol. 3, p. 3, 2016. URL: <https://www.frontiersin.org/articles/10.3389/frobt.2016.00003/full/>.
- [32] Valve Corporation, “Steam hardware & software survey: March

- 2021,” 2021. URL: <https://store.steampowered.com/hwsurvey/Steam-Hardware-Software-Survey-Welcome-to-Steam> (visited on 2021/04/19).
- [33] D. D’Orazio and V. Savov, “Valve’s vr headset is called the vive and it’s made by htc,” 2015. URL: <https://www.theverge.com/2015/3/1/8127445/htc-vive-valve-vr-headset> (visited on 2021/04/19).
- [34] I. Hamilton, “Oculus insight: Facebook details quest’s inside out tracking system,” 2019. URL: <https://uploadvr.com/oculus-insight-details-quest/> (visited on 2021/04/19).
- [35] Unity Technologies, “Unity,” 2021. URL: <https://unity.com/> (visited on 2021/02/04).
- [36] D. Buckley, “Unity vs. unreal – choosing a game engine,” 2020. URL: <https://gamedevacademy.org/unity-vs-unreal/> (visited on 2021/04/28).
- [37] S. K. Arora, “Unity vs unreal engine: Which game engine should you choose?,” 2020. URL: <https://hackr.io/blog/unity-vs-unreal-engine> (visited on 2021/05/06).
- [38] Upwork, “C# vs. c++: Which language is right for your software project?,” 2019. URL: <https://www.upwork.com/resources/c-sharp-vs-c-plus-plus> (visited on 2021/05/06).
- [39] Unity, “Gameobjects,” 2017. URL: <https://docs.unity3d.com/Manual/GameObjects.html> (visited on 2021/05/13).
- [40] Unity Technologies, *Scenes*, 2021. URL: <https://docs.unity3d.com/Manual/CreatingScenes.html> (visited on 2021/05/13).
- [41] Unity Technologies, “Type of lights,” 2021. URL: <https://docs.unity3d.com/Manual/Lighting.html> (visited on 2021/04/18).
- [42] Unity Technologies, “Shadows,” 2021. URL: <https://docs.unity3d.com/Manual/Shadows.html> (visited on 2021/04/29).
- [43] Unity Technologies, “Light mode: Baked,” 2021. URL: <https://docs.unity3d.com/Manual/LightMode-Baked.html> (visited on 2021/04/29).
- [44] Unity Technologies, “Occlusion culling,” 2021. URL: <https://docs.unity3d.com/Manual/OcclusionCulling.html> (visited on 2021/04/18).
- [45] J. Lampel, “Creating 3d assets for virtual reality.” URL: <https://cgcookie.com/lesson/creating-3d-assets-for-virtual-reality> (visited on 2021/03/31).
- [46] SketchUp, “Sketchup,” 2021. URL: <https://en.wikipedia.org/wiki/SketchUp> (visited on 2021/04/20).

- [47] C. Graham, “Sketchup vs blender | what’s the difference?,” 2021. URL: <https://www.buildercentral.com/sketchup-vs-blender/> (visited on 2021/04/06).
- [48] Blender Foundation, “About blender,” 2021. URL: <https://www.blender.org/about/> (visited on 2021/04/22).
- [49] S. Rosna, “Why use blender when you’re new to 3d modelling?,” 2019. URL: <https://optimalbi.com/why-use-blender-when-youre-new-to-3d-modelling/> (visited on 2021/04/26).
- [50] Blender Foundation, *Duplicate Linked*, 2021. URL: https://docs.blender.org/manual/en/latest/scene_layout/object/editing/duplicate_linked.html (visited on 2021/04/29).
- [51] Unity, “Unity asset store.” URL: <https://assetstore.unity.com/> (visited on 2021/04/28).
- [52] Valve Corporation, “Steamvr plugin,” 2021. URL: <https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647> (visited on 2021/04/27).
- [53] ousttrue, “UniglTF.” URL: <https://github.com/ousttrue/UniGLTF> (visited on 2021/04/27).
- [54] Sketchfab, “Sketchfab for unity,” 2018. URL: <https://assetstore.unity.com/packages/tools/input-management/sketchfab-for-unity-14302> (visited on 2021/04/27).
- [55] Github, “Github for unity,” 2019. URL: <https://assetstore.unity.com/packages/tools/version-control/github-for-unity-118069> (visited on 2021/04/27).
- [56] Valve Corporation, “Steamvr,” 2021. URL: <https://store.steampowered.com/app/250820/SteamVR/> (visited on 2021/05/13).
- [57] Valve Corporation, “Steamvr input.” URL: https://valvesoftware.github.io/steamvr_unity_plugin/tutorials/SteamVR-Input.html (visited on 2021/05/13).
- [58] vis2k, “Mirror,” 2021. URL: <https://assetstore.unity.com/packages/tools/network/mirror-129321> (visited on 2021/04/27).
- [59] Software Freedom Conservancy, “Git.” URL: <https://git-scm.com/> (visited on 2021/02/05).
- [60] GitHub Inc., “Github: Where the world builds software.” URL: <https://github.com/> (visited on 2021/04/17).
- [61] E. Zilberman, “Top github best practices guide for developers,” 2019.

- URL: <https://www.datree.io/resources/github-best-practices> (visited on 2021/05/13).
- [62] Atlassian, “What is agile?.” URL: <https://www.atlassian.com/agile> (visited on 2021/04/05).
- [63] Agile Alliance, “Agile 101.” URL: <https://www.agilealliance.org/agile101/> (visited on 2021/04/28).
- [64] ProductPlan, “Fibonacci agile estimation.” URL: <https://www.productplan.com/glossary/fibonacci-agile-estimation/> (visited on 2021/04/28).
- [65] Scrum Inc., “Scrum board.” URL: <https://www.scruminc.com/scrum-board/> (visited on 2021/05/13).
- [66] Atlassian, “Trello 101.” URL: <https://trello.com/guide/trello-101> (visited on 2021/05/13).
- [67] Hotjar, “What is usability testing?,” 2021. URL: <https://www.hotjar.com/usability-testing/> (visited on 2021/02/05).
- [68] AGConsult, “User testing: what, why and how?.” URL: <https://www.agconsult.com/en/usability-blog/user-testing-what-why-and-how/> (visited on 2021/04/28).
- [69] J. Nielsen, “Thinking aloud: The #1 usability tool,” 2012. URL: <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/> (visited on 2021/04/28).
- [70] V. Yocco, “Ethical considerations in ux research: The need for training and review,” 2020. URL: <https://www.smashingmagazine.com/2020/12/ethical-considerations-ux-research/> (visited on 2021/05/12).
- [71] arcmini phuc, “Thuya smagard free 3d model,” 2019. URL: <https://www.cgtrader.com/free-3d-models/plant/conifer/thuya-smagard> (visited on 2021/04/22).
- [72] J. Burman, J. Eksberg, M. Ibrahim, T.-R. Karlsson, J. Pettersson, and A. Rohdell. URL: <https://github.com/jpettson/CinemaReconstructionInVr> (visited on 2021/06/02).
- [73] “Flamman,” 2013. URL: <https://kinnebg.wordpress.com/2013/02/23/flamman/> (visited on 2021/04/19).
- [74] Kjellbergska flickskolans kamratförening, “Gunnar erik ström i aulans tak.” URL: <https://flickskola.wordpress.com/gunnar-erik-strom-i-aulans-tak/> (visited on 2021/04/19).
- [75] Skarredghost, “How to make steamvr input work with unity xr interaction toolkit in unity,” 2020. URL: <https://skarredghost.com/2020/09/25/steamvr-unity-xr-interaction-toolkit-input/> (visited on 2021/04/29).

A

Appendix 1

A.1 Evaluation of User Tests

First Evaluation of User tests

	Task	Problem	Solution	Difficulty
2	ID: Functionality			P1 P2 P3
2.2	Start a movie	Unclear instructions Unclear if movie is playing or not	Add manual/popup explanations	1 1 1
2.6	Movements	1. No clear instructions how to move 2. Teleport points too far away(especially the one around the posters) 3. No walking between chairs 4. Changing between movements not intuitive	Add a manual Add more teleport points	1 1 1
2	Interaction Design			
1.2	Navigation	1. Some confusion with finding the door under the stairs	Add navigation	1 1
2	Interacting with objects	1. Not being able to turn the door knobs, touch the movie rolls, hold the movie menu	Make objects interactable	1 1 1
	Motion sickness	1. Felt sick		1 1
1 2	Aesthetic			
2.5	Level of darkness in CR	1. Did not get dark enough when movie played 2. When lights were off some reflections became apparent	Increase darkness Decrease reflections	1
1.2	Materials	Surface of some objects looks too even	Add more textures	1
1.2	Size	Doors to big Projector too small	Adjust sizes	1

Second Evaluation of User tests

	Task	Problem	Difficulty
2	ID: Functionality		P1 P2 P3
2.2	Start a movie	Too little information on how to start the movie	1
2.6	Movements	Some instructions did not match the controllers accurately	1
2	Interaction Design		
2	Interacting with objects	1. Movierolls disappearing sometimes when grabbing them 2. Movierolls disappearing from the view randomly 3. Door in cinema room not interactable	1 1 1
	Motion sickness	1. Felt sick	1 1
	Manual	1. Hard to read the text in picture	1 1
1 2	Aesthetic		
2.5	Lights	1. Color palette in cinema room very black&white 2. Lights to evenly spread 3. Lights in the foaje was turned off during movie 4. No light coming from the screen 5. Light coming from projector not round but square	1 1
1.2	Materials	1. Surface of some objects looks too even 2. Texture on the floor should be more even	1
1.2	Size	Doors to big Scale of controllers wrong	1
	View	View outside Flamman too monotone looking	1

B

Appendix 2

B.1 Images of Flamman

B.1.1 Side by side comparison of the current, renovated Flamman and the application



(a) Picture from the current, renovated Flamman

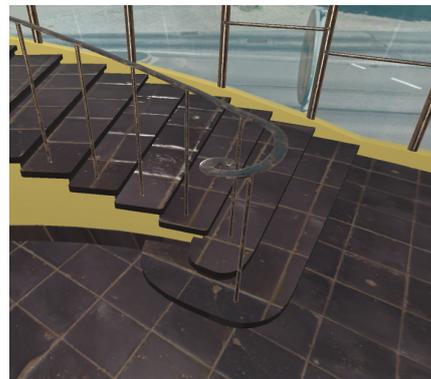


(b) Picture from the application

Figure B.1: Side by side comparison of the lobby



(a) Picture from the current, renovated Flamman



(b) Picture from the application

Figure B.2: Side by side comparison of the railing



(a) Picture from the current, renovated Flamman



(b) Picture from the application

Figure B.3: Side by side comparison of the projector door



(a) Picture from the current, renovated Flamman

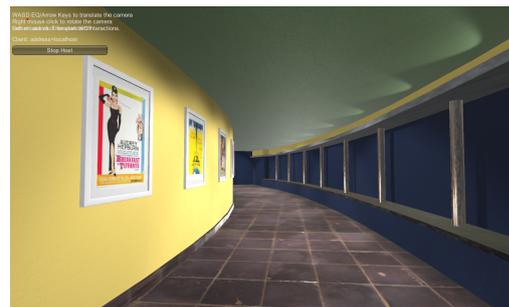


(b) Picture from the application

Figure B.4: Side by side comparison of the access to the saloon



(a) Picture from the current, renovated Flamman



(b) Picture from the application

Figure B.5: Side by side comparison of the corridor

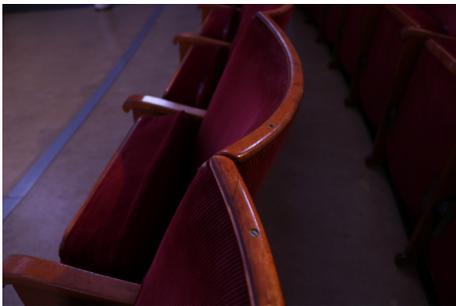


(a) Picture from the current, renovated Flamman

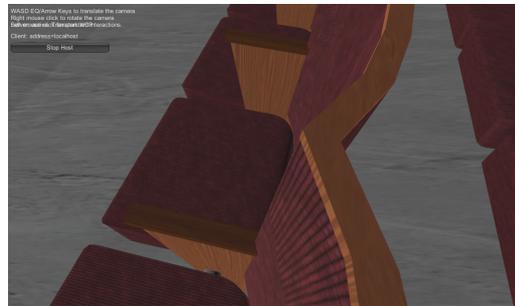


(b) Picture from the application

Figure B.6: Side by side comparison of the saloon chairs



(a) Picture from the current, renovated Flamman



(b) Picture from the application

Figure B.7: Side by side comparison of the saloon chairs



(a) Picture from the current, renovated Flamman



(b) Picture from the application

Figure B.8: Side by side comparison of the saloon scene



Figure B.11: Ticketbooth in the lobby



Figure B.12: One of the two trees in the lobby



Figure B.13: Table and shelf for movie rolls in the projector room

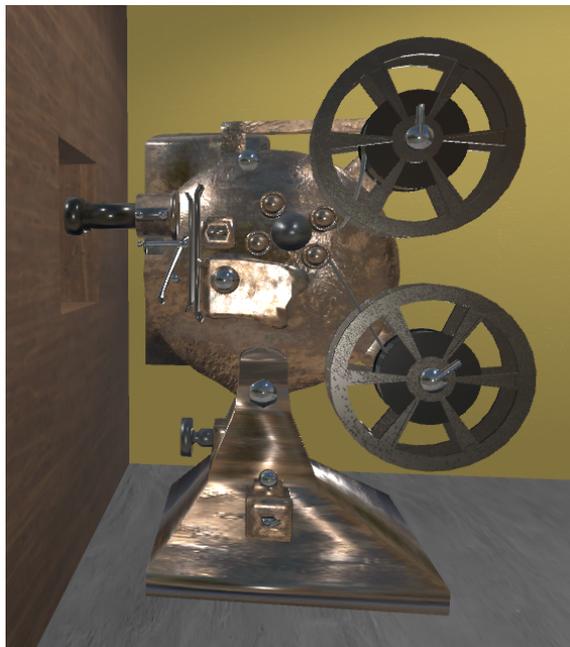


Figure B.14: The projector in the projector room

B.1.3 More pictures of Flamman

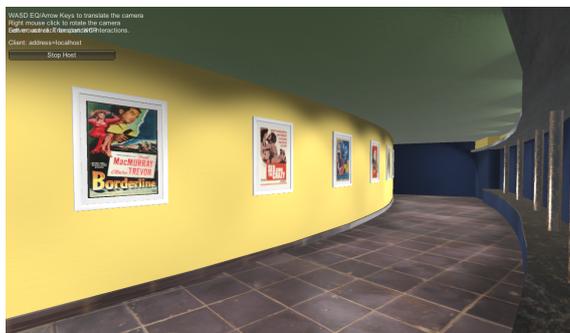


Figure B.15: Corridor with posters



Figure B.16: The view from the projector room out to the saloon

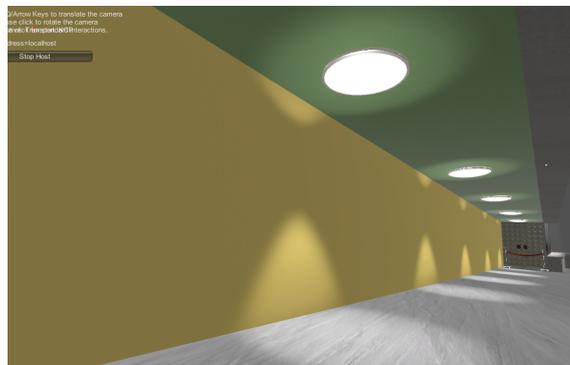


Figure B.17: The corridor on the left side of the saloon



Figure B.18: The entrance doors to Flamman



Figure B.19: From the lobby to the stairs

C

Appendix 3

C.1 Epics

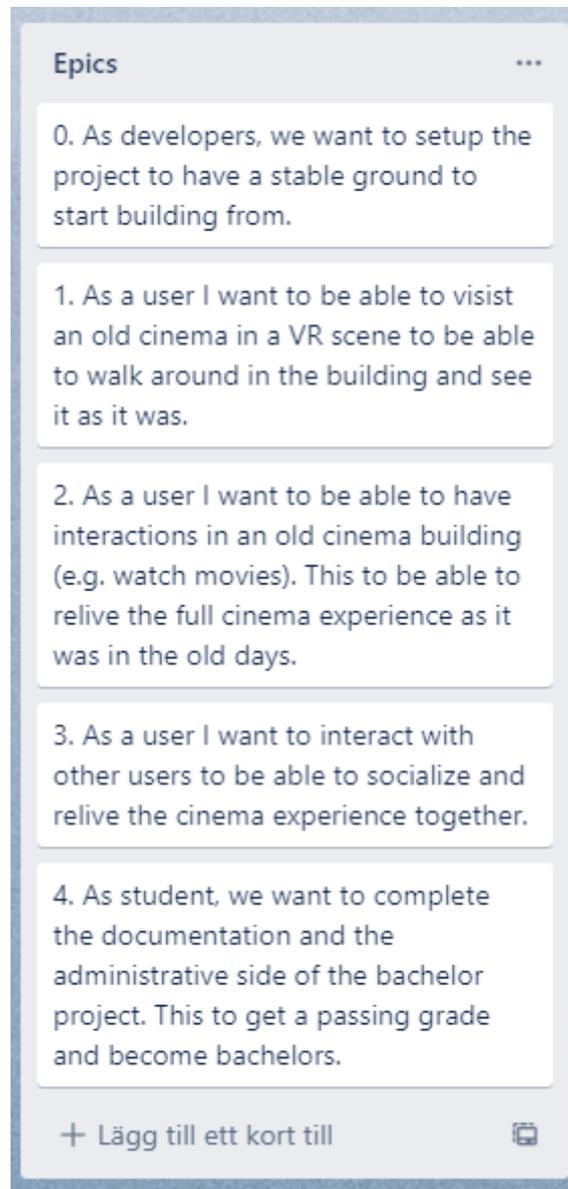


Figure C.1: Epics created to achieve the project goals

C.2 Scrum Board

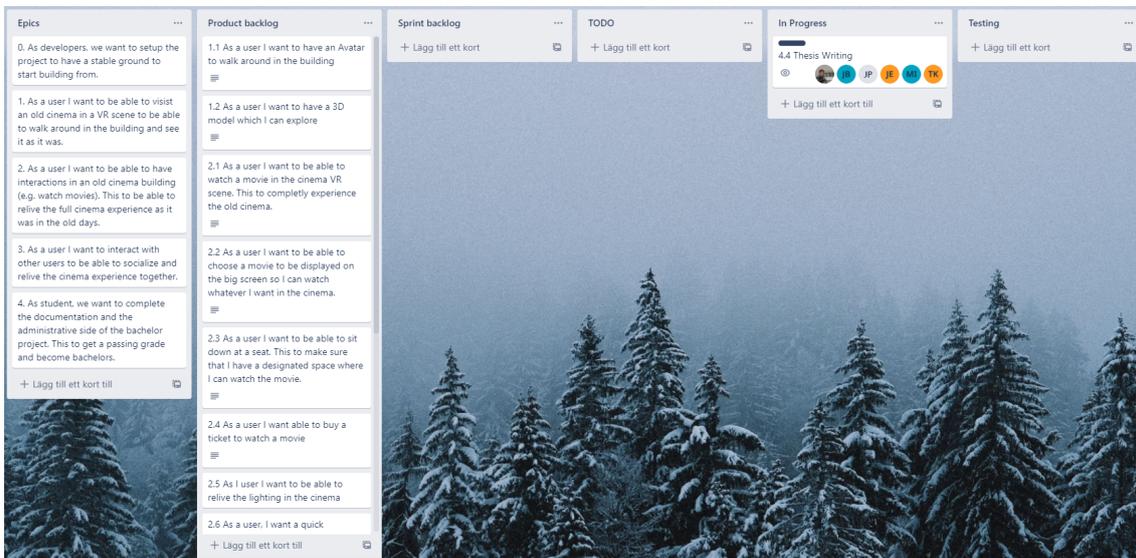


Figure C.2: The Scrum Board hosted at Trello

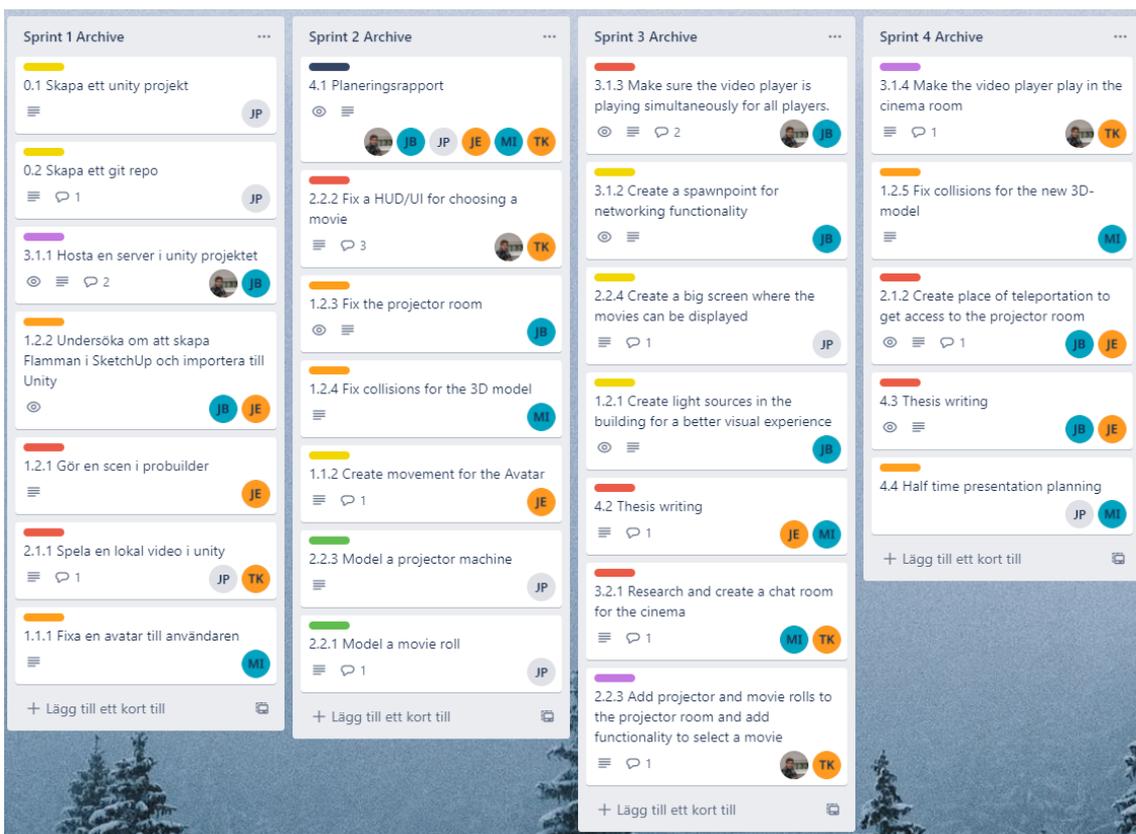


Figure C.3: Part of the sprint archive