Bachelorarbeit im Studiengang Audiovisuelle Medien

The Impact of Customization on Immersion in Virtual Reality

Vorgelegt von:

Name: Sophie Giez
E-Mail: info@sophiegiez.com

an der Hochschule der Medien Stuttgart am 21.03.2024
zur Erlangung des akademischen Grades Bachelor of Engineering.

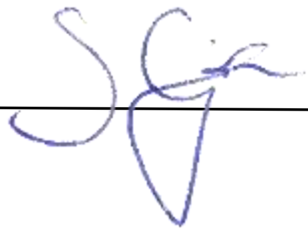
Erstprüfer: Prof. Dr. Stefan Radicke
Zweitprüfer: Prof. Dr. Bernhard Eberhardt

Ehrenwörtliche Erklärung

Hiermit versichere ich, Sophie Giez, ehrenwörtlich, dass ich die vorliegende Bachelorarbeit mit dem Titel: “The Impact of Customization on Immersion in Virtual Reality” selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Ebenso sind alle Stellen, die mit Hilfe eines KI-basierten Schreibwerkzeugs erstellt oder überarbeitet wurden, kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§ 24 Abs. 2 Bachelor-SPO) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.

Stuttgart, der 21.03.2024



1. Abstract

This study looks at how customization of the hands used in VR affects the experience within virtual reality. It details the prior findings on this topic, that found that personalization of a virtual avatar can affect the felt presence as well as the body ownership and how gender plays into the experience depending on the avatar used. It then details the specifics of designing the study and creating the environment for it to take place in, and ends with the findings and results that also seem to align with the difference in perception in relation to gender.

German

Diese Studie untersucht, wie sich die individuelle Anpassung der in VR verwendeten Hände auf das Erlebnis in der virtuellen Realität auswirkt. Darin werden die früheren Erkenntnisse zu diesem Thema detailliert beschrieben, die ergaben, dass die Personalisierung eines virtuellen Avatars sowohl die gefühlte Präsenz als auch den Körperbesitz beeinflussen kann und wie das Geschlecht je nach verwendetem Avatar in das Erlebnis einfließt. Anschließend werden die Besonderheiten der Gestaltung der Studie und der Schaffung der Umgebung für ihre Durchführung erläutert und abschließend die Erkenntnisse und Ergebnisse aufgeführt, die offenbar auch mit der unterschiedlichen Wahrnehmung in Bezug auf das Geschlecht übereinstimmen (translated via google translator).

2. Table of Contents

3. Introduction

I’m very interested in virtual reality and how it affects people, especially in how it relates to gender and diverse identities. Therefore, in this study I want to look at how customization of the hands used in VR affects the experience within virtual reality.

4. Foundation

There are multiple factors that shape the experience of users in virtual reality as well as in virtual environments generally, so to gain an understanding on these influences and how they impact users, some previous work has to be covered.

4.1. Character Customization in Gaming

Character Customization has long been a staple in video gaming, beginning with the choice between different characters and expanding toward customization of stats and abilities as well as the customization of the character’s appearance. The ability for players to create your own character to be able to see yourself in an environment can be a huge draw to a game. Players interested in role playing games often spend vast amounts of time in character creators, and the character creation itself can be a fun experience.

Let’s Play YouTubers like Gronkh consistently dedicate whole episodes just towards character creation (Gronkh, 2020, 2023). Recently Capcom has released the character creator for the upcoming Dragon’s Dogma 2 as a free standalone tool, in what seems to be a very popular move as it has a rating of “*Very Positive*” on Steam with 82% positive ratings out of 1,178 overall votes (as of 17.03.24). There are whole reviews about just the Character Creator of a game, such as Kotaku’s Review of Cyberpunk 2077’s Character Creator (Parrish, 2020) with the author writing in her review “*The thing that interested me the most about Cyberpunk 2077 was its character creator. I’m someone who’s spent a lot of her time thinking about character creators and how they work for people who are not cis, white, able bodied, or male.*” (Parrish, 2020).

Looking at Customization overall through a more general lens shows that it has a big impact on psychological ownership. One study letting participants customize a game with only 4 parameters in a few minutes found that “*despite this very limited personalization, students still feel a strong sense of ownership, originality and pride in their creations.*” (Schanzer et al., 2018, pg. 1).

To be able to create and inhabit your own character can offer players the freedom to step away from societal norms through the anonymity that taking on another persona provides. Players can step away from their everyday persona without the fear of disapproval by their peers (Turkle as cited in Turkay & Kinzer, 2014). This is something I, as a trans woman, experienced firsthand, where Character Creators gave me permission to play as a woman and interact with the virtual world through a female lens, long before I was able to even give that permission to myself.

Character Customization can also be a lens through which the players experience the world, which games like Disco Elysium sometimes make very literal, as the character traits a player picks in that game, determine what your character thinks and notices. That can extend to identity formation as well, leading players to think about what type of character they want to play as, and how the character should function as a vessel for their identity (Arnett as cited in Turkay & Kinzer, 2014). Character Customization therefore is a significant component for players to connect to and project themselves into virtual worlds.

4.2. Virtual Body Ownership

Virtual Body Ownership describes the experienced Body Ownership that is caused from artificial virtual stimuli, like virtual bodies being perceived and accepted as the user's own body. The acceptance of these virtual bodies is called the illusion of virtual body ownership (Waltemate et al., 2018, pg. 2). This is an important psychophysical component to the experience of virtual reality for its users as it factors into their self-perception in immersive virtual environments.

Scientific work on virtual body ownership makes differentiations between bottom-up factors and top-down factors contributing to the illusion of body ownership. Bottom-up factors consist of synchronous visual, motor and tactile sensory inputs perceived by users, whereas top-down factors are related to the similarity of form and appearance. (Waltemate et al., 2018). Research indicates that bottom-up factors seem to be stronger triggers for virtual body ownership to occur (Waltemate et al., 2018) like in the Rubber-Hand Illusion, in which simultaneous stroking of a rubber hand (or a virtual hand) and the participants own hand makes them accept the rubber hand as their own body part (Botvinick & Cohen, 1998). This effect is limited though, by anatomical and postural plausibility, where mismatches with the expected model of participant's own bodies censor the effect (Ma & Hommel, 2015b). Though it can also extend to non-corporeal objects, such as size modulated balloons, to which the ownership isn't any weaker than to virtual hands (Ma & Hommel, 2015a). Top-Down factors of virtual body ownership seem to make less of an impact compared to bottom-up factors (Waltemate et al., 2018). This is something a different study showed as well when demonstrating that bottom-up factors can temporarily override top-down factors of body ownership (Slater et al., 2010).

4.3. Uncanny Valley

The Uncanny Valley effect was first proposed by Masahiro Mori in 1970 and describes the sudden drop in affinity, related to human likeness, turning toward revulsion when something is not quite, but almost human, which he ascribed to our human instinct, protecting us from the danger of corpses or disease (Mori, 2012).

This effect is important to consider when dealing with human likeness and it has become all the more important as cg animation and video games are aiming toward photorealism. The uncanny valley effect has a considerable impact on virtual body ownership and users accepting a virtual body as their own. This happens in top-down factors in virtual body ownership, as one study found the ownership decreasing with a more human-like resemblance, opposed to inhabiting a robot or a block-figure (Lugrin et al., 2015). Though not only top-down factors can produce the uncanny valley effect, as a similar effect seems to occur in dependance of different levels of virtualization of sight and touch in regard to the embodiment of hands (D'Alonzo et al., 2019).

The uncanny valley effect seems to be even more pronounced and more easily triggered in Virtual Reality (Schwind et al., 2018, pg. 48). The high level of immersion resulting from the fusion of visual and tactile sensations can lead to a great discomfort from small deviations like women using a male avatar, since users are highly familiar with their own body and what it should look like (Schwind et al., 2018, pg. 48). This effect seems to relate most to transition as a habituation effect may occur after a few minutes where users get used to their avatar, but are then disturbed when their body is changed again (Schwind et al., 2018, pg. 49).

4.4. Immersion and Presence

Immersion is an objective property of virtual reality systems linked to both software and hardware components and based on the subjective feeling of presence, related to the sensation and experience which users of VR systems perceive (Selzer & Castro, 2022, pg. 2). There are a lot of different approaches to measuring presence, that can get quite expansive, like the 32-item questionnaire by Witmer & Singer (as cited in Schwind et al., 2017, pg. 2) but research has found that a single item measure of presence can be used reliably as a measure (Bouchard et al., 2004, pg. 3). The understanding of immersion is very complex since there are a lot of different factors contributing to immersion. Selzer and Castro list the factors affecting immersion they found in prior research:

> *"Previous works present several variables related to the immersion and the visual features provided by the system. These include the field-of-view [20, 21], the screen resolution [20, 22], the stereopsis [20, 22], the response time or latency [23], brightness, contrast, saturation, and sharpness [24], the level of detail of the 3D models [25], the lighting of the virtual environment [26], and the use of dynamic shadows [26]. Regarding the variables related to audio, these include the use of sound compared to not using sound [27, 28], the ambient sound [29], the 3D spatial sound [30, 31], the use of headphones compared to the use of speakers [31], and the echo or reverberation [29]. Finally, regarding the variables related to the user's tactile system and tracking, these include the sensory bandwidth [32], the level of body tracking [33], the degrees of freedom [29], the affordance of the controls [34], the response time or latency of the tracking [35], the locomotion mode used to navigate through the virtual environment [36], and the temperature and wind [30]."* (Selzer & Castro, 2022, pg. 2).

Selzer and Castro listed rather technical factors impacting immersion, but immersion can also be impacted by psychological factors. One of these factors is anxiety, where a study found that anxiety increases the feeling of presence in virtual reality, through immersing participants in environments that were either anxiety inducing and not anxiety inducing (Bouchard et al., 2008).

Another psychological factor, that is more relevant to this study, is the factor of virtual body ownership, though the research on virtual body ownership lacks statistical power due to a lack of standardization, measurement instruments and analysis approaches, as a recent meta study finds (Mottelson et al., 2023). Though in studies looking at these factors there often seems to be a correlation of presence and virtual body ownership (Jung et al., 2018; Schwind et al., 2017; Waltemate et al. 2018) (even if Schwind et al.'s study doesn't directly talk about ownership, I would still count their findings towards that: "*We found that participants' presence in VR was affected by three different deviations from their own hands*" (Schwind et al., 2017, pg. 3)), which seem to be affecting each other, as Waltemate et al.'s (2018) study found that "*personalized avatars significantly increase body ownership, presence, and dominance*" (Waltemate et al., 2018, pg. 1) as well as "*the degree of immersion significantly increases the body ownership, agency as well as the feeling of presence*" (Waltemate et al., 2018, pg. 1) and Schwind et al.'s (2017) study finds "*women perceive lower levels of presence while using male* *avatar hands and male perceive lower levels of presence using non-human avatar hands*" (Schwind et al., 2017, pg. 1), which arguably describes the impact of top-down virtual body ownership manifesting in a congruence or mismatch in perceived gender, as Slater et al. (2010) describe their achieved effect of body transfer illusion into the opposite gender as overriding top-down virtual body ownership (Slater et al., 2010, pg. 1).

4.5. Customization in Relation to Virtual Body Ownership and Presence

Two are studies I found are related to personalization, which is arguably one step removed from customization. In Jung et al.'s (2018) study participants personalized their hands with paint, which significantly increased their felt presence as well as the illusion of body ownership. Waltemate et al. (2018) were implementing a fast 3D-scan workflow to create personalized avatars of participants in their own clothes as well as in a generic motion capture suit. They used an L-Shaped part of a CAVE System as well as a Head Mounted Device for different levels of immersion and found the personalization significantly increasing virtual body ownership as well as their felt presence (Waltemate et al., 2018).

A different part of character creation is getting to inhabit a different character, and because of the illusion of virtual body ownership in virtual reality – a different body altogether. This is something that Slater et al. observed 2010 as mentioned previously with bottom-up factors overriding top-down factors, leading to a body transfer illusion, which Slater and Sanchez-Vives, who were both part of the previous study, further researched in 2014 in another paper. There they proposed the concept of 'body semantics' as the brain driving attitudes and behavior according to the individual's body ownership and the type of body they inhabit, with participants changing their motor behavior, their implicit racial bias, the perception of object sizes as well as their self-categorization as child- or adult-like (Slater & Sanchez-Vives, 2014, pg. 7). The difference in perception of object sizes was also noted by Jung et al. (2018) in regard to the personalization of hands. Similar to the way Slater and Sanchez-Vives (2018, pg. 5-6) observed the decrease in implicit bias after the embodiment of a dark-skinned avatar, another paper found that the illusion of inhabiting a body of the opposite sex leads to a decrease in gender-stereotypical beliefs, as well as a more balanced identification with both genders, i.e. a more fluid experience in gender (Tacikowski et al., 2020).

These ideas were further explored by the YouTuber Straszfilms in a 2021 video essay looking at the culture within VRChat, a VR social platform, for which users can create custom content. That means, users can't directly adjust their avatar, but if they possess knowledge of 3D modelling, texturing and rigging, they can create their own avatars, or use avatars created by other people. Straszfilms' (2021) video essay argues the unique position of VRChat enabling a stronger sense of embodiment through the focus on virtual reality and the inherent high degree of immersion the medium has, in combination with it's focus on social interactivity: "*In VR Chat […] the social interactivity is the point - if you pick a gender-swapped character, you are deliberately saying: this is how I want to be seen*" (Straszfilms, 2021, 8:20). They further argue, that since a strong anime subculture exists on VRChat, which normalized the use of anime girls as avatars (Straszfilms, 2021), the culture on VRChat enables its users to explore their gender while retaining plausible deniability:

> "[gender play is] unserious on the surface, in a way that allows deflection, but past that, it can be more meaningful. It might be someone, say, feeling out the limits of what they're comfortable with, but for others it could be finally experiencing something that they've always wanted and finding joy in it." (Straszfilms, 2021, 38:36).

They also observed from anecdotal evidence that due to this culture, men were able to distance themselves from gender expectations, enabling them to show emotional vulnerability (Straszfilms, 2021, 45:55 - 54:44), which aligns with the findings of Tacikowski et al. (2020) in regard to the decrease of gender-stereotypical beliefs after the embodiment of an avatar of the opposite sex.

As far as I was able to tell in my research, I couldn't find any study looking at how immersion or virtual body ownership are affected by customization specifically as presented in character creators, in which participants could adjust different aspects of their avatar directly, which led me to explore this topic within this study.

5. Study

This study is based around the implementation of customizable hands within a virtual reality environment and how they affect immersion, while also looking at the effects on ownership, likeability and the uncanny valley. Since customization can be a great tool to allow for diversity in race, gender and self- expression, creating the opportunity for a diverse range of people to see themselves within a virtual environment, this study also wants to examine the experience related to identities usually not catered to within VR (i.e. non-white and non-male).

5.1. Methodology

This study is conceptualized as being of exploratory nature, since the lack in temporal and organizational resources available, as well as the lack of experience in statistical analysis, combined with the small available sample size, since the study, reliant on students as participants, had to be conducted during semester holidays with a lot of students absent, would make it difficult to draw absolute, significant and conclusive results from the study. Because of those reasons, the study is focused on gathering a range of relevant data points as well as comments from participants and exploring this data to draw conclusions about what could be investigated further in later research, and what can be learned about how studies could best look into the topics.

The data that is collected is split into a few different categories, first there is demographic data collected, specifically age, gender and skin color, next there is data to the prior experience with videogames, customization and virtual reality and the rest of the data is related to the experience within virtual reality in regards to enjoyment, presence, embodiment, the uncanny valley and customization, collected via a questionnaire and optional verbal comments. Supplementary the technical data of the customized hands is collected as well.

This study is modelled after Schwind et al.'s 2017 study that detailed how a gender mismatch on hands in virtual reality affects immersion, by exposing their participants to six different hand models, with which participants had to complete three different tasks each, and answer a questionnaire in VR after each task, with another questionnaire handed out on paper afterwards (Schwind et al., 2017, pg. 2-3).

5.1.1. Task

The goal was to implement a few different types of interaction of the hands with the environment, to involve the hands as much as possible as well as to have the hands in the field of view while interacting with the environment, in a similar way, in which Schwind et al. (2017, pg. 2) designed their tasks. They included three different tasks, specifically typing on a keyboard, drawing and building a pyramid out of cubes (Schwind et al., 2017, pg. 2). Since especially the first two tasks require more complex functionality, I chose to focus on similar types of interactions, while relying on simpler functionality, and to combine them into just one task instead of the three different tasks. The interactions I landed on are the interaction with buttons, grabbing objects and inserting them into slots or a different area, and interacting with a sliding door, also by grabbing the handle and pulling it up or down.

To make a thematic connection to the study itself, the task was conceived as a precursor to the customization, basically as completing some steps in a futuristic laboratory to prepare the creation of a pair of artificial hands to be modified in the customization process. The tasks and level were created in the style of a puzzle game. The steps participants have to take are as follows: a power-core is inserted into the workstation to provide power for the process, then the power button is pressed to power it on. Next, a cube is retrieved from a cube dispenser by pressing a button to release the cube. This cube is then inserted into a 3D printer, for which the printer's sliding door has to be opened before and closed after inserting the cube. Then another Button is pressed to activate the printer which then spawns two blank static hands while destroying the cube, which then can be inserted into the hand display, that holds the hands there in a sort of floating position in which the hands are displayed later, during customization.

The tasks are tracked and displayed as a checklist on a display on the workstation, so participants can figure out what they need to do, though they are given hints if they are confused, as the ability to complete the task isn't of substance within this study. The steps detailed on the checklists are as follows:

- Get a Power-Core and put it in the Socket
- Turn on the Workstation
- Get a Cube from the Dispenser
- Put the Cube in the Printer and activate the Printer
- Insert the Hands into the Display Unit

After the completion of all of the tasks, another message appears "Completed – Press Power Button to Exit", with which the participants can then exit the task level.

5.1.2. Study Setup

For the study an Oculus Rift S headset was used, in combination with its controllers. The use of controllers instead of a more high-fidelity kind of hand tracking was chosen since most VR applications use controllers, as well as for availability and easier implementation on the technical side. The application created was run from a Windows 10 PC with an AMD Ryzen 7 5800X processor, 64GB RAM, using a Nvidia GTX1080 graphics card. The screen- and audio recordings were created using OBS and a Blue Yeti Microphone. The program was created on the same PC and headset, to ensure no compatibility problems would arise. The application was made within Unreal Engine 5.3 and run from the editor as a VR preview directly, so that if technical issues with the program occurred, they could be speedily addressed. The study was conducted with the participants standing in the middle of the room, because the tasks required them to walk around. It was designed for physical locomotion only, since virtual locomotion within VR, could trigger cybersickness (Clifton & Palmisano, 2019).

5.1.3. Procedure

In the study, participants are first informed about the data collected and the study generally and sign a consent form, then they fill out a small questionnaire in Excel about their demographic as well as prior experiences. During the study, participants could interrupt or abort at any time for any reason. The screen- and audio recording is started, and next, the participants put on the VR headset and pick up the controllers, familiarizing themselves with the controls and with being in VR, after which they can start the test. Due to the small sample size, the study was conducted in the same way for all participants, exposing them first to the generic hands, that are male-leaning and white, with which they start into the task level, in which they complete a small task, involving the hands as much as possible. After the task the participants complete a questionnaire within VR and then go on to customize their hands. After customization, they enter the task level again, now with their customized hands, and repeat the same questionnaire, with some additional questions about the customization process. They are also asked what they felt missing in customization as well as for further comments. During the process, the participants are encouraged to think out loud, during the task as well as answering the questionnaires. The questionnaires about the experience were conducted within VR for the same reasons Schwind et al. brought forward:

> *"Post-test questionnaires are the most frequently used measures of presence in previous work. One disadvantage of post-test questionnaires is that they rely on subjects' memories, which reflect an inconsistent and incomplete picture of the VR-experience. We, therefore, developed a VR questionnaire which appears in front of the participant within the virtual environment. Thus, participants filled the virtual questionnaire using the virtual hands whose influence we measured"* (Schwind et al., 2017, pg. 2).

5.1.4. Questionnaire

For the demographics the questions used for age and gender were pretty straightforward, for skin color, the participants were asked to pick the closest match to their skin color out of 7 different options presented as colored squares. The questions regarding prior experience were giving multiple options toward the regularity played for video games, binary options toward familiarity with character customization, as well as virtual reality, and multiple options for the overall experience within VR.

To be able to differentiate the effect that customization has the questions on the experience of presence and the hands are repeated before and after customization. It starts out with general enjoyment of the experience, and after that, presence. Since this study is exploratory and looking at a wider array of data points than Schwind et al.'s (2017) study, I opted against using the 32-item questionnaire to measure presence used in their study (Schwind et al., 2017, pg. 2) and instead opted for the single item proposed by Bouchard et al. (2004). Next to presence the participants get to rate the environment's realism, followed by questions addressing the hand. First there is a question towards ownership, then there are three questions addressing the uncanny valley from different perspectives, the first asks about the degree of naturalness and humanity, next is the likeability, and then users are asked whether the hands make them feel more comfortable or uneasy. It then finishes with two questions on gender and skin color congruence.

In the last survey there are additionally a few questions related towards the customization experience. Two address the freedom provided by customization to adjust the hands to the participants wishes, one asks how close the customized hands feel opposed to the participants physical hands, then there is a question on the intent behind the customization, giving binary options between recreated their own hands or created other hands they liked. After that is a question toward the degree of personality expressed in the hands, a question towards the ease of use and last the enjoyment of the customization process.

Except for the question about the intent of customization, all other questions were presented in a 7-point Likert-scale, with the minimum and maximum each labeled. In the following is the full questionnaire, with the corresponding labels and dimensions, split in different sections.

Demographic and Prior Experience (Taken on Desktop PC)

Measuring	Question	Label
Age	What is your Age?	
Gender	What Gender do you identify as?	Male / Female / Non-Binary / Other
Skin Color	Which skin color best resembles yours?	
Extent of Gaming Experience	How often do you play Videogames?	Almost every day A few times a week A few times a month A few times a year less
Experience with Character Customization	Have you played games with Character Customization before?	Yes / No
Experience with Virtual Reality	Have you had experience with Virtual Reality before?	Yes / No

Extent of VR Experience	If so, how much experience do you have with Virtual Reality?	1-10h / 10-30h / 30-60h / more

Experience, Presence and Hands (Taken Pre and Post Customization)

Measuring	Question	Min Label	Max Label
Enjoyment	How much did you enjoy the experience?	Not at All	Completely
Presence	To which extend do you feel present in the virtual environment?	Not at All	Completely
Scene Realism	To which degree did you perceive the virtual environment as realistic?	Artificial	Photoreal
Hand Ownership	To which extend do you feel like these are your own hands?	Not at All	Completely
Hand Humanity	How human / natural do these hands appear to you?	Not at All	Completely
Hand Likeability	To which extend do you like these hands?	Not at All	Completely
Hand Uncanny Valley	How do these hands make you feel?	Uneasy	Comfortable
Hand Gender	To which extend do these hands match with your gender?	Not at All	Completely
Hand Skin Color	To which extend do these hands match with your skin color?	Not at All	Completely

Customization Experience (Taken Post Customization)

Measuring	Question	Min Label	Max Label
Customization Freedom 1	To what extend do you feel you could make these hands your own?	Not at All	Completely
Hand Similarity	How close do these hands resemble your own physical hands?	Not at All	Completely
Customization Intent	Did you try to reproduce your own hands or create other hands you liked?	My Own Hands / Other Hands	
Customization Freedom 2	How close are these hands to the hands you wanted to create?	Not at All	Exactly
Hand Personality / Expression	To which degree do you feel your personality expressed in these hands?	Not at All	Completely
Customization Accessibility	How easy was it for you to create these hands?	Very Hard	Very Easy

Customization Enjoyment	How much did you enjoy the customization process?	Not at All	Completely

5.2. Creation of the Virtual Testing Application

To test out the effects that customization has on the user's experience in the virtual environment, there was a need for customizable hands, a way of customizing them and a task in a virtual environment, to test the hands in.

5.2.1. Technical Terminology

Since the latter part of this subchapter goes into the technical intricacies of creating a virtual reality application as well as custom hands for said application, some technical terms need to be clarified.

The first part of this process involved **Autodesk Maya**, which is software used for 3D visualization and animation. In it, 3D **models**, which are 3D objects or shapes, can be created through a process called **modelling**. Said models are made up of **vertices** (sg. **vertex**), which are points in 3D space. When two vertices are connected, they make up an **edge** which is a line between two points in 3D space. When three or more edges are connected by vertices, most of the time in a triangle or square, that square, for example, can be filled, to create a **face**, or polygon which describes a simple surface in 3D space. More complex surfaces are described by a so-called **mesh**, which is made up of multiple faces. Mesh and model can sometimes be used synonymously, though a model can be made up of multiple meshes. To create a mesh, different approaches can be used: **hard-surface modelling** is usually used for, as the name suggests, hard surface objects, like a table for example, as it is a very precise form of modelling. **Sculpting**, on the other hand, is utilized for organic shapes, and uses similar methods to clay sculpting, and it results in a very high polygon (-count) mesh, that is impractical for animation. To enable animation, sculpted meshes are usually **retopologized**. That is a process, that changes the topology of a mesh. **Topology** describes the structure a mesh has, in the way it's edges and faces are aligned and structured. When utilizing retopology, 3D modelers are drawing a new topology on an existing mesh, which is usually done in a strategic way, to reproduce the shape of the object in the most accurate way, while ensuring that it still stays accurate during animation.

To be able to display textures on an object, there needs to be a way to get 2D textures on a complex 3D object. **Textures** are in the most basic form, an image that is projected onto 3D objects, to basically "paint" color onto them, they can get very complex with a lot of different kinds of texture used for different cases. The solution for this is **UVs**, called that way, as supplementary coordinates: X, Y and Z are already in use to describe the 3D space, so U and V make up the coordinates in which the 2D textures are placed. But to actually get textures onto complex 3D objects, a process is utilized called **unwrapping**. That process is best imagined as the way a cube is glued together from a flat paper, but in reverse: a 3D cube would be cut along certain edges and then unwrapped to form a flat surface in the UV coordinate system. That way textures can be projected onto the cube without distortion. The process of creating textures is called **texturing**.

To enable the mesh to be deformed for animation, a process called **rigging** is employed. A **rig** provides the framework with which a mesh can be animated. To animate, for example, human characters, it is best to emulate the ways in which humans move, to be able to create realistic animations. Since human movement is defined by our skeleton, a similar structure is built up in rigging, this structure is made up by **joints**, that can be seen as similar to bones which are connected in a **hierarchy** to form a **skeleton**. When one joint sits under another in the hierarchy, the joint further up the hierarchy is called a **parent** and the joint furthest up the hierarchy is called the **root joint**. This can be done not only with joints but with all objects and when one object is placed under another, it is called **parenting** the object to its parent. If a skeleton of joints is constructed, it inherently doesn't move anything yet, to enable the skeleton actually deforming the mesh, a process called **skinning** is used. In this process the mesh, now also called **skin**, is tied to the skeleton by placing weights to define which vertices get influenced by which joints in which strength. To refine the initial binding done by the program, in a process called **weight painting**, the weights for the different joints are painted onto the vertices, by adding or subtracting weightings. This process can be further utilized by weighing the vertices for the way the weights are translated from the joints. There are two major ways the program can interpret weights: **linear blend** and **dual quaternion**. To stay succinct, I won't further go into these, and will just state, that dual quaternion better retains volume when bending off or twisting a joint, for example when crooking a finger, but could add too much volume, resulting in bulges, therefore painting a mix between the two where they're most needed is a good way to ensure a better deformation of the mesh.

Sometimes our skin doesn't follow our bones 100% though, because of our muscles and organs sit in-between the bones and the skin, therefore skinning and weight painting can only get you so far. In addition to those, **blend shapes** can be utilized. Blend shapes are in the most basic form a different version of the mesh, that can be blended to, for example you could make a version of an arm, in which the biceps is bulging, but that should only be present during the bending of the arm, so you could blend your mesh, to the version that is bulging when the arm is bent. To make this process easier, Maya has a feature called the **Pose Editor**, in which you can directly create blend shapes that are tied to certain angles of the joints and blended to automatically when rotating the joint. Blend shapes in maya are called **morph target** in Unreal Engine, therefore both names are used later. The joints as a skeleton make up the so-called **bind rig**, as the skin is directly bound to it. Since the bind rig is often impractical and tedious to use for animators, there is often another rig created on top of the bind rig, called the **control rig**. The bind rig is parented to the control rig, but the control rig can offer further functionality to the animators, like giving them one parameter to adjust if they want to roll the foot, for easier walk animations, and much more. It can also implement two different ways to move the joints: **forward kinematic** or **FK,** is the way the joints already move rotating from the top of the hierarchy down, but that can be tedious and look unnatural when for example an arm is grabbing something, and animators have to precisely rotate the elbow in the opposite way of the shoulder, which looks really stiff. Therefore, the other way, **inverse kinematic** or **IK**, is a system that is given the position of a part lower in the hierarchy, for example the hand, and it calculates the way the joints have to rotate to get the hand where it needs to be. It is quite complex and wasn't needed in this project, but it needed to be mentioned to better explain FK.

The second part of this process involved **Unreal Engine**, a real-time game engine that is optimized for the best visual quality with the least computing power needed to render (compute) a frame (image) in the shortest time possible. It also provides a framework for game developers to create a game, providing the most important functionality most often needed within games, though Unreal Engine can be used outside of gaming as well. One such feature is a node-based system that's implemented, called a **Blueprint**. In short, Blueprints in Unreal are an alternative to writing code in C++, that offer Functions that could be called within code as nodes, that can be plugged together with values fed into the nodes or returned by them. It is a more visual way to establish functionality.

Since Unreal Engine is a game engine, and games usually function in a reactive way (i.e. if I do *this*, then *that* happens), the Blueprints are built around **Events**. Events are a way to react to things, like when the frog dies, I can call the Frog_ Dies-Event, but for that Event to do anything it needs to be listening if the event is being called or not, that is the implementation of a **Listener**, with which Events can be reacted to, like I can build functionality off of the Frog_Dies Listener, so that if the frog dies, it falls over, for example. This is something done in the **Event Graph**, a part of the Blueprint where the functionality is defined. But there are also some things that need to happen right at the beginning of a level, which can be done with the **Begin Play** Event, or there are things that need to happen every single time the game updates, which basically means every Frame, which can be done by building off of the **Tick** Event, that is called every time the game updates. Something like that are the animations for example, which are created in a special **Animation Blueprint**. That is one kind of a list of special blueprints, that need special functionality around a specific thing, so it's easier to use a special blueprint than to re-build all of that functionality yourself. Within the Animation Blueprint there is not only an Event Graph, but an Animation Graph as well, which outputs the animations that should be displayed. In this project I'm using a **State Machine**, that outputs a certain animation based on the state that it is in, i.e. holding an object. Another special Blueprint is the **Widget Blueprint**, that is used to create and display User Interfaces, which can be interacted with in VR via the **Widget Interaction** functionality, that can for example determine, where a controller is pointing. The controllers in VR are called **Motion Controllers** and to read out the buttons, pressed for example, within Unreal, **Input Actions** negotiate all the possible different devices with a unified function to react to.

For this project I started off with the **VR Template** that is already implemented in Unreal Engine and makes the process of creating a VR Application easier, by already providing a lot of functionality around VR. Two important Blueprints that come with Unreal's VR Template are for one, the **VR Pawn** Blueprint, that is practically the "Player Character" and provides functionality for the player to interact with the world. The second important Blueprint mentioned above, is the **Grab Component** Blueprint that provides grabbing mechanics for the player to pick things up.

Next, I want to get into materials, similar to other render engines, unreal has a shader system, in which you can create a **Material**, which is similar to a Blueprint, because it is also node-based, but instead of acting on Events, it outputs the parameters for the engine to calculate how an object will look like and how it interacts with light. I will now rapid-fire through some terms for brevity: **Normals**, or **Normal Maps** are Textures that define the way a certain pixel on a surface is pointing, if a surface is smooth, the Normals are distributed equally pointing in the same direction, but a rough stone surface would have a lot more complexity within that. **Alpha** is an additional channel, to the Red, Green and Blue channels (**RGB** to RGBA), that is a greyscale defining the Opacity of a certain point, often used for Masking. There is also alternative to the RGB model, the **HSV**, or **HSB** model, that defines a color over 3 values for Hue, Saturation and Value / Brightness. **Subsurface Scattering** or **SSS**, is the way light can scatter under a surface and produces a different appearance of an object, it is often used in skin, i.e. to create the look where your skin lights up red when shining a flashlight through your fingers, that makes the material look more realistic. **Emission** is an attribute of a parameter that defines if a material glows, i.e. emits light. Materials in Unreal use **Parameters**, which are usually float variables, that can be changed from outside the Material Editor, in order to change a material, for example to react to an Event.

In Unreal there are **Static Meshes**, that can't be deformed / animated, and **Skeletal Meshes**, that contain a skeletal rig on which animations can be played. A **Physics Object** is usually made up of **Simple Collision Objects**, very basic shapes (cube, capsule, …), held together by **Physics Constraints**, conceivable as springs holding two parts together, that negotiate physics simulations and especially collisions in the most efficient way possible. Skeletal and Static Meshes can have Physics Objects as a tied to them, to have them interact in a physical way, simulated via the Physics Engine. Whereas the Physics Constraint can be seen as a sort of connector, the **Physics Handle** can be seen as a sort of Holder, but otherwise works in a similar way. There are some Physics Objects that just exist to detect collisions, such as the **Box Collision** or the **Sphere Collision**, that can be used in a Blueprint, for example to detect a certain object entering a certain space. There are classifications of things, like Pawn and Actor to adjust to which objects in a certain scene should be interacted with and how. The options for this are **Block**, blocking an object like a wall, **Overlap** like detecting an object entering another collision object or **Ignore**, which is self-explanatory. There are **Scene Components**, which are like containers for almost anything located a certain transform in the 3D scene. And finally, you can **Spawn** things, which means loading them into a level they weren't previously in, and making them appear, as well as **Destroy** things, that then vanish and are unloaded from the level.

Now that the terminology needed is covered, we can get to the actual creation of the virtual reality application.

5.2.2. Creating the Customizable Hands

Since the hands needed to work in various ways specific to this study, the simplest way to incorporate a wide range of functionality was to create them from scratch to ensure that no incompatibilities arose further along the way, that could arise from the use of existing hand models.

The first step in this was sculpting the hands, to ensure a realistic base shape, but also to enable the first dimension of customizability: gender. To make this dimension possible the basis for the hand model was an androgynous hand that couldn't be placed into either gender and expanding from that base, the model was respectively feminized and masculinized into two further models. The feminine model was made by softening the edges, reducing the knuckles, slimming the fingers and the wrist, to create a more soft, dainty and feminine look. The masculine hand however was modified in the opposite way, with a thicker wrist, thicker, bulkier fingers and harder edges, to give a rougher, thicker and muscled feel to them.

These sculpted models were then imported into Autodesk Maya where a custom topology was created based on the androgynous model. The goal of creating this topology was to ensure high enough spacial resolution to not just accurately render the hand model, but ensure there was plenty resolution to deform the model later on, in the rigging and animation stages.

Next was another dimension in which the hands could be customized: the nail shapes, though there first needed to be nails that could be modified. The hand model was created without nails to enable an easier way of customizing them as well as separating the UVs and the materials in Unreal Engine. The fingernails consist of a quite basic shape, to be modified further on in the creation process. To make it easier to create the different nail shapes later on, the initial nails were longer than needed, so they could more easily be remodeled into the intended shapes.

With the nail shapes finished, that concluded the basis for the androgynous hand, which now could be UV-unwrapped to ensure consistency over textures across the three models used for gender customization. The different nail shapes were unwrapped all on top of another, with the long nail as a basis so one created texture would work on all the different shapes in the same way, since the long nail contains all the other nail shapes.

To end the modelling process, finally the androgynous hand was adapted to the masculine and feminine sculpt respectively, with the topology staying the same, and only the vertices snapped onto each sculpts' surface. The nails had to be placed and modified manually, since the sculpts were created without them. Since the UVs were already finished and the topology didn't change, they now provided a unified UV space across all of the different models, which concludes the modelling stage.

Following the modelling stage was the rigging process, which didn't have to be overly complex, since the human hand moves in predictable and restricted ways, which were even further restricted by the use of controllers in VR, which only need a few simple animations (Quinn Kuslich, 2022a). After constructing a skeleton from joints, with the wrist as the root joint the skin could be bound to the skeleton. That was done to the hand only, since the rigid nails are not deformed, they could simply be parented to their respective end joint on each finger. After a few tries to figure out the best settings for the binding process I could begin the weight painting process, in which I tried to create the best possible deformations for all the joint movements. With additional weight painting between linear blend skinning and dual quaternion, which better maintains volume when bending the mesh, the deformations were further improved, until the skeleton worked quite well on its own, deforming the mesh. With some deformations, the skinning can only do so much, the best example of this being the rotation of the thumb over the palm of the hand. To fix these issues the rotations of the different bones were improved with the use of the pose editor, which creates blend shapes and connects them to certain angles on certain bones. Regrettably these improvements didn't make it into the final project, as the method to transfer animations into Unreal Engine only transfers the Animations of the Bones, and not the ones on the blend shapes. Even though the blend shapes are present in Unreal as so called "morph targets" they would need to be connected manually, one by one, to make them work with the animations, which would be too much effort for the scope of this project.

After the joints were working well a basic forward kinematic control rig was created to ease the creation of animations. The rigging was all done on the androgynous model, with the masculine and feminine models incorporated now, in the control rig, with the use of blend shapes, to be able to morph the hand into the masculine or feminine direction, which created the possibility to slide seamlessly between genders. This was done for the hand, as well as the nails, so the nails always matched the gender selected. Here as well, animations on the blend shapes didn't translate into Unreal but opposed to the hands, since these were only two morph-targets, one for each gender direction, it was easier to directly control them within Unreal Engine.

Now as a last step before moving the hands to Unreal Engine, the animations needed to be created. These were fairly basic since they only needed to be simple poses, as the transition between these poses is controlled in Unreal Engine based on controller inputs. There were 18 poses created, of which 10 ended up being actually used in the project. Some of the unused poses were for example different kinds of grab animations, which were mostly redundant and could've only been of use with a more elaborate grabbing system, which would've gone beyond a viable scope for the project.

In Unreal Engine, of which I used Version 5.3.2, the hands first needed the general functionality of virtual reality hands, so moving with the controllers, animations and the ability to grab or interact with objects and devices (Quinn Kuslich, 2022a). For the interaction with objects there needed to be some physics to the hands, so first there were simple collision objects created and tied to corresponding bones to enable the hand to trigger collisions (for being stopped by a wall or pushing away a cube) and overlays (for buttons). After these were constructed, connected by physics constraints. I attempted to implement physical interactions with the fingers (VR Playground, 2022), like the fingers bending under pressure or fingers not clipping into a cube, for a better grab mechanic, but sadly the physics constraints connecting the different joints were causing a lot of issues, so the physics weights are set to 0, causing the constraints have no impact on the hand itself.

As for the further physical interactions, I tried to use physics constraints from the hands to the motion controllers, which also didn't work, so I shifted to using physics handles, which the hands are tied to (VR Playground, 2021). These have a similar effect as the physics handles, holding the hands with a certain strength (stiffness) and damping interpolated movements. The physics handles are then tied to empty scene components I called physics parents (as in parent objects), to enable the hands to move away from the motion controllers, for example, when moving with the handle of a sliding door while operating it. This concludes implementing physics for the hands, though the whole process was involving a good amount more trial and error than described here.

The next thing the hands needed was the implementation of animations. These were mostly implemented inside of an animation blueprint, with a state machine powered by the states set in the event graph. For these states enumerators were used, listing the different poses, or the object-type with which the grab action can be used. In the VR Pawn blueprint some Boolean variables are set, dependent on which, controller inputs are being used and these are then checked in the animation blueprint's event graph every tick and based on the Booleans the states are being set. The previous steps in this paragraph were taken from a tutorial by Quinn Kuslich (2022a). This provides basic functionality, with the state machine now switching between the different poses, but since it is, at this point, only based on Booleans, users can't, for example, have their hand half-closed. For a better experience and immersion, I therefore fine-tuned the animations by driving the hand and index animations inside Blend Spaces, with the Action Value provided by the input actions of the controllers in the VR Pawn blueprint. Since inputting the Action Values directly into the animations would result in the animations being too fast, there was an additional step of interpolation added in the event graph. This slows the animations down by making the animation values approach the action value over time. As previously mentioned, I experimented for a while with implementing better grab animations that respond to the way an object is held. Another approach than the physics constraints were animations for each finger being driven by collision with the object separately (Quinn Kuslich, 2022b), though this also didn't work, which leaves the grabbing animation now at one basic pose for each object (cube and sliding door). The animations were now implemented with the thumb animation being driven by a Boolean and the index and grab animations by Action Values.

At this point the basic functionality of the hands was established, with the grabbing mechanic in large parts based on the Unreal Engine VR Template which was used as the core of this project. As a next step the customizability needed to be implemented. The customization in this project is split in two major customizable parts, the hands themselves and the nails, with the customization of each taking place in two ways, part of it transforming the physical form and part within the material.

The physical hand transformations had the following customizable dimensions: gender, scale, finger length and finger thickness. The gender dimension was realized by using the feminine and masculine Morph Targets (Blend Shapes). The corresponding slider had the range of -1 to 1, which was mapped by clamping the value between 0 to 1 for the masculine Morph Target and the value multiplied with -1 before clamping the value was driving the feminine Morph Target. This is not only driving the Morph Targets on the Hand Mesh itself, but also on each nail, so it matches the hand.

The scale is simply adjusting the overall scale of the hands between 0.9135 and 0.1335. The finger length and thickness are being applied within the Animation Blueprint after the State Machine that is driving the animations. These are being realized by Transform Bone nodes, with the finger length driving the X value and the thickness is input into the Y and Z dimensions.

The nails had a simpler physical transformation. They only needed to react to gender and switch nail shapes. Adjustments in Gender are achieved by transforming all of the different nail shapes with morph targets by the same system with which the hand is transformed. The nail shapes are modified by switching the visibility of the different models depending on which one is selected.

The Material of the hand was initially planned for more customizability, which had to be reduced due to the limited time frame available to develop this tool. The adjustable parameters of the material are all related to the skin color, of which you can input the melanin, which controls how dark or light the skin is, the undertone, which gives the skin a more olive or rosy taint and the saturation. Initially there were more texture-based adjustments planned, like adjusting the age by adding wrinkles or adding secondary gender components, such as the visibility of veins and hair. These possibilities fell flat though, because the creation of a convincing skin texture with overlay textures for hair, wrinkles and veins would have inflated the scope too much. Therefore, the only textures used for the hand material were a normal map that was baked from the initial sculpt to give more shape and details to the hand model, as well as basic alpha masks for the fade at the wrist and the palm of the hand to mask the difference in skin color from back to palm in darker hands. The settings for the skin color are based on a gradient texture I created in Adobe Photoshop, on which the X axis maps the Melanin of the skin from very light to very dark, and the Y axis maps the undertone with neutral tones in the middle, rosy tones towards the bottom and olive ones toward the top. By mapping the skin color on a 2D texture, the skin color could be driven by inputting the melanin and tone parameters into the UV coordinates of the created texture. To achieve a difference in color for the palms in darker skin, the texture was created twice with a brighter skin tone falloff toward the darker colors, driven by the same values and interpolated based on the mask mentioned before. Finally, the saturation parameter is driving a Desaturation Node, which acts on the final skin color. And last the subsurface scattering color is added to make the material look like real skin and not plastic. At first, I tried using red as the subsurface scattering color, but that resulted in darker hands being completely red, so I modified it a bit by darkening the red based on the selected melanin and mixing the red with the final skin color, based on the Melanin parameter as well.

The nail material is more expansive than the hand material. There are more parameters that act on it, specifically Melanin and Skin-Tone, the selectors for Natural Nail color and Nail style, and for the primary and secondary colors the Hue, Saturation, Value (Brightness), and selectors for Metallic, Matte and Glitter Finish respectively. The Natural Nail color is created similarly to the hand, by using the skin color texture for the palms to select melanin and skin tone, but other than the hand, the nails had a texture, which was easier to create, because of the simple shape of the nail, and therefore viable. I created the texture by photographing my own thumb nail, cleaning up the picture with Photoshop and extending the tip out to ensure it would work for all of the nail shapes. Then to enable the different skin colors, the Texture was divided by a matching skin tone from the skin color texture in photoshop, and the resulting texture is later multiplied in unreal with the selected skin color, to get a matching color for the nailbed. The nail tip is added with a mask from the original nail texture, because the tip doesn't change with skin color. Then the subsurface scattering is calculated similar to the hand, except for the tip, which just uses the same base color again.

The colors for painted nails are created rather straightforward, as the parameters are already values for a HSV color model, which are just translated to RGB. Based on the selected Nail style, which include Solid color, French Tip and Ombre, the primary and secondary colors are merged by alpha masks. With the first style, solid color, the secondary color isn't used at all, French Tip colors the tip of the nail in a different color, and Ombre creates a gradient, that fades from the primary color at the cuticles to the secondary color at the tip. Depending on the selector for natural nails, the primary color is replaced by the natural color, drawn from the shader above.

Last, to enable the use of different finishes, the metallic and matte finishes are cut out by the masks described above and plugged into the Material's Metallic and Roughness inputs respectively. The Glitter finish is a bit more complex and overrides the metallic finish. To create the glitter finish, I loosely followed a tutorial by Let's Make FX (2021), but deviated from it since it is for a fantasy material, whereas I wanted the Glitter to be more grounded in reality. The Glitter shader is based on a texture of random noise created in Photoshop, which is used for the normal map and metallic inputs, to simulate a lot of particles in different angles, but that alone wasn't sufficient to make the material really sparkle. To simulate the sparkling of the particles, the brightest parts of the noise were isolated and multiplied with a Fresnel, which is acting on the angle the camera is looking at the material, and returning higher values for a more direct angle. Based on this system practically isolating the "sparkling particles", the base color, created in the way described above, is modified with the sparkling particles turned brighter and even adding a slight amount of emission to make it look like they are sparkling. This, powered by the Fresnel results in different particles lighting up depending on the angle they're looked at, which forms the glitter finish.

The last part of enabling customization of the hands lies in the UI with which users can change the different parameters of the hand as well as a save system to transfer the hands between levels. The UI was created loosely based on a tutorial by Virtus Learning Hub (2022), as a Widget Blueprint in Unreal and consists out of Sliders, Selectors, Checkboxes and Buttons. It was split in two parts: customizing the hands and customizing the nails, for better legibility. I created some basic shapes in Photoshop, like arrows, a checked and unchecked checkbox or a circular selector for the Sliders.

Most Sliders are rather straightforward, because they're already included within Unreal's Widget Palette. The ones to pick the Melanin and Skin Tone had a texture applied to indicate the value being picked and just the Sliders for selecting the Hue, Saturation and Value (Brightness) are a bit more complex. To make them more intuitive the Selector on the three sliders features the color that was selected and the texture on the Slider is being updated with the other selected Values, so the Hue Slider gets darker as the Brightness decreases, it isn't being updated with Saturation to make the Hue still legible, the Saturation Slider reacts to the Hue as well as the Value, and the Brightness Slider receives its color from the other two. Since there aren't Selectors in Unreal already, they had to be created from a Text Box displaying the selected option and two buttons, which rotate the selectable options and set the parameters accordingly. The Checkboxes and Buttons were both used out of the box from Unreal's own Palette.

Then at last there was another invisible overlay created out of Buttons to make hitting the different UI elements with the VR controller easier. These Buttons expanded, for example, over the Text of the Checkboxes, as well as over the padding of the elements or over the Text Box of the Selectors. To better visualize where to hit, a cursor was created as well (Max SMoke, 2019), that responds to the Widget Interaction of the controller that last interacted with the Widget.

The save system relies on a simple Structure, that contains all of the different parameters being changed on the hand. That Structure is being stored as a variable within a Blueprint created called "SaveAttributes". This Blueprint acts not only as the Save Game Object being saved, but also contains functionality around saving, like constructing a valid slot name to be saved to, checking for and returning valid existing save slots to load, as well as deleting saves and reordering the others so the saves are all ordered in a number series from 1 to the last existing save.

5.2.3. Creation of the Task

Now that the Customizability was established, the task itself needed to be implemented. For this task, a few things needed to be created: a button, which could be re-used multiple times, a power-core and a corresponding socket, the hand-display, the printer and the cube dispenser. To limit the scope of the project the creation of haptic or auditive feedback systems to indicate interaction or progress were foregone and instead I focused on visual feedback by implementing colored emissive light elements, comparable to led light strips, into the objects. This was based on an emissive material that outputs different colors depending on a parameter with which to select the status of the object in question. The states and colors were powered off - indicated by a dark blue, emitting no light, powered on - indicated by a light blue glow, ready - indicated by a green glow, processing – indicated by a yellow glow, and finally error – indicated by red glow.

All of these objects were modelled in Maya as well, in a minimalist futuristic style. Some of the modelling could be used for multiple objects to cut down the total workload, like the printer is using the same frame the dispenser is using, only slightly modified, and the power-core could be turned inside out to speed up modelling the socket for it to slot into. After they were modelled, they too needed some UV layout, for Lightmaps to work, as well as to be able to apply different materials to the different parts of the mesh, like glass or the light strips mentioned before. When transferred into Unreal the moving parts were imported separately, like the button and the button base or the dispenser and its sliders releasing the cube, as well as the printer and the sliding door.

To provide the functionality, each of the interactable objects was created as an Actor Blueprint based on the meshes modelled earlier. The Button is based on a tutorial by VirtualRook (2023) and works based on a Box Collision, which detects when objects overlap into it, records their initial position and updates the position of the button in Z direction by the distance the overlapping object has moved from its initial position. When it has moved down around 1cm distance and the button is active and wasn't triggered yet (based on variables) an Event is triggered, which can be reacted to from any blueprint that has a reference to the triggering button object. When the button has moved far enough up again, the variable to check if it's been triggered is reset. This prevents the button constantly triggering once pressed down. There is also a function setting the status of the button, which activates or deactivates the button and also sets the status parameter for the glow material, changing the color of the lit elements.

The Power-Core and Power-Socket work in the same way the Hand Display and the static hands do. The basic functionality of attaching components using sockets in Unreal Engine is based on a tutorial by VR Playground (2021). The Power-Core itself has a very basic Actor Blueprint, just incorporating the Grab Component which provides functionality to grab and release objects. It is already existent functionality that came with Unreal's VR template, together with the Grabbable Cube Blueprint. The Power-Socket has a Socket placed in its Static Mesh, an Unreal feature that is practically just a locator that provides a quick way to get a certain location and rotation that deviates from the location of the object itself. Inside the blueprint there is a Sphere Collision placed, that works in a similar way to the Box Collision mentioned above. When it detects overlap though, the overlapping object is first checked for being a Power-Core Actor. If a Power-Core is detected the references inside the VR Pawn Blueprint to an Object being held are cleared out and the Release function of the Grab Component is called. Then the Collisions of the Power-Core are disabled for a short time, so it doesn't get stuck as it is moved to the Location of the Unreal Socket mentioned above. Last an Event is triggered, so other objects can react to the Socket being powered on. The Sockets Glow Material is also updated to indicate the new Status. When the Overlap ends because the Power-Core is pulled out of the Socket again, another Event is fired, and the Glow Material is updated as well. The Hand Display works in the same way, although it keeps track of two objects (the left and right hand), though it only works if it has been powered on first. It also reacts to being powered off by dropping the hands it was holding, as well as turning it's glowing inside ring off.

The Dispenser's functionality is based on an Event that can be called from the main level. It has a socket in its static mesh as well that indicates the spawn location for the cubes, it's located above inside the chute. Initially it spawns one cube, using the Grabbable Cube Blueprint from the VR Template, which then lays inside, above the release doors. If the Event is called and it is powered on and not currently releasing a cube, it changes the status color to yellow and then moves the release doors outward, so the cube drops down, and closes them again after one second, then changing the color back to blue.

The Printer is more complex as it has to account for more states, as well as provide the functionality of the sliding door. The functionality of the sliding door is built upon a tutorial by VR Playground (2021). The sliding door also implements Grab Component, although, since the movement of the door is restricted Grab Component had to be modified to accommodate for this. Since Grab Component reacts to a Grab Type, I could define a custom Grab Type I called Slider Door, with which object references for the grabbing controller, hand mesh and its physics parent could be transferred to the Printer Blueprint as parameters of a created Grab function. The Grab function saves these references into variables for the Printer and disables physics simulation and collision for the hand in question. Then every Tick there is a check whether the references are valid and if the Printer isn't currently printing, the Slider Door's Location is calculated and set. This is done via a spline that defines the way the door can move, on which the function "Find Closest Point on Segment" outputs the door position based on the Motion Controller's Location. This Location is then additionally used as the driver to outputs from 0 to 1 how open the door currently is, and the opacity of the glass is changed based on that, so that the glass is fully opaque when the slider is closed, to obfuscate the despawning of the cube and the spawning of the hands. In addition to that, a Boolean variable is set, that indicated whether the door is fully closed or not.

After this, the new hand position is calculated, because the hand should be holding onto the handle, even if the controller is moved further back or forth. This is where the Physics Parent comes into play, it can be moved away from the controller and it's set to the location of a socket in the handle of the Slider Door, slightly adjusted for whether it's the right or left hand. This way the location of the hand will now stick to the door, but to make the rotation still react to the way the controller is held, the X Rotation is isolated, so the hand can only roll around the bar, but doesn't tilt sideways.

There is also a Release function, to which only the grabbing controller is transmitted to check whether the same controller is releasing the door. If that's the case, the references for the Controller, the Physics Parent and the Hand Mesh are cleared, after sweeping the Physics Parent back to its original location and resetting the Hand's physics, as well as the collision responses.

For the printer's main functionality there's a Box Collision inside the printer, which detects if the overlapping (inserted) object is a cube, if that's the case a Boolean variable is set to indicate that. Then the cube is saved as a reference and its color is changed from the typical yellow to blue. If the overlap ends, the color and the Boolean are changed back, and the reference is cleared. There is an Event that can be triggered, to activate the printer, which first checks the printer's current status. If it's not powered on, it does nothing, if it is, it checks whether the cube is placed and the door is closed, if that's not the case, the printer's status lights flash red for half a second to indicate an error occurring. Otherwise, it's state is set to printing, the lights are changed to yellow and the reference of the cube is used to destroy the cube and the Boolean to indicated a cube being placed is set to false again. Now the two static hands are being spawned within the printer and there is an overall delay of 2 seconds until the status is changed back from working to ready to conclude the process and enable the door to be opened again. Last an Event is called for the Print having been completed.

Now that all of the different actors had their functionality, they only needed to be connected in the Level Blueprint itself so they could interact with one another as well as get tracked for Level Progression. To track progression another Widget Blueprint was created, though this one couldn't be interacted with, it simply showed the tasks along with checkboxes and provided functions with which the completion of a task could be set or revoked. The functions checked the checkbox in question and strike the text to mark completion. When all the tasks were complete it showed a message on how to exit the level.

Within the Level Blueprint at the start of the Level the last available Hands are being loaded and then Listeners to the different Events of the Buttons, the Print's completion, the Power-Socket and the Hand Display are being created to be able to react to their Events. The Power-Socket causes two events, one when the Power-Core is inserted, from that the Power-Button is activated. The other Event triggers if the Power-Core is removed, and then every actor is being deactivated. The Power-Button checks first if the Level was already completed, which loads the next level, if not it checks if the Power is already turned on and depending on that, every actor is either activated or deactivated. If the Dispenser Button is triggered, the Dispenser's Event "Release Cube" is called, and it's the same for the Printer and it's assigned Button. Each of the steps is also updating the Widget to display the progress. Last the Hand Display's Events for putting in our pulling out the hands is setting the Variable for the Level being complete after which the Power Button will trigger the loading of the next level.

The level now had full functionality, but the environment was rather bare and artificial, as it was the standard environment within Unreal's VR Template. Therefore, a basic environment and a few supporting objects needed to be built. With the Environment itself, I wanted to provide a few different lighting situations in which to see the hands in, so I thought of partial sunlight partial shade, provided through a skylight. I also didn't want the experience to feel claustrophobic but also to naturally restrict the area where the task is taking place, so I thought of an elevated mezzanine area as part of a larger room, to give a feeling of space and openness with the restrictions of the smaller mezzanine. The Architecture was inspired by the 2019 game "Control", which is using a lot of brutalist inspiration, exposed concrete and light shafts, though its light shaft's source is left mysterious, while I used the sun (as a directional light) to provide direct illumination through the light shafts in the ceiling and more indirect illumination through light shafts in the walls. This makes for diverse lighting that the hands are exposed to during the task. Since I wanted a futuristic and also scientific feel to the experience, in combination to the concrete, the items, as well as the furniture were conceptualized as metallic, with light elements creating the futuristic feel. To not expand the project further and limit modelling, I used some elements of Jonathon Fredericks "Modular SciFi Season 1 Starter Bundle" (2017), which was available for free on the Unreal Engine Marketplace, for the Task Level, specifically a metal lab desk, metal railings, to fence off the mezzanine, and a small shelf, on which I located the Power-Cores.

For better and easier reach, I wanted the items to all be situated around waist level. Out of this reason, as well as to create a more believable environment, I modelled a few more objects without functionality. For the Dispenser, I created a wall mount with a kind of sink and an arm next to it, to catch the cubes being released and to hold the Button, with which the dispenser is operated. I had also created a simplistic desk, which I ended up using for the customization and survey levels, but since it has a lighter design, and I wanted something heavier holding the devices, I chose the desk I mentioned earlier for the task level. To make the Power Socket more obvious and better accessible, I created a slanted element holding the socket and the corresponding Power Button. And last, to not have the UI float in the air, cubes with a dark material were used as a sort of display behind the UI Widgets, for which I modelled a simplistic base holding the display. I also created some basic light strips to decorate the desk with, which use the same glowing material the other devices use, which just light up to indicate everything being powered on, since I didn't implement auditive or haptic feedback.

5.2.4. Questionnaire and Structure

Since I wanted the survey to be taken in VR, to have the experience still fresh in mind as well as to enable the participants to look at their hands for more ease answering the questions, I also created a Widget Blueprint to conduct the Survey with. This was mostly based on the Widget created for customization, with one specific widget element consisting of a question, 7 radial buttons, and a scale for the buttons, created specifically for the survey. This element could be reused for all of the questions and could be populated via a CSV data table (Dev Enabled, 2018a) inside the Survey Widget. The Survey Widget could then after completion save out all of the answers together with their corresponding questions, as a CSV file again (Dev Enabled, 2018b), to easily gather the data collected.

Overall, there were four distinct levels with seven levels overall: Intro and Outro Levels, the Task Level featured twice, the Questionnaire Level also twice and the Customization Level. The Intro Level just had a button to start the study with, which I included so the participants could get used to the VR Environment, the Hands adjust the headset and start when they were ready. The Intro Level loads the first Task Level, which, after completion, loads the first Survey. After these answers are saved, it starts up the Customization Level, which, in turn, leads to the second Task Level, that leads to the second Survey Level, concluding with the Outro Level thanking for participation.

5.3. Participants

For the next point, I want to get into the participants used for the study. The participants were sourced from the student and employee body of the Hochschule der Medien Stuttgart at which the study was conducted, as well as from my circle of personal friends. Because of the exploratory nature of the study, the selection of participants was very open and the only requirements for partaking were being over 18 years old and possessing a moderate understanding of the English language, since the questionnaire, as well as the UI in the VR application were in English. The participants were not compensated for partaking.

In total, 15 people took part in the study, 8 identifying as male, 5 as female and 2 as non-binary. The participants had a mean average age of 27.46 (SD = 9.133) ranging from 19 to 59. They classified themselves as mostly white, with 6 participants choosing category 1 ( ) as closest to their skin color, 7 choosing category 2 ( ), and 1 participant each identifying with category 3 ( ) and category 4 ( ).

When looking at the participants' prior experience, their gaming habits had a wide range, with 4 participants stating they play videogames almost every day, 3 declaring themselves to play a few times a week, 2 playing a few times a month, 1 indicating a few times a year and 5 participants citing they play less. Of those 5 participants, 2 declared never playing a game with character customization before, but all of the other 13 participants had exposure to character customization previously.

In regards to virtual reality, 4 participants said that they were never in VR before, among the other 11 participants 5 stated having little experience with virtual reality (1-10h), a further 4 had limited experience within VR (10-30h) and the remaining 2 cited having moderate experience (30-60h) and substantial experience (60+h) respectively, meaning that in conclusion, the majority of participants (9 out of 15) had none or very little experience with virtual reality.

5.4. Expectations

5.4.1. Presence

In relation to the felt presence within the virtual environment, I expected to see a substantial increase among all participants after customizing their hands, since Jung et al. (2018) and Waltemate et al. (2018) found that personalization, significantly increases the presence participants felt within the virtual environment. Since I would consider personalization to be a less potent form of customization, I expected to find a significant increase in felt presence among all participants.

In addition to that, I expected the increase in felt presence to be even higher for non-male (i.e. female and non-binary) participants, since the generic hands participants are first exposed to, were designed to be read as male. Therefore, I expected in accordance to Schwind et al.'s (2017) findings, that due to the incongruence in gender the initial felt presence would be lower for non-male participants, while the increase in felt presence after the customization would be higher.

Depending on the participants who signed up for the study, I also expected similar results like ones described in regards to gender to happen in relation to the participant's skin color creating an incongruence to the white generic hand first exposed to, something Schwind et al. (2017, pg. 4) also proposed as possible findings in further studies. Due to the my dependance on volunteers, this was a factor that I could only substantially investigate if a large enough group of non-white participants would partake in the study.

5.4.2. Virtual Body Ownership

In regards to virtual body ownership I expected a substantial increase in felt ownership among all participants after the customization, since the process of customization itself already creates a strong sense of psychological ownership (Schanzer et al., 2018), in addition to the increase in virtual body ownership observed in previous research in relation to personalization (Jung et al., 2018; Waltemate et al., 2018).

Additionally, I expected a similar effect, as the one expected for presence in regards to gender and skin color, i.e. that the initial ownership would be lower, and the increase in ownership would be higher for these demographics.

5.4.3. Uncanny Valley

For the Uncanny Valley, I didn't particularly expect much, neither in relation to possible experiences of the Uncanny Valley, nor did I expect it to change, since the hands don't gain properties during customization, which relate to them looking more or less human.

5.4.4. Use and Experience of Customization

With the different ways users can approach customization, I expected some users to just create other hands they thought looked cool, instead of remaking their own hands, for which I added the corresponding questions to the questionnaire, asking about perceived closeness to the participant's own physical hands, as well as the felt freedom of customization from different angles and straight out their intent within customization.

5.5. Execution

The execution of the study went rather smooth, with only slight technical difficulties arising once or twice related to SteamVR, which was used to run the VR Preview on, that could be resolved within minutes, and in regard to OBS where the audio for the last participant somehow wasn't recorded.

Two participants reported developing headaches and were asked if they wanted to abort the study, but they wanted to keep going. The headsets are possibly related to the Oculus Rift S missing optical features causing a strain on their brain, matching with the symptoms that were reported.

Overall, the participants spent on average 20:31 min in VR of which on average 6:08min, or 27% of their overall time was spent within customization. The participants also in most cases enjoyed the experience, with the average enjoyment overall (pre, post and customization) rated at 6.04 out of 7, with the lowest ratings used being 4, i.e. neutral.

5.6. Biases

There are certain biases weighing upon this work, I want to briefly get into, though, since this is an exploratory study, the findings will have to be investigated further anyway.

As mentioned above in Procedure (5.1.3.), because of the use of questionnaires within VR itself, recall bias is being avoided. But other biases affect this study's validity.

As an interviewer conducting the study all by myself, I tried to stay as unbiased as possible, but reviewing the footage gathered there is certainly some degree of interviewer bias present. I tried to give everybody the same hints and clues, but in some cases, I was more explicitly encouraging towards some participants to play around with the environment in general, like spawning a lot of cubes and playing around with them, which might have led them to have more fun or be more immersed in the environment.

I may also have unconscious researcher bias, because I am modelling this study after other studies previous findings, I bring certain expectations with me, that might affect my interpretation of the data.

There might be demand characteristics present among participants, since I was open upfront about what was being researched, though that would only affect their relation to immersion, and that only in cases where participants make the connection between immersion and presence.

Since I also drew participants from my personal friend group, courtesy bias might occur for them, that they might convey a certain outcome because they would not want to disappoint me, although I deem their answers as honest. It could also affect their behavior during the study, because of the rapport they have with me, they might be more willing to mess around and try out things than other participants and therefore also have more fun. They also might have a predisposition toward liking the hands, for example, because it is something I made.

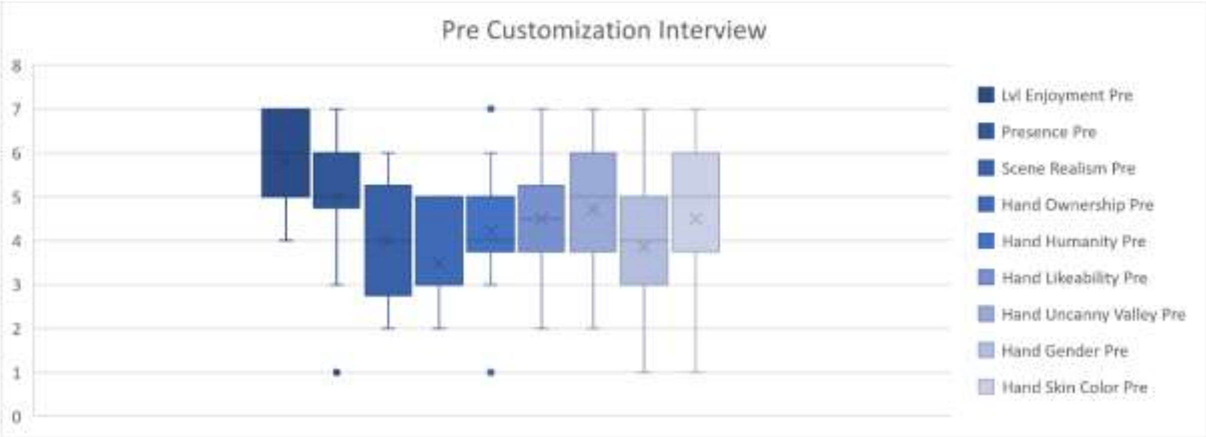
There is also the possibility of extreme response bias, since I am using Likert scales. This might be especially prevalent in regard to immersion, since VR is inherently already a very immersive medium, that might lead participants to answer very high in regards to presence toward the beginning already, making it harder to detect changes.

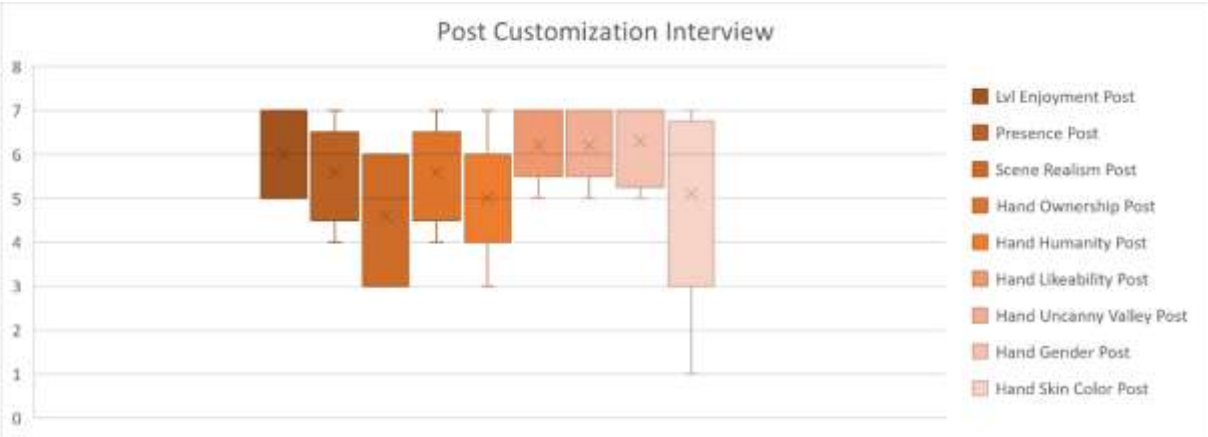
Sampling bias certainly occurs, since I couldn't sample participants representative of the population, as well as the small number of participants overall.

Volunteer bias is also possible, since people are likely to have volunteered to participate because they are interested in the topics and virtual reality, affecting them to behave or react different than the general population would.

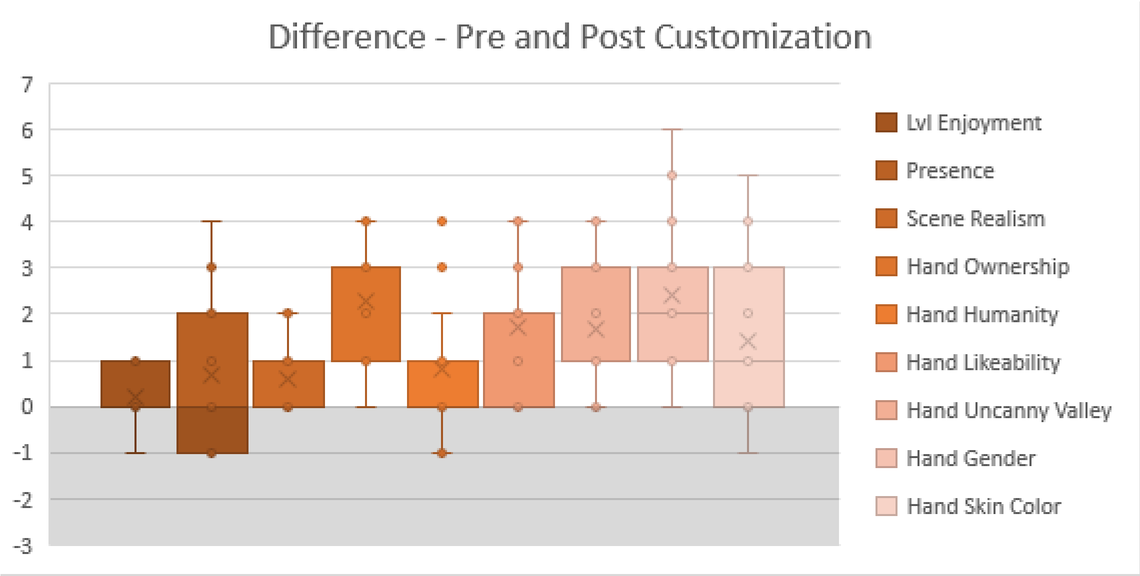
5.7. Results and Discussion

Now to look at the results found within the study, first, I want to present the answers, that people gave as answers to the questionnaires within Virtual Reality. Therefore, I created some box plots to best get an overlook over the data:



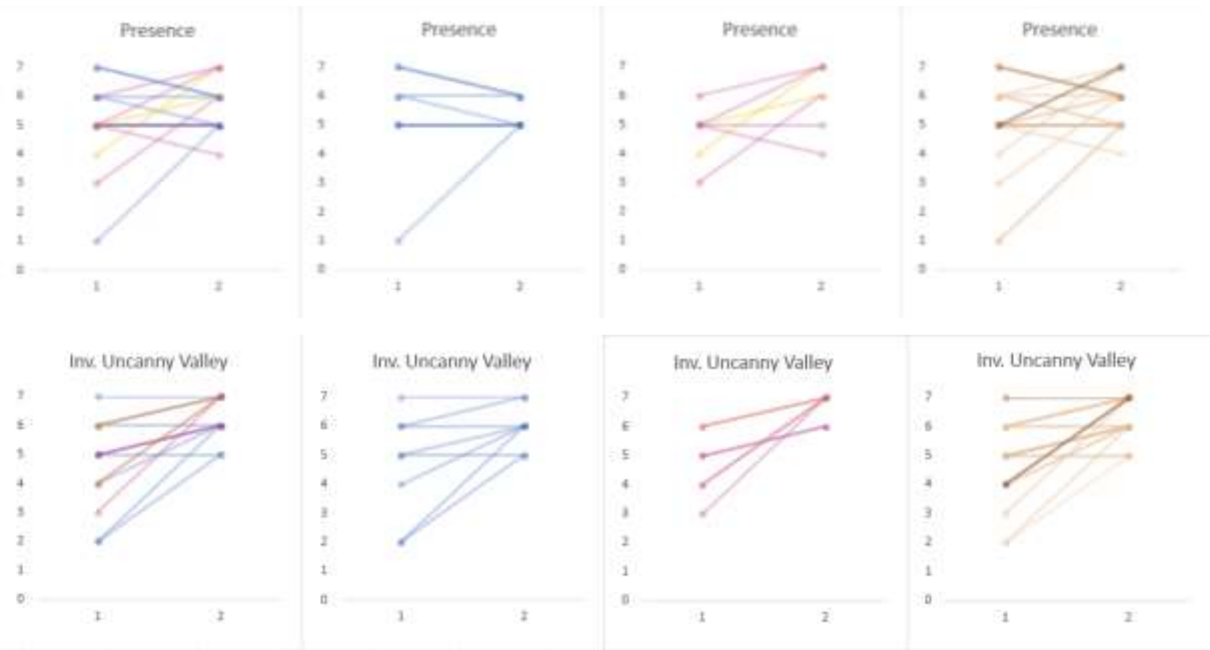


Next, to track how the answers changed from the generic hands to the customized hands, I subtracted the pre-answer from the post answer, which gives this overall change between the initial hands and the customized ones:



This shows an overall positive effect of the customization on the experience, and it interestingly affects the uncanny valley as well, where people seem to be more comfortable with the hands, they customized than generic ones.

To further track in which ways the customization changed their experience, I tracked specifically the presence and uncanny valley in regard to gender and skin color.



In regard to skin color, there weren't any particular differences among participants, which probably is the case due to the lack of diversity in skin color among participants.

Among Gender though there was a big effect between the experiences of the different genders, as this calculation of averages shows, listing likeability, ownership, uncanny valley and presence averages from before and the difference in which they changed.

	Like Pre	Own Pre	Unc Pre	Pres Pre
M	4.5	3.125	4.625	5.25
F+NB	4.5714286	4	4.7142857	4.7142857
F	4.2	3.6	4.6	4.8
NB	5.5	5	5	4.5

	Like Diff	Own Diff	Unc Diff	Pres Diff
M	1.375	2.625	1.375	0.125
F+NB	2.1428571	1.8571429	2	1.2857143
F	2.4	2	2	1
NB	1.5	1.5	2	2

Interestingly the Ownership changed more among male identified participants than the other genders, but the presence had basically no change among male individuals but a, still small, but significantly larger than the male participants.

The difference on the gender question was extremely pronounced in non-binary participants, having an increase of 5.5 points, with both being extremely excited about the customizations, one saying “Yes! My gender is turquoise glitter nail-polish!” (translated from German) and the other saying “ohh, I feel so good, I feel more myself in this than I feel in reality. Can you make a gender-game, where you can be whatever you want to be?”

Generally, 12 of the 15 participants remarked upon the realism of the hands especially in relation to that they were missing the textures on the hands and would wish for further texture-based customization options, like age, scars, hair and tattoos, which probably was the biggest inhibitor of accepting the hands as real, whereas it was also mentioned a couple times that they were still “just floating hands”, and that arms at least would probably lead to better immersion: “I think it would be even better if you had more of your arm, they’re still two floating hands”.

5 people noted that they found the customization challenging because they couldn’t see their own hands “if I could just see my hands, as a comparison […] it’s interesting, if you try to build yourself in Sims, you’re always looking in the mirror, because you ask yourself, how do I even look? I mean, we see our hands the whole day, but does anyone notice them?” (translated from German).

6. Conclusion

In conclusion, I think that the results seem to align with the findings pointed out in the research section. Customization is a very powerful tool, though within VR you have to tow a very delicate line, to make people accept the virtual environment and their avatar. If Customization is looked at further, it should probably be done on at least full arms as models, as well as with a larger and more diverse sample of participants, to get a better estimate for how customization in VR can affect the population.

7. References

7.1. Bibliography

Barnes, T. (Ed.) (2018). Proceedings of the 49th ACM Technical Symposium on Computer Science Education. *ACM Conferences.* ACM.

Bouchard, S [S.], Robillard, G [G.], St-Jacques, J [J.], Dumoulin, S., Patry, M. J., & Renaud, P [P.] (2004). Reliability and validity of a single-item measure of presence in VR. *In Proceedings. Second International Conference on Creating, Connecting and Collaborating through Computing* (pp. 59–61). IEEE. https://doi.org/10.1109/HAVE.2004.1391882

Bouchard, S [Stéphane], St-Jacques, J [Julie], Robillard, G [Geneviève], & Renaud, P [Patrice] (2008). Anxiety Increases the Feeling of Presence in Virtual Reality. *Presence: Teleoperators and Virtual Environments, 17*(4), 376–391. https://doi.org/10.1162/pres.17.4.376

Clifton, J., & Palmisano, S. (2020). Effects of steering locomotion and teleporting on cybersickness and presence in HMD-based virtual reality. *Virtual Reality, 24*(3), 453–468. https://doi.org/10.1007/s10055-019-00407-8

Jung, S., Bruder, G., Wisniewski, P. J., Sandor, C., & Hughes, C. E. (2018). Over My Hand. *ACM Conferences, Proceedings of the Symposium on Spatial User Interaction* (pp. 60–68). ACM. https://doi.org/10.1145/3267782.3267920

Ma, K., & Hommel, B. (2015). The role of agency for perceived ownership in the virtual hand illusion. *Consciousness and Cognition, 36*, 277–288. https://doi.org/10.1016/j.concog.2015.07.008

Mark, G. (Ed.) (2017). Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. ACM Digital Library. ACM.

Mottelson, A., Muresan, A., Hornbæk, K., & Makransky, G. (2023). A Systematic Review and Metaanalysis of the Effectiveness of Body Ownership Illusions in Virtual Reality. *ACM Transactions on Computer-Human Interaction, 30*(5), 1–42. https://doi.org/10.1145/3590767

Parrish, A. (2020, December 15). Cyberpunk 2077's Character Creator: The Kotaku Review. Kotaku. https://kotaku.com/cyberpunk-2077-s-character-creator-the-kotaku-review-1845882374

Schanzer, E., Krishnamurthi, S., & Fisler, K. (2018). Creativity, Customization, and Ownership. *In T. Barnes (Ed.), ACM Conferences, Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 161–166). ACM. https://doi.org/10.1145/3159450.3159471

Schwind, V., Knierim, P., Tasci, C., Franczak, P., Haas, N., & Henze, N. (2017). "These are not my hands!": Effect of Gender on the Perception of Avatar Hands in Virtual Reality. *In G. Mark (Ed.), ACM Digital Library, Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 1577–1582). ACM. https://doi.org/10.1145/3025453.3025602

Selzer, M. N., & Castro, S. M. (2022, June 15). Immersion Metrics for Virtual Reality. Institute for Computer Science and Engineering (UNS-CONICET) and VyGLab Research Laboratory (UNSCICPBA), Department of Computer Science and Engineering, Universidad Nacional del Sur, Argentinia. http://arxiv.org/pdf/2206.07748.pdf

Waltemate, T., Gall, D., Roth, D., Botsch, M., & Latoschik, M. E. (2018). The Impact of Avatar Personalization and Immersion on Virtual Body Ownership, Presence, and Emotional Response. *IEEE Transactions on Visualization and Computer Graphics, 24*(4), 1643–1652. https://doi.org/10.1109/tvcg.2018.2794629

7.2. Videography

Gronkh. (2020, December 9). *CYBERPUNK 2077 [4K60/RTX] 🤖 000: Prolog // Charaktererstellung* [Video]. YouTube. https://www.youtube.com/watch?v=ymax6WjrMIQ&list=PLGWGc5dfbzn92rrXktK35Mx7MraxIf_kD&index=2

Gronkh. (2023, August 7). *Alles eine Frage des Charakters 🐉 Baldur's Gate III #001* [Video]. YouTube. https://www.youtube.com/watch?v=ctdOWQ0PIjA&list=PLGWGc5dfbzni1vTIwAbHDNtnL0G2rqYy

Let's Make FX. (2021, May 1). *Glitter Shader Tutorial UE4* [Video]. YouTube. https://www.youtube.com/watch?v=pSRM3n14k3A

Max SMoke. (2019, November 23). *Make a VR mouse cursor in UE4 on a widget* [Video]. YouTube. https://www.youtube.com/watch?v=LLBSuxIBZV8

Quinn Kuslich. (2022, April 11). *VR Hands and Animations in UE5 & UE4.27 | Tutorial Part 1 | Adding Hands and Gesture Animations* [Video]. YouTube. https://www.youtube.com/watch?v=mee_6vlWZ8c

Quinn Kuslich. (2022, April 11). *Procedural VR Hand Animation in UE5 | Tutorial Part 3 | Half Life: Alyx Procedural Grip Animations* [Video]. YouTube. https://www.youtube.com/watch?v=Lq65clDAttA

Straszfilms. (2021, February 3). *Identity, Gender, and VRChat (Why is everyone in VR an anime girl?)* [Video]. YouTube. https://www.youtube.com/watch?v=5v_DI7i4Bcw

VirtualRook. (2023, October 20). *Let's Make A Physical VR Button In Unreal Engine 5.3* [Video]. YouTube. https://www.youtube.com/watch?v=m8DGSmDUHfk

Virtus Learning Hub. (2022, July 23). *Build User Interfaces In VIRTUAL REALITY For Unreal Engine 5* [Video]. YouTube. https://www.youtube.com/watch?v=kM27HYbpvc0

VR Playground. (2021a, May 21). *Attaching Components with Sockets - Unreal Engine VR* [Video]. YouTube. https://www.youtube.com/watch?v=kahjV2SjFWE

VR Playground. (2021b, September 3). *Hand Actor Collision - Unreal Engine VR Tutorial* [Video]. YouTube. https://www.youtube.com/watch?v=Ly1XMXVdPlU

VR Playground. (2021c, December 10). *Make a Slider for Virtual Reality - Unreal Engine VR Tutorial* [Video]. YouTube. https://www.youtube.com/watch?v=If4JQOQYeD4

VR Playground. (2022, April, 8). *Physical VR Hand Animations in Unreal Engine* [Video]. YouTube. https://www.youtube.com/watch?v=_n6Nk4NNsDc

7.3. Web Content

Dragon's Dogma 2 Charakter-Erstellung & Speicher bei Steam. (2024, March 17). https://store.steampowered.com/app/2674810/Dragons_Dogma_2_CharakterErstellung__Speicher/#app_reviews_hash