Avatar: The Legend of VR

1st Hannah Köster
Institue of Cognitive Science
University of Osnabrück
Osnabrück, Germany
hakoester@uos.de

2nd Janeke Nemitz
Institue of Cognitive Science
University of Osnabrück
Osnabrück, Germany
jnemitz@uos.de

3rd Jule Körner
Institue of Cognitive Science
University of Osnabrück
Osnabrück, Germany
jukoerner@uos.de

4th Michael Hüppe
Institue of Cognitive Science
University of Osnabrück
Osnabrück, Germany
mhueppe@uos.de

5th Nicolai Hermann
Institue of Cognitive Science
University of Osnabrück
Osnabrück, Germany
nihermann@uos.de

***Abstract*—As large parts of our lives increasingly take place online, with platforms like Zoom, Google Meet, and Facetime being used everyday by millions of people, the world of virtual avatars as well as their design gets more and more important. This study investigates whether people prefer to interact with virtual avatars that match their visual preferences. We conducted a VR experiment with 13 participants where they had to make several decisions between two avatars that matched their preferences to different degrees. Afterwards, participants filled out a questionnaire to assess the reasoning behind their decisions. Our analysis shows that participants were more likely to chose and keep those avatars that matched their preferences to a higher degree and that visual appearance in the form of color and style was the main influence on their selections. We conclude that people indeed seem to prefer interacting with avatars that match their own visual preferences, however, more research needs to be done on this topic to determine which visual factors have the largest influence on how people perceive virtual avatars and also how much importance movement and posture have in that regard. Furthermore, our entire project and analysis is publicly available on GitHub here.**

***Index Terms*—VR, HRI, avatar, uncanny valley**

I. Introduction

The world, particularly the online world, is ever-changing. Nowadays, our lives are heavily influenced by the internet. For practically everyone, daily online meetings via platforms like Zoom, Facetime, or Google Meet are on the agenda. That is why a virtual world devoid of genuine human relationships, as is the case with today's social media, is out of date. The pandemic forced us to become increasingly socially isolated, reinforcing the desire for many to create a more cohesive and integrated digital social reality. Your digital identity is more than just an avatar; it's a passport that connects all of your online actions in one place. It's all about using data and visual things to convey the concept of identity into the digital realm. Companies like META and Microsoft began to use Avatars to generate such online identities as a result of their newfound relevance. This opens up a whole new universe of possibilities. These avatars can be used in gaming, as well as for presenting ones physical appearance and style of fashion. In the metaverse and virtual worlds in general, beauty, aesthetics, fashion, and the entire realm of self-care and well-being have a large field of self-expoloration. Of course virtual avatars can also be helpful in terms of education, training and even social studies. Additionally, simulating a marketing campaign or an advertisement in the digital real can be done more safely, controlled and cost efficient.Moreover, gauging public interest in a product, and organizing a contest are all examples of possible simulations in the virtual world. [3] But why should we keep or have a virtual environment (even when the pandemic is gone)? Remote workers have complete control over their schedules, making balancing personal and work life much easier. People cope with less distractions and become more productive when they can more easily mix work and personal lives and operate in their own regulated environment. Recent statistics from the Global Workplace Analytics suggest that companies with a large number of workers that work remotely have lower operational costs and an increase in productivity [2]. These statistics also show that with fewer employees commuting and reduced energy consumption for transportation, virtual offices are more environmentally beneficial [2]. When working from home is an option, employing personnel is no longer limited by geography. Bringing in the best applicant, not just the best local applicant, can assist a company. It is no longer essential to relocate.

There has already been a great deal of research on the appearance of robots, making it hard to avoid the so-called uncanny valley effect, which defines a phenomena that occurs when humans engage with robots. It states that the affinity of an robot increases with it's similarity to humans. Think of a mechanical assembly robot and a somewhat humanoid toy robot (here to the degree that it has two arms and two legs but is no where indented to actually resemble the human form). According, to studies for the uncanny valley effect most people would feel more positively drawn to the toy robot. However, this correlation between human-likeness and affinity has a harsh drop when the robot in question resembles a human too closely. Thus, a lot of people have an unnerving or unpleasant sensation that some individuals have when they see not-quite-but-closely-to-human creatures such as humanoid robots or lifelike computer-generated characters.

Following this previous research the idea for our research was based on the idea that people prefer interacting with other people(robots) that match their preferences [6]. Keeping that in mind our research paper discusses the question of weather humans prefer interacting with an avatar that is specifically designed after their visual preferences.

II. Avatar

To create custom avatars which are designed after the preferences of a participant we searched for a way how we could dynamically alter the visual appearance of an avatar. Our initial plan was to use the Unity Multipurpose Avatar System (UMA) to instantiate the avatars in Unity. UMA is a character creation and modification system that, among other things, allows to modify parts of avatars (e.g. changing the shape or size of body parts) and also provides a simple way of adding or changing additional features like clothes. We planned to use UMA to adjust the body of the avatar (i.e. the height and width of bones) as well as allow for different clothing and hairstyles, adjusting colors, and allowing different accessories like glasses, hats and jewelry. UMA would provide a simple solution for making these adjustments, provided we had a so called UMA "base race" and fitting accessories. To avoid the uncanny valley effect, we decided to make the characters have a unrealistic and more specifically cartoony look [4]. That means that the avatar should be recognized as a human character, having a distinctly human appearance, but it should also be clear that it is not intended to look realistic. As no free assets were available in the style that we wanted to use we decided to create our own base models, one male and one female one, and set them up as UMA races.

A. Creating the base avatar

To achieve the desired cartoony look, we modified the proportions of the body, for example making the head larger compared to the rest of the body thus different to a realistic depiction of a human. Additionally we enlarged the eyes and avoided too many details. This is also reflected in the choice of materials, which had no details and simply consisted of a single color. Clothes and hair were also stylized.

Initial concept drawings were created on paper to lay down the style, proportions, and general look of the avatar. The best ones were then digitized to be used as reference images in Blender – a free, open source 3d software [5] –, which was used to create the 3d model.

Although we originally planned to create two base models, during the creation of the female model it became apparent that the process was more time consuming than we anticipated and at some point we decided that one base model has to suffice. The model was mostly created in Blender's sculpt mode, where the 3d model can be shaped using different tools called brushes to extend and retract parts of the mesh and change the topology in a way that is reminiscent of sculpting something out of clay. This way organic shapes can be created much more easily than with modeling. However, this process is different from modeling, so even if one has some experience with that, sculpting will still be difficult and require a lot of practice. One should not underestimate the difficulty and the amount of time this takes. Sculpting organic shapes such as a human avatar is more similar to drawing than modeling – and best done using a drawing tablet which also enforces this feeling –, but one should not forget to be aware that one is creating a 3d sculpture, and that it has to look good from more than just one angle. We made most of these – in hindsight obvious – mistakes, underestimating the difficulty of sculpting and therefore the time that creating an avatar would take. Getting usable results required a lot of practice, we also had to learn that instead of starting from scratch, trying to redo the whole thing every time we were unhappy with the current look of the avatar and thought it beyond saving, it is more efficient to just take a few hours practicing a specific part like a mouth or nose and then try to incorporate that into the whole avatar. Conforming to the agreed upon style is a whole other issue, in our case it was easy to get lost in details, reference images can help with this and also taking breaks and then comparing to the reference and checking if the style is still appropriate, but multiple iterations will probably still be required to achieve the intended look if one is not already familiar with the style. It will be hard to match the reference perfectly, so some deviations should be accepted if the result still looks acceptable to save time that can better be spent on other parts of the experiment. As seen in figure 1, our final avatar did not completely match our reference drawings, but it looked like a human and conformed to the cartoon style and perfecting it would have taken too much additional time that we did not have.

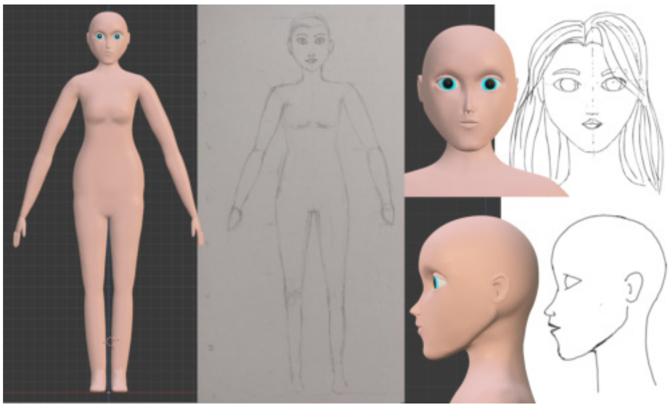

Fig. 1. The finished avatar model compared to the reference images.

After the avatar mesh was finished, we used the pre-installed Rigify add-on to generate a human rig which we then aligned with our model. To generate the weights which affect which parts of the mesh are influenced by certain bones and how much, we used the automatic weights function, but we ended up having to adjust them manually later in some areas to allow for certain movements without deformation of wrong body parts that should not be affected. For accessories we also tried to use automatically generated weights but often needed to adjust them and sometimes completely painting them manually was required and the results are still not ideal with skin coming through in some animations.

We encountered some issues importing the model into Unity and especially when trying to create an UMA race from it. These issues seemed to be caused by problems with the rig – more specifically with the weights missing or additional bones that Unity or UMA either did or did not expect, and also the naming of the bones which did not conform to expectations of Unity or UMA. Ultimately, we were unable to set up an UMA race of the model in the time that we were willing to spend on that part and decided to import the model with all the accessories into Unity and just controlled the visibility of the features. While this allowed for easier ex/import between Blender and Unity and also adjustable accessories/clothing we are not able to satisfyingly adjust the avatars body.

III. Clothes

As soon as we had the first character prototype ready, we started developing the idea for the clothing design. In general, we decided on having three different clothing styles, namely, casual, chic, and sporty. In addition to that, our intention was to also reflect the cartoon theme in the design of our clothes. We defined a cartoony clothing style as not looking too realistic and figure-hugging, but rather simple. How we realized this will be made more explicit in the following sections.

A. The cut

We agreed on designing at least one outfit for each style type:

- A Loose-fit pants and a t-shirt for the casual look.
- A suit for the chic style.
- A pair of sports pants and a top for the sporty look.

With this rough sketch ready we started the realization in Blender.

In the end, we used two different techniques to come to our final outfits: (1) Download existing pieces from different stores, and (2) Design the pieces from scratch.

1) Downloading existing clothes: First, this idea seemed most efficient and simple to us. We downloaded free clothes from different stores. Turbosquid and CGTrader were the ones we used most frequently. They offer pieces in many formats, like Filmbox (.fbx) and OBJ (.obj) files, which are good to import to Blender. The offer of pieces was huge, but due to our design idea, we were quite restricted. Many clothes looked too realistic and thus contradicted the cartoon style. Nonetheless, we found some pieces.

Our general steps to put the clothes on the character included roughly scaling the piece such that the size seems to fit the character, moving it in the right position to cover the body shape properly, scaling separate parts again.

Unfortunately, in many cases, the initial expectation regarding simplicity was not met. It costs quite a lot of time to move and scale the clothes such that they fit the character.

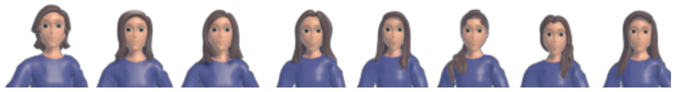

Fig. 2. hairstyles

Furthermore, the results were often quite sobering. Due to the fixed cut of the clothes, they often did not fit the shape of the character properly. Meaning, we ended up with unnaturally looking, deformed outfits or ones which did not cover each part of the character's body. Despite the difficulties we had with some clothes, we got the shirt and our accessories, like glasses and cap with this technique. Additionally we used a similar technique to acquire the different hairstyles which can be seen in figure 2.

2) Designing the pieces from scratch: As already mentioned, due to the mismatch of the body shape of the character and cut of the clothes, we did not get satisfactory results for a T-shirt/top and trousers/sports pants design by using the former technique. That is why we decided to design these clothes directly on our customized character. We became aware of cloth simulation, or to be more precise, cloth sewing in Blender.

`cloth` is part of the physics simulation Blender offers. Applying a cloth modifier adds cloth properties to the respective object. To give one example for a cloth property, an object with a cloth modifier will not look rigid, but soft. This modifier together with the collision modifier is the basic building block for cloth sewing.

As the name already suggests, sewing clothes in Blender is comparable with real sewing. That is why this technique seemed so intuitive to us. Furthermore, we found amazing step-by-step videos on YouTube [1] that let us learn the basic sewing procedure relatively quickly. The good thing was that independent of the clothing we wanted to design, the general sewing procedure stays the same: First of all, we had to place a plane in front of the character and roughly form it to the desired shape. After that, this mesh had to be doubled and the resulting part was placed behind the character. Thus, behind and in front of the character were identical planes, which are then connected with edges. These edges later served as threads that pull the two parts together to form one clothing. In order to get the pulling force, the respective modifiers had to be added. First, a collision modifier was needed for the character, otherwise the clothes would fall through the body instead of colliding with it. Second, the cloth modifier was added. For the shrinking, we had to enable sewing in the `cloth` section, such that the two planes were pulled together along the threads when starting the simulation. By that, we got the clothing. Applying the modifiers gave us the final clothing objects.

When animating the character, we observed that holes appeared in the cloth when the avatar did certain movements. Presumably this was caused by sub-optimal weight assignments from parts of the clothes to the avatars rig. These

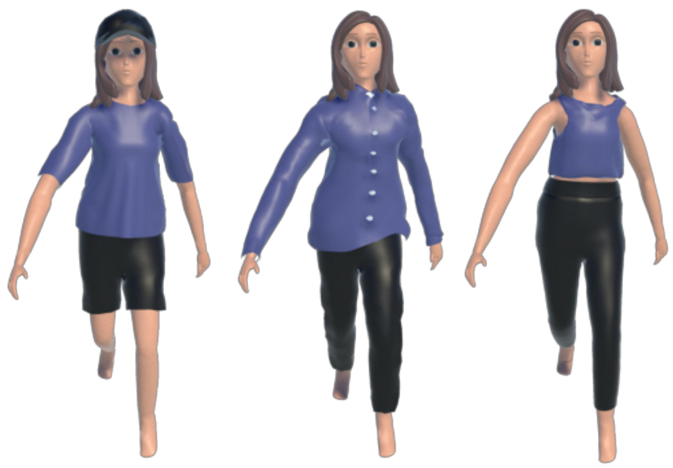

Fig. 3. The three styles (left to right): casual, chic, sporty

weights define how strong the parts of the cloth are attached to certain body regions of the avatar and subsequently how strong they stick to these body regions while the body is moving. When they are slightly out of balance it can happen that the cloth is drawn into the body which we then perceive as a hole. Due to time constraints we did not manage to solve this issue completely. But altogether, this technique delivered us good and quick results and we were even able to add shorts to our outfits which we originally did not even plan for. But luckily this technique was easily applicable when getting the hang of it. The final different types of outfits can be seen in figure 3.

B. The Cartoon Style

Regardless of the technique we used, we always remained dedicated to the cartoony look. Thus, all cuts of the clothing were kept simple without any additional details.

The same applies to the material and texture we chose. We avoided using complex fabric textures like jeans or cotton. The only thing we changed later appropriately to the questionnaire was the color of the clothes.

C. Remarks

As an overall summary of our clothing-design process, we can say that we got all our planned outfits. Although they were not perfect, they served the purpose and we did not perceive the gaps in the clothes as being a significant distraction during the experiment. Getting all the outfits ready was a learning process with many ups and downs taking quite some time. Besides the fact that we were new to Blender, coming up with a new technique after finding out that downloading will not give us the intended result was time-consuming. But all in all, we are happy about the result we got. We learned many new Blender skills and are sure that we would be able to design new pieces quickly when extending the experiment.

Fig. 4. Overall room scene

D. Room

We wanted our avatar to look human-like without reaching the ”uncanny valley”. In order to avoid reaching the “uncanny valley“, we agreed on an overall cartoon-like design, especially with respect to the character’s appearance. In addition to that, we assumed that reflecting the cartoony pattern in the room design would amplify this effect as well.

To impart a pleasant atmosphere to the experiment, we chose a living room as our basic environment. The basic room was downloaded from the internet and we adjusted it to look more cartoony by changing floor texture, wall, and furniture materials as well as adding some additional decorations, like plants and pictures. You can find the basic room here

In general, we can say that there is a large selection of room designs available online for free. Taking them is much easier than designing them on your own in Blender and it will save a lot of time that can be spend on other parts of your study. We designed most of the textures and materials of the room on our own even though there is a large offer of them online. Nevertheless, we learned a lot by doing it on our own, but looking back we could have invested this time better. Our final room can be seen in figure 4

IV. Experimental Setup

From the Hypothesis to the experiment

Coming up with an experiment is the most essential part of the whole study. It is used to test the hypothesis and in the best case, lets you confirm the hypothesis in the end. For that, an experiment has to be found that allows measurements that can be statistically analyzed to test the hypothesis.

We had a quite turbulent way to our final experimental setup. Having decided on our hypothesis, we started with a lot of brainstorming in our weekly group meeting. We came up with many ideas and while some of them had to be immediately discarded, others became more concrete. Next to working altogether, we read papers, to find out what measurements might fit our context. Perugia et al stated in their paper, that finding a robot more appealing increases the number of saccades to this robot [8]. That is why we first decided to use eye tracking in our study, to measure how often the participant is looking at each avatar and whether we can see differences between them. In the end, we decided on a simplified method to assess this (More about the method can be found in Section V).

In addition to our internal group exchange, we benefited a lot from the input we got after our weekly stand-up talks. Not only because we received general tips, but also the questions that were asked made us aware of some aspects we sometimes had not considered and therefore pushed us in the right direction. After multiple missteps, idea scratching and restarting we finally ended up with our current hypothesis and the way to test it. Fortunately, we managed to finish in time but looking back we could have avoided this kind of resulting stress. How we would improve our strategy when doing it again and what we actually had to discard will be discussed in section VII. The final implementation and design of our experiment are shown in detail in the following sections.

A. About the characters

The hypothesis that how much an avatar is liked increases with its level of fit to our preferences presupposes the ability to create avatars of different levels of match. Of course here knowing the preferences of the participant is the most important part. Thus we started our experiment with a questionnaire about ones favourite physical attributes. We gathered information about the participants favourite hair color/-style, clothing style or eye color. Additionally, we asked the participant to indicate their own skin color. We did this to avoid questions that might be insensitive. Additionally, there are studies suggesting that people like other people more when they look similar to themselves [7] Every question presented the option to skip it to avoid making the participant feel uncomfortable. Moreover, the questionnaire emphasised that all the information was anonymous, meaning we are not able to trace any information back the participant themselves. To ensure the validity of the experiment a skipped question was assigned a random value from its possible answers. When more than 50% of the questions had to be filled out randomly (due to the participant not wanting to answer them) we disregarded that run of the experiment. Fortunately, in most cases the participants answered all of the questions so we didn't have to discard any run because of that. This questionnaire was filled out in an Unity scene, thus all the answers were easily saved and send to the script handling the avatar instantiation. The Scene is depicted in figure 5

In total there were 8 attributes representing the participants favourite physical attributes. Namely:

- favourite hair style (short, middle, long)
- favourite hair color (18 possible hair colors ranging from natural tones (blond, brown black) to dyed hair color (pink, blue, green)
- favourite clothing style (casual,sporty, chic)
- favourite overall color (every possible color in the RGB color space)
- their own skin color (30 possible skin tones)

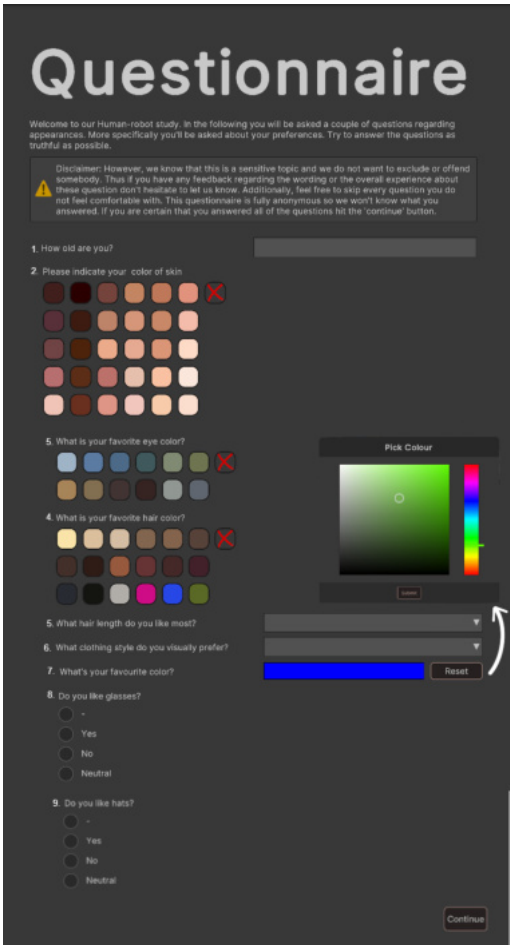

Fig. 5. Starting questionnaire to pick physical preferences.

- eye color (12 color variations of blue, green, brown and grey)
- whether they liked glasses (like, dislike, neutral)
- whether they liked hats (like, dislike, neutral)

In order to test our hypothesis we wanted to create avatars that match the participants preferences to five different degrees. Hence, there was a gradual increase of attributes the created avatar fit with each level. Meaning, while the lowest level did not fit any of the attributes, the highest level perfectly fulfilled all the specified preferences.

In figure 6 you can see how the different avatars might look for each level of preference fit. Here the participant preferences formulated would be something like this: I like brown hair that is not too long or to short. I like a more casual clothing style and hats. However, I dislike glasses. My favourite color is a mix between blue and purple. These preferences actually correspond to the "average avatar", meaning taking the preferences of all participants these were the average answers. An animation of this example can be seen here

1) Implementation Attributes: Each modifiable attribute was associated with different game objects which can be activated or deactivated. Meaning, to change the appearance of an avatar, we just had to specify which attributes we would want to be active and more importantly visible. The same holds for the colors of those attributes. Thus we would just have to specify the game objects and the color we want them to be. Thus, we were able to dynamically create a lot of different avatars by just changing the different attributes. An animated illustration of that can be seen here).

Thus, to adjust the avatar based on the participants preferences, we just have to activate the attribute that is associated with the option the participant chose. For example, if the participant chose that they prefer short hair, the model for short hair was added onto the base avatar, which was stripped from all characteristics and consisted purely the 3D mesh of the body. Since the previous scene (the questionnaire) saved the preferences in an object with each attribute representing one preference this is as easy as calling: `Activate(particpantPreferences.favHairStyle , particpantPreferences.favHairColor)`. This of course is only the case if the avatar was supposed to fit that specific category. Therefore, if the goal was to construct the most optimal/fitting avatar, the base model would get all of the attributes the participant chose. However, if we wanted to achieve the opposite effect all of the attributes were either chosen randomly or in a way that they contradicted the participant's. For example, if the participant stated that he favoured glasses and a more casual look, we made sure that the avatar that was supposed to fit the preferences the least had no glasses and a chic look.

As already stated the numbers of attributes that differ from the specified preferences increased with decreasing level of match. The basic method is illustrated in figure 7.

2) Implementation Attributes Color: In addition to adjusting the clothing style, accessory and hairstyle of the model we also made the color of those attributes relative to the choice of the participant. In the case of a wanted full match for instance, when the participant specified their favourite hair color as blond and their favourite overall color as blue we gave these attributes the corresponding tone. However, to achieve a different look for the other matches we colored those attributes distant to the chosen (most favourable) color. To be more consistent we chose the CIELAB color space. This particular color representation fits the human perceived color assignment [10]. Therefore, colors

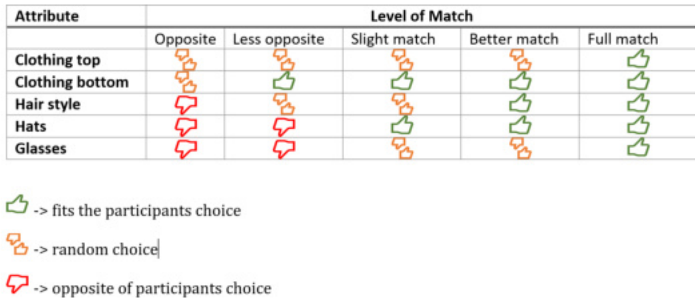

Fig. 7. Attributes to fit the participants preferences based on different levels of match

Attribute	Level of Match (distance in CIELAB space)				
	Opposite	Less opposite	Slight match	Better match	Full match
Favorite color	15	13	8	5	0
Hair color	8	6	5	3	0
Eye color	4	3.5	3	1.5	0
Skin color	4	3.5	3	1.5	0

Fig. 8. Distances between colors for different levels of match (LoM)

that humans typically consider similar looking are close in distance in this color space. However, colors that are easily distinguishable from one another have a larger distance. We employed this color space to have a representation of how far off one color is from the participants favourite one. Here, the distance was defined by the level of match (LoM). The used distances can be seen in figure 8.

B. About the game

The incentive of the game was to move across a huge game board, accompanied by different avatars. The end of the trial was reached when the participant reached the goal field at the end of the board which was recognizable through its size. It was approximately four times the size of a regular field. Along the way the participant was faced with four decisions. On every decision he had to choose between two avatars specifically designed for him. The chosen avatar would then accompany the participant up to his next decision. At the beginning of each trial the participant found himself in an over dimensional large room standing on a huge table, giving him the impression that he was shrunk and is a figure of the game himself. He was equipped with playing cards held by his left hand which

Fig. 6. Avatars of different level of match left to right: Full match, better match, slight match, less opposite, opposite

Fig. 9. Current game cards

he was able to play by pressing the grip button of his left VR controller. The cards had numbers from 1-10 on them in various designs which are illustrated in figure 9.

By moving the left thumb on the controller's touch pad, the thumb in game moves as well, thereby pointing to one of the cards being held. This gives the participant the possibility to select the cards he could then play by pressing the grab button with his pointing finger. Whenever a card is selected via the touch pad, the fields in front of him are colored, indicating the number of fields he would walk if the card is played. After some moves the participant reached a so-called avatar field. When an avatar field is reached the participant automatically stops and all remaining steps are disposed. At that point the participant will find avatars in front of him with different levels of match (LoM). His right hand is now emitting a laser beam that he can direct onto one of those avatars. If one avatar is targeted, the laser beam will change color indicating that a selection is possible. By pressing the grip button with his right pointing finger, the selection will be conducted. From now on the selected avatar will accompany the participant to the next avatar field. While the participant moves by teleportation the accompanying avatar will walk and thus take some time. Therefore, the participant is forced to wait for the avatar before he can pick the next card. Because of that the participant must show some patience with his companion. There are a total of four avatar fields in each trial. In the first decision the participant is presented with two new avatars and for the remaining three stations there will only be one new avatar while the second choice is always your accompanying avatar. Throughout the experiment the participant will see a total of five different game boards which vary in design and location inside the room. For example, one board has a winter theme while being located on the table in the middle of the room, whereas another board is located at the shelf on a wall. By changing the environment, we intended to make the whole experiment more interesting, preventing the participant from getting bored and absent. All the different boards and environment are depicted in figure 10. An example video for one experiment run can be seen here.

1) Implementation:

a) The board: If you look at figure 11 you can see an illustration of the game board. To start off, each white circle here is a field prefab which holds a reference to the next field in front of it thus creating the route of the board. The large circle is a field additionally tagged as `GoalField`. The goal field does not have a next. The green rectangles are virtually identical to the circles but resemble the route of the accompanying avatar and are invisible to the participant. This basic setup makes it easy to navigate across the board, where the movement is defined by the number of steps, depicted on the hand cards of the participant.

b) The game flow: The game procedure is implemented by utilizing the state pattern as described in [11]. This code pattern lets you dissect your game into multiple states which act independently from each other. Furthermore, you define transitions between states to decide when a state should be changed. Checking if a transition is possible and subsequently performing them is executed by the state machine class. The implementation of the state machine and the state interface was taken from Jason Weimann [11]. The state interface defines a `OnStateEnter`, `OnStateExit` and a `Tick` method. The `OnStateXXX` methods are called respectively when the state is entered or left to setup the state or to clean up references and memory. The `Tick` method is called every `MonoBehaviour.Update()`. In the `Tick` method

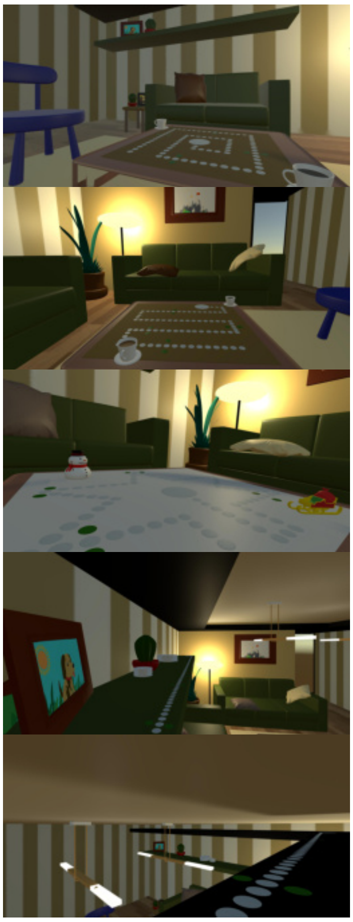

Fig. 10. Different boards positioned in the room
(top to bottom): Starting board, Second board, Third board (Winter wonderland edition), Fourth board (on the shelf), Last Board (underneath the ceiling)

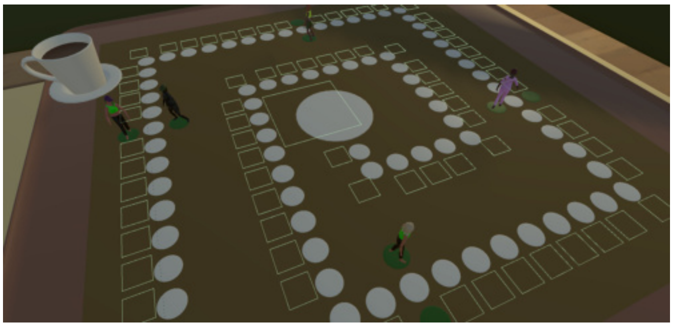

Fig. 11. Board Game setup

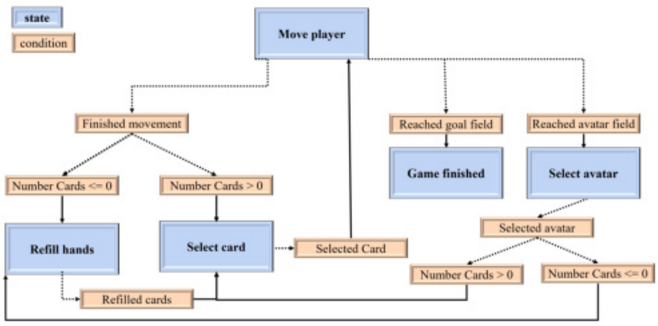

Fig. 12. Graph structure of game states

the actual state behaviour is performed. For example in the `AvatarSelectState` the `tick` method tracked the view direction of the participant to constantly measure where he's looking at. We would strongly suggest using this or a similar pattern when dealing with behaviors with more than a few different states or phases. It saves you from many switch cases and checking countless conditions which is usually error prone and makes the code hard to extend and maintain as new states are added. By encapsulating the states, the code is a lot clearer because there are no edge cases and interference's with other states, and you can easily add more states without the fear of breaking any other states. Another benefit is that you can visualize the game nicely with a graph structure that can be directly implemented without a lot of additional mental effort. Ours can be inspected in the figure 12.

This brings conceptualization and implementation closer and opens up for the possibility to distribute the state implementations across multiple developers as every state is completely independent from all the other ones.

V. Metrics

In order to draw reasonable conclusions from our experiment we decided to measure in three different phases. The participant preferences questionnaire before the experiment started, numerous variables during the experiments trials and a final questionnaire to investigate the decision-making strategies of each participant after the experiment was conducted. The participants preferences were simply stored as they were submitted. During the experiment we recorded the following variables:

- As soon as the participant reached the avatar select state a timer went off, recording how long it took until one of the two proposed avatars were chosen. Later we will refer to this measurement as the participant's *thinking time*.
- In parallel to the thinking time timer there were two other timers, measuring how long the participant looked at each avatar. We went with a naive approach by tracking the forward axis of the head instead of the eyes. Whenever the participant rotated his head towards one avatar we checked if its forward axis intersected with one of the avatars by casting a ray. This technique might not be ideal, but the error of this method was reasonably small. Our empirical tests showed that you had to align you forward axis towards one of the avatars because both of them were in the periphery when looking straight in between both avatars. Due to the nature of the VR googles everything in your periphery was blurred, so you were forced to turn your head for a clear image. In our setting the heads forward axis always intersected with the avatar's bounding box as soon as you were able to perceive a clear image. We tested that on the HTC VIVE Pro and exclusively used those during all experiments.
- After the participant decided upon one avatar, we recorded the level of match of the chosen avatar and the level of match of the rejected one.
- The last variable was a boolean indicating whether the participant picked the same avatar from his last decision again or if he chose the newly proposed avatar. This of cause was only possible after the first decision of each trial as you only received the companion after the first decision. The variable is called *kept the avatar* and is true if you stuck to your companion, otherwise it is false.

After the participant finished all trials, he was asked to complete our post questionnaire where we asked him questions about his motives, thoughts and arguing during the experiment. The aim was to understand the reasoning of the participant when choosing an avatar. What perks of the avatar were how influential on the decision or where there other aspects we haven't anticipated beforehand? More on that in the results section.

A. Study execution

We recruited a total of 13 participants over two days for our pilot study. After some minor initial difficulties, we ran the experiment on two computers in parallel to be more flexible with scheduling appointments with participants. This was especially useful as it was storming and raining outside so we were asked to reschedule appointments which would not have been possible if we did not setup more than one computer. Although the experiment only takes about 30 minutes, it still demands a lot of time executing the pilot study.

VI. Results

Before we dive into the results of this study, we want to present some baselines first. We recruited 13 participants, three of which were from our group. Knowing the research hypothesis surely introduced a bias but we argue that the bias is reasonably small. Only one of the three knew how the avatars were created and still there was some randomness in the creation of each avatar such that it was fairly hard to predict which level of match an avatar had. Therefore, relying on gut feeling usually lead to results that were more consistent with our hypothesis.

One participant decided not to rely on his intentions. In the post questionnaire he revealed that he systematically picked the right avatar in every decision. As we try to investigate the influence of the visual appearance of a virtual avatar his

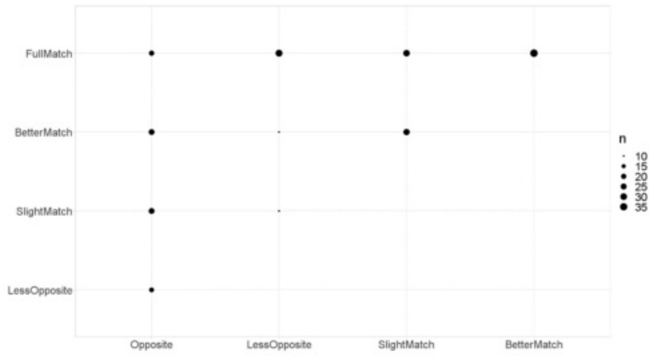

Fig. 13. How often the participants had to decide between different pairs of LoMs.

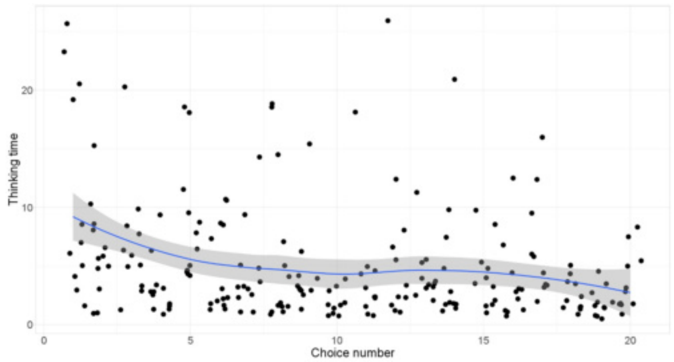

Fig. 14. How long participants took until deciding on an avatar throughout the experiment.

strategy did not allow us to draw any conclusions for our research question, so we excluded him from the following analysis. All upcoming plots and test were made with R [9].

A. Decision space coverage

Although the appearance of all avatars was specific to the participants preferences, the order in which he was presented with the different level of matches (LoM) was controlled. For every trial it was previously determined which LoM was encountered on which avatar field. This was the same for all participants. One thing which was only partially controllable was the quantity of decisions between different combinations of the LoMs. As an example, if the participant selected his optimal avatar on the first decision and stuck with it until the end of the trial, we sampled a lot of data about decisions including the optimal LoM but non about decisions not including the optimal LoM. For example, the participant was never asked to decide upon a slight match and a better match in this trial. Therefore, the combination of different LoMs presented were not equally distributed among all categories as depicted in 13. We can deduce from this plot that decisions between *LessOpposite* and *BetterMatch*, and *LessOpposite* and *SlightMatch* were underrepresented. This can be explained in two different ways. Firstly, the trail setup may have facilitated this effect when for example a *SlightMatch* was placed at the beginning of a trial and the *BetterMatch* towards the end of the trial. This makes it unlikely that the participant has do decide between those two classes as it is quite unlikely that he kept the *SlightMatch* throughout the whole trial. The second argument will get more descriptive after the next results as we found that you are more likely to pick an avatar that is designed for, thereby you'd be more likely to swap an avatar for a higher LoM leaving us with less data for classes in between the extremes.

B. Thinking Times

When we look at the mean thinking times of our participant across the whole experiment, we excluded a total of five decisions which took more than 40 seconds to choose. The mean thinking time before removal was 6.25 seconds and the median 3.3 seconds. After removal the mean was 5.0 seconds and the median 3.27 seconds. As depicted in the graph below we can see that there was no significant change in reaction time across all trials. The correlation coefficient between the thinking time and the running index of decisions is -0.26 therefore, we cannot claim any correlation between those variables. We wanted to check that because it could have been an indication that the participants got bored while progressing through the experiment. They could have started to randomly pick avatars to reach the end of the experiment quicker thus reducing their thinking times. Luckily we found no evidence for that.

C. Which LoMs were chosen?

This question is crucial to answer our research question, whether the mere adjustment of the visual appearance of an avatar to the participants preferences, increases the enjoyment of interaction within a virtual environment. As we hypothesize that this has a positive impact, we should find that the participant is more likely to choose a better LoM over lower LoMs and thereby also have an increasing probability of keeping an avatar if its LoM is higher than the newly proposed one.

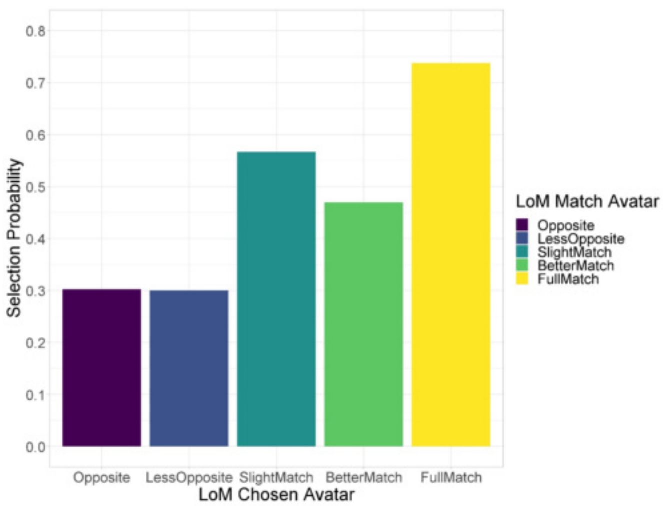

Fig. 15. The probability of choosing a LoM when it was selectable.

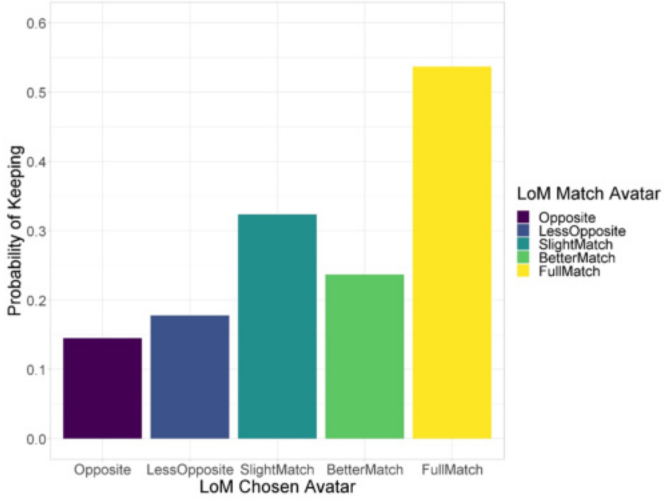

Fig. 16. How probable it was to keep your companion based on its LoM.

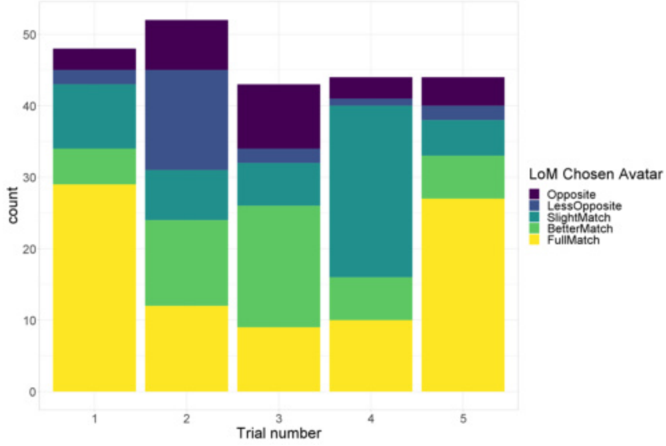

Fig. 17. How often a LoM was chosen for each trial.

In 15 we see the probability of choosing a certain LoM if it was present in the proposed decision. We can clearly see that there was a strong preference for the *FullMatch*. If a *FullMatch* was available, it was chosen in 73.7% of cases. This seems to decrease with the level of match, although it seems like the *SlightMatch* was overall more likable than a *BetterMatch*. This might be due to the minimal changes between these two levels. Looking at the algorithm for the avatar instantiation the only attribute that was changed from preference fitting (*BetterMatch*) to a random option (*SlightMatch*) was the hair style. Since the three hairstyles were short, middle and long and the participant could only choose between the three and not the actual 8 hair models associated with them, thus the probability that the *SlightMatch* is the same (attribute-wise) as the *BetterMatch* is 33%. Hence, the only other distinguishable features are the colors of the attributes. Here, we observed that, while the color for the *BetterMatch* was indeed more similar to the chosen color, it often was just a change in hue sometimes resulting in unpleasant neon-colors. This of course is only speculation but it might be an explanation why the *SlightMatch* was more favourable then the *BetterMatch*.

Figure 16 shows how likely a participant was to stick with an avatar (by selecting the same avatar that they chose in the previous decision) of a certain LoM whenever it was possible. This excludes each first decision of every trial as the proposed avatars were both new so there was none to keep. Aligning with our hypothesis you can clearly see that the participants were more likely to keep an avatar with an increasing LoM. We can again observe that the SlightMatch did better compared to the BetterMatch which can be explained in the same terms as previously.

Another explanation for why the *SlightMatch* seemed to perform better than the *BetterMatch* is that it could have been due to what was available for selection. For trial four we can see a clear domination of the *SlightMatches* which was probably induced by the setup of this trial. Firstly, the participant could decide between *SlightMatch* and *Opposite*, followed by *BetterMatch*, *FullMatch* and finally *LessOpposite*. It seems like the majority picked *SlightMatch* and stuck to it for a while. This shows that the order of presentation has an influence on our results, as certain LoMs seem to be favored depending on whom they are setup against leaving them at a benefit. In trial one and five the first decisions were between *FullMatch/LessOpposite* and *FullMatch/BetterMatch*. On both occasions we can observe a domination of the *FullMatch*s strengthening our hypothesis as is shows that they were usually kept over the most decisions over two dissimilar permutations of LoMs. In the other trials the *FullMatch* was the second to last choice therefore not leaving much potential for keeping. However, *LessOpposite* and *Opposite* consistently did not receive much attention regardless of the permutations. Nevertheless, the setup of the trials has a noteworthy effect on the selection behavior due to the determination of choices. To make up for this one would have to think about making more trials as we only covered five out of 120 possible permutations.

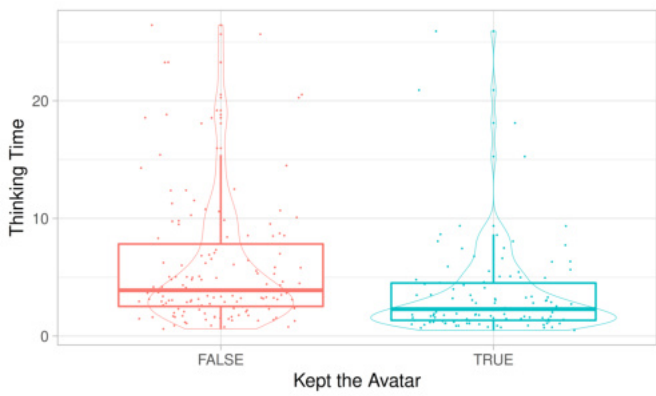

Fig. 18. The thinking time with respect to the decision to keep the current avatar.

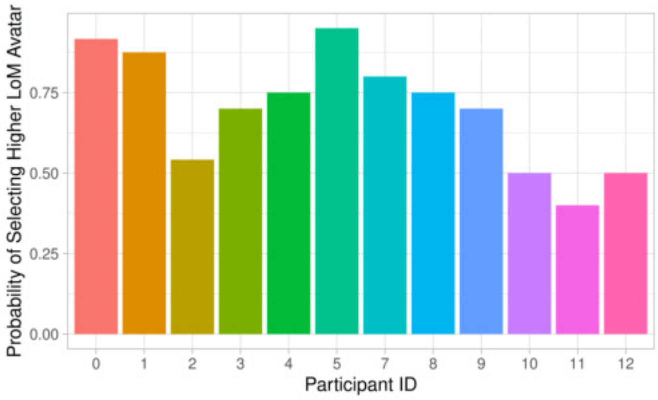

Fig. 19. Percentages of decisions where participants chose the avatar with a better LoM

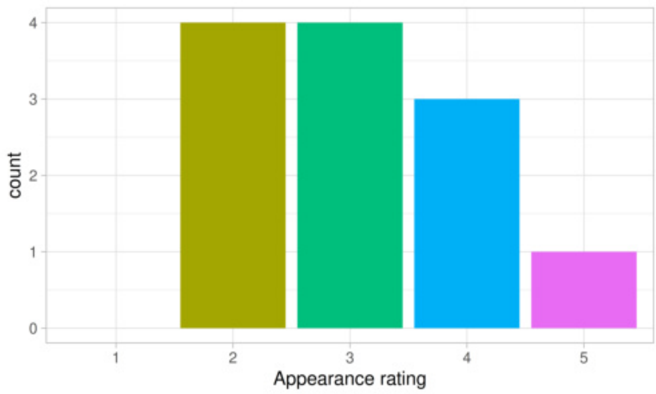

Fig. 20. How participants rated the general appearance of the avatar on a scale from 1 (pleasant) to 5 (scary/creepy).

D. Thinking time with respect to keeping an avatar

18 shows the thinking time - i.e. the time that a participant spent on the selection field until they selected an avatar - with respect to whether they then chose to keep their current avatar or replace it. As it is evident from the plot, participants took more time to select an avatar if they chose to replace the current one. A possible explanation for this is that the participants took more time to think about their decision, maybe compared the avatars. It is also possible that deciding to keep the current avatar is easier than replacing it as the participant might have already ”bonded” with the current avatar. A Welch two sample t-test shows that the average thinking time was significantly greater when the avatar was replaced opposed of keeping him. ($t = 2.2428$, $df = 762.33$, p-value = 0.0126)

E. Selection of better LoM avatars

19 shows for all participants the percentage of decisions where they chose an avatar that better matched their preferences. An example of such a choice would be if a slight match and a full match could be selected, if the full match was chosen, the decision would increase the percentage as the full match fits the preferences better than the slight match. As depicted, most participants tended to chose the avatars that matched their preferences to a higher degree, with the average participant choosing the better avatar around 70 percent of the time (mean 0.6986, range 0.4 – 0.95). This supports our hypothesis that participants are more likely to chose those avatars that better match their preferences. It is also worth noting that the participant that decided to always switch avatars (not shown in figure 19) selected the better avatar in 55 percent of the choices, indicating that the choices were relatively well balanced.

F. Post Questionnaire Results

1) Appearance rating: Participants were asked to rate the general appearance of the avatars on a 5-point Likert scale from 1 (pleasant) to 5 (scary/creepy). We asked this to assess how well the design of the avatar avoided the uncanny valley effect. figure 20 shows that most participants found the avatar neither pleasant nor creepy, or slightly pleasant, however, 3 participants found the appearance slightly creepy and one found the appearance scary/creepy. That participant also noted in the remarks that the legs were wide spread, so this selection might be due to the animations rather than the style of the avatars. Bad ratings could also stem from the errors regarding the weights with skin showing through clothes, mesh deformation and not ideal animations (one participant did note that the legs were spread apart a bit wide, this was due to the animation) so participants might not have thought of the avatars as creepy but rather as bad looking/unpleasant.

Therefore, it seems that our general approach of using a cartoony style is warranted, but more work should be done to improve the look of the avatars by fixing the still remaining weight problems and choosing more suitable animations. The questionnaire could be improved by asking participants to separately rate style and pose/animations and give reasons for their rating.

2) How participants selected avatars: We asked participants to indicate whether their avatar selection was random, based on a gut feeling, or with reasoning. As can be seen in figure 21, eight participants responded that they chose

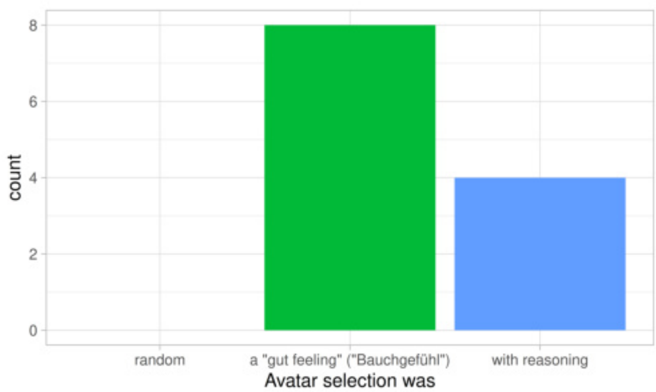

Fig. 21. How participants generally selected the avatars.

avatars based on a gut feeling while four specified that there was reasoning behind their choices. The others specified that their decisions were strongly influenced by hair color, posture, clothing, hair, skin color and hair length. 6 of the 8 participants who indicated that their avatar choice was based on gut feeling also provided text answers when asked about specific aspects that had a strong influence on their decision and all included visual aspects, mostly color, but also hairstyle and clothing. This already shows that the visual appearance is perceived as an important factor when selecting the avatars. However, one participant also explicitly stated that the posture of the avatars influenced their decision, raising an important limitation/pitfall of this study. The avatars had different poses which were selected randomly which apparently influenced the selection. However, this was not accounted for as the original plan had been to use the same pose across all avatars so that only the appearance would differ. A future study should examine the influence of appearance while keeping the same pose for all avatars and maybe investigate the influence of pose.

3) How much different adjustments influenced the participants' avatar selection: For the four aspects clothing style, hairstyle, clothing color and hair color the participants were asked to rate how much they focused on them / how important they were for their selection on a 5-point Likert scale from 1 (not at all) to 5 (a lot/very important). 22 shows the results. As one can see, all aspects apparently had some effect on the selection process, with both clothing style and hairstyle having a median rating of 3 – somewhat important – (ranges 1-4 and 1-5 respectively), and clothing color and hair color having a median rating of 4 (ranges 2-5). Therefore, it seems that color has slightly more influence on the avatar selection than hair or clothing style. This is also supported by the free text answers that participants gave, where 6 out of 10 who answered the question explicitly stated color as one of the strongest influences.

Some possible limitations are that we did not include eye color in this, even though it was also adjusted. However, from testing it ourselves, we noticed that we usually did not pay attention to their eyes and did not notice that avatars had different eye colors, but still, it might have been an influence and should have been checked in the post questionnaire. We also did not ask about the influence of skin color as we did not want to make the participants feel uncomfortable, but some participants mentioned that this did have an influence. Additionally it might have been beneficial to test whether the styles were perceived as intended, e.g. whether the casual look was perceived as casual, in a small pre-study. If participants did not think that the clothes conformed to a certain style that might explain that we did not find that the style was perceived as very important. That the styles might not have been perceived as intended was hinted by a participant noting that the ”pyjamas looked atrocious” but no pyjamas were available as clothes and we suspect that the chic style was meant.

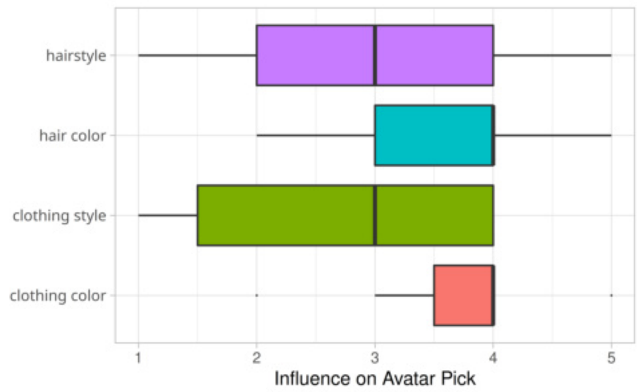

Fig. 22. How much different aspects of the avatar appearance influenced participants' selection on a scale from 1 (not at all) to 5 (a lot / very much).

VII. Resume

In this section we want to reflect our work and discuss the popular question ”If we do it all over again, what would we do differently and what went well?”. As it is important to learn and grow from failures we want to share what we have learned over the past four months.

For better comprehension we want to quickly summarize the schedule and demands of this course. First of all, we were asked to deep dive into the existing human robot interaction literature, eventually finding an unanswered question that we'd like to investigate. Knowing the state of the art around our topic, we should come up with an experimental paradigm in VR which has the potential to answer our novel question. Implementing this paradigm usually involved modeling objects in Blender and programming in Unity. For compatibility reasons we had to work with SteamVR for the VR component within Unity. Depending on previous knowledge about these programs, the basics of them had to be learned first. Once the experiment's implementation was finished, a pilot study had to be executed. This usually involved inviting friends and family to the VR lab to take part in the experiment. We were instructed to then analyse and descriptively plot the previously collected data. Finally, the results were presented to all groups in the form of a talk and at the very end all work should be documented in an article like this. During the literature review and implementation phases we had weekly meetings with the course leaders to show our progress and receive feedback and guidance.

A. Prove concepts and sketch ideas before realizing them

1) Scratching our first idea: Even though you have a weekly supervision it is key to know your goal as precise as possible. If you have a clear idea of where you want to go, the steps on how to get there will be obvious. Spend more time on conceptualizing your ideas and prove that they are functional instead of starting the development immediately. If your idea presupposes some features that you're unsure about, make sure that these features are actually feasible and do not leave it for later.

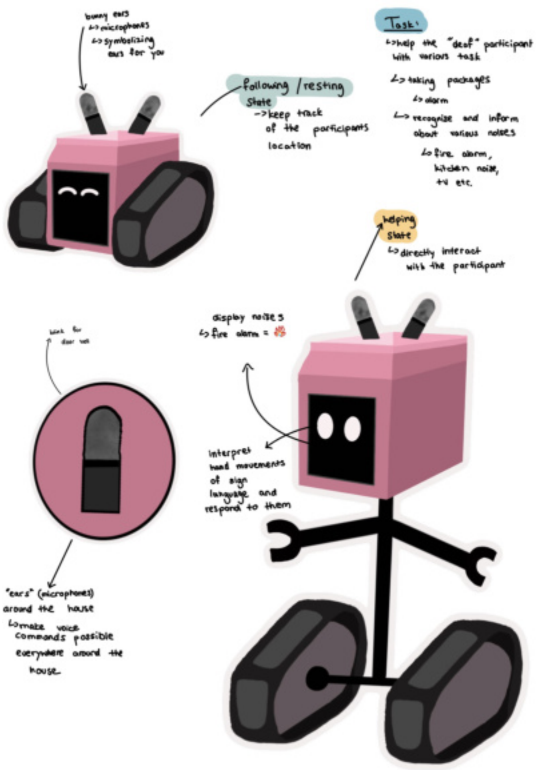

Fig. 23. First rendition of our hearing bot

As an example, we want to give a quick illustration of our first major hardship and analyse what exactly went wrong. At the beginning of the course most of us had more the implementation part of the project in mind. More specifically, we were more interested in creating something interesting than actually constructing a study. At first, we had a lot of meetings to come up with an idea for this course. We often disregarded an idea because we considered it as not flashy or revolutionary enough. When we finally agreed on an idea all of us had interest in and we were really stoked to do the research for it. More specifically our first idea was the following: A robot for people that are hard of hearing or completely deaf. This idea is something that we really consider as important and thus the motivation for it was easily found. We often disregarded an idea because we considered it as not flashy or revolutionary enough. While coming up with the main implementation of this idea we only focused on the design of the robot. Meaning, how it should look and what it can do. This can be seen in one of our first sketches in figure 23.

As you can see we focused on the details of this specific product we wanted to create and completely disregarded the actual study around it. After two weeks of intensive research and finally moving towards building the experiment it dawned us. Not only did we heavily focus on the details, but we also did not think about an actual research topic, let alone a proper hypothesis.

After that we tried to come up with something to research based on our robot but we quickly realized that everything we might want to research would be infeasible. To gather proper data we would have to invite people that are actually hard of hearing or could relate to the problems our robot should solve.

In addition to that none of us could actually speak any sign language nor did we have any idea how to record and analyse the hand movement of the participant in VR. There were a lot of things that were wrong with this idea but all stem from the same problem we had: heading straight into an idea without actually considering its implications or thinking through every detail.

After trying to somewhat modify this experiment we had to admit that it was not feasible for this course. After this harsh realisation we had to come up with a new idea.

2) Scratching and modifying our second idea: The time we spend on this first idea resulted in a lot of time pressure at the later stages of our final experiment. We quickly found our new topic and thought about possible experiment implementations. This time we wanted to be sure to focus on the hypothesis and the research question first before thinking about anything else. So we were determined to find something interesting to fit our research question and not the other way around. Most of them weren't really satisfying until we came to the idea to play "Dogs". "Dogs" is a game similar to "Mensch ärger dich nicht" but you play it in teams. At the beginning of each round you swap one of your cards with your teammate. Because there are cards that are better than others (jokers, special cards and cards with higher numbers are generally better than lower cards) we argued that we could use this card-swapping as a measurement for the likeness of your teammate. In short: If you give one avatar better cards than another you must have liked him more. We were stoked and started implementing the game in Unity without ever proving that this idea is meaningful. We were just glad that we finally found something and to be fair, no one had a better idea.

In the following no one really payed attention to the metric we planned to utilize to test our hypothesis until two weeks before our final presentation was scheduled. By then everything around it was almost done and it was time to come up with card and game compositions which allow to draw conclusions through the swapping behaviour of the participant. We quickly realized that our metric of likeness was hard to analyze. As stated above, our hypothesis was that the participant gives an avatar they like more good cards than an avatar they dislike. However, since they played as a team and one can only succeed if the other team member does equally well as oneself most of the decisions were based on strategic reasoning rather than likeness to the team partner. Additionally it would have been hard to compare results from different participants as all of them would have been in different scenarios depending on how the game progressed based on their decisions. Therefore, much of the work we had done up until this point was rather unusable for the study itself.

Namely, the entire logic behind the game play. This not only includes the movement of the player pieces, the different teams but also the implementation of all of the cards and their

Fig. 24. Original game cards and board along with player pieces to play with

logic. Since the game should be stripped from all possible strategy influence all of the game rules (reaching the goal field or kicking off a piece from an opponent) and their implementation had to be removed as well. To that point the game and its environment was essentially a completely working VR board game. The original setup can be seen in figure 24. Again here you can see our biggest mistake we had throughout the course: working on something because we thought it's "cool" rather than sketching out the idea and it's implications in advance.

However, getting rid of everything would have been not only a shame but also nonsensical because there still was a lot of material we could work with. Therefore, we improvised and focused on the real heart of the study and the hypothesis while also keeping as much material as reasonable from our first try. And from this the idea of choosing a team partner for an over dimensional board game was born. This way we were able to keep the game movement and the board itself as well as the room and the logic of the cards. However, if you compare the version below with the one on top you can see that we still had to change cards which were "special" (i.e. had more logic to them like swapping players) because we only used the cards as the way to move across the field.

B. Working in a large group

Working in large groups is always difficult. Our group had five members with various expertise. Some of us were already experienced with Unity but only one of us had prior knowledge of Blender. The remaining group members haven't had much prior experience with these programs but were motivated to learn them. Right from the beginning, we decided to divide our group into two expert groups. While two of us dived deep into unity, the others dealt with Blender and designed the base character and the clothes. By that we efficiently divided the overall workload.

1) Agile workflow:

2) Collaboration with GitHub and git-lfs: When collaborating in a group and working on the same project and files, version control is a must have. Even when working alone, making backups regularly is strongly recommended. For this purpose we used git for version control and GitHub for collaboration and backup. Working with git usually comes with pain in the form of merge conflicts, corrupted or suddenly broken files and other unforeseen errors. Therefore, we want to give some insights into our project design, problems and what you might want to watch out for.

The setup of a projects repository is usually pretty straightforward. One team member creates the repository in GitHub and invites the remaining members as contributors. After that you have to make a decision or rather a trade-off on how you want to structure your collaboration. Either all of you are pushing directly to the main branch of the main repository or each team member creates a fork of the main repository to obtain their own online repository. Alternatively to creating forks you could also create an own branch for every team member in the main repository. But why are those extra repositories useful and what's the trade-off?

In the first scenario, all team members directly push their local changes to the main version of your program, located in the main branch. Like this every team member is at the risk of breaking the main version for everyone with just one command: `git push`. While this method might seem uncomplicated and fast at the first glance, it really won't if someone breaks something and pushes it. You only have this one main version which also serves as some kind of an online backup, meaning if you mess up with git locally, (which is almost guaranteed to happen at least once) and you push it to the main repository, everyone has to deal with it and unfortunately, it is hard to get rid of it once it's online. Usually, this implies hours of googling error messages and executing commands from StackOverflow that you do not even understand and mostly are not helping either. Trying to constantly be aware of what you are pushing is desirable but sometimes git prompts you with errors out of nowhere and you don't have a clue how this error came to be. Those types of errors are nearly impossible to foresee as a beginner, but we made some and figured out their cause so we will share them in the do's and dont's section with you. To conclude, this strategy allows you to make fast changes but with a high risk of breaking things. Doing backups regularly is crucial for this strategy.

The second strategy reduces the risk of breaking the main program by adding additional steps until your changes end up in the main branch of the original repository. Ideally every team member creates his own fork. What is a fork? If you press the fork button on the web page of a repository that you do not possess, it will make a functional copy (or fork) of that repository to your account. Loosely spoken this repository now exists twice, under two different accounts - but one of them is now possessed by you so you can change whatever you want. But a fork is not just a simple copy. The original repository is linked with all its forks in a special way. If the original repository was updated, every fork can fetch and integrate those changes with a button press to their fork. If a fork was modified it can't simply update the original repository. For this the fork has to create a pull request to the original repository which can be understood as asking the owner of the original repository if he wants to integrate your changes while he can review all your changes before accepting or denying. In our scenario you were invited to the original repository therefore you could hypothetically review and accept the pull request which you created from your own fork. But that would contradict the idea of pull requests. The idea is that someone else ("a second pair of eyes") reviews your changes and can potentially catch bugs before your changes were deployed. Furthermore you can discuss about the changes and ensure good code quality and more. In theory you could do all of this without forks by just creating an own branch for every team member and create pull requests to ask for integration of the members branch into the main branch. This can work just as good but you have to be aware of how branches work and what you could mess up on the way. Since you are all on the same repository you could easily checkout the main branch and make direct changes. If you do this you are back at scenario one. By creating forks you can pretend like you are in scenario one but you only mess up for your fork and not for everyone. And even if you broke your fork and your pull request went through, subsequently breaking the main program as well. The other team members are still not affected until they fetch your changes to their fork. Like that it takes steps from all team members until everything received your broken version, leaving a high chance that someone did not fetched the broken version and thereby serving as a backup. With this strategy you implicitly have a lot of backups without actually doing so much different. The trade-off here is definitely your time efficiency. You have to invest time into reviewing changes and building the infrastructure but on the other hand you drastically reduce the risk of breaking everything and you have the possibility to ensure good code and design quality which could as well help to save time in the long run.

To conclude, both strategies have their advantages and disadvantages. Which one you want to choose is up to your project and team. The second strategy is definitely the most robust one but it will also slow down the development. The idea of pull requests is great but unfortunately you have to wait until your changes got reviewed which might take some days until the reviewer finds the time to do so. Therefore this strategy benefits the more members are working simultaneous on the code. Thus, if your project demands a lot of programming done from multiple team members, you should go with the second strategy. If only one or two members are programming or there is not too much demand for programming the overhead of strategy two is too much. In this case be responsible with what you are pushing and make backups!

In our group, four out of five members were involved with programming at some point. Initially we went with the second, save strategy which was indeed a blessing as we messed up more than once but luckily were always able to restore our project from backups and other forks. Towards the end we had severe time problems so three of us started to work directly on the main branch while constantly being aware of what the others were changing. This allowed us to make fast progress with a conceivable risk as we knew at all time what everyone was doing.

a) Dos and Dont's:

1) Do not be afraid of git. Git is a version control tool for different versions of your files, it is not a backup tool for your project where you create checkpoints every now and then. Therefore use git to modulate your changes, meaning that you commit files together which belong together and give the commits descriptive names. This allows you to easily revert commits which introduced bugs without having to redo the commit because you included other components into the commits which have nothing to do with the files in question. Also many nerve wracking problems start by committing a lot of files without exactly knowing what you've added. More on that later. The take away of this point is to use git frequently and be aware of your changes than using git rarely and committing many files at once.
2) Do not use git-lfs when you can avoid it. GitHub will warn you whenever you try to push files larger than 50 MB but if a file exceeds 100 MB GitHub will reject your push. If you want to push anyway you have to use git-lfs (lfs: large file storage). This tool is super easy to install and configure but a nightmare if you want to remove from your repository again. But why would you want to remove it? The problem which is not documented well enough is that you have a quota that you can exceed. You have a limited amount of up and downloads of your git-lfs files for free. If you exceed it you cannot push nor pull anymore and that's certainly not a good thing. The ridiculous thing with git-lfs on GitHub is that the quota is for your repository including all forks and not per user. This means every time you clone the project or if any member pushes or pulls a git-lfs object the quota will go down. With a large group with limited time and a lot of work to do you will exceed that quota in no time and suddenly your project will be frozen. Therefore, don't use git-lfs if you don't have to. If an avatar is too large, try reducing its fave/vertices count or if textures are too large consider using different ones with lower resolution. We exceeded that quota and since it is seemingly impossible to remove git-lfs again the only solution was to create a new repository and disregard our entire git history.
3) Do not add and commit files that are larger than 100 MB or are confidential. Git will not complain but as soon as you try to commit GitHub will refuse your push because of that file. Furthermore, if you committed a large file, deleting it will not suffice. Git must forget that this file

ever existed so you must rewrite gits history by undoing the commit that added this file and every subsequent commit which modified it. Rewriting this history is time consuming therefore it is sometimes faster to delete the `.git` folder of your project, clone the repository again and copy the old project to the newly cloned one while choosing `replace files`. This will reset your git history to the point it was before you did your changes and by selecting `replace files` you are reapplying all of your changes again. Now you can commit everything like described in 1. & 2. but be aware that if someone else pushed something which you did not include so far this approach will override their changes if you changed the same files (... you chose `replace`).

4) Useful references
   - Introduction to git
   - About large files with GitHub and how to rewrite your git history

VIII. Outlook

While developing a project within a fixed time schedule it is inevitable to make compromises along the way. Some of them elicit more consequences than others. As an outlook we want to quickly go over them and serve guidance on how this study could be carried out successfully.

First of all, the naive eye tracking approach worked well in our opinion but it should have recorded already much earlier, not just during the selection phase. This would lead to much more accurate results because is was sometimes possible to see the other avatars before actually being on the avatar field thus distorting our results. When we have a look at our clothing again you can spot some gaps in the clothes every now and then while the avatars are moving. Presumably this was caused by sub-optimal weight assignments from parts of the clothes to the avatars rig. These weights define how strong a parts of the cloth are attached to certain body parts of the avatar and subsequently how strong they stick to these body parts while the body is moving. When they are slightly out of balance it can happen that the cloth is drawn into the body which we then perceive as a hole. Correcting that is tedious but it would factor out the gaps as possible distractor. However, our post questionnaire did not revealed them as a major distractor. Still, doubts can be addressed. Furthermore, there are two aspects which could further increase the accuracy of this research. First of all a pre-study should be conducted to confirm that the created clothes are indeed perceived as their labeled category and secondly more cloth pieces and accessories could be added to adapt to the participants preferences even more precisely. Alongside new pieces one could also think about varying body proportions, shape and height. Adding a male avatar might be beneficial too, but also cause a lot of additional work as all clothes have to be designed for both sexes. Having only a female character might add a little bias to the experiment since one might prefer to interact with a male or female character. In addition to that, we are eager to see if and what difference the additional character might cause in our results. Besides knowing what a participant likes might not be insightful enough. We used this superficial information about the participant to also construct avatars which we supposed he would not like. But certainly that is a bold assumption. For example someone might like more than one clothing style or eventually adore a color which is perceivable distant from your favourite color as well. Nevertheless, our results showed clear evidence for our hypothesis although they might become even more explicit if are able to reliably create dislikable avatars. Finally, we noticed something really interesting. During the experiment we randomly altered animations of the avatars to make them feel more alive and active. Like this they sometimes waved or gesticulated a bit. Interestingly the post questionnaire revealed that this had an influence on their decision making. Therefore it would be exciting investigate this further and maybe use those animations in a controlled manner. Interactions with avatars face a large challenge as they often lead to unpleasing experiences if the avatars do not react naturally all the time. Such passive motions like waving which do not necessarily demand a complex interaction between the human and robot could help making virtual environments more fun and engaging. We would love so see some more research into this direction.

REFERENCES

[1] PIXXO 3D. Blender — sewing cloth in blender 2.93 — beginner tutorial. https://youtu.be/p4Xwo3ygaCM, 2021.

[2] Global Workplace Analytics. Advantages of agile work strategies for companies. 2021.

[3] Jonas Mahlmann Anna Kalinowsky. Faq: Was ist das metaverse? https://www.heise.de/ratgeber/FAQ-Was-ist-das-Metaverse-6331112.html, 2022.

[4] Debaleena Chattopadhyay and Karl F. MacDorman. Familiar faces rendered strange: Why inconsistent realism drives characters into the uncanny valley. *Journal of Vision*, 16(11):7–7, 2016.

[5] Blender Foundation. Blender - a 3d modelling and rendering package, 2021.

[6] Christian Felix Deuchler Jonas Wölfel Matthias Hepperle, Daniel Purps. Aspects of visual avatar appearance: self-representation, display type, and uncanny valley. 2021.

[7] R. Matthew Montoya, Robert S. Horton, and Jeffrey Kirchner. Is actual similarity necessary for attraction? a meta-analysis of actual and perceived similarity.

[8] Alanenpää Madelene Castellano Ginevra Perugia Giulia, Paetzel-Prüsmann Maike. I can see it in your eyes: Gaze as an implicit cue of uncanniness and task performance in repeated interactions with robots. 2021.

[9] R Core Team. R: A language and environment for statistical computing, 2021.

[10] in Colour Design (Second Edition) T.M. Goodman. 2012.

[11] Jason Weimann. Unity bots with state machines - extensible state machine / fsm. https://youtu.be/V75hgcsCGOM, 2020.