Aalborg University Copenhagen, Medialogy

P10

- Master Thesis -

Blink-induced change blindness as a means of redirection in Virtual Reality

Authors:

Theo Khumsan

Jonas Valvik

Mikkel Rosholm

May 24, 2023

Aalborg University Copenhagen

Aalborg University Copenhagen
A.C. Meyers Vænge 15, 2450
Copenhagen SV, Denmark

Secretary: Christine Pedersen
Phone: 93562054
cpe@create.aau.dk

Semester: 10th Semester

Title: Blink-induced change blindness as a means of redirection in Virtual Reality

Project period: Spring 2023

Semester theme: Master Thesis

Supervisor(s): Niels Christian Nilsson

Members:

Theo Khumsan

Mikkel Rosholm

Jonas Valvik

Abstract:

This report explores a new method in redirected walking (RDW) by exploiting blink-induced change blindness to hide architectural manipulations. Two within-subject designed experiments were conducted to test this method.

Firstly, thresholds were investigated and analysed using a two-alternative forced method, indicating that front facing walls can be moved up to 0.38m towards, or 0.90m away from users as they blink, without the vast majority noticing. Secondly, to demonstrate the effectiveness of said thresholds, an experiment comparing blink-induced change blindness to existing methods, were developed. Results of a Wilcoxon signed rank test showed no significant change between blink- and turn-induced change blindness for RDW, providing a foundation for future use cases of this method in VR applications.

Ultimately, the findings are valuable as they provide thresholds for detecting front facing manipulations during blinks, while also demonstrating that the method allows users to explore a larger virtual space than physically possible, with minimal noticeability. This technique has potential as a technique within RDW, and offers a unique approach to architectural manipulation.

Table of Contents

1 Introduction 1

2 Related Work 2

2.1 Redirected Walking 5

2.1.1 Subtle perspective manipulation 6

2.1.2 Subtle architectural manipulation 8

2.1.2.1 Change blindness 9

2.1.2.2 Change detection 9

2.1.3 Platforms for RDW 11

2.1.4 Challenges with RDW 11

2.2 Physiology and Psychology of Blinking 11

2.2.1 Eye tracking 13

2.3 State of the art 15

2.3.1 Induced blinking 15

2.3.2 Blink-induced redirection 16

3 Methods 17

4 Experiment 1 (E1): Estimation of detection thresholds for blink-induced change blindness 18

4.1 Methods - E1 19

4.2 Design - E1 21

4.3 Implementation - E1 22

4.4 Participants - E1 23

4.5 Equipment - E1 24

4.6 Results - E1 24

4.6.1 Qualitative results 26

4.7 Discussion - E1 27

5 Experiment 2 (E2): Utilizing blink-induced change blindness for RDW 28

5.1 Method - E2 28

5.2 Design - E2 31

5.3 Implementation - E2 34

5.4 Participants - E2 37

5.5 Equipment - E2 37
5.6 Procedure - E2 38
5.7 Results - E2 38
5.7.1 Post exposure questions from turn- and blink-induced change blindness 38
5.7.2 Click data 40
5.7.3 SSQ 40
5.7.4 Qualitative feedback 41
5.8 Discussion - E2 42

6 General Discussion 43

7 Conclusion 45

8 Bibliography 46

9 List of Figures 52

10 List of Tables 54

A Experiment 1 Questionnaire 55

B E2 Questionnaire 57

C Experiment 2 results 64

D Task screens from Experiment 2 69

1 Introduction

As Virtual Reality (VR) gradually becomes more accessible to the average consumer, and more people try the technology in their homes, there is a growing need for better VR navigation options than what is currently available. The latest research in VR navigation suggest that real walking is a more easy and natural form of locomotion, when compared to other options such as flying, teleportation or walking-in-place [1] [2]. Moreover, Slater et. al. [3] found that subjective presence from real walking was higher when compared with the abovementioned alternatives, flying, teleportation or walking-in-place. Their experiment solidified the results from an earlier experiment in 1995 [4], stating that subjective presence is highly correlated with users' degree of association with their virtual body. Finding the optimal form of navigation in virtual environments, where the user is supposed to travel by foot, is therefore a complex discussion that involves many factors such as user preferences, user comfort, sensation of presence, real room size, target user experience, etc. When investigating novel approaches for locomotion the question arises: how should the user walk around in the virtual room while remaining safe in the real world, and avoiding any unwanted side effects such as nausea or headaches?

As a possible means, Redirected Walking (RDW) has been proposed. RDW is a VR locomotion manipulation technique developed in 2001 by Razzaque et. al. [5]. The technique is able to create the illusion of walking in a significantly larger virtual area than the real world permits. Within the concept of RDW there are two branches of manipulations, namely perspective manipulation and architectural manipulation [6]. Perspective manipulation covers the subtle or overt techniques that alter the users direct movement or the users perception of movement in terms of speed and rotation in the virtual environment, i.e. manipulating the mapping between real and virtual movements. Architectural manipulation covers the techniques where the environement itself is manipulated to produce self-overlapping virtual architecture, either occurring overtly in plain sight for the user or subtly outside the field of view of the user to remain unobserved and possibly taking advantage of the concept, change blindness. Change blindness is a cognitive psychology concept related to how we perceive and remember the world around us and the objects within it [7]. When the two are combined, they can create the illusion of moving around in a space that is larger than the real world space. Change blindness and RDW both demonstrate how our perception of the world can be manipulated and influenced in unexpected ways. RDW demonstrates how changes in our environment can alter our sense of movement and space, whereas change blindness demonstrates the limitations of our attention and perception, as well as how we can fail to identify or remember important details in our surroundings.

While the amount of research in RDW is growing at a rapid rate, the focus appears to be on subtle perspective manipulations, sometimes with the help of overt perspective manipulations. Leaning on the concept of change blindness, this report demonstrates whether architectural manipulations as translations or rotations of doors, objects or walls within the users field of view, can go unnoticed during the momentary blindness that takes place when the use is blinking.

To what degree is it possible to redirect a users movement in a virtual environment utilizing change blindness during blinks keeping said user unaware of the redirection

The report is based on two seperate experiments in VR that investigate the possibility questioned above, where both compare the results to the status quo. The first experiment aims to establish detection thresholds for the degree of translation on the Z-axis possible for objects inside the users field of view compared to identical translations outside the user field of view. The second experiment employs the thresholds found in the first experiment to implement architectural manipulations during eye-blinks in a real scenario based on and comparing to an existing study by Suma et. al. [7] similarly leveraging change blindness, albeit not in the users field of view.

2 Related Work

The following subsections will cover theory and practice of VR, RDW and the physiology and psychology of blinking, to build the theoretical foundation on which the project is based.

The backbone of this project, which enables exploration of the aforementioned components, is VR. According to Zheng et. al. [8] VR is a human-computer interface that enables users to interact with computer-generated environments. Its main objective is to provide the feeling of "being there", and to achieve this feeling Zhang et. al. [8] argues that the muscle and perceptual system of the user must be linked to what is perceived in the virtual environment. In terms of hardware, this requires three components; sensors, effectors and reality simulators. These components work together to create said feeling of "being there", which we are able to utilize to examine the concept of RDW. Briefly put, VR involves a user wearing a sensor, often a head mounted display to track the user's head movements, with an effector, often a screen visualizing something to the user, and a reality simulator linking these two, whereby the user sees an image-stream that adapts to their head movements be it translated or rotated, thus making them feel present in the virtual environment setup.

To be present or to experience presence in VR is depicted as the psychological sense of "being there", in the virtual world [4], with this sense being a function of the match between a users sensory data in terms of proprioception and internal representation, i.e. the degree of association between the users real body and the users virtual body.

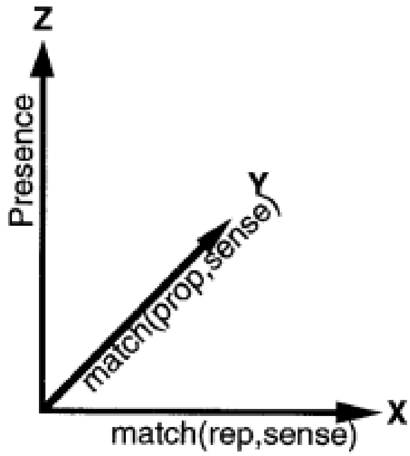


Figure 1: prop = proprioception; rep = internal representation; sense = sensory data [4]

Slater et. al. [9], describe the concept as an illusion composed of two dimensions, place illusion and plausibility illusion. The place illusion depict users feeling of being in the place where the virtual experience takes place. The plausibility illusion depict the users belief in whether the events and interactions in the virtual world actually take place. In both of the dimensions, the user is initially well aware that they they are in a virtual world, and the degree of presence then describes how well the users senses are convinced that the illusion is real. The more elements inside the virtual environment that create a realistic and immersive experience for the user, the more said user will feel present in the virtual world, and as a result, detach themselves from the real world [10] and suspend their disbelief. And it all comes back to the brain - how does the human brain perceive and process the virtual, and can we turn the functionality of perception into our benefit by exploiting its quirks. Technological research in disciplines like psychology, neurology, and human-computer interaction has made it possible for researchers to better pinpoint the elements involved with immersion and presence in virtual environments that relate to the users perception in order to find areas of improvement and exploitation relating to theses disciplines. The human brain is an advanced organism, that processes variables such as light, shapes, positions, sizes, etc. as cues to form the environment we see through our eyes and give it meaning [11]. These cues might be assisted by patterns from memories, words, written or spoken, or even actions that give meaning to a certain environment to solve the puzzle of what we see and how it makes sense [11]. In VR it is possible to play with these variables to make the user perceive the environment as intended by the developer.

The concept of perception encompasses how a person experience, interpret and respond to the sensory information surrounding them [12]. To make the user feel present in the virtual world, the world and the objects within it must, to some degree, faithfully imitate the relevant sensory inputs that affect the users perception [13]. That means, in the case of realism, do the textures and colors of objects appear realistic, and does the movement of the virtual body correspond to the movement of the users real body. These questions incorporate both subjective- and objective aspects. What does the user feel when subjected to visuals, audio, touch, etc. and what is the quality of the technology [14], and is it for instance, able to deliver graphics on a realistic level with a smooth frame rate.

RDW takes advantage of human perception and the bias it has when processing, predicting and adapting to the world. It is enabled as subtle or overt changes to specific objects or interactions, where the brain becomes convinced that the change is real as it does not appear otherwise. The visual and vestibular systems do not store all incoming information from our proximate environment, meaning it does not hold precise object placements or accurately corresponding movements. It has a rough and seemingly accurate idea, but not perfect [11]. RDW is promising because it enables the experience of 1:1 locomotion, meaning that wherever the user navigates or naturally moves to in the real world, the same translation and rotation will be applied in the virtual world, mapping the locomotion accurately to the real world. The benefits from this is presence-enhancing and nausea-decreasing [15] because either the users' walking direction is subtly manipulated in order to redirect the user without them knowing, or the architecture of the virtual world is changed without them seeing.

2.1 Redirected Walking

Coined by Razzaque et. al. [5], RDW is an umbrella term covering the techniques used in VR to manipulate a user's path of locomotion. Typically, it takes advantage of how a user perceives their path of travel or how well they remember the details in the scenes. RDW makes it possible to redirect the user with tolerable gains or changes, wherein the user is not able to distinguish the inconsistencies between the real path and the virtual path [10]. The figure shown below shows the usage of gains in the form of translation and curvature to reroute the user around a bigger virtual space than physically possible.

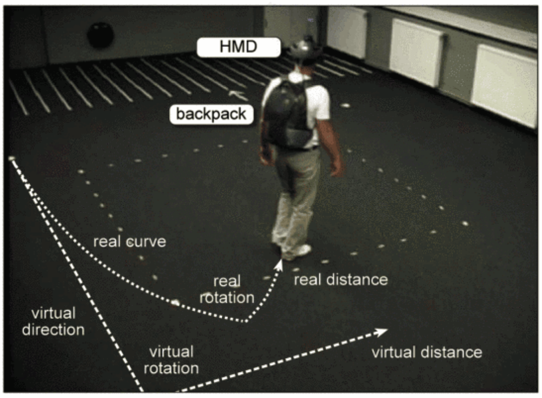


Figure 2: A visualization of applied translation gain and rotation gain to change their walking path. [15]

Within RDW, the manipulation techniques covered by the concept are distinguished mainly by whether they are subtle or overt [6]. The objective of subtle manipulations is to stay hidden from the user, whereas the objective of overt manipulations is not to stay hidden, but instead to enlarge the virtual exploration area or to keep the user out of harms way. Both subtle and overt techniques encompass two manipulation subcategories; perspective and architectural. Between all the types of manipulation, the goal of keeping the user withing the physical boundaries of the real world remains the same. Perspective manipulation attempts to achieve this goal by manipulating the user's perception of self-motion and works by subtly changing the visual presentation of the virtual world to create the intended illusion [10]. Architectural manipulation on the other hand involves altering the layout or structure of the virtual environment to create the illusion of a larger virtual environment than possible in the current real environment and thereby achieve its goal [10]. Seeing as this project aims to implement architecture manipulation while blinking and keeping the changes unnoticed, the report will mostly cover subtle manipulations. Examples of common overt perspective manipulations could be the 2:1

turn and the freeze-and-turn methods [10]. The former has the user actively rotating themselves by 180 degrees while rotating the virtual world by 360 degrees to let the user move further along a path in the virtual world. The latter has the user perform the same rotation in the real world, but freezes the virtual world instead of rotating it [6]. Overt redirections are widely used for when a user encounters a wall or a blocked path in the real world. A prompt of some kind is then initiated to let the user know that they should rotate in the real world to keep moving inside the virtual world. However the technique has the effect of decreasing the feeling of presence when it noticeably redirects the user [2].

2.1.1 Subtle perspective manipulation

There are four widely used subtle perspective manipulation techniques, rotation gain, translation gain, curvature gain and bending gain. They all implement the same core idea, which is to guide the user away from boundaries or obstacles by means of gains. These gains all rely on the limitations of human perception, and take advantage of how well humans determine whether they move by themselves (self-motion) or whether their avatar or the objects around them are moving (external motion) [5]. When the user perceives the redirected movement or external motion as self-motion then the manipulation has been successfully implemented [5].

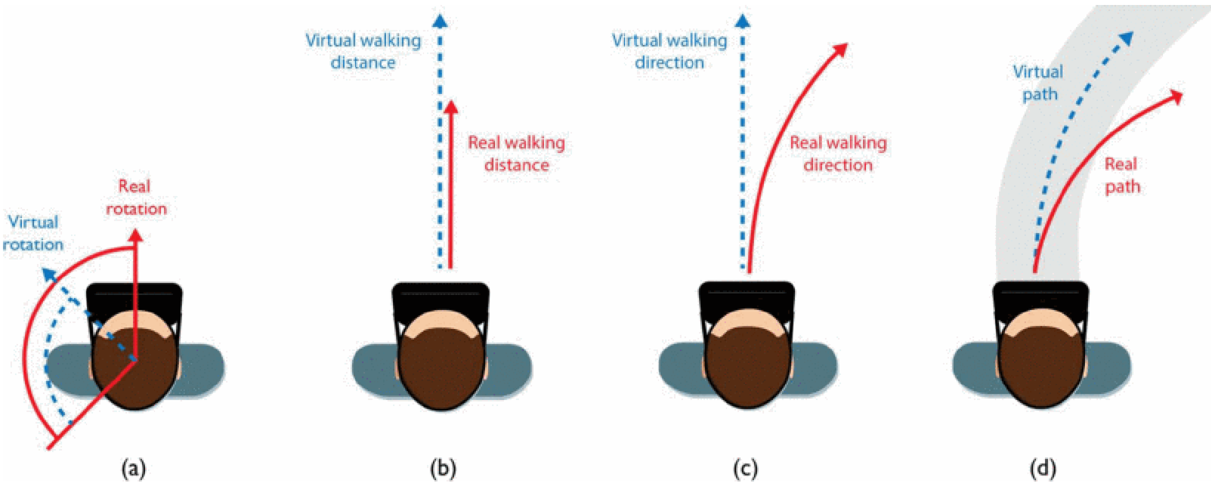


Figure 3: Visualizing the four manipulation techniques [10]. The user's path in the actual world is indicated by the red arrows, while their path in the virtual environment is indicated by the blue dotted arrows. (a) shows rotation gain. (b) shows translation gain. (c) shows curvature gain. (d) shows bending gain.

Rotation gain involves subtly rotating the virtual environment in response to the user's real world movements thereby up-scaling or down-scaling their real world rotation in order to give the appearance of a larger space. For instance, the virtual world may rotate slightly to the left if the user rotates their head to the right, giving the impression that they have turned more than they have. The virtual environment may also be subtly compressed or stretched as the user advances to give the impression that the distances are getting longer or shorter [6]. If a rotational gain of 50% are being implemented into the VR application, then the user has to turn their head 180 degrees in the real environment in order to rotate 90 degrees in the VR environment. While a rotational gain of 200% means that the user has to turn 45 degrees in the real environment, for them to turn 90 degrees in VR. Steinke et al. estimated a threshold of rotation gain of 49% more and 20% less[16]

Translation gain gives the impression that a space is bigger or smaller by slightly shifting the virtual environment in response to the user's motions. The virtual environment, for instance, may be slowly shifted to the side as the user advances, giving the impression that they are traveling along a longer road than what is actually possible in the real world [6]. Both up-scaling and down-scaling translation gain can be applied to the users movement E.g. A 300% translation boost will result in the user moving three times as far in the virtual world. Both a up-scale threshold and an down-scale threshold have been developed as a consequence of research into the detection thresholds of translation gain. The down-scale barrier is 14%, whereas the up-scale threshold is 26%, according to the findings of Steinicke et al. Accordingly, a 3m x 3m tracking space can be converted to a 3.78m x 3.78m space using a translation gain of 26% [16].

Curvature gain is applied as a slight continual rotation when the user moves forward. With the aid of this method, users may move in circles while physically following a virtual path endlessly [6]. Depending on different factors, the threshold of curvature gain can vary. A study of Neth et al. suggested that walking faster can increase the sensitivity of the curvature gain.[17]

Similar to curvature gain, bending gain change how the direction of movement in the real world is leading the user to steer left or right as they go along a curved virtual route. Both curvature and bending gains depend on correct forecasts drawn from the user's recent motions or prior information on the user's intended path [6]. A study by Langbehn et al. on bending gain imply that the virtual curvature can be bent up to 4.35 times its radius in the real world. E.g. a radius of 3m in the real world can be mapped as virtual curvature of 13.05m [18].

In a review of redirection methods by Li et. al. they list several other subtle redirection methods under the sub-category; view-based, describing the following manipulations; reactive, predictive, resetting, learning-based and saccadic suppression [19]. Reactive methods manipulate the users walking path away from the real world boundaries without considering the users future next step. The algorithms steer to center (S2C) and steer to orbit (S2O) are reactive redirection techniques. Respectively they steer the user towards the center of the real world or around the center the real world [5]. Predictive redirection is a more recent idea, where the redirection technique considers which elements the user will encounter in the short future and where they are moving towards based on the current walking path, current facing direction and current environment layout. The predictions then make it possible to steer the user differently, depending on the current trajectory. Resetting incorporates the 2:1-Turn and freeze-turn techniques explained earlier. Learning based manipulations are based on learning algorithms such as reinforcement learning to optimize existing techniques as S2O by dynamically calculating the best target to steer towards and around in order to avoid obstacles (S2OT) [20]. Saccadic suppression takes a different approach than the other techniques and rely on manipulation of the view while the user is temporarily blinded during a saccade, the rapid movement from one gaze fixation point to another. Bolte et. al. [21] explored this approach in terms of user reorientation and repositioning during saccades and found that imperceptible manipulation was entirely possible albeit not significantly large manipulations.

2.1.2 Subtle architectural manipulation

The techniques often explored within subtle architecture manipulations are impossible places [22] and change blindness [7]. Both impossible spaces and change blindness are well-established techniques to subtly reroute walking as an alternative to gains or in combination with gains.

Developing virtual surroundings that defy physics or that physically do not make sense in the real world, such as rooms that overlap is known as impossible places. Using this technique, the user cannot rely on their preconceptions of how physical environments are mapped in the real world, which can lead to disorientation thus making it harder for them to keep track of their walking path [6].
Change blindness is another approach to hiding architectural manipulation [7] entailing

small adjustments to objects or elements in the virtual environment while the user's attention is diverted elsewhere in order to influence how they perceive the virtual environment. In the context of RDW, this means that developers can discreetly alter the virtual environment to change the user's direction without the user being aware of the change. Developers can produce more immersive and compelling virtual worlds that can more successfully control the user's movement without interfering with their sense of presence by fusing impossible spaces, change blindness, and RDW. This may result in a more seamless and enjoyable experience for the user and is simultaneously one of the project goals; making a seamless RDW experience using blink induced change blindness and impossible spaces.

2.1.2.1 Change blindness

"To see or not to see" [23], that is the basis of Change Blindness. The illusion takes advantage of the fact that people never really form a complete and detailed map of the world in their field of view and in extension the world surrounding them. It covers the inability to detect significant changes to visual scenes, seeing as relatively little information about the details of a humans' visual input stream is stored in the working memory. More specifically, humans have a hard time detecting these changes because the selective attention allocates cognitive resources to the elements that appear most salient in the view and because the visual system does not process what the view as a continuous stream of information, but rather as a representation of the information in the form of snapshots of the views. These snapshots are rapidly changing and only about four to seven chunks of informations are really noticed and stored in the working memory [7]. The complexity or distinctiveness of the elements in the scene play a big part in noticing them. Changes in rotations and colors of objects are often the least noticed, where placements are most likely to be noticed. As the visual stream of information keeps flowing, the older snapshots decay and are eventually forgotten to make room for new information.

2.1.2.2 Change detection

Whether the user detects a change comes down to various factors such as experience, attention and noticability, cognitive load, mental maps, etc. As mentioned above the concept of attention plays a vital role where it is essential to grab the users attention and redirect it away from the change happening in or to the environment. Attention is a broad neuroscientific concept that covers how the brain processes and filters through sensory information to focus on what is important or relevant. When a user explores a virtual environment, sensory information in the form of sounds, visuals and sometimes touch all bombard the brain as signals that will be processed and filtered to prioritize the most important signals and to filter out the less important signals [24]. A signals priority is rated by the amount of distinct sensory information within it, i.e. saliency of the stimuli. Examples of salient attention-grabbing signals include but is not limited to visual cues like flashing lights, interactive elements that call to action or audio that is emitted from specific locations. Knowledge of attention and what captures attention can help tremendously in the development of an RDW experience to hide changes to a scene.

As mentioned, the user is at all times bombarded with sensory information and this means that the brain is always working. Cognitive load describes the amount of mental work needed to perform a task or to process information [25]. Understanding the factors that contribute to cognitive load and how they can be taken advantage of is important for developing RDW experiences using architectural manipulation. Knowing that a more cognitive taxing task is better at grabbing attention than one that is less taxing can help to hide a change to a scene better [24]. Moreover, virtual environments are often quite dynamic and realistic, meaning that users need to digest a lot of sensory information when exploring the virtual environment. A high cognitive load plays well with change blindness and makes it easier to hide changes to the environment. However, a high cognitive load can also negatively impact the illusion of natural walking because it may draw attention to the inconsistencies in mapping between the real and the virtual world. Balancing and maintaining cognitive load to grab attention while ensuring a seamless and natural experience is crucial to a successful implementation of RDW [24].

Martin et. al. [26] recently tried to explore the phenomenon of change blindness to find when and why changes are detected. Change blindness research has so far mostly employed 2D graphics, but this study used immersive 3D environments to more closely mimic real-world situations. Two experiments were performed in the study. In the first experiment, the researchers looked at how change blindness was impacted by the kind, distance, complexity, and range of vision of changes. In the second experiment, they varied the number of changes in the scene to evaluate the connection between change blindness and visual working memory capacity. Around 770 trials were used to gather data for the study, and the detection ratio was 46.22% overall, implying that more than half of the changes were missed. These results solidify the psychophysical foundation for change blindness and its relation to the limitations of the visual working memory, and support the working of change blindness in VR and in RDW.

2.1.3 Platforms for RDW

Li et. al. [19] also mention the use of platforms to simulate customizable RDW techniques thereby giving researchers and developers toolkits or methods to employ for their own innovative ideas. One platform is the OpenRDW toolkit [27], which is an open source library that includes a collection of functions and algorithms that make it possible for developers to implement novel perspective redirection ideas without the need to setup the virtual environment from scratch. The toolkit can drastically simplify the development process as it includes head tracking and scene management and algorithms for translation gain and rotation gain that can all be customized to implement the developer's new idea.

Besides this, the exploration of machine learning algorithms in the field of RDW has also recently gained traction, e.g. with a novel implementation of reactive steering algorithms using reinforcement learning that are able to outperform the steering algorithm S2C in its current form [28]. The paper demonstrates that machine learning algorithms are capable of optimizing human engineered redirection algorithms with enough training.

2.1.4 Challenges with RDW

A major drawback to current RDW techniques lies in the development process. The virtual world must be carefully mapped to deliver the experience sought by the developer. The changes to the virtual world must be subtle enough to not draw the user's attention to themselves or make them feel uncomfortable, yet substantial enough to produce the desired manipulation of the walking path. In addition, developers must be aware of any potential safety issues, such as making sure that users do not unintentionally move into real-world walls or other obstructions while in the virtual environment [6].

2.2 Physiology and Psychology of Blinking

As mentioned, this project aims to uncover, whether it is possible to utilize blinking to induce change blindness in RDW. Blinking is a habit that humans are mostly not consciously aware of in daily life. It is an essential procedure that is enforced for the eyes to work properly. The primary physiological function of blinking is to lubricate and clear the eye of dust particles[29], and spontaneously happens at a frequency ranging from 6 to 30 times per minute[30][31].

The blink rate is correlated with several mental processes, one of which is the mental burden on the brain, with the blink rate declining as the mental load rises. In a 1972 study by Holland and Tarlow[32], the participants were asked to look at a plethora of numbers and then to recall either 4, 6, or 8 digits. Their results showed that the rate of blinking increased when there were few digits to remember and decreased when there were eight to recall, implying that a state of concentration decreases blink rate. A few other studies have looked at the correlation between blinking and social behavior, where research have shown that conversation and blinking are tightly connected. It was discovered that blink rates were higher during conversation than what was required for eye lubrication and that short and lengthy blinks had quite different purposes[33]. Another social communication study examined how nonverbal response engagement is perceived by people blinking; the test subjects in this study would engage with an on-screen agent that would blink at various intervals. The findings revealed that blinking at higher rates could produce a higher sense of being watched than slow or immediate blinking[34].

Regulating the physiological process of blinking is handled by a sophisticated system in the body. It is an fast involuntary movement which usually lasts 100-400ms[35], and involves the eyelids closing and then opening again. Blinking is mostly used to keep the eyes safe, by having the eyelids distribute a thin film of tears over the surface of the eye with each blink. This aids in keeping the eyes lubricated, which is necessary for clear vision and to stop eye damage[29]. However, the blinking rate can change based on a number of variables including age, gender, use of contact lenses, psychological state, drugs, diseases[30], and as mentioned also the surrounding environment, e.g. the weather, the people around, the time of day, etc. For instance, people tend to blink more frequently when they are sleepy or feeling their eyes getting dry, and less frequently when they are engaged or concentrating on a job, as the brain works on getting more information about the environment and therefore lowers the need of blinking. Studies exploring this found, that in simulated flights, the pilot's blink rate is lower than the co-pilot's, and drivers on busy city roads have lower blink rates than those on quiet city roads, as the visual information they need to process is more demanding[36].
Additionally, a decrease in blink rate has been tied to the viewing of immersive content. A study by Kim et al.[37] investigated how different screens affected blink rate, and found that, amongst natural blinking, a traditional monitor, a VR HMD, and AR glasses, the VR HMD

showed the lowest blink rate of close to 11 blinks per minute, compared to natural blinking which showed around 18 blinks per minute. Being aware of how VR consumption affects the rate of blinking is an important aspect when working with blink-induced designs, as well as being aware of how a lower than usual blink rate affects such things as simulator sickness.

As mentioned, blinking serves the primary purpose of protecting the eyes, but it also plays a part in psychological instances such as emotions, attentiveness, social contact and communication. A protracted or pronounced blink, might be used to express feelings like surprise, skepticism, or flirtation. Long-term eye contact or a lack of blinking may be interpreted as an aggressive or disrespectful gesture in some cultures. Below is a short description of the effect on psychological elements related to blinking[33], being important factors to have in mind depending on the application design.

- Emotions: Studies have shown that emotions can affect how often and how long eye blinks last. For instance, people typically blink more frequently when they are tense or nervous and less frequently when they are unhappy or depressed.
- Attention: The degree of focus or participation in a task can also affect how often your eyes blink. While engaged in an activity that needs continuous attention, like reading or watching a movie, people tend to blink less frequently.
- Social cues: Nonverbal communication and eye contact are two examples of social cues that might affect eye blinks. For instance, in some cultures, maintaining lengthy eye contact or failing to blink is a symbol of hostility or dominance.

The anatomy of the eye and the brain, as well as social and cognitive aspects, are all involved in the physiology and psychology of eye blinks. Knowing the natural functions of the body and how our emotions, attention, and social interactions affect them, enables us to comprehend the process of blinking, and what factors are necessary to integrate it most effectively for redirection manipulations to work seamlessly.

2.2.1 Eye tracking

As an extension to the section on blinking, we explore eye tracking, which is a technology able to track and measure gaze variables. Eye tracking is typically explored and used in studies relating to marketing, psychology and UX-design where it can be used to understand how people interact with their environment and the objects around them.

Eye-tracking devices are able to output variables, identifying where a person is looking by measuring the angle of the pupil and the reflection of an infrared light source on the cornea[38]. Eye-tracking in VR has become a popular technology to improve the experience in the virtual world. Devices like the Meta Quest PRO or the HTC Vive PRO Eye that are able to perform eye-tracking in VR, track the user's eye movement using small infrared cameras, which enables developers to use the output variables as raw data, to manipulate the virtual environment or to optimize the system[39]. This also includes blinking, which is output as whether the pupil is visible or not. If not then the eye must be closed, otherwise open. The following points are some of the primary advantages of eye tracking in VR[39]:

- Enhanced performance: To render visual quality on par with that of human vision would require an immense data stream of over 100 GB/sec[40], a very challenging achievement for modern HMD's. However, the eye's fovea, the central part of the retina, has such a high acuity that only around 4% of pixels in a VR setting are perceived at a high level of detail[41]. The remaining 96% of rendered pixels are not processed with high resolution. By only generating high-resolution visuals in the parts of the display that the user is looking at, known as foveated rendering, eye tracking can assist and lessen the computational strain on the system.

- Increased interactivity: Eye tracking can be utilized to enable organic interactions with the virtual environment, such as aiming the gaze at specific items to target them or initiating actions by looking at certain regions of the screen. Particularly useful would such interactions be for people with motor disabilities, that does not allow for the use of hand or head movement. In a more general sense, eye movement has shown to be a less accurate, but faster interaction method than other input devices[42][43], and can be utilized to reduce 'gorilla arm syndrome'[44] - a fatigue caused by lifting ones arms to perform hand-based gestures for a prolonged amount of time - something that has shown to reduce the time wanted to spend in VR[45].

- Improved learning experiences: Since its inception VR has been widely used for learning purposes because of its immersive simulated environments, that are fail safe and allows for repetition. The inclusion of eye tracking has lead to recent studies in educative domains such as sports[46][47], medicine, military[48], and transportation[49]. For evaluating the performance of such simulated learning, methods like post-experiments interviews and questionnaires are usually performed. But because this data is acquired post-experiment, it can be subject to being irrelevant if the participant's memory of events are reconstructed or fades due to factors such as level of fatigue, complexity or duration of the experiment[50]. In this context, eye tracking helps by enabling the researcher to objectively assess quantitative information of the user's visual, attentional, and cognitive capabilities, without interrupting their virtual learning experience.

The Meta Quest Pro has inbuilt eye tracking [51], where the HTC Vive Pro Eye instead uses its own cameras but rely on the Tobii SDK to enable development with eye tracking. The Tobii Eye Tracker can be used for a variety of purposes, including gaming, research, and user experience design, and as mentioned, it tracks eye movements using infrared cameras. It is known for its high accuracy, low latency and easy integration with the HTC Vive Pro Eye [52]. VR eye tracking is a technology that has the potential to increase system performance using foveated rendering, improve user experience with gaze detection, and offer insightful data on user behavior. It is likely to become a more significant part of the VR experience as VR technology develops.

2.3 State of the art

The purpose of the section is to describe three different studies related to blinking in VR. The first study focuses on finding effective methods to trigger blinks in VR. The second and third explores the use of blinks for reorienting or repositioning the user's viewpoint or the scene in VR. By exploring these studies, we aim to improve our understanding of blinking in VR to optimize our solution.

2.3.1 Induced blinking

In order to lessen the dependence of blink-based strategies on spontaneous and voluntary blinks, Zenner et al. [53] began this study to discover the most effective and efficient method to trigger blinks in VR. The study sought to compare the advantages and disadvantages of six distinct blink trigger implementations; flash trigger, blur trigger; approaching object trigger, sound trigger, glabella trigger and airpuff trigger.

Before evaluating the triggers, the authors found that both the sound trigger and the glabella did not meet the minimum requirements to be a part of the implementation, therefore these were not evaluated. The participants was instructed to remember the colors of four cubes positioned within a meter of them in a virtual world. 2 seconds later the cubes turned grey and the participant had to touch a colored sphere close to them with one hand while touching the cube they thought had the same color as the sphere with the other hand. Upon reaching 30% of the way, the trigger would activate and upon touching the cube the participants would be questioned with a 2AFC (Two-alternative forced choice) question: "Did you notice any visual or physical stimulus?", "Yes or No". The comparison between the four remaining triggers uncovered that an object approaching the user at a fast pace activates the human danger reflex, which resulted in the quickest and most similar response times of all triggers amongst all participants. This suggests that the danger reflex is efficient, making it particularly intriguing for inducing blinks in VR. Following the approaching object trigger, both the flash trigger and the airpuff trigger had a blink response time significantly shorter than the baseline. The investigation also discovered that although the Blur trigger was the least noticeable, all reliable trigger configurations were perceptible. The comfort of the triggers was also reported to vary, with the Approaching Object trigger being the most disruptive. The other triggers were experienced with low to medium levels of distraction and were regarded as being noticeably less intrusive without any visible symptoms of being startled. Both the flash trigger and the airpuff trigger could be good options for induced blinking in an architectural manipulation scenario to have the user blink when needed. The flash trigger was implemented using only software, having the HMD displays light up and brightly illuminate the eyes. The airpuff trigger was implemented with hardware, where a small air blowing mechanism was integrated into the inside of the HMD. Combining any of these triggers with our approach would surely be interesting, however the flash trigger is seemingly the most fitting trigger for our approach.

2.3.2 Blink-induced redirection

Langbehn et. al. [35] explores the use of blinks to either reorient or reposition the viewpoint of the user in VR, for the purpose of RDW. The study examines these manipulations through two similar psychophysical experiments, focused on subtle perspective manipulations, where the users should not notice the manipulation. The first experiment analysed how much rotation could be applied to the user's avatar and thereby their viewpoint in VR, during a blink. Sixteen participants (three females and 13 males) took part in the experiment, all with normal or corrected-to-normal vision and no reported disorders of equilibrium. The participants were tasked with completing 198 randomized trials with 11 different offsets ranging from 0 to $\pm$ 15 degrees on all three axes. The results show that users could tolerate up to ± 4.763 degrees of rotation around the up axis, ± 2.358 degrees of rotation around the right axis, and ± 3.703 degrees of rotation around the forward axis without noticing. The participants reported a slight increase in VR sickness symptoms and a mid-high sense of presence. Most participants tried to focus on a particular point or feature in the virtual environment to compare their position before and after the rotation. Overall, the users did not notice the reorientation. The second experiment analyses how much translation can be applied to the users viewpoint in VR, during eye-blinks. The same sixteen participants completed the second experiment, and it was found that the detection thresholds for translations were approximately 4-9 cm, and there were differences between the three axes. The participants did not report any significant differences in bias when compared to the offset of 0.0. The experiment showed an increase in VR sickness symptoms, and most of the participants tried to focus on a certain point to compare their position before and after blinking. The sense of presence in the VE was similar to the first experiment.

Nguyen et. al. [54] explored as a similar manipulation, but rather than rotating the users viewpoint, they rotate the entire scene around the user, whereby the manipulation becomes an architectural manipulation. The idea of rotating the scene during blinks was employed to assist the use of existing algorithms as the steer-to-center and steer-to-orbit algorithms mentioned earlier. The study aims to find thresholds for how much scene rotation is possible either during blinks or with eyes open. Fourteen participants (ten male and 4 female) walked around a virtual forest to collect targets while experiencing discrete scene rotation every time they blinked. The study found a detection threshold of 9.1 ± 3.2 degrees of scene rotation during blinks and a threshold of 2.4 ± 0.97 with eyes open. The average blink frequency was found to be 13 times per minute. More than half of the participants reported an increase in simulator sickness parameters such as sweating, general discomfort and nausea after the experiment, however the study rationalized that this is the effect of a long time spent in a small virtual environment with many resets.

3 Methods

In this project, we propose a novel approach to architectural manipulation within RDW, utilizing change blindness during the momentary blindness of a blink (detailed descriptions of the methods and materials of each study can be found in Section 4 and 5).
To do so, two separate experiments are presented. E1 (4) aims to estimate distance-based

detection thresholds for how much a wall can be moved, either closer or further away during the temporary blindness of a blink. Based on the existing research on noticability of turn-based manipulations [7][55], and blink-induced manipulations[54], we hypothesise that it is entirely possible to mask the movement of a wall, both closer and further away from the user, during their momentary blindness of a blink. The question is what distance is it possible to move this wall. The users will experience a total of 84 varying rooms, seated on a swivel chair, trying to detect a manipulation of the placement of the wall in front of them. The wall will either move 5, 10 or 15 % closer to or further away from them either while they blink or while they look away, comparing this novel approach to the approach by Suma et. al. in 2011 [7].
E2 (5) aims to evaluate these thresholds in a real scenario, and just as how E1 (4) compared blink-induced change blindness to turn-based change blindness, experiment 2 will also compare this novel approach to the approach by Suma et. al. [7]. The question we seek to answer in experiment 2 is whether blink-induced change blindness can hide changes in architectural manipulation as well as turn-based change blindness in a scenario previously tested by Suma et. al.

4 Experiment 1 (E1): Estimation of detection thresholds for blink-induced change blindness

This initial experiment explored threshold detection of architectural manipulations, to investigate how much we would be able to move a wall either closer or further away, before the user took notice of the movement. Two types of change blindness manipulations were compared this way, blink-induced and turn-based or "turn your back", the original approach proposed by Suma et al.[7]. An existing study on thresholds of turn-induced manipulations has been explored by Hwang et. al[55], who applied small gains to walls with doors that was either 1-, 2-, or 3-meters from the user. Their results indicate that the noticability thresholds increase with the distance between the user and the wall.
As earlier mentioned, many factors can affect how well manipulations in 3D environments are noticed such as their distance to the user, visual complexity, and whether they happen inside or outside a their FOV[26]. Since an visually similar environment, to the one by Hwang et al.[55], was built in this experiment, we expected similar thresholds for

turn-based manipulation. However, for blink-induced manipulations, a larger distance makes room for more objects, which affect changes that are immediately presented to the user right in front of them, such as difference in lighting and relative position between virtual objects. For this, distances of 2-, and 4-meters between the user and manipulated wall were used, to presented a significant change in the user's FOV.
Additionally, as mentioned in section 2.2.1, a user's gaze affects how detailed the virtual world is perceived at any given time. Areas or objects that are looked straight at become much clearer, while their peripheral vision becomes unclear. If not guided, the gaze may significantly impact the noticability of blink-induced visual changes in front of the user.

4.1 Methods - E1

To acquire data on thresholds we are essentially interested in if users notice a change of an architectural manipulation or not. To do so, a within subjects study based on a 2x2 factorial design was performed, as seen in table 1. This allowed us to investigate the effect of our two independent variables (type of change blindness and distance from the user to the manipulation) on a single dependant variable (noticability).

		Distance to manipulation	
		2 meters	4 meters
Type of change-blindness	Blink-induced	Noticability	Noticability
	Turn-induced	Noticability	Noticability

Table 1: 2x2 factorial design of E1. Two independent variables (type of change-blindness and distance to manipulation), each with two levels (respectively, blink-, and turn-induced vs. 2 and 4 meters) are being compared to see their effect on a single dependent variable, noticability.

Like Hwang et. al[55], movements were applied to the manipulated wall. Through early testing, appropriate numbers of -15% to 15%, in increments of 5 percentage points, were applied to both distances, including no movement. These numbers or percentages translate to the translation gain set on the wall, e.g. -15% translation in the 2m distance condition, has the wall move 15% or 0.3m closer to the user. For our use case, a task that worked for both types of change-blindness while also keeping the users' vision focused on the manipulated wall was needed. This was done by having a light on the wall in front of, and behind the user, that were either showing green or red. Users were then given the instruction of looking at the green light. The experiment consisted of 84 total trials for each user (2 types of change-blindness × 2 wall distances × 7 wall movements × 3 repetitions). Half of the users started the trials in the 2m wall distance room and the other half started in the 4m wall distance room wherein they encountered 42 randomly sequenced types of change-blindness and amount of wall movement. Upon completing the 42 first trials the participant is given a small break before trying the other wall distance 42 times. When the user completed the test, they were asked to answer a few post-test questions regarding their experience in the virtual environment.

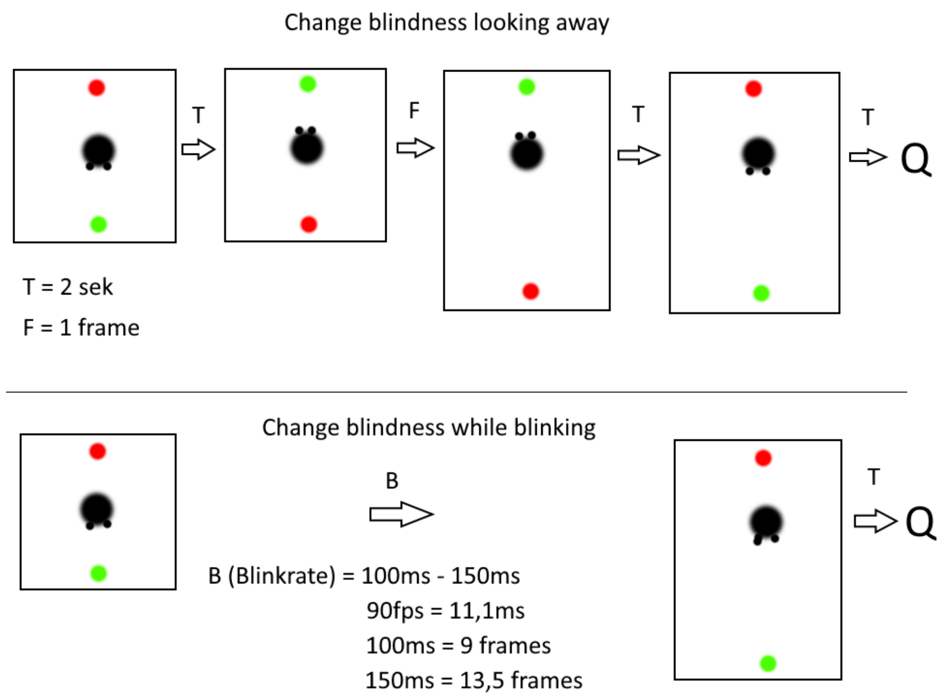


Figure 4: In condition one, change blindness looking away, participants looked at a green dot on a wall for two seconds, then turned to another wall where a manipulation occurred. After waiting two seconds, they faced the original wall and were asked about its movement. In condition two, change blindness while blinking, participants blinked, experienced the manipulation, and then waited two seconds before responding.

Similarly to Hwang et. al.[55], the participant was asked a question following each manipulation. For our experiment, they were faded to a separate environment beforehand, to reduce their remembrance of the manipulated environment. Here, they were presented with the question: "Did the wall with the door move closer or further away?". With a controller the user could click either closer or further, meaning their response was gathered using a two-alternative forced choice (2AFC) method that forced them to select one of the answers and eliminated the neutral option. The data was then analyzed by processing it through a fitted psychometric function using MatLab, with the form $f(x) = \frac{1}{1+e^{a*x+b}}$. The Point of Subjective Equality (PSE), or 0.5 in the psychometric function, is the point at which the individual is unaware of any wall movement. In a study exploring perspective gains in RDW, Steincke et. al [16] argued that thresholds for perspective gains can be found at the point where the manipulation is only detected some of the time and therefore defines detection thresholds for gains smaller and greater than the point of subjective equality (PSE). This is based on the subject's probability of choosing the "smaller" response correctly. This project defines wall movements as gains, which is the amount of wall-translation closer or further away from the user, and focuses on the range of gains where subjects cannot reliably detect the difference in change between scenes, i.e. where they perceive the scene pre-gain and post-gain as identical. In this experiment, the resulting detection thresholds range from the lower point of 0.25, i.e. the halfway point between PSE and 0, to the higher point of 0.75, i.e. the halfway point between PSE and 1. This 25-75 percent range of gains will provide an interval of possible architectural manipulation implementations for future use in RDW.

4.2 Design - E1

The design of E1 had the participant sitting in the middle of a virtual room, surrounded by four walls, one having a door. The wall with the door had a light placed on it, and so did the wall opposite the door. As mentioned, these lights served as tasks, to ensure that the participants keep their gaze at a similar point, in front of the manipulated wall. For comparability, the simple environment design is based on Hwang et. al.[55], as well as research showing that less cognitive load is linked to a higher blinking rate, i.e. the simpler the room design, the more the participant should blink. Point of views (POV) from the two different room sizes in E1 can be seen in figure 5a and figure 5b.

(a) Point of view of from a trial with a room size of 2 meters from the user to the manipulated wall.

(b) Point of view of from a trial with a room size of 4 meters from the user to the manipulated wall.

4.3 Implementation - E1

Firstly the conditions were setup with a for-loop in the start function to get a list of the 84 scenarios. In the update function they would then be randomly selected based on how many there were left in the list and then removed (see listing 1 and 2).

```
TestConditions = new List<Conditions>();
int Id1 = 0;
for (int c = 1; c <= 2; c++) // cond
{
    for (double g = 0.85; g <= 1.15; g += 0.05) // gain
    {
        for (int r = 1; r <= 3; r++) // reps
        {
            TestConditions.Add(new Conditions()
            {
                Id = Id1++,
                Name = "W1",
                Condition = c,
                WallDist = 1,
                Gain = Math.Round(g, 3),
                Reps = r,
                Choice = 0,
                CorrectGuess = 0,
                StartTime = DateTime.Now,
                WallMovedTime = DateTime.Now,
            });
        }
    }
}
```

Listing 1: MoveWall.cs - Add conditions

```
if (TestConditions != null && TestConditions.Count != 0)
{
    Conditions conditionRemove = TestConditions[testCondition];
    TestConditions.Remove(conditionRemove);

    if (TestConditions != null && TestConditions.Count != 0)
    {
        testCondition = UnityEngine.Random.Range(0, TestConditions.Count);
    }
}
```

Listing 2: MoveWall.cs - Remove condition

The blinking simple took advantage of the Tobii eye tracking SDK and combined the usage of the continuous variables IsLeftEyeBlinking and IsRightEyeBlinking to output a true boolean and an counter integer to see when both eyes were closed (see listing 3).

```
if (TobiiData.IsLeftEyeBlinking && TobiiData.IsRightEyeBlinking)
{
    if (blinkBoolCounter == 0)
    {
        blinkBoolCounter = 1;
        blinkingBothEyes = true;
    }
}
```

Listing 3: MoveWall.cs - Blinking

Moving the wall happened when the user was blinking or when the participant looked away and moved the wall either closer or further away based on the condition in the list (see listing 4).

```
moveWall.transform.localPosition = new Vector3( moveWall.transform.localPosition.x,
moveWall.transform.localPosition.y,  moveWall.transform.localPosition.z *
    (float)condition.Gain);
```

Listing 4: MoveWall.cs - Moving the wall

4.4 Participants - E1

20 people (8 female and 12 male) participated in E1, being between 20 and 30 years of age (median age 26, min. 21, max. 30). Selection of the participants was based on convenience sampling where most were students or researchers at Aalborg University Copenhagen. The participants were all asked about their experience with VR before commencing the experiment, whereas 8 answered "No experience", 10 answered "A little experience", 2

answered "Experienced". None of the participants were aware that blinking was the area of exploration of the project beforehand.

Figure 6: Participant taking part in E1.

4.5 Equipment - E1

The experiment took place in the multisensory experience lab at Aalborg University Copenhagen. The participants were seated on a swivel chair during the entire test, wearing a HTC Vive Pro Eye HMD. A rotating chair was important to allow them to turn to face the relevant virtual walls during the trials while still staying in the same place in the same distance from the relevant virtual walls. The playspace was 3x3 meters large, sectioned in a secluded part of the lab. The HMD was plugged into a PC running Unity3D, Steam VR, SR-Ranipal and Tobii to use eye-tracking and detect gaze and blinks. The PC had an NVIDIA 3080 graphics card and a AMD Ryzen 7 5800x processor. Frame rate was at a constant 90 frames per second.

4.6 Results - E1

Table 2, figure 7 and figure 8 shows the result of the threshold detection for each distance between the manipulated wall and the user. The x-axis on figure 7 and figure 8 represents the wall movement gain, and the y-axis represents the participants' rating of expansion. The grey area represents the wall gain from the lower threshold (25%), PSE (50%) and the upper (75%).

Table 2 shows, from left to the right, turn-induced 2 meters have a detection threshold ranging from 0.758 to 1.269 with a PSE of 1.013. The data shows that the participant does not notice the change in wall movement until it has moved towards them at approximately 0.5 meters and 0.52m away. In the 4 meters turn-induced the detection threshold range from 0.910 to 1.120 with a PSE of 1.014 the notice of wall movement is 0.36 m towards the user and 0.8m away from the user. The blink-induced 2 meters has a threshold ranging from 0.810 to 1.198 with a PSE of 1.004, the notice of wall movement is 0.38m for both ways. lastly the 4 meters blink-induced range from 0.898 to 1.123 with a PSE of 1.010, notice of wall movement is 0.4m towards the user and 0.9m away from the user.

Table 2 shows the threshold gains that can be applied in an application before the user will notice the change. Later in this project, these numbers will be applied for a more in-depth experiment.

Type of change blindness	**Distance from wall**	**Lower**	**PSE**	**Upper**
Turn-induced	2 meters	0.758	1.013	1.269
Turn-induced	4 meters	0.910	1.014	1.120
Blink-induced	2 meters	0.810	1.004	1.198
Blink-induced	4 meters	0.898	1.01	1.123

Table 2: Turn- & blink-induced thresholds across conditions.

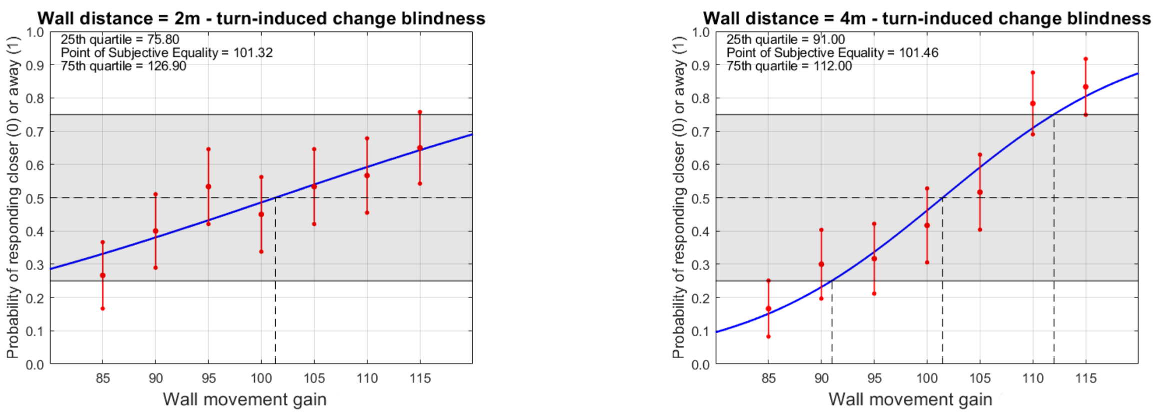


Figure 7: Turned-induced blinking for 2 & 4 meters. x-axis shows the applied gain on the wall, where 100 = no gain applied. y-axis show the probability of the participants guess.

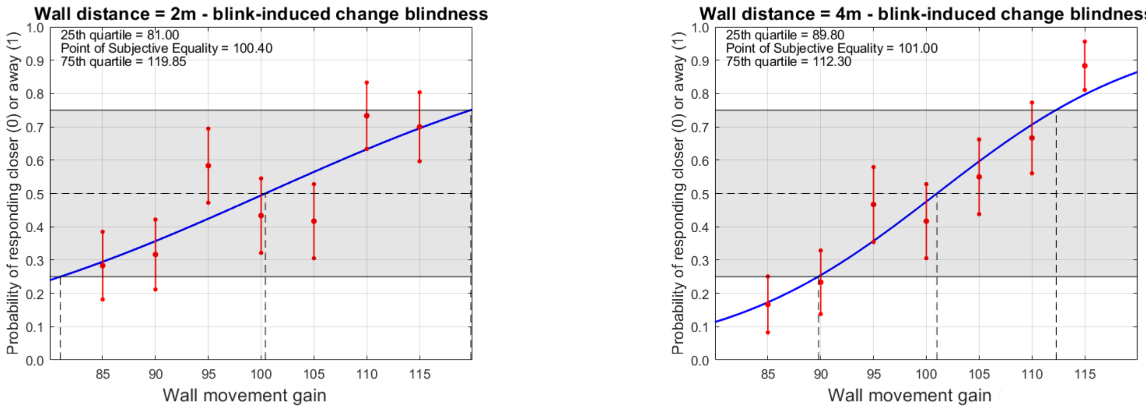


Figure 8: Blink-induced blinking for 2 & 4 meters x-axis shows the applied gain on the wall, where 100 = no gain applied. y-axis show the probability of the participants guess.

Type of redirection	Distance from user to wall	Gain towards	Gain away
Turn-induced	2 meters	0.50 meters	0.52 meters
Turn-induced	4 meters	0.36 meters	0.80 meters
Blink-induced	2 meters	0.38 meters	0.38 meters
Blink-induced	4 meters	0.40 meters	0.90 meters

Table 3: Turn- & blink-induced thresholds across conditions, in meters

4.6.1 Qualitative results

The question 'Did you notice what made the door move?' 7 people noticed that when they blinked it would activate the manipulation and only 1 participant guessed that it was when they had their back turned. Many different strategies was used by the participants when asked the question 'Did you use any strategies for noticing the movement?'. Some used their peripheral vision to notice the manipulation "... Using the peripheral vision" " It was easier to see when the door filled more of my field of view." Others used the placement of the door, the wall corners or the floor "To look at the floor and the doorframe while blinking ... ", "I tried to look towards corner or where wall and ceiling meet", "I looked at the bottom of the wall and the sides to see the change".

4.7 Discussion - E1

This experiment aimed to investigate and compare the detection thresholds of various wall movements, caused by turn- and blink-induced change blindness from a distance of 2 and 4 meters.

The results from the 2m distance trials showed turn-based thresholds ranging from 0.758 to 1.269 and blink-based thresholds ranging from 0.810 to 1.198. These results were quite surprising, seeing as neither the lower nor the upper thresholds of both were within the range of the 0.85 to 1.15 gains implemented. This might suggest that there may be a limited range of sensitivity or variability in the observed responses, or that the detection difficulty was to high thus resulting in a great deal of guessing. Either way, in the case of the 2m distance, it might have been more fruitful to use a different range of gains, e.g. 0.7 - 1.3 with 0.1 increments.

The 4m distance results on the other hand appear to be very relevant for further use. The results indicated that the detection thresholds ranged from 0.910 to 1.120 with turn-based change blindness and that the detection thresholds ranged from 0.898 to 1.123 with blink-based change blindness. These findings suggest that participants did not notice the change in wall movement until it had moved towards them at approximately 0.36 meters towards and 0.8 meters away from the user when turn-based and 0.4 meters towards the user and 0.9 meters away from the user when blink-based. Interestingly, when comparing to the findings of Hwang et. al.[55], who found larger ranges in thresholds as the distance from user to manipulation increased, our distance of 4 meters showing a significantly lower range than theirs of 3 meters.
When looking at the results, the findings suggest that the detection of wall movement induced by turns and blinks is affected by distance, with lower thresholds observed at shorter distances. However, as mentioned above, there is a fair chance that the 2m distance results are not useful seeing as the results indicate larger gains possible than ones tested. The results from this study have practical implications for the design of virtual and augmented reality applications, where the detection of environmental changes is critical for user experience.

In conclusion, the study aimed to investigate the detection thresholds and perception of wall movement induced by turns and blinks at 2- and 4-meters distances from the participant. The findings suggest that blink-induced change blindness works to the degrees mentioned earlier, however elements such as the visual appearance of the virtual environment might have affected the thresholds gathered in this experiment along with other elements. Therefore future studies are required to determine whether these results will generalize beyond what is presented here. Meanwhile, the results found and presented in Table 3 could have implications for the design of virtual and augmented reality applications, possibly allowing for a more seamless and immersive experience for users.

5 Experiment 2 (E2): Utilizing blink-induced change blindness for RDW

By looking at the data from the detection threshold experiment, E1, we have obtained possible imperceptible gains for blink-induced change blindness that may be utilized for architectural manipulation. In a prior experiment by Suma et. al. [7], they found that turn-induced change blindness could be successfully implemented for RDW without the vast majority of users noticing it. This experiment aims to see if blink-induced change blindness has the same potential as turn-induced.

Section 4 shows that it is, to some extent, possible to implement blink-induced change blindness without the user noticing wall-translation gains in their FOV. In E2, a VR environment closely resembling the one in Suma's change blindness experiment [7] was developed with two conditions; turn-induced change blindness, and blink-induced change blindness. This way, a comparison based on an established design could be done, to see if there are any differences in change detection between the two conditions.

5.1 Method - E2

The data was gathered by using a within-study design with two conditions; turn-induced and blink-induced change blindness. All participants in the experiment would experience both conditions, with half of them starting in the turn-induced condition, and the second half starting in the blink-induced condition, to counterbalance. Each one consisted of four adjacent rooms, that the participants had to chronologically navigate through. These rooms were separated by a hallway, with each room having a task that needed to be completed in order to be able to move to the next. The task, in both conditions, were the same and would trigger the architectural manipulation depending on the condition (turn- or blink-induced change blindness). Similarly to Suma et al.[7], the triggered architectural manipulation would change the room layout by moving one wall closer to the user, another away, and flip the open room door to a perpendicular wall, but keeping the area of the room the same. Before starting the test, a diversionary explanation was told to every participant, about how the environment and its objects would be procedurally generated and that visual glitches might or might not appear, to mask the real intent of the experiment. The choice was made, so the user would divert the attention from the blink-induced change blindness. At the end of the test, an explanation of the real intent of the experiment would be told to the test participants.

In this experiment, multiple points of data were gathered to reach a conclusion. The first data gathered was from the simulator sickness questionnaire(SSQ)[56]. The SSQ is a self-report questionnaire, that is designed to assess a individual and their symptoms of simulator sickness or motion sickness. It consists of 16 questions, where the participants has to rate each question/symptoms on a likert scale (0="None." to 3="Severe."). Each participant's answers would be further divided into 4 sub-scores; Nausea, Oculomotor, Disorientation, and a Total-score. Each participants would answer the questionnaire before and after each condition, a total of three times, to see if the experiment gave them any simulator sickness, or if one condition affected their ratings more than the other.

Each condition would also have a 'post-exposure questionnaire' regarding RDW. This questionnaire is based on Suma et al.'s[7] study on turn-induced change blindness in VR, where nine questions were asked to assess if the participant noticed the architectural manipulation of the RDW. Within the questionnaire only two of the nine question were primary outcomes to asses the RDW, while the rest were installed as decoy questions. The participant would be asked to rate the following question on a likert scale (0="I did not notice." to 6="Very obvious." The primary questions are italicized.

- I saw the virtual world get smaller or larger.
- *I felt like I was turning in circles.*
- I saw the virtual world flicker.
- I saw the virtual world get brighter or dimmer.

- *I saw that something in the virtual world had moved*
- I saw the virtual world rotating.
- I felt like I was getting bigger or smaller.
- I felt like I was being moved around.
- I saw that something in the virtual world had changed size

The participants would be equipped with a controller that had a virtual pointing ray attached, as seen in figure 9, and were asked to click at anything in the VR environment, if they felt like an object glitched by changing size, position, rotation or color. By pressing a button, they had the ability to turn the ray on and off for their liking. They would then be able to precisely point at an object and click on the trigger, to log the targeted object. This data would be used to see if they noticed the architectural manipulation while inside the VR environment. As each condition has four manipulations, the participants can only guess correct four times per condition.

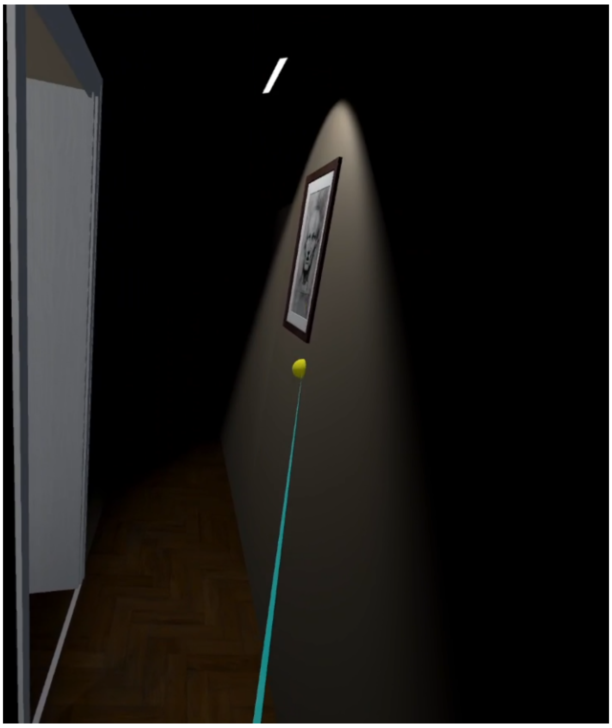


Figure 9: POV from a participant, showing the virtual ray attached to their controller. Pointing the yellow ball at an object, and clicking the trigger button, would log information about the object, and if it was clicked before or after a manipulation had happened.

At the end of the test, the participants had the opportunity to give qualitative feedback. The questions that were asked was; "Can you describe the path or the layout of the virtual environments you just experienced?", to get a sense of their spatial awareness, and "Do you have any other comments regarding the experience in terms of walking around, glitches happening, changes or anything else out of the ordinary going on?"

5.2 Design - E2

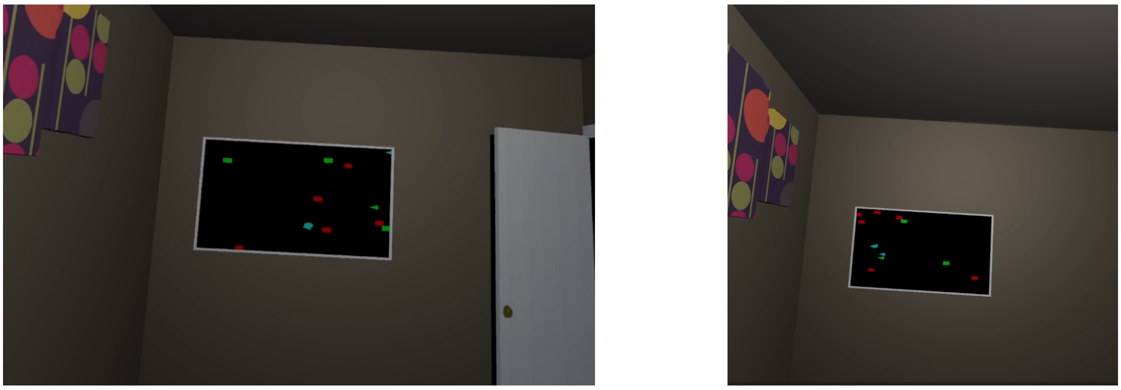
The design of the second experiment of this report incorporates the detection thresholds found as a result of Section 4. Participants now have to actively move and navigate through different rooms in the VR environment. As mentioned, each room had a task that needed to be completed, before they could continue. Just like our initial experiment, a task that kept the gaze at a specific area as the manipulation happened, was desired, to ensure the same noticability conditions for each participant. The task was a simple video, in which participants had to count specific colored shapes moving across a TV screen, as seen on figure 10b. Building on research from section 2.3.1, a white flickering effect was applied to the video, in hopes of inducing blinks on the participant, triggering the blink-induced manipulation more frequently. To activate the task, the participant had to

(a) Task screen. (b) Television playing the task video.

Figure 10: POV's of the tasks in room one. The participant would walk up the task screen (a), and turn around and count how many green shapes would appear on the television, facing the manipulated wall (b).

go near a computer screen, placed in the corner of each room, as seen on figure 10a. Once they were close enough, a task would appear on the screen, telling the participant to count how many specific shapes that would appear on the television, attached to the manipulated wall. Once the video stopped, they needed to turn back to the task screen, and say out loud how many shapes they counted, into the virtual microphone. As presented in Section 4, the reason behind this simple task is to lessen the cognitive load and induce blinking on the participants.

Like the initial experiment, the participants were exposed to two conditions, one applying turn-induced change blindness and the other blink-induced change blindness. The former followed Suma et al.'s[7] manipulation design and happened behind the participant's back, as they approached the task screen. Like them, our virtual corridor was also 0.9 meters of our slightly bigger playspace of 5x5 meters (theirs being 4.25x4.25 meters), close to 20% of. This meant that for the blink-induced condition, the participant would be facing the wall from exactly 4 meters, before the manipulation happened, making it plausible for our thresholds to be applied. Although, this also meant that a total movement of 1 meter had to be made, in order to rearrange the architecture. Through early testing, we found that an increment of four blinks, a move closer of 0.25m, were acceptable in terms of task length and noticability. The television also scaled down 5% at every increment, to have it fill the same amount in the participant's FOV. A comparison of this manipulation can be seen in figure 11. Another design addition was to add fixed paintings outside



(a) No blinks have been made. Nothing has been manipulated.

(b) Four blinks have been made. The room has been fully manipulated.

Figure 11: POV's of the incremental manipulations of the blink-induced change blindness condition. From no blinks and manipulations(a), to after four blinks and a fully manipulated room where the wall has moved 20% closer, the television has reduced 20% in size, and the door has rotated 90 degrees to the front facing wall(b).

each room, to further the illusion of walking down a straight corridor. The participant would faced with the same image before turning down a corridor that had been rotated 90 degrees. Figure 12a and 12b showcase the architectural manipulation that happened in both conditions. Subsequently, figure 12c to 12h show how each room layout were designed pre- and post-manipulation, with free 3D assets found on the Unity asset store[57].

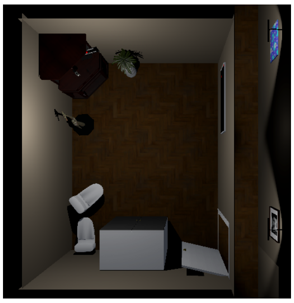
(a) Room one, pre-manipulation

(b) Room one, post-manipulation

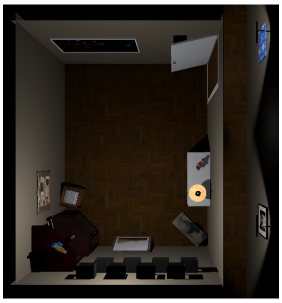
(c) Room two, pre-manipulation

(d) Room two, post-manipulation

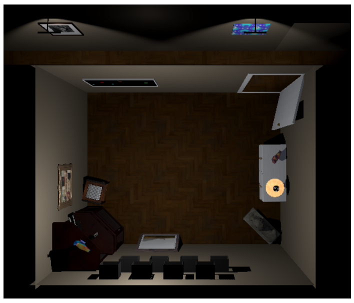
(e) Room three, pre-manipulation



(f) Room three, post-manipulation



(g) Room four, pre-manipulation



(h) Room four, post-manipulation

Figure 12: Overview of the four different rooms pre- and post-manipulation, side by side

5.3 Implementation - E2

Section 5 is implemented in four separate scripts: Two similar scripts to control the architectural manipulation based on whether the user is testing the turn-induced condition (condition 1) or the blink-induced condition (condition 2). One script is put on the objects that would be focusable by the users gaze, in order to see what they are looking at. Lastly, one script continuously logs clicked input to a .csv file for whatever the user might have seen changing in the environment, during the test.

Note, due to area limitations, the implementation itself was done using a smaller playspace of 3x3 meters. To ease internal testing, the implementation of entry- and exit- routes for the rooms, are based on reference points. This allows for the base environment (walls and doors) to be scaled by simple parameters, such as virtual corridor length and playspace size (in meters).

The two scripts in charge of manipulating the environment underway, use triggers to know when to commence the manipulation. The script in charge of turn-induced change blindness uses an 'OnTriggerEnter' function to ensure that the change only happens once when the user walks into the Box Collider setup, where the task takes place. In this script, the manipulation happens as the user enters the task area while looking at the task screen. This is to make sure that the user does not back into the task area and then notice the manipulation to the environment (see listing 5 and 6).

```
private void OnTriggerEnter(Collider other)
{
    if (scriptRunning != 1) {
        return;
    }

    if (other.tag == "roomTrigger" && !triggeredOnce) {
        triggerEnterOnce = 1;
    }

}
```

Listing 5: RedirectionLogic.cs - Turn-Away manipulation part one

```
private void TriggerEnterChecker()
{
    if (gazeObject == null) {
        return;
    }

    if (triggerEnterOnce == 1 && gazeObject.tag == "monitor") {
        firstTrigger = true;
        triggerCounter++;
        DeactivateOldObjects();
        ScaleTV();
        ActivateNewObjects();
        PlayVideo();
        timerChecker = 1;
        timerRandom = Random.Range(20, 25);
        triggeredOnce = true;
        triggerEnterOnce = 0;
    }
}
```

Listing 6: RedirectionLogic.cs - Turn-Away manipulation part two

Colliding with the box collider placed around the task, enables the function 'TriggerEnterChecker.cs' to do its purpose, which is to deactivate and activate gameobjects to move the wall, the TV, the door and the furniture. The use of activation instead of translation is used, to move away from very specific translations, to having gameobjects already in place and then simply activating and deactivating based on where the user walks and which box colliders they enter and trigger. The difference between the scripts in condition 1 and 2 is largely in this part of the script. Condition 2 uses the 'OnTriggerStay' function for the tag "roomTrigger" instead, where the manipulation is activated in four steps while the user is looking at the task on the TV-screen. Both conditions use 'OnTriggerEnter' for exiting the room, and walking into the next room, where new gameobjects are deactivated and activated along the way (see listing 7).

```
private void OnTriggerStay(Collider other)
{
    if (other.tag == "roomTrigger") {
        if (triggerCounter == 0) // Same for triggerCounter 2, 4 and 6 {
            EnableTasks("taskOne"); // Same for taskTwo, Three and Four
        }

        firstTrigger = true;
        if (gazeObject == null) {
            return;
        }

        if (gazeObject.tag == "blinkTag")
        {
            if (blinkBoolCounter == 1)
            {
                if (wallMoved == 0)
                {
                    blinkCounter++;
                    if (blinkCounter == 1 || blinkCounter == 2 || blinkCounter == 3)
                    {
                        MoveRoom();
                        ScaleTvInIncrements();
                        wallMoved = 1;
                    }

                    if (blinkCounter == numOfIncrements)
                    {
                        triggerCounter++;
                        ScaleTV();
                        DeactivateOldObjects();
                        ActivateNewObjects();
                        wallMoved = 2;
                        timerChecker = 1;
                        timerRandom = Random.Range(5, 10);
                    }
                }
            }
        }
    }
}
```

Listing 7: BlinkRedirectionLogic.cs

The remaining code is largely the same in both scripts. They both utilize the Tobii SDK to locate gaze and to measure eye-blinks, and they both use functions to play and pause the task-video based on which triggers the user activates underway. In order to locate gaze, the fourth script is implemented to put onto the gameobjects that should be focusable, this would be the monitor, the TV, the walls, the floor, the furniture, etc. This script also utilizes the Tobii SDK, where the script inherits the IGazeFocusable component and uses its function GazeFocusChanged() to update a boolean from false to true, when a gameobject equipped with the script is looked at. The last script is implemented to allow the user to point out any changes they might see underway and log them to a csv file (see listing 8).

```
private void CheckSelectedObject()
{
    if (reticlePoser != null && pointerState)
    {
        selectedObject = reticlePoser.hitTarget;
        var selectedName = "";

        if (selectedObject != null)
        {
            selectedName = selectedObject.name;
            var parentName = "";
            var parentObjectTransform = selectedObject.transform.root;

            if (clickingTrigger)
            {
                writerChanges = new StreamWriter(filePathChanges, true);
                writerChanges.WriteLine(selectedName + ";" + parentName + ";" +
 triggerCounter + ";" + reticlePoser.hitDistance);
                writerChanges.Close();
            }
        }
    }
}
```

Listing 8: VivePointerSelect.cs

The Start function creates a new csv file and checks if a controller is active. Whenever the user clicks the trigger button on the controller while pointing at an object, an additional line will be added to the csv file with the root object, the current object, the counter which checks in which room the user finds themselves in and the distance to the object. The user can at any time hide or show the pointer by clicking the pad on the controller and thereby changing the boolean pointerState.

5.4 Participants - E2

To gather participants for the experiment, convenience sampling was utilized with a sample size of twenty test participants (nine female, eleven male; mean = 34.9, median = 27, SD = 14.8, minimum age 24, maximum age 66). eight participants reported no VR experience, nine had little VR experience, and three had experience with VR.

5.5 Equipment - E2

The experiment was done at Aalborg University Copenhagen, in an open space. The participants were wearing a HTC Vive Pro Eye HMD. The HMD was connected to a PC that ran Unity, Steam VR, SR-Ranipal and Tobii to use the eye tracking and detection of gaze and blinks. The HMD was plugged into a PC running Unity3D, Steam VR, SR-Ranipal and Tobii to use eye-tracking and detect gaze and blinks. The PC had an NVIDIA 3080 graphics card and a AMD Ryzen 7 5800x processor. Frame rate was at a constant 90 frames per second.

5.6 Procedure - E2

The experiment was done in a quiet area that slightly exceeded that of the play-area of 5x5 meters. This area would be kept hidden by a curtain. The participants would fill out a consent form before a short briefing of the experiment would take place. The diverted explanation of 'procedural generated environment' was told to the participants, before trying the VR application. After the first condition, the participants would then answer a SSQ and a post-exposure questionnaire. Thereafter, they would experience the second condition, with a SSQ and post-exposure questionnaire. The HMD would be equipped before removing the curtain, and a test conductor would guide the participant by their shoulder, to their starting position, before each condition. After the experiment was done, the real intention of the experiment was revealed to the test participants.

5.7 Results - E2

Unless otherwise written, all statistical results use a significance value of $\alpha = .05$.

The following result that will be presented is;

- Post exposure questions from turn- & blink-induced change blindness
- Point and click data
- SSQ
- Qualitative feedback

5.7.1 Post exposure questions from turn- and blink-induced change blindness

The first results of the experiments are those of the post-exposure questionnaire. Here, a Wilcoxon signed-rank test was done to compare the two conditions, to see if there

were any significant difference between the two groups. Table 4, shows all the analysed measurements of the post-exposure questions. Figure 32, compares the two conditions with a boxplot of each questions.
The decoy questions did not show any significant difference between the two groups, with p-value ranging from 0.06 to 0.83. The primary measurement questions also did not show any significance difference, with question 2; "I felt like i was turning in circles." having a median = 0 on both conditions, p-value = 0.88 and z-statistic = 16, and question 5; "I saw that something in the virtual world had moved." having a median = 0.5 on both conditions, p-value = 0.99 and z-statistic = 23. One decoy question had a higher median rating than the rest, "I saw the virtual world flicker." with a median = 1 in turn-induced change blindness and a value of 2 in blink-induced change blindness.

Questions	p-value	Turn-induced median	Blink-induced median	z-statistic
I saw the virtual world get smaller or larger.	0.14	0	1	20
I felt like I was turning in circles.	0.88	0	0	16
I saw the virtual world flicker.	0.28	1	2	29
I saw the virtual world get brighter or dimmer.	0.67	0	0	17.5
I saw that something in the virtual world had moved.	0.99	0.5	0.5	23
I saw the virtual world rotating.	0.06	0	0	15
I felt like I was getting bigger or smaller.	0.40	0	0	8
I felt like I was being moved around.	0.83	0	0	20
I saw that something in the virtual world had changed size.	0.47	0	0	20

Table 4: Measurements of post-exposure questions.

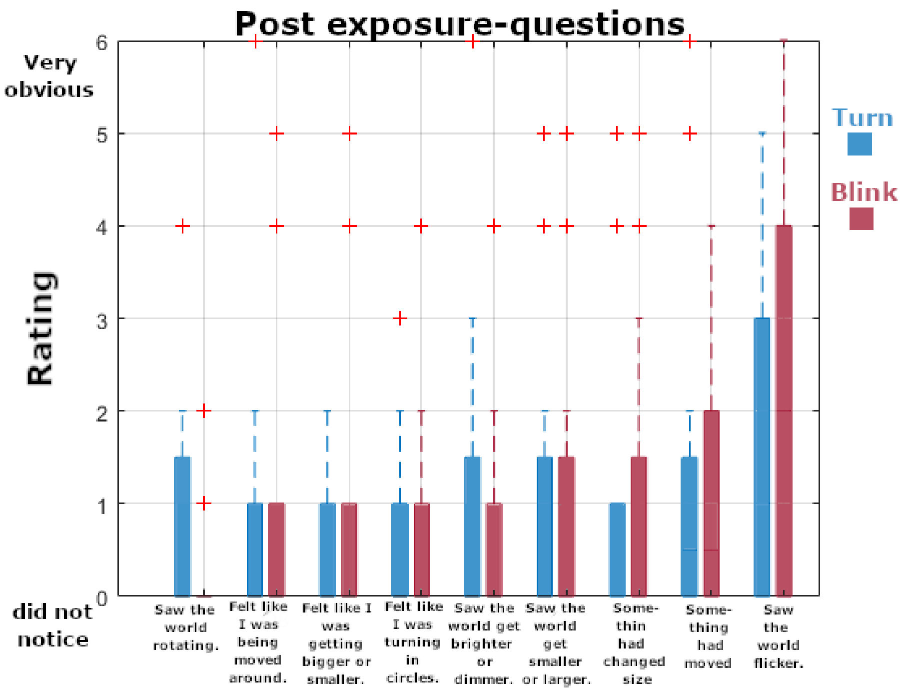


Figure 13: Comparison Questions

5.7.2 Click data

In the turn-induced change blindness condition, three different people had at least one correct click, compared to 5 different people in the blink-induced condition. The dataset did not pass the test of normality, therefore a wilcoxon signed-rank test was used, with both groups having a median of 0, and a p-value = 0.35.

5.7.3 SSQ

Figure 32 shows a comparison of the SSQ sub-scores; Nausea, Oculomotor, Disorientation and total-score, of the two conditions.
The turn-induced condition shows nausea shows a mean = 7.63, median = 4.77 SD = 10.07. Oculomotor; mean = 8.33, median = 7.58, SD = 8.48. Disorientation; mean = 9.04, median = 0, SD = 12.18. Total-score a mean = 9.16, median = 5.61, SD = 9.27.

blink-induced condition shows nausea; mean = 8.58, median = 9.54, SD = 9.73. Oculomotor a mean = 10.23, median = 7.58, SD = 11.07. Disorientation mean = 12.52, median = 0, SD = 19.10. Total-score a mean = 11.78, median = 7.48, SD = 13.25

To compare the two conditions for the SSQ, a Wilcoxon signed-rank test was used, as it did not pass the test of normality. All the sub-scales in the SSQ shows no significant difference between the groups; Nausea p-value = 0.76 and z-statistic = 10.5. Oculomotor p-value = 0.46 and z-statistic = 25. Disorientation p-value = 0.21 and z-statistic = 4 Total-score p-value = 0.28 and z-statistic = 20.

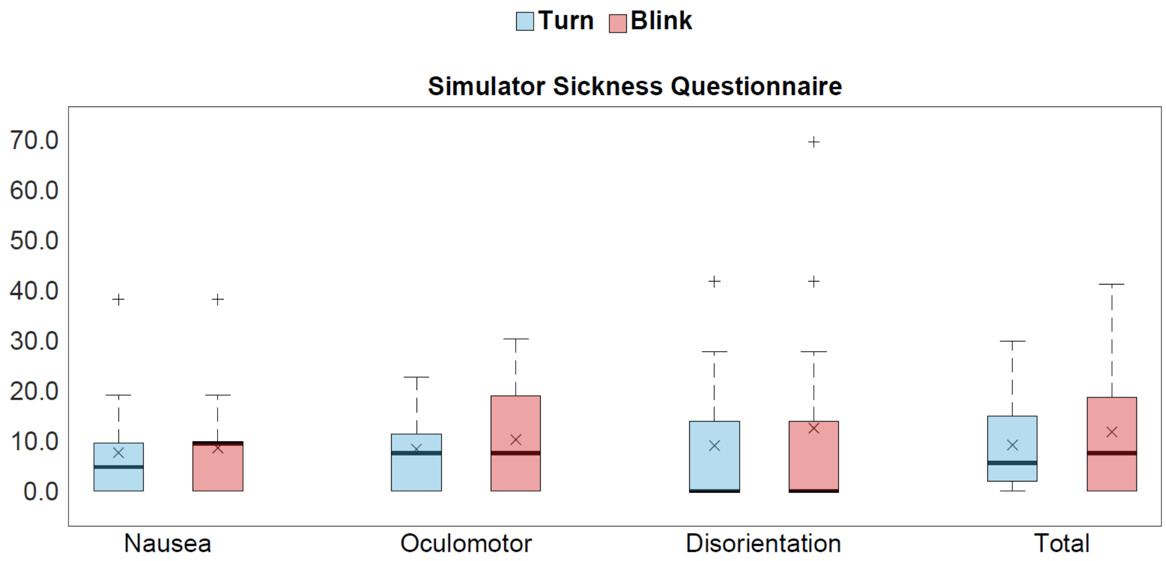


Figure 14: SSQ

5.7.4 Qualitative feedback

The qualitative feedback gathered from the first question 'Can you describe the path or the layout of the virtual environments you just experienced?', revealed how eleven participants mentioned they were walking down a long hallway with rooms with answers such as "A long corridor with multiple rooms", "A long hallway with rooms" and "I walked into a hallway with some rooms with a computer screen and a TV screen"(translated from danish), with no one mentioning any rotated or changed paths. The rest commented on the room decorations, or answered "no".

The question 'Do you have any other comments regarding the experience in terms of walking around, glitches happening, changes or anything else out of the ordinary going on?', showed that some participants thought they saw glitches with answers such as; "I noticed some flickers/glitches while I was counting shapes on the television", "The tv glitched a few times in each room ... " and "There was some flickering on the television where the shapes moves", perceiving the intentional flicker as one of the non-occurring glitches.

5.8 Discussion - E2

The results from the experiments showed promising outcomes regarding utilizing blink-induced change blindness in architectural manipulations. The results from the post-exposure questionnaires showed that there was no significant difference in the ratings, when comparing the two conditions. Interestingly, in the blink-induced condition, the door would disappear from the users' FOV, with most not detecting any changes. Although the blink-induced condition had some higher ratings in question 5, regarding movements in the virtual world, it showed no difference between the two conditions.

The diverted explanation of a procedurally generated environment, together with the task, might have caused enough cognitive load on the participants, which made them shift attention to the procedurally generated glitches that in fact did not happen, and also the given task of counting shapes.

The decoy question of "I saw the virtual world flicker." had the highest median rating of both condition. This high rating might have been a result of the given task of counting shapes, as the video that was played had some intentional flicker implemented. This is also supported by the answers that some participants gave from the qualitative feedback.

It has to be noted that Suma et al.'s[7] study had a higher pool of participant of thirty-seven, compared to our sample size of twenty. Their study had a positively outcome as a redirection technique which proved to be effective. As an effective technique, this experiment sought to achieve same effectiveness as the one done by Suma et al. By looking at the results of this project, the blink-induced change blindness was surprisingly effective when compared to turn-induced, as there were no questions serving as a significant outlier in the post-exposure questionnaire.

E2 gathered some valuable results that shed some understanding of applying blinks as a change blindness technique for architectural manipulations. However, most of the analysis did not pass the test of normality, which in the end resulted in analysing the data with non-parametric methods. For a more clear conclusion, parametric methods would be ideal to use.

The hardware that was used limited the experiment capabilities, as the wire from the VR headset could only reach certain amount of space, and might have been an apparent element to the participants, although not mentioned. A wireless HMD could differ this experience of the experiment, to an extent. The implementation of a scaleable virtual environment also worked without issues, making possible studies in larger, similarly designed environments, possible.

Distractors and attention in this experiment was a key factor, that would help the manipulation to an extend which would mask the manipulation for the participants. It would be interesting to experiment further into these factors, to get a better understanding of how much cognitive load is needed to mask manipulations for blink-induced change blindness.

6 General Discussion

Through this section we will elaborate and analyse on our approach for the project and the findings that we have uncovered in the two experiments, based on their possible application in the field of RDW.

The first experiment (see Section 4), revealed that it is possible to move a wall both closer and further away from a user seated at distances of 2 m and 4 m, keeping the manipulation unnoticed while the user is blinking. The wall positioned 2 m from the user could be translated up to 0.38 m closer or further away from the user, without them noticing the change. The wall position 4 m away from the user could be translated up to 0.4 m closer and 0.9 m away from the user, without them noticing. The users blinked unconsciously, meaning that the experiment time for some of them were much longer than for others. Data regarding the blink frequency, collected in the first experiment, showed that the users' blink frequency are similar to research by [37], with an average frequency of 10,54 blinks per second over an average duration of 29,51 minutes.

The second experiment (see Section 5), showed positive results, with no significant difference across conditions from both the post-exposure questionnaire and SSQ. The vast majority of participants did not notice changes when they blinked. Again, this is possibly due to the task and how attention grabbing it was.

There are, of course, limitations to keep in mind. The time it takes to get users to blink can play a significant role, depending on the application, and seeing as we did not log the blink-frequency in E2 we are unable to compare the blink frequency from E1 to that of E2, to see whether the task reduced or increased the this frequency. Simultaneously, there is no telling whether the results are the product of a distractor that is too powerful. It is compelling to explore whether the manipulations would go as unnoticed either with a less powerful distractor or even without a distractor.

This project did lean on Suma et al.'s[7] study regarding turn-induced change blindness, and Hwang et al.'s [55]'s study on turn-induced thresholds, to investigate and implement blink-induced thresholds. Those studies were reliable and valid enough to compare results with regard to this project's unique investigation on architectural manipulations caused by blink-induced change blindness.

There is still much room to grow for this novel technique. In in terms of a developer's work load when it comes to customization of the environment, it provides a less intense work load than other redirection techniques, like steering algorithms, while not setting as many limits in terms of fixed waypoints where users have to go. The constant unconscious action of blinking can, to a certain extent, dynamically change the environment behind and in front of the user, without them noticing. For future projects, the idea of combining blink-induced and turn-induced change blindness with procedural generation of an environment with distractors is gripping and could possibly be very powerful.

7 Conclusion

In this project, we proposed a new architectural manipulation technique in RDW, exploiting blink-induced change blindness, wherein we make large scene changes inside users' field of view. We have established detection thresholds for moving walls in rooms of varying sizes, showing that blinking is able to mask changes in wall position of up 0.9 m away from the user.

These results were put to the test in a second experiment, where we built a RDW scenario, in which the user had to walk through four rooms, each of which included a task wherein an entire room would incrementally change after each blink. After four blinks the architecture had been manipulated to the extent that they could walk to the next room, with a total front facing manipulation of 20%, and with minimal noticability. The results from the second experiment showed, that through careful planning, it is very possible to use blinks to hide large scene changes in a RDW scenario, and allow users to explore a larger virtual space than physically possible, though with the attention-grabbing power of the task and its distracting effect, kept in mind.

8 Bibliography

[1] M. Wang, Z.-Y. Chen, W.-C. Cai, and F. Steinicke, “Transferable virtual-physical environmental alignment with redirected walking,” *IEEE transactions on visualization and computer graphics*, vol. PP, Nov 2022.

[2] S.-B. Jeon, J. Jung, J. Park, and I.-K. Lee, “F-rdw: Redirected walking with forecasting future position,” *arXiv preprint arXiv:2304.03497*, 2023.

[3] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks, “Walking > walking-in-place > flying, in virtual environments,” in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’99*. Not Known: ACM Press, 1999, p. 359–364. [Online]. Available: http://portal.acm.org/citation.cfm?doid=311535.311589

[4] M. Slater, M. Usoh, and A. Steed, “Taking steps: the influence of a walking technique on presence in virtual reality,” *ACM Transactions on Computer-Human Interaction*, vol. 2, no. 3, p. 201–219, 1995. [Online]. Available: https://dl.acm.org/doi/10.1145/210079.210084

[5] S. Razzaque, Z. Kohn, and M. C. Whitton, “Redirected Walking,” in *Eurographics 2001 - Short Presentations*. Eurographics Association, 2001.

[6] N. C. Nilsson, T. Peck, G. Bruder, E. Hodgson, S. Serafin, M. Whitton, F. Steinicke, and E. S. Rosenberg, “15 years of research on redirected walking in immersive virtual environments,” *IEEE Computer Graphics and Applications*, vol. 38, no. 2, p. 44–56, Mar 2018.

[7] E. A. Suma, S. Clark, D. Krum, S. Finkelstein, M. Bolas, and Z. Warte, “Leveraging change blindness for redirection in virtual environments,” in *2011 IEEE Virtual Reality Conference*, Mar 2011, p. 159–166.

[8] J. Zheng, K. Chan, and I. Gibson, “Virtual reality,” *IEEE Potentials*, vol. 17, no. 2, p. 20–23, Apr 1998.

[9] M. Slater, D. Banakou, A. Beacco, J. Gallego, F. Macia-Varela, and R. Oliva, “A separate reality: An update on place illusion and plausibility in virtual reality,” *Frontiers in Virtual Reality*, vol. 3, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frvir.2022.914392

[10] C. Anthes, R. J. García-Hernández, M. Wiedemann, and D. Kranzlmüller, "State of the art of virtual reality technology," in *2016 IEEE Aerospace Conference*, Mar 2016, p. 1–19.

[11] B. Sharma. (2022) Sensation perception in virtual reality. [Online]. Available: https://immersivedirectory.com/sensation-perception-in-virtual-reality/

[12] U. of Florida. (2023) Sensation and perception. [Online]. Available: https://pressbooks.online.ucf.edu/lumenpsychology/chapter/reading-what-is-perception/

[13] J. Diemer, G. W. Alpers, H. M. Peperkorn, Y. Shiban, and A. MÃ$\frac{1}{4}$hlberger, "The impact of perception and presence on emotional reactions: a review of research in virtual reality," *Frontiers in Psychology*, vol. 6, Jan 2015. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fpsyg.2015.00026/abstract

[14] W. J. Norton, J. Sauer, and D. Gerhard, "A quantifiable framework for describing immersion," *PRESENCE: Virtual and Augmented Reality*, vol. 29, p. 191–200, Nov 2022. [Online]. Available: https://direct.mit.edu/pvar/article/doi/10.1162/pres_a_00361/113514/A-Quantifiable-Framework-for-Describing-Immersion

[15] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe, "Estimation of detection thresholds for redirected walking techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 1, p. 17–27, Jan 2010.

[16] ——, "Estimation of detection thresholds for redirected walking techniques," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 1, pp. 17–27, 2009.

[17] C. T. Neth, J. L. Souman, D. Engel, U. Kloos, H. H. Bulthoff, and B. J. Mohler, "Velocity-dependent dynamic curvature gain for redirected walking," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 7, pp. 1041–1052, 2012.

[18] E. Langbehn, P. Lubos, G. Bruder, and F. Steinicke, "Bending the curve: Sensitivity to bending of curved paths and application in room-scale vr," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 4, pp. 1389–1398, 2017.

[19] Y.-J. Li, F. Steinicke, and M. Wang, "A comprehensive review of redirected walking techniques: Taxonomy, methods, and future directions," *Journal of Computer Science and Technology*, vol. 37, no. 3, p. 561–583, Jun 2022. [Online]. Available: https://doi.org/10.1007/s11390-022-2266-7

[20] D.-Y. Lee, Y.-H. Cho, and I.-K. Lee, "Real-time optimal planning for redirected walking using deep q-learning," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, Mar 2019, p. 63–71.

[21] B. Bolte and M. Lappe, "Subliminal reorientation and repositioning in immersive virtual environments using saccadic suppression," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 4, p. 545–552, Apr 2015.

[22] E. A. Suma, Z. Lipps, S. Finkelstein, D. M. Krum, and M. Bolas, "Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 4, pp. 555–564, 2012.

[23] R. A. Rensink, J. K. O'Regan, and J. J. Clark, "To see or not to see: The need for attention to perceive changes in scenes," *Psychological Science*, vol. 8, no. 5, p. 368–373, 1997. [Online]. Available: http://journals.sagepub.com/doi/10.1111/j.1467-9280.1997.tb00427.x

[24] G. Bruder, P. Lubas, and F. Steinicke, "Cognitive resource demands of redirected walking," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 4, p. 539–544, Apr 2015.

[25] J. G. Frederiksen, S. M. D. Sørensen, L. Konge, M. B. S. Svendsen, M. Nobel-Jørgensen, F. Bjerrum, and S. A. W. Andersen, "Cognitive load and performance in immersive virtual reality versus conventional virtual reality simulation training of laparoscopic surgery: a randomized trial," vol. 34, p. 1244–1252, Mar 2020.

[26] D. Martin, X. Sun, D. Gutierrez, and B. Masia, "A study of change blindness in immersive environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, p. 2446–2455, May 2023.

[27] Y.-J. Li, M. Wang, F. Steinicke, and Q. Zhao, "Openrdw: A redirected walking library and benchmark with multi-user, learning-based functionalities and state-of-the-art algorithms," in *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Oct 2021, p. 21–30.

[28] R. R. Strauss, R. Ramanujan, A. Becker, and T. C. Peck, "A steering algorithm for redirected walking using reinforcement learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 5, p. 1955–1963, May 2020.

[29] T. Bedinghaus, "Why you blink your eyes," 2020. [Online]. Available: https://www.verywellhealth.com/why-do-we-blink-our-eyes-3879210

[30] M. G. Doane, "Interactions of eyelids and tears in corneal wetting and the dynamics of the normal human eyeblink." *American journal of ophthalmology*, vol. 89, no. 4, pp. 507–516, 1980.

[31] S. Duke-Elder, "System of ophthalmology," *The ocular adnexa*, p. 414–419, 1974.

[32] M. K. Holland and G. Tarlow, "Blinking and mental load," *Psychological Reports*, vol. 31, no. 1, pp. 119–127, 1972.

[33] M. J. Doughty, "Consideration of three types of spontaneous eyeblink activity in normal humans: during reading and video display terminal use, in primary gaze, and while in conversation," *Optometry and Vision Science*, vol. 78, no. 10, pp. 712–725, 2001.

[34] Y. Yoshikawa, K. Shinozawa, and H. Ishiguro, "Social reflex hypothesis on blinking interaction," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 29, no. 29, 2007.

[35] E. Langbehn, F. Steinicke, M. Lappe, G. F. Welch, and G. Bruder, "In the blink of an eye: leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 66:1–66:11, Jul 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201335

[36] J. A. Stern, "What's behind blinking?" *The Sciences*, vol. 28, no. 6, pp. 43–44, 1988. [Online]. Available: https://nyaspubs.onlinelibrary.wiley.com/doi/abs/10.1002/j.2326-1951.1988.tb03056.x

[37] J. Kim, L. Hwang, S. Kwon, and S. Lee, "Change in blink rate in the metaverse vr hmd and ar glasses environment," *International Journal of Environmental Research and Public Health*, vol. 19, no. 14, p. 8551, 2022.

[38] B. T. Carter and S. G. Luke, "Best practices in eye tracking research," *International Journal of Psychophysiology*, vol. 155, p. 49–62, Sep 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167876020301458

[39] I. B. Adhanom, P. MacNeilage, and E. Folmer, "Eye tracking in virtual reality: a broad review of applications and challenges," *Virtual Reality*, Jan 2023. [Online]. Available: https://doi.org/10.1007/s10055-022-00738-z

[40] B. Bastani, E. Turner, C. Vieri, H. Jiang, B. Funt, and N. Balram, "Foveated pipeline for ar/vr head-mounted displays," *Information Display*, vol. 33, no. 6, pp. 14–35, 2017.

[41] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn, "Towards foveated rendering for gaze-tracked virtual reality," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–12, 2016.

[42] L. E. Sibert and R. J. Jacob, "Evaluation of eye gaze interaction," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 2000, pp. 281–288.

[43] O. Špakov, P. Isokoski, and P. Majaranta, "Look and lean: accurate head-assisted eye pointing," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 35–42.

[44] A. Cockburn, P. Quinn, C. Gutwin, G. Ramos, and J. Looser, "Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback," *International Journal of Human-Computer Studies*, vol. 69, no. 6, pp. 401–414, 2011.

[45] S. Jang, W. Stuerzlinger, S. Ambike, and K. Ramani, "Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 3328–3339.

[46] J. M. Bird, "The use of virtual reality head-mounted displays within applied sport psychology," *Journal of Sport Psychology in Action*, vol. 11, no. 2, pp. 115–128, 2020.

[47] S. Pastel, J. Marlok, N. Bandow, and K. Witte, "Application of eye-tracking systems integrated into immersive virtual reality and possible transfer to the sports sector-a systematic review," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 4181–4208, 2023.

[48] B. Xie, H. Liu, R. Alghofaili, Y. Zhang, Y. Jiang, F. D. Lobo, C. Li, W. Li, H. Huang, M. Akdere *et al.*, "A review on virtual reality skill training applications," *Frontiers in Virtual Reality*, vol. 2, p. 645153, 2021.

[49] Y. Lang, L. Wei, F. Xu, Y. Zhao, and L.-F. Yu, "Synthesizing personalized training programs for improving driving habits via virtual reality," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2018, pp. 297–304.

[50] T. Renshaw, R. Stevens, and P. D. Denton, "Towards understanding engagement in games: an eye-tracking study," *On the Horizon*, 2009.

[51] Meta. (2022) Meta quest pro: Built with privacy in mind. [Online]. Available: https://www.meta.com/da-dk/blog/quest/meta-quest-pro-privacy/

[52] Tobii. (2023) Tobii xr devzone. [Online]. Available: https://developer.tobii.com/xr/

[53] A. Zenner, K. Ullmann, O. Ariza, F. Steinicke, and A. Krüger, "Induce a blink of the eye: Evaluating techniques for triggering eye blinks in virtual reality," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23. New York, NY, USA: Association for Computing Machinery, Apr 2023, p. 1–12. [Online]. Available: https://dl.acm.org/doi/10.1145/3544548.3580888

[54] A. Nguyen and A. Kunz, "Discrete scene rotation during blinks and its effect on redirected walking algorithms," in *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '18. New York, NY, USA: Association for Computing Machinery, Nov 2018, p. 1–10. [Online]. Available: https://doi.org/10.1145/3281505.3281515

[55] J.-Y. Hwang, S.-U. Kwon, Y.-H. Cho, S.-B. Jeon, and I.-K. Lee, "Redirected walking in infinite virtual indoor environment using change-blindness," *arXiv preprint arXiv:2212.13733*, 2022.

[56] P. Bimberg, T. Weissker, and A. Kulik, "On the usage of the simulator sickness questionnaire for virtual reality research," in *2020 IEEE conference on virtual reality and 3D user interfaces abstracts and workshops (VRW)*. IEEE, 2020, pp. 464–467.

[57] B. P. Studio.

9 List of Figures

1 prop = proprioception; rep = internal representation; sense = sensory data [4] 3
2 A visualization of applied translation gain and rotation gain to change their walking path. [15] 5
3 Visualizing the four manipulation techniques [10]. The user's path in the actual world is indicated by the red arrows, while their path in the virtual environment is indicated by the blue dotted arrows. (a) shows rotation gain. (b) shows translation gain. (c) shows curvature gain. (d) shows bending gain. 6
4 In condition one, change blindness looking away, participants looked at a green dot on a wall for two seconds, then turned to another wall where a manipulation occurred. After waiting two seconds, they faced the original wall and were asked about its movement. In condition two, change blindness while blinking, participants blinked, experienced the manipulation, and then waited two seconds before responding. 20
6 Participant taking part in E1. 24
7 Turned-induced blinking for 2 & 4 meters. x-axis shows the applied gain on the wall, where 100 = no gain applied. y-axis show the probability of the participants guess. 25
8 Blink-induced blinking for 2 & 4 meters x-axis shows the applied gain on the wall, where 100 = no gain applied. y-axis show the probability of the participants guess. 26
9 POV from a participant, showing the virtual ray attached to their controller. Pointing the yellow ball at an object, and clicking the trigger button, would log information about the object, and if it was clicked before or after a manipulation had happened. 30
10 POV's of the tasks in room one. The participant would walk up the task screen (a), and turn around and count how many green shapes would appear on the television, facing the manipulated wall (b). 31
11 POV's of the incremental manipulations of the blink-induced change blindness condition. From no blinks and manipulations(a), to after four blinks and a fully manipulated room where the wall has moved 20% closer, the television has reduced 20% in size, and the door has rotated 90 degrees to the front facing wall(b). 32
12 Overview of the four different rooms pre- and post-manipulation, side by side 33

13 Comparison Questions . . . 39
14 SSQ . . . 41
15 Consentform experiment 1 . . . 55
16 Qualitative questions post-test for experiment 1 . . . 56
17 Consentform experiment 2 . . . 57
18 Post-exposure questionnaire; question 1-5 . . . 58
19 Post-exposure questionnaire; question 6-9 . . . 59
20 SSQ; question 1-4 . . . 60
21 SSQ; question 5-8 . . . 61
22 SSQ; question 9-12 . . . 62
23 SSQ; question 13-16 . . . 63
24 Post-exposure question 1 . . . 64
25 Post-exposure question 2 . . . 64
26 Post-exposure question 3 . . . 65
27 Post-exposure question 4 . . . 65
28 Post-exposure question 5 . . . 66
29 Post-exposure question 6 . . . 66
30 Post-exposure question 7 . . . 67
31 Post-exposure question 8 . . . 67
32 Post-exposure question 9 . . . 68
33 Task ready screen. Approaching this would change image to the appropriate task. . . . 69
34 Task of room one. Made in GIMP. . . . 69
35 Task of room two. Made in GIMP. . . . 70
36 Task of room three. Made in GIMP. . . . 70
37 Task of room four. Made in GIMP. . . . 71

10 List of Tables

1 2x2 factorial design of E1. Two independent variables (type of change-blindness and distance to manipulation), each with two levels (respectively, blink-, and turn-induced vs. 2 and 4 meters) are being compared to see their effect on a single dependent variable, noticability. 19
2 Turn- & blink-induced thresholds across conditions. 25
3 Turn- & blink-induced thresholds across conditions, in meters 26
4 Measurements of post-exposure questions. 39

A Experiment 1 Questionnaire

P10 - Virtual Reality Experiment

Thank you for participating in our experiment.

This experiment involves a series of scenarios presented in virtual reality. After each scenario a question will be presented and can be answered with the included controller. Estimated duration of the test is approximately 30 minutes and will consist of two sessions, separated by a small break.

If you feel any discomfort during the experiment, please notify us or simply remove the head-mounted display.

After answering the following preliminary questions, you are ready to start. Please notify the test conductor.

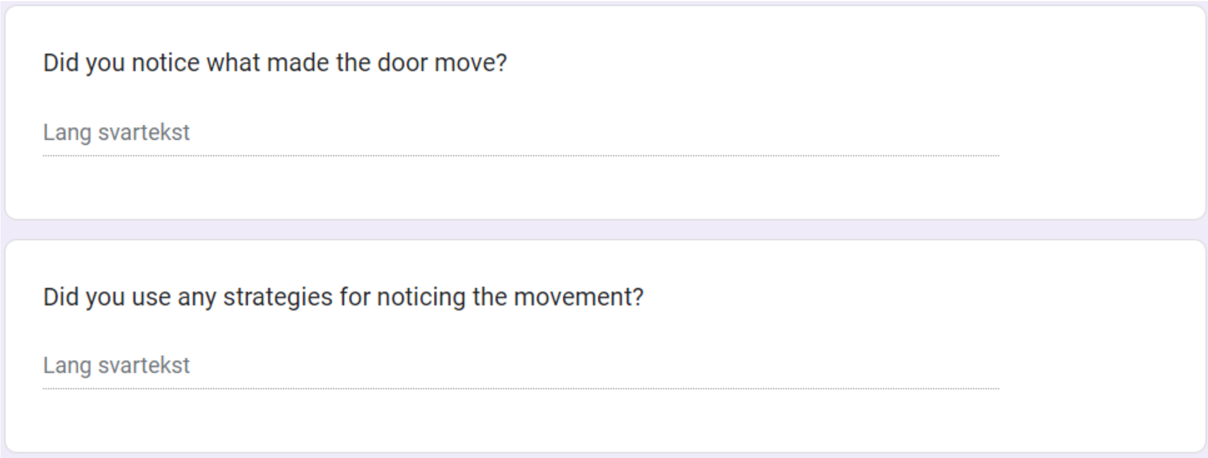
By participating, you consent to your data being used for our research project and purely academic purposes. It will not be distributed elsewhere. You are above the age of 18 and are okay with us potentially recording the experiment.

If you have questions at any point, do not hesitate to ask us.

For withdrawal of any provided information or for further questions please send an email to: jvalvi17@student.aau.dk

Thank you very much!

Figure 15: Consentform experiment 1



Did you notice what made the door move?

Lang svartekst

Did you use any strategies for noticing the movement?

Lang svartekst

Figure 16: Qualitative questions post-test for experiment 1

B E2 Questionnaire

Informed Consent to participants

I agree to participate in the study conducted by Theo Khumsan as part of a master thesis at Aalborg University Copenhagen. I understand that the purpose of this study is to gain insight and data into participants' views
on Virtual Reality of precedualy generated environments.

I understand that my participation in this research will involve wearing a VR headset and physically walking around. I might be inconvenienced by the time required to be involved in this study, but no
other harm is likely to result from my participation.

I understand that the research activity may require a recording in audio, video, or otherwise for help with transcription or other necessary data computation.

I hereby give consent to the recording of my activity. I understand that the research activity requires the taking of a screenshot or photograph of
me during the activity, used to provide evidence to the teachers. The image will not be used for anything else.

I am aware that I can contact Theo Khumsan (phone: [redacted], email: tkhums17@student.aau.dk) if I have any concerns about the research. I also understand that I am free to withdraw my participation from this study at any time I wish, without consequences, and without giving a reason. I will not be penalised in any way for declining to take part in any stage of the research.

I agree that Theo has answered all my questions fully and clearly.

I agree that the research data gathered from this study may be submitted in a form that does not identify me in any way.

By pressing "Next" I concent to participate.

Figure 17: Consentform experiment 2

I saw the virtual world get smaller or larger. *

	0	1	2	3	4	5	6	
Did not notice or did not happen	○	○	○	○	○	○	○	Very obvious

I felt like I was turning in circles. *

	0	1	2	3	4	5	6	
Did not notice or did not happen	○	○	○	○	○	○	○	Very obvious

I saw the virtual world flicker. *

	0	1	2	3	4	5	6	
Did not notice or did not happen	○	○	○	○	○	○	○	Very obvious

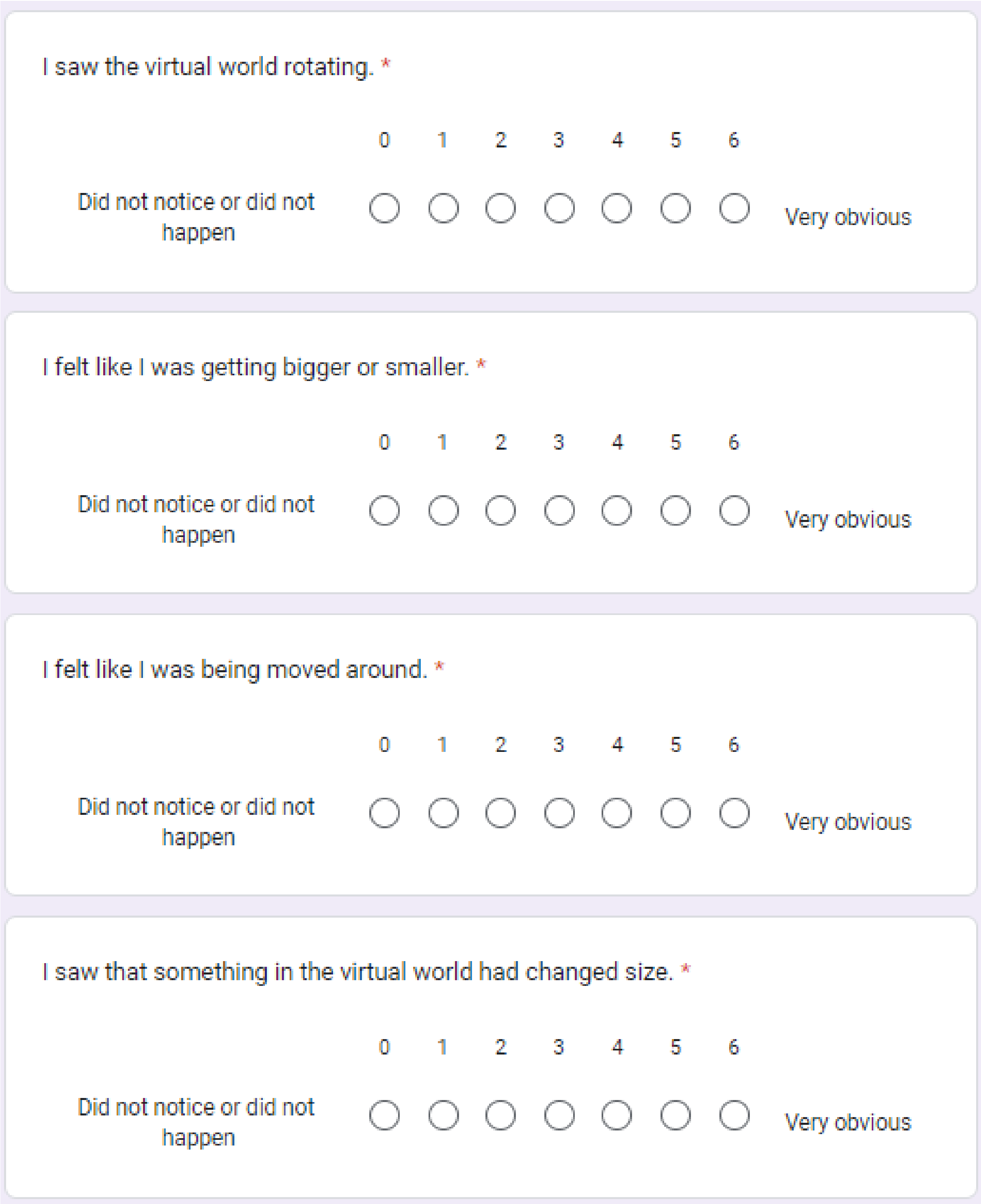
I saw the virtual world get brighter or dimmer. *

	0	1	2	3	4	5	6	
Did not notice or did not happen	○	○	○	○	○	○	○	Very obvious

I saw that something in the virtual world had moved. *

	0	1	2	3	4	5	6	
Did not notice or did not happen	○	○	○	○	○	○	○	Very obvious

Figure 18: Post-exposure questionnaire; question 1-5



I saw the virtual world rotating. *

	0	1	2	3	4	5	6	
Did not notice or did not happen	○	○	○	○	○	○	○	Very obvious

I felt like I was getting bigger or smaller. *

	0	1	2	3	4	5	6	
Did not notice or did not happen	○	○	○	○	○	○	○	Very obvious

I felt like I was being moved around. *

	0	1	2	3	4	5	6	
Did not notice or did not happen	○	○	○	○	○	○	○	Very obvious

I saw that something in the virtual world had changed size. *

	0	1	2	3	4	5	6	
Did not notice or did not happen	○	○	○	○	○	○	○	Very obvious

Figure 19: Post-exposure questionnaire; question 6-9

1. General discomfort *

- ○ None
- ○ Slight
- ○ Moderate
- ○ Severe

2. Fatigue *

- ○ None
- ○ Slight
- ○ Moderate
- ○ Severe

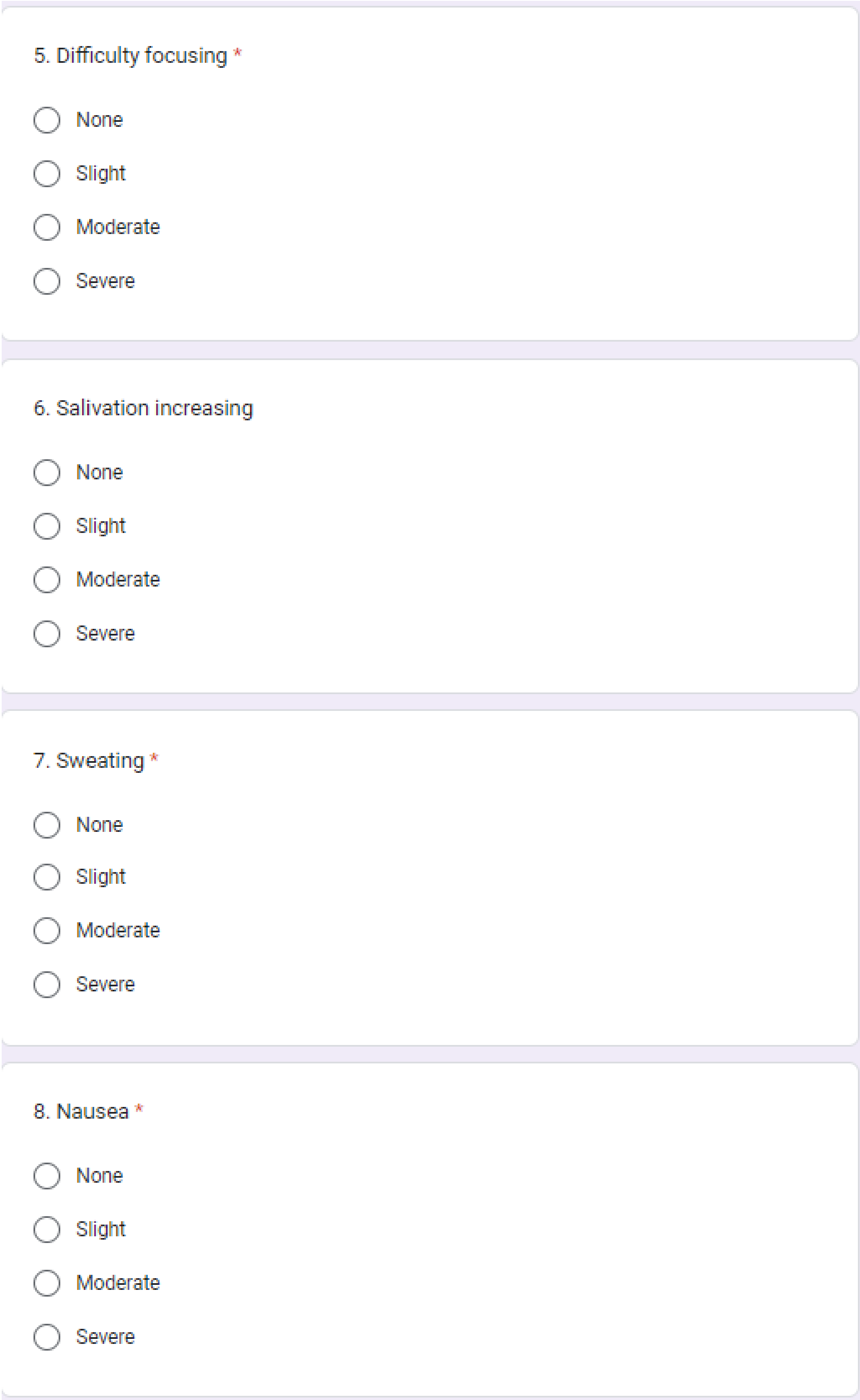
3. Headache *

- ○ None
- ○ Slight
- ○ Moderate
- ○ Severe

4. Eye strain *

- ○ None
- ○ Slight
- ○ Moderate
- ○ Severe

Figure 20: SSQ; question 1-4



5. Difficulty focusing *

- None
- Slight
- Moderate
- Severe

6. Salivation increasing

- None
- Slight
- Moderate
- Severe

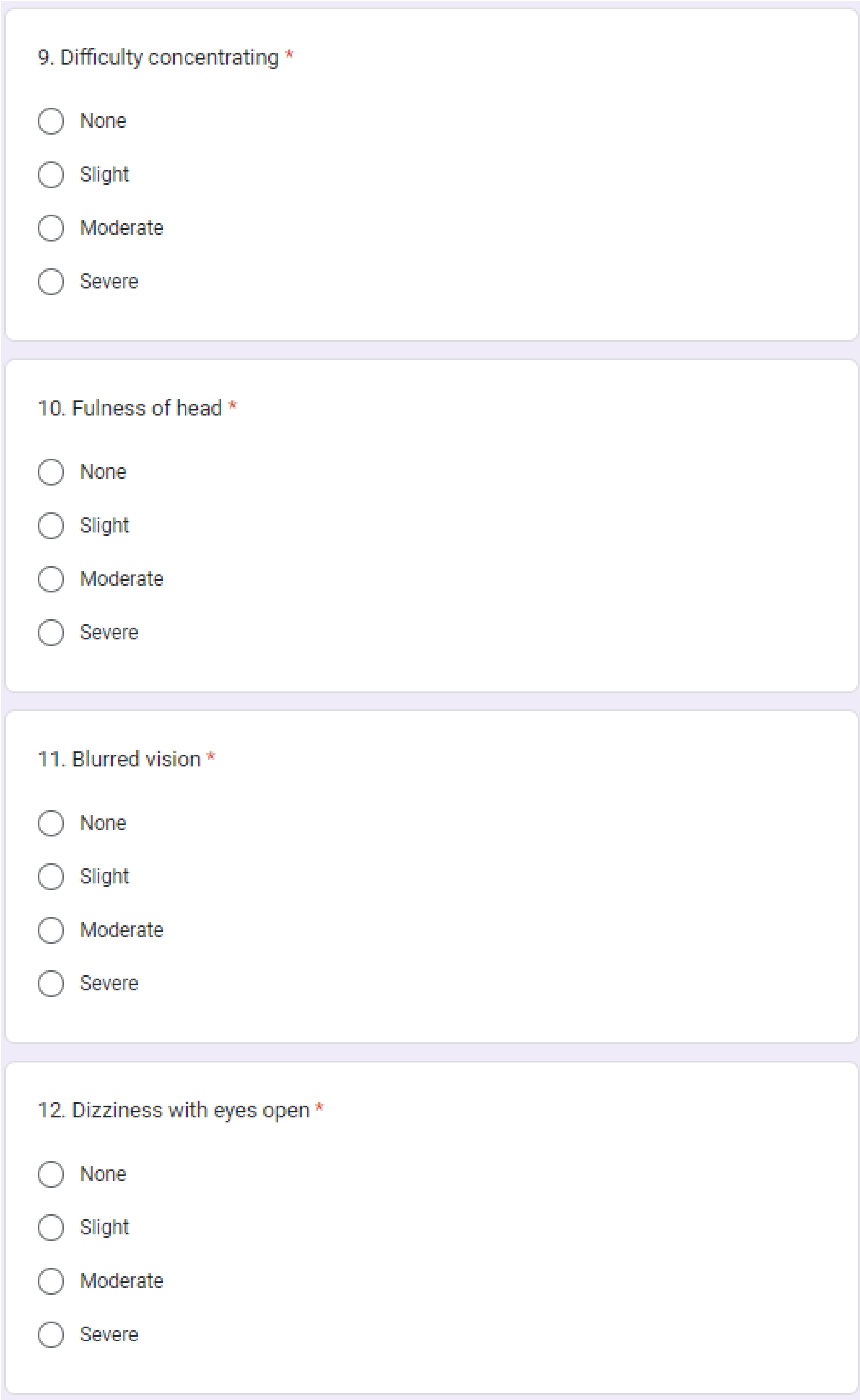
7. Sweating *

- None
- Slight
- Moderate
- Severe

8. Nausea *

- None
- Slight
- Moderate
- Severe

Figure 21: SSQ; question 5-8



9. Difficulty concentrating *

- None
- Slight
- Moderate
- Severe

10. Fulness of head *

- None
- Slight
- Moderate
- Severe

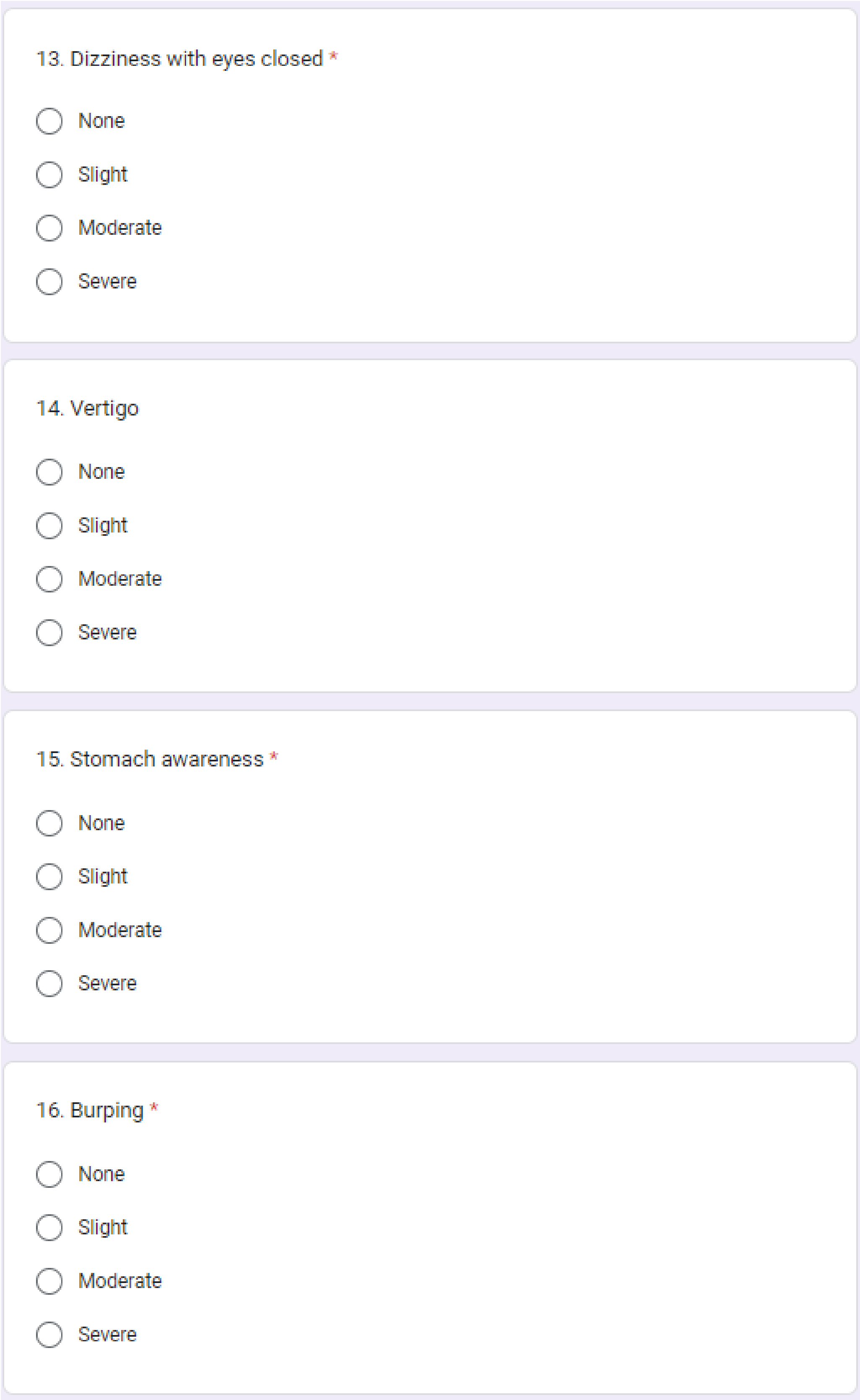
11. Blurred vision *

- None
- Slight
- Moderate
- Severe

12. Dizziness with eyes open *

- None
- Slight
- Moderate
- Severe

Figure 22: SSQ; question 9-12


13. Dizziness with eyes closed *

- None
- Slight
- Moderate
- Severe

14. Vertigo

- None
- Slight
- Moderate
- Severe

15. Stomach awareness *

- None
- Slight
- Moderate
- Severe

16. Burping *

- None
- Slight
- Moderate
- Severe

Figure 23: SSQ; question 13-16

C Experiment 2 results

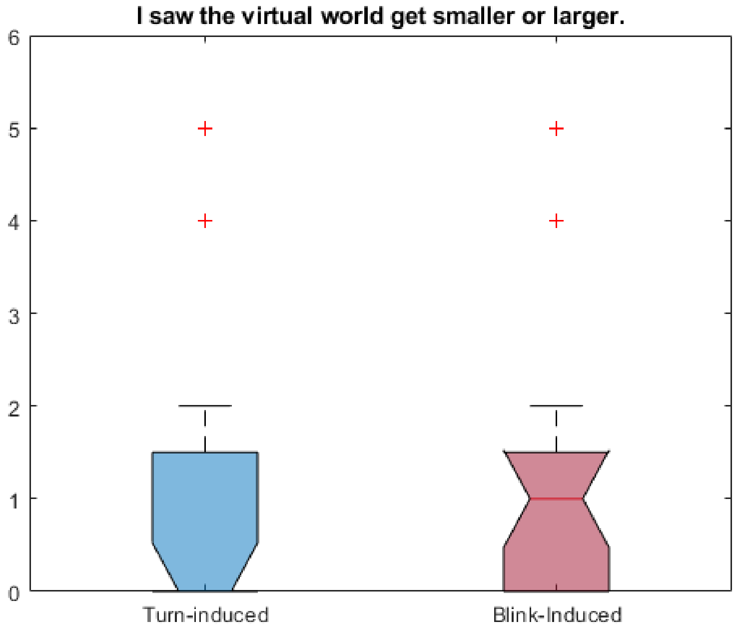


Figure 24: Post-exposure question 1

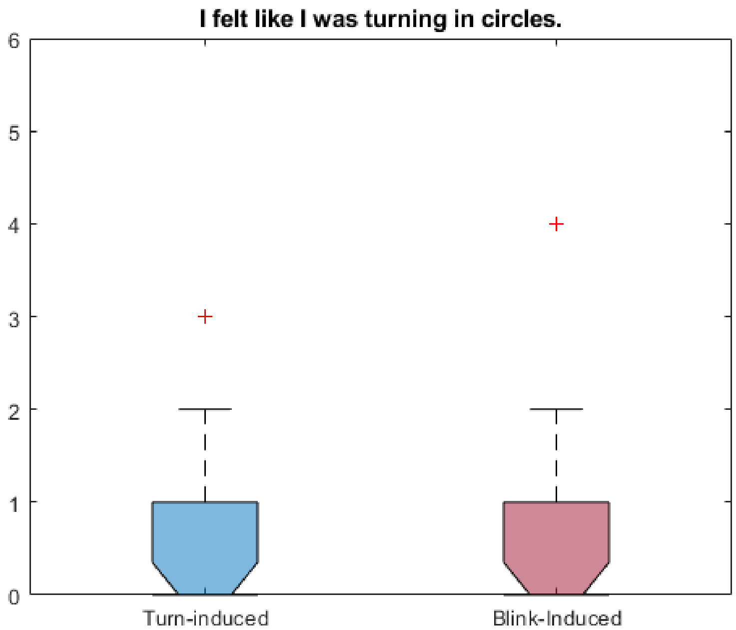


Figure 25: Post-exposure question 2

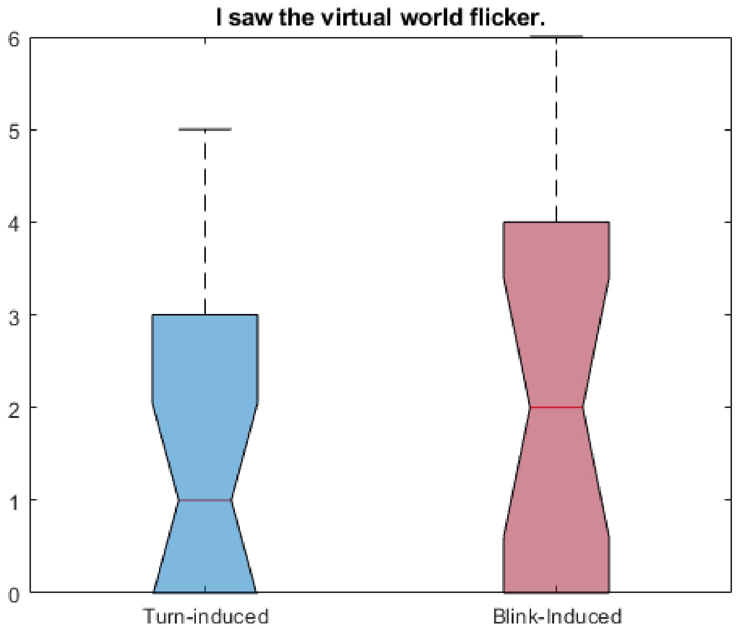


Figure 26: Post-exposure question 3

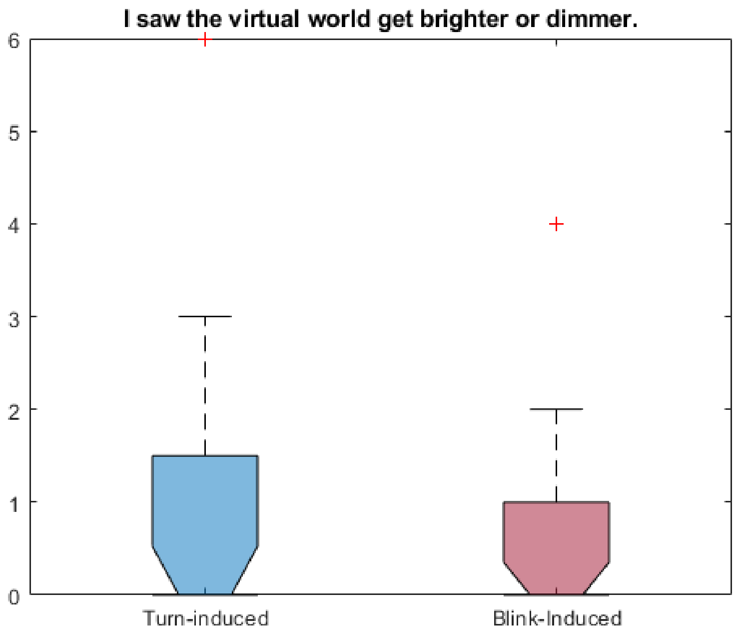


Figure 27: Post-exposure question 4

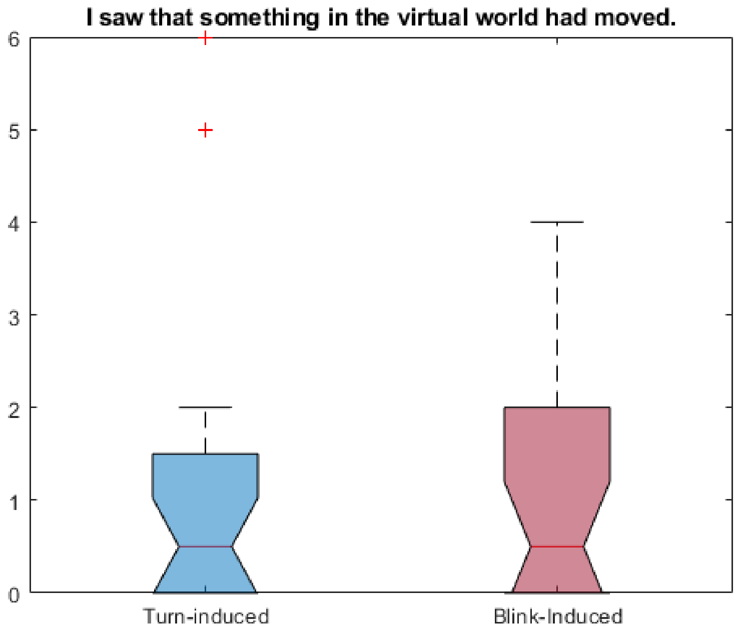


Figure 28: Post-exposure question 5

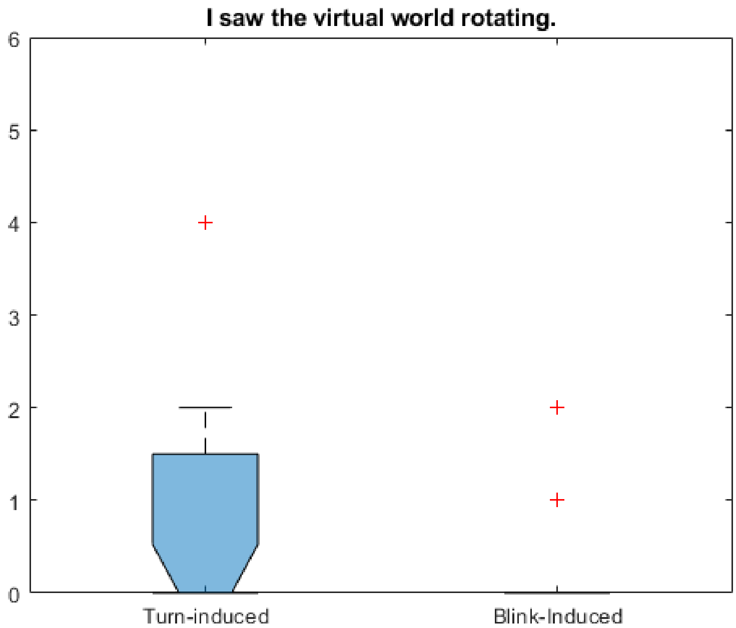


Figure 29: Post-exposure question 6

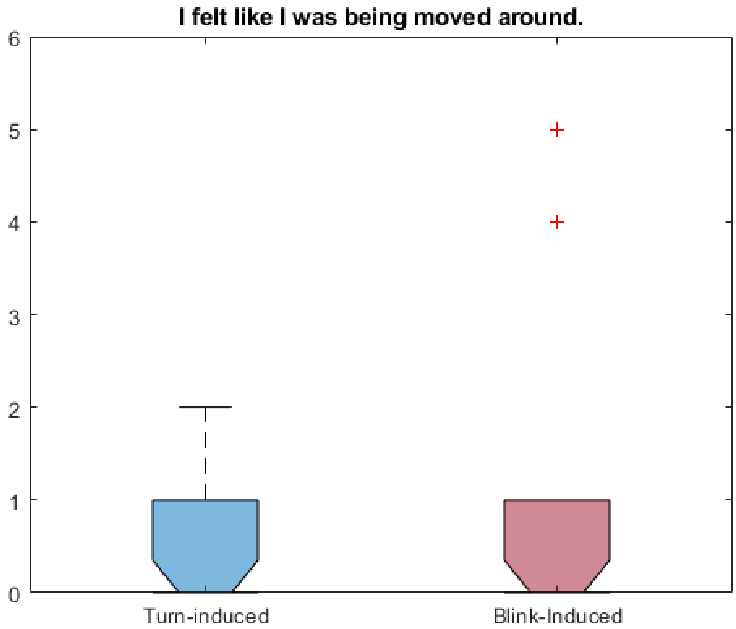


Figure 30: Post-exposure question 7

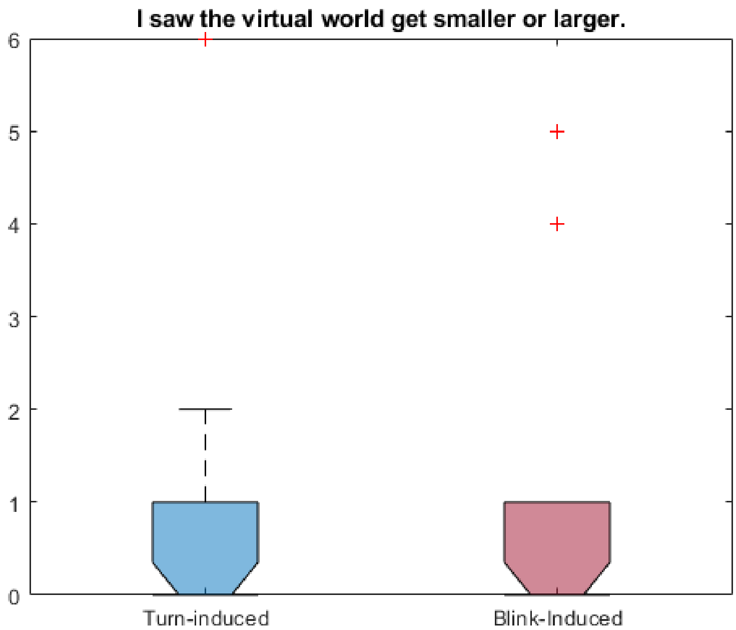


Figure 31: Post-exposure question 8

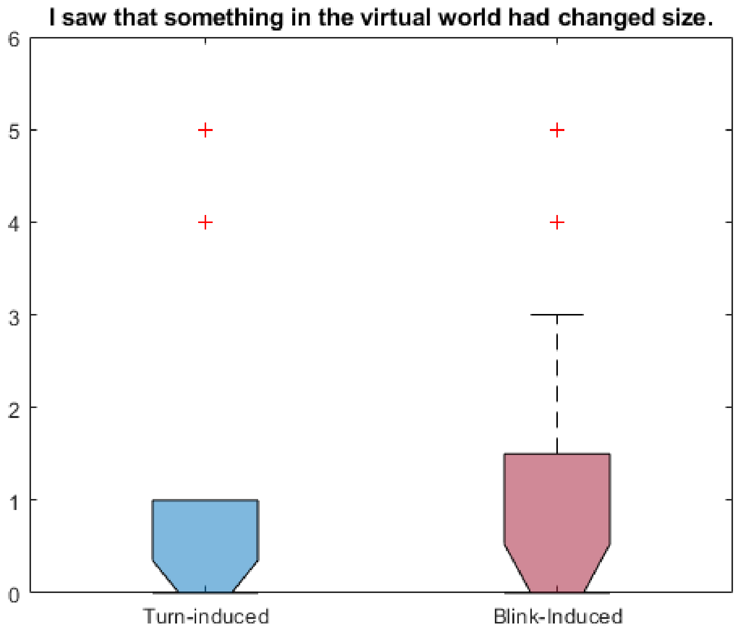


Figure 32: Post-exposure question 9

D Task screens from Experiment 2

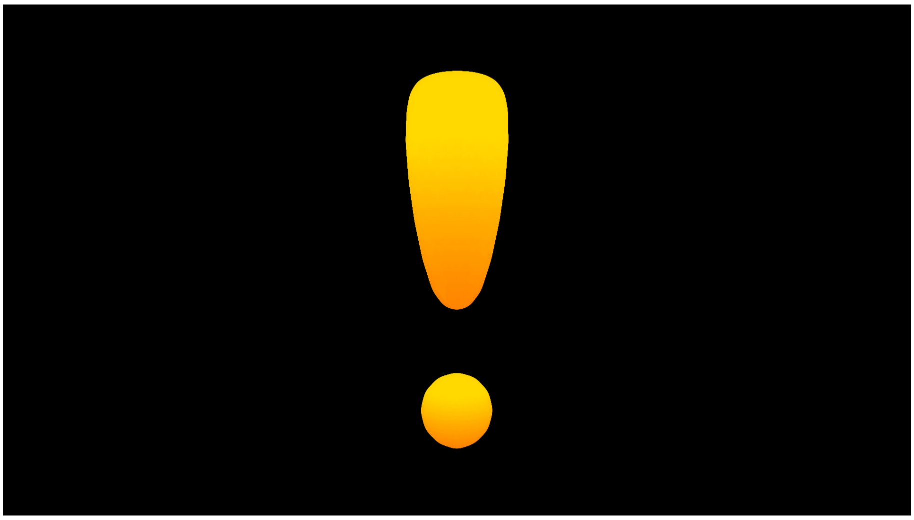


Figure 33: Task ready screen. Approaching this would change image to the appropriate task.

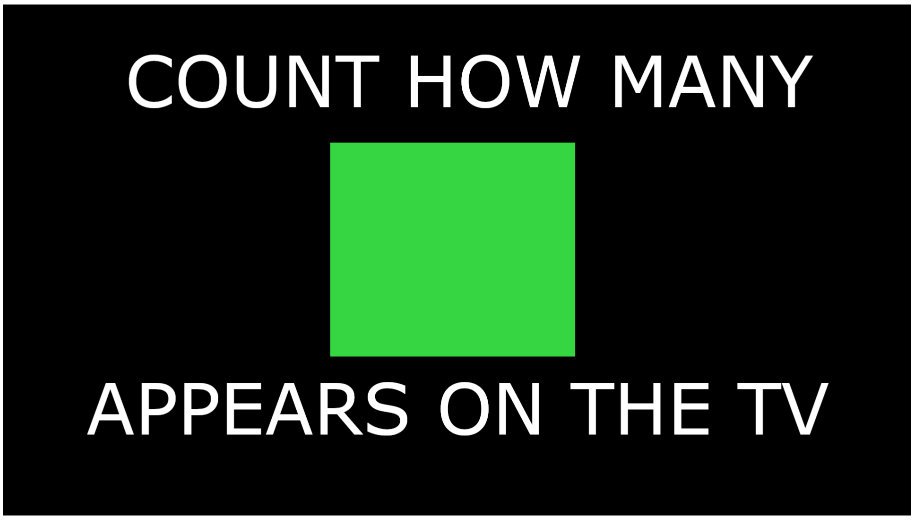


Figure 34: Task of room one. Made in GIMP.

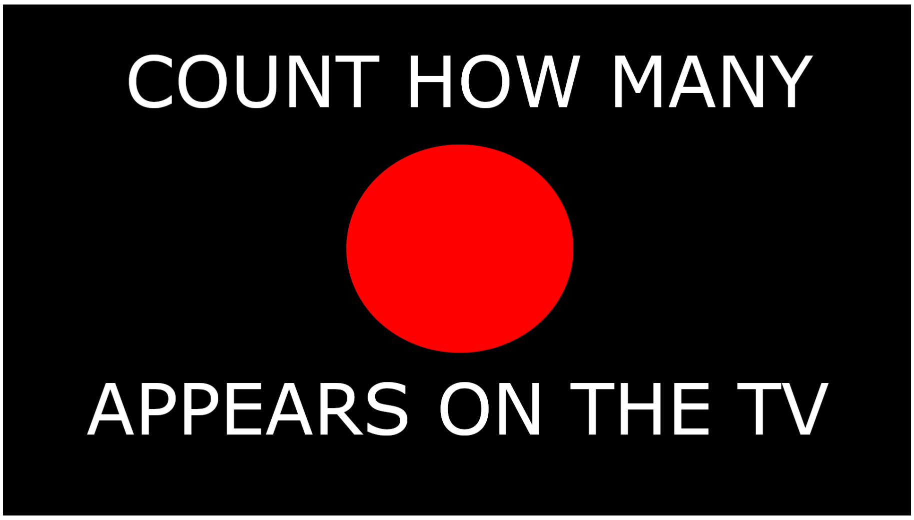


Figure 35: Task of room two. Made in GIMP.

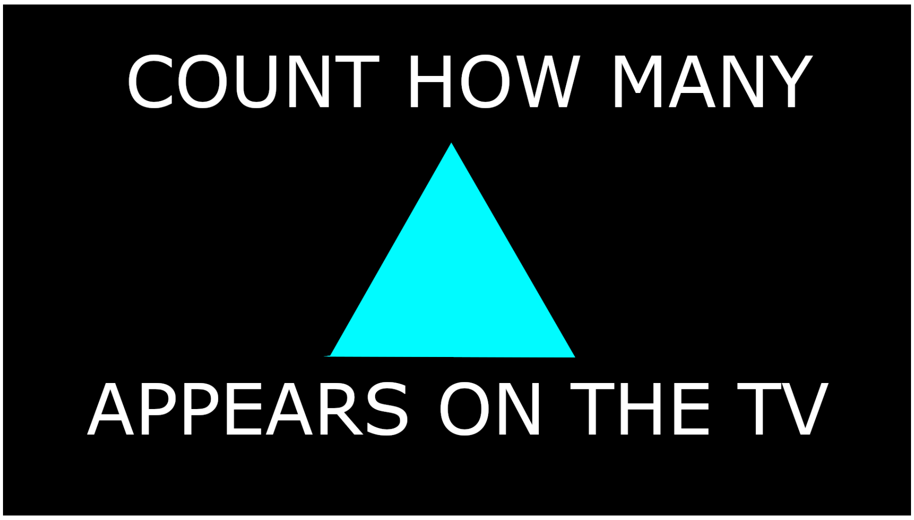


Figure 36: Task of room three. Made in GIMP.

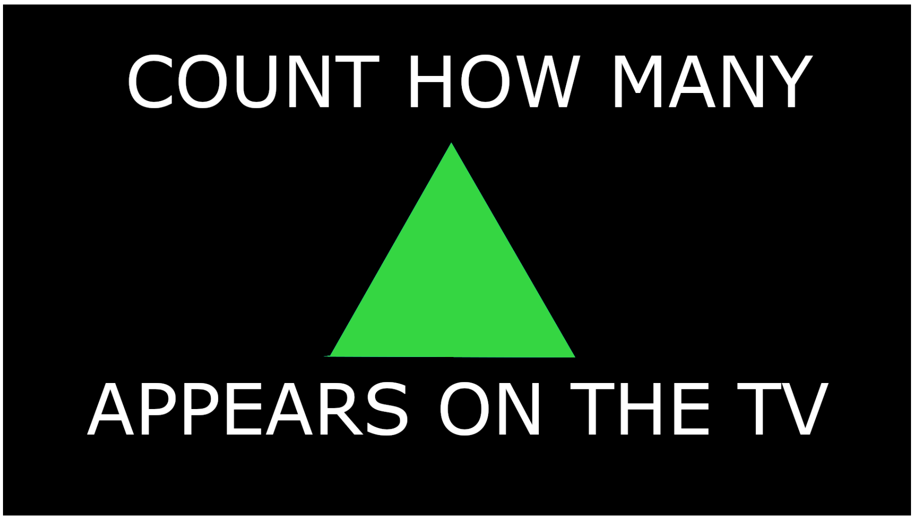


Figure 37: Task of room four. Made in GIMP.