TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Zain Ali Khan
IASM195988

Assessment of User Interfaces for Control System in an Immersive Virtual Reality Environment

Master's thesis

Supervisor: Aleksei Tepljakov, PhD
Department of Computer Systems, TalTech

Saleh Alsaleh, MSc
Department of Computer Systems, TalTech

TALLINN 2022

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Zain Ali Khan
IASM195988

Automaatjuhtimissüsteemide kasutajaliideste hindamine immersiivses virtuaalreaalsuse keskkonnas.

magistritöö

Supervisor:

Aleksei Tepljakov, PhD
Department of Computer Systems, TalTech

Saleh Alsaleh, MSc
Department of Computer Systems, TalTech

TALLINN 2022

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature, and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Zain Ali Khan

[03.01.2022]

TALLINN 2022

Abstract

The purpose of this thesis is to look at various applications of user interfaces (UI) in virtual reality to control system applications. The purpose is to evaluate available UI control systems in VR, which, after being integrated with a 3d simulation crane model in an unreal engine VR environment to test and identify its quality and thus simplify industrial adoption of such control systems primarily centered on eye-tracking as a new UI controller.

To attain the above-mentioned aim, the following experiments have been carried out:

• Conducting a literature review to identify potential gaps in the research area.

•Designing and implementing experiments with VR UI control systems including the modification of control parameters and the consequent modification of system behavior.

• A comparison of theoretical predictions and real findings is performed to ensure that system control is valid.

• Conduct a detailed study and comparison of the control systems employed in these research studies using SUS.

Keywords: Virtual Reality, VR Control Interface, Eye Tracking, Human Control Interaction, Computer Simulation,

This thesis is written in English and is [47] pages long, including [10] chapters, [16] figures, and [8] tables.

TALLINN 2022

List of abbreviations and terms

VR	Virtual Reality
UI	User Interface
DI	Digital Twin
DES	Discrete Event Simulation
VRE	Virtual Reality Environment
HMD	Head Mounted Display
EDT	Experientable Digital Twin
EDTs	Experientable Digital Twins
SteamVR	Steam Virtual Reality
EC	Eye Gaze Controller for HTC Vive Pro Headset
HC	Handheld Controller for HTC Vive Pro Headset
EG	Eye Gate
EGOff	Eye Gate Off
EGOn	Eye Gate On
SUS	System Usability Score

Table of Contents

TALLINN 2022

List of figures

List of Tables

1. Introduction

As the whole industry is going through the industry 4.0 revolution, using old analog or traditional controller-based systems into digitally advanced systems is becoming more attractive and easier to integrate with current usage than applying or building whole systems from scratch, avoiding expensive investments. With the current epidemic situation, solutions that enhance automation and remote work possibilities have become necessities, giving the working environment a new perspective and functional approach. The gaming industry plays a vital role in covering and maturing user interface-based controllers, providing high performance, displays, and graphic card utilization with computer hardware environments. which helps smart manufacturing and using such innovations to be applied to industrial usage. As a result of these advancements, Virtual Reality has become a highly important and more accessible research tool.

Along with these aspects, assessing gaze control-based interface over handheld control system in VR, the purpose of this research paper is to propose the use of eye gaze-based control interface as assist for disabled person opportunities in the industry.

This data presented in this research paper can help researchers develop and integrate Eye Gaze User Interface controls for VR systems, which can help provide jobs for disabled people as one potential application.

Virtual reality offers the benefit of allowing the participant to move freely and in a very natural setting while still allowing for a well-controlled experiment setup. The person may look in all directions simply moving his or her head, much like in real life. The location of the objects in respect to the subject's position may be assessed precisely. while the subject's head moves.

In practice, whole-body movements like turning towards things or even walking may be used. High immersion of the subject with the virtual world may be achieved by the synchronization of body motions and pictures supplied to the eyes.

In this research, we investigate and describe the results of using VR Eye-tracking as a controller for an existing 3D crane model in VR simulation and analyzing its usefulness by comparing it to a handheld controller by completing a variety of trials with distinct data sets.

VR and its various forms:

Virtual Reality

Virtual reality is defined as the use of computer technology to create an interactive three-dimensional world in which things have a feeling of physical presence. This can be further explained in terms of Virtual Reality Environment which can be described as a wide range of computer-based programs with immersive, highly visual 3D qualities that let the user gaze about and navigate inside a supposedly real or physical world. It is typically defined by the technology utilized, such as head-mounted displays, stereo capabilities, input devices, and the number of sensory systems engaged [1]. Immersive experiences include all depending on the type of content made, it might be based on only real-world content, purely computer-generated content, or both.

Augmented Reality

AR is a blend of the digital and physical worlds. It is a real-time overlay of computer-generated information on the actual world that may interact with it visually. A type of virtual reality in which synthetic stimuli are overlaid on real-world objects to make information that would otherwise be barely noticeable to human senses. It combines real and virtual objects in a real environment, runs interactively and in real-time, and registers (aligns) real and virtual objects with each other [15].

Extended Reality and Mixed Reality

In Extended Reality (XR), the physical and virtual worlds are combined using computer graphics and wearable technology. Computer-altered reality is referred to as XR, which includes Augmented Reality (AR), Mixed Reality (MR), and Virtual Reality (VR).

MR is a subset of AR. Mixed reality can be defined as a more powerful form of AR or as a hybrid of AR and VR (potentially bound to specific hardware) [16].

VR is an evolving consumer market technology that will provide many fresh opportunities and development for research [34]. This presents a laboratory atmosphere with a high degree of Immersion and alignment nearby. An experiment that uses VR takes place in a highly supervised setting that facilitates the collection of a more comprehensive amount of information. While many VR apps currently allow users to 'pull' items at a distance by pointing to and activating a catch, eye-tracking may make it easier and more precise, enabling users to simply look and steer. Using eye-tracking for this purpose will potentially be much more effective, and our eyes are much better at pointing to distant targets than using a laser pointer from our fingertips. This method has significant potential for improving or providing a viable alternative for the user experience in a VR environment.

2. Background

2.1 DT in VR technologies

This research [5] gives a better definition for this title as Digital twins are digital representations of genuine items or subjects, complete with data, functions, and communication capabilities. They enable networking and, as a result, the automation of complex value-added chains as nodes on the internet of things. The use of simulation methods brings digital twins to life and allows them to be experimented with; digital twins become experientable digital twins (EDTs). Initially, these EDTs only communicate with one another in the virtual environment.

2.2 3D-Crane Model

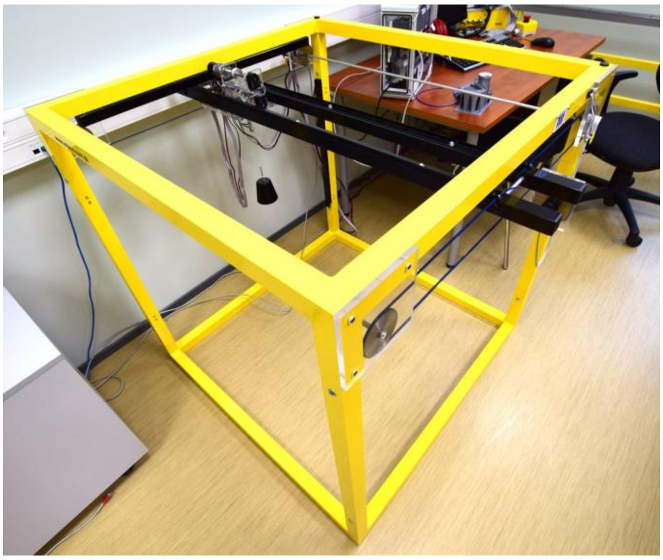


Figure 1 A laboratory-scale model of a gantry crane used as the basis for the digital twin. The crane has 3 degrees of freedom and can be used for control experiments in several ways including conducting laboratory experiments in university courses related to control systems.

The three-dimensional industrial crane model is a highly nonlinear MIMO system outfitted with a specific sensor system and a unique 2D angle measurement unit. The system is completely integrated with MATLAB/Simulink and runs in real-time.

The crane model was developed for purpose of learning and teaching students to understand controls systems more interactively. Hands-on experiments with control objects are extremely beneficial in teaching system theory and control system design; consequently, as a proof of concept, a digital twin of a 3D crane was developed, and students were able to interact with this twin via a VR application [7]. (Existing 3D-Crane controls) Math Works created MATLAB, a proprietary multi-paradigm programming language, and a numeric computing environment. The current 3D Crane system is completely integrated with MATLAB/Simulink and runs in real-time.

Based on the physical model simulation in Unreal Engine 4 has been made and was used in this research. 3D- crane simulation is designed using the following technologies.

Hardware:

HTC Vive Pro Eye 2

This VR headset is a virtual reality headset that allows you to see the world through the eyes of Track and interpret eye movements to allow lifelike interactions, better control GPU burden, and simplify input and navigation in a VR headset with eye-tracking features. It features sensors that are monitored by base stations[23].

Computer System

A desktop station required to run all hardware and software's was used with the following specifications:

Processor: Intel i7

Ram: 32GB DDR4

GPU: Nvidia GT1080ti

Software:

MATLAB

The MATLAB language, which is intended for scientific computing involving matrix-based computations ("MATLAB" actually expands into "MATrix LABoratory")., is the most intuitive way to represent computer mathematics in the world. Data may be seen and analyzed using built-in graphics. Experimentation, exploration, and discovery are encouraged in the desktop environment. All these MATLAB tools and capabilities have been thoroughly tested and are designed to function together. MATLAB allows to run larger data sets and scale up to clusters and clouds for analyses. MATLAB code may be combined with different programming languages, allowing the deployment of algorithms and applications in online, corporate, and production systems[22].

Simulink

Simulink is a multidomain simulation and Model-Based Design block diagram environment. It allows for system-level design, simulation, automatic code generation, and continuous testing and verification of embedded systems. Simulink is a dynamic system modeling and simulation tool that includes a graphical editor, customizable block libraries, and solvers. It is compatible with MATLAB[22].

Simulink is a graphical programming environment for modeling, simulating, and analyzing multidomain dynamical systems that are based on MATLAB. The current 3D Crane system is completely integrated with MATLAB/Simulink and runs in real-time.

Unreal Engine 4 (UE4)

Unreal Engine is the most open and powerful real-time 3D development tool in the world. It is constantly expanding to fulfill more than just its original function as a cutting-edge gaming engine, and it now provides artists across industries with the freedom and power to generate cutting-edge content, interactive experiences, and immersive virtual worlds [5]. The program will be used to create UI and interact with Simulink to control a 3D crane using a VR headset.

It is a highly strong framework that may provide excellent outcomes[6]. Many people have said, and I can now concur, that it has a very short learning curve and that it takes a long time to have a strong foundational understanding of the framework.

Using the Unreal Engine game engine for Integrating Assembly Process Design with VR-based Evaluation gives an abstraction layer and flexibility in terms of functionality, graphic fidelity, and VR systems. The designed workflow intends to lower the barrier of entry for industrial applications so that it may be easily integrated into current manufacturing processes[13].

The simulation environment of 3D-crane runs on unreal engine and MATLAB simultaneously. All physical models and functionalities of the 3D crane are defined in an unreal engine blueprint as near to the physical model of the crane.

Steam VR

VR content can be experienced on any device with SteamVR[20]. SteamVR is compatible with the Valve Index, HTC Vive, Oculus Rift, Windows Mixed Reality, among other headsets. SteamVR is Valve's virtual reality platform that works with Steam. SteamVR offers full-room, 360-degree VR experiences. The device was launched on March 1, 2015, at Mobile World Congress. SteamVR's first device, the HTC Vive, is like the Oculus Rift, created by Valve and HTC. In contrast to Oculus, SteamVR offers a holodeck experience that allows users to freely position themselves within a limited space. Additionally, SteamVR supports alternative HMDs such as the Rift, as well as its HMDs such as the Vive [20][21].

OBS Studio

It’s a Video recording and live streaming software that is a free and open-source program in real-time [27], which was used to document experiment procedures for this current research paper analysis. These documented recordings’ purpose was to make them useful and help in future projects for researchers. And to have a better understanding or revisit the experiments.

2.3 3d-Crane motion controller

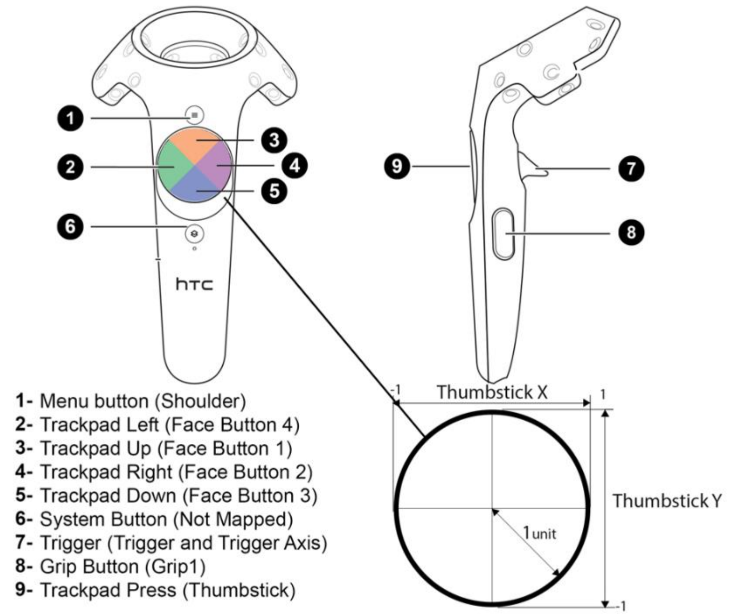

1
2
3
4
5
6
7
8
9
htc
1- Menu button (Shoulder)
2- Trackpad Left (Face Button 4)
3- Trackpad Up (Face Button 1)
4- Trackpad Right (Face Button 2)
5- Trackpad Down (Face Button 3)
6- System Button (Not Mapped)
7- Trigger (Trigger and Trigger Axis)
8- Grip Button (Grip1)
9- Trackpad Press (Thumbstick)
-1
Thumbstick X
1
Thumbstick Y
1 unit
-1

Figure 2 A depiction of the HTC Vive wireless handheld controller through which the user interacts with the virtual reality environment; provided are also functions of the buttons present on the controller.

The controllers used in the experiment are HTC Vive Pro 2 headset standard controllers. The picture illustrates all controls that can be used as default and mapped according to requirement. The existing mapped controls were used for aiming and triggering (which refers to the execution of command when aimed at desired target).

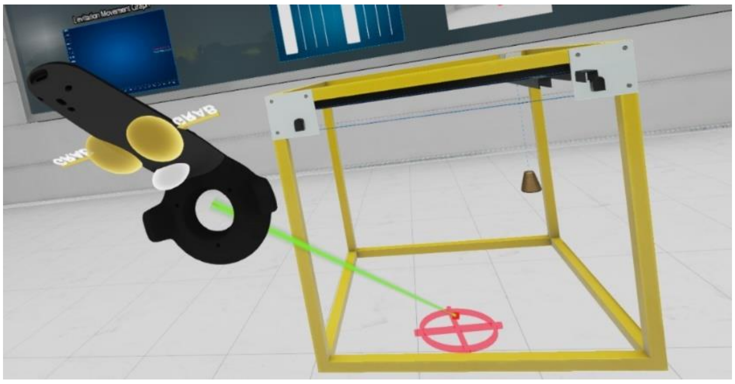


Figure 3 A depiction of the simulation environment in VR with the digital twin of the handheld Vive controller (left) and the 3D crane (right).

Figure 3 above can be seen which shows the hand controller being used in the experiment for aiming and targeting the crane load to specific points in the simulation environment. The picture is taken from a running simulation of a crane with a hand control interface without eye gate assistance being used. When Key 7 (trigger) as shown in (Fig 2) is squeezed the ray of green pointing beam is visible to participants to help and guide them to the desired position where a load of crane can be moved.

2.4 Eye tracking controller

An eye-tracking or gaze projection system monitors eye movements and continually assesses the direction of a subject’s attention. This feature of eye-tracking systems allows them to be used in applications such as mental health, gaming, marketing, vehicle safety, and human-computer interaction[14].
These systems include a camera for recording eye pictures as well as light sources to illuminate the eyes. This research focuses on head-mounted systems that are developed for extended reality systems. Head-mounted systems bring the camera and light sources closer to the user's eyes [14].

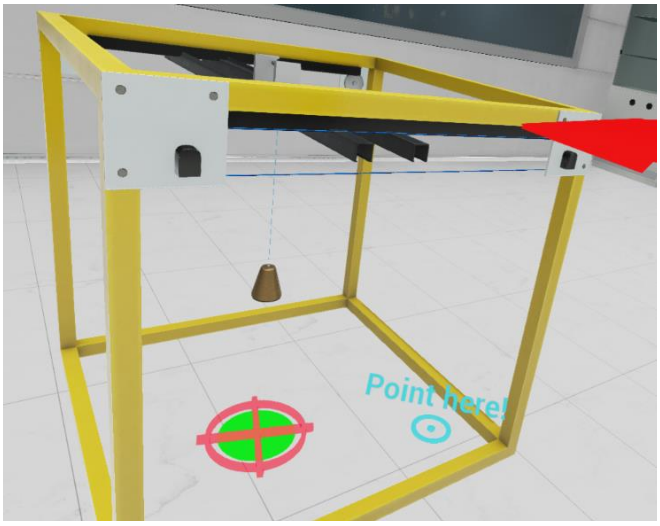


Figure 4 A depiction of Eye Control Interface being used in virtual reality environment.

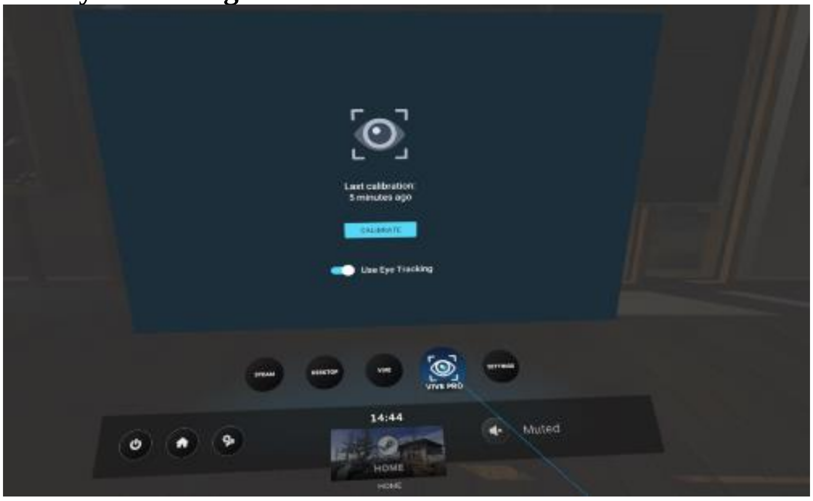
The above figure shows interface control in eye gaze control mode, which was used in Experimentation for this research paper, as it can be seen from figure 3 and figure 4 the difference, there is no aiming beam to point on the target, by moving gaze direction the aim for crane load maneuvers entirely, but still, to execute the trigger command, hand controller trigger was used.

Making it 100 percent eye gaze control-based control can be a part of future work along with the data of this experiment set.

The article [5] presents data that demonstrates the efficiency of the technology and what kinds of outcomes may be produced when employing eye tracking in VR. It aims to explore the process of incorporating VR and eye-tracking into the lab. VR presents certain hurdles to the researcher in terms of natural movement and motion sickness, for which no ideal answer has yet been discovered. Despite this, we predict that solutions to these challenges will be identified shortly because of the rapid advances in the field of VR technology. In the end, the benefits of using VR exceeded the drawbacks. It has the potential to be very useful in solving more issues regarding human cognition and behavior.

2.5 Identified gaps

VR-Eye-Tracking Calibration Failure



Last calibration:
5 minutes ago

Use Eye Tracking

Figure 5 A depiction of Eye Tracker Calibration during HTC Vive Pro Headset calibration is being set in the beginning.

As thoroughly discussed in [3]Calibration and validation are standard techniques used to assure the accuracy of eye tracking readings. The participant has presented target points in various positions on the screen and was asked to focus on them. The eye-tracker utilizes these targets as reference points during calibration to alter its estimation of the gaze to match the position where the patient is gazing. The visual angle between the coordinates given by the eye-tracker and the actual coordinate of the target is determined during validation and serves as a quality assessment of the calibration.

Calibration and synchronization are common techniques to ensure accuracy in eye-tracking measurement. Slight movement of the headset can de-calibrate and result in connection break or wrong data input as the precision of eye-tracking slowly deteriorates due to drifts, As the precision of eye-tracking deteriorates over time owing to movement.

Gaze Accuracy Error

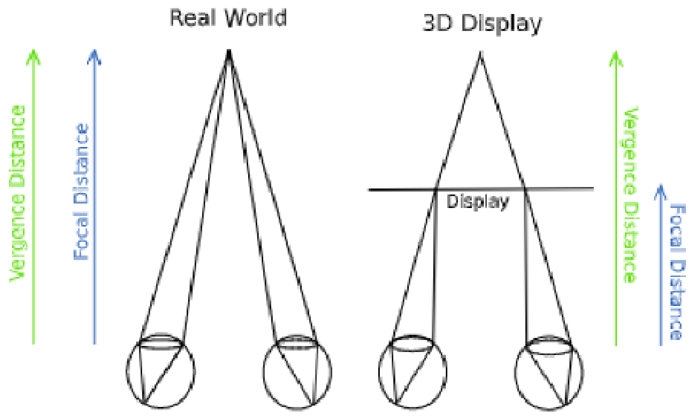


Figure 6 A depiction of Eye Gaze line projection comparison showing Real World vs 3D display projection in HMD :(left) real world: vergence = focal distance. (right) Focal distance always stays the same in VR [12].

Another known problem for eye movement and focus on VR is known as vergence accommodation conflict. In the real world, our brain receives depth information from vergence and the focus of the lens of the eyes. In contrast, we only receive information about vergence in VR. Since the virtual scene is only presented on a single plane at a fixed distance, namely the HMD display, one cannot extract any depth information out of the focus of the lens. The additional lack of focus blur can lead to a different perception of the size and distance of objects in the virtual environment [12] [7].

This research identifies a potential source of visual stress that has not before been examined in the context of VR systems: an insufficient vertical gaze angle. It was investigated to what degree variations in vertical gaze angle influence the demands put on the vergence eye movement system. The findings revealed that visual stress may be influenced in part by vertical gaze angle. Because of the close distance of the display within an HMD, a VR headset should be at the proper vertical position for every particular user. Because the display panels in an HMD are so close to the eyes, relatively slight changes in headset position result in substantial changes in gaze angle. This process might explain the complexities of past study findings and has implications for those who design HMD[24]s.

2.6Goals/Aims

The use of VR control signifies to help with rapid prototyping, furthermore, virtual reality-based previews and assessments eliminate the requirement for physical workstation prototypes, allowing for faster feedback and evaluation as well as early customer integration[13].

The present thesis aims to assess, analyze and compare existing user interface control systems as; motion controllers and new eye gaze-based controllers developed in Unreal Engine 4. Furthermore, we perform experiments based on these setups which will be explained later in a paper in the experiments design and implementation section, collecting data and doing a thorough analysis on both controller's performance and results. Additionally, we collect feedback from participants and have their experience be considered to conclude the outcome of experimentations. Collecting data to check how smooth and accurate experience participants had used both controllers, also if they were able to hit the target as close as possible factors that affect the usability of the controller. Identify them and after identifying checking how much they fact the proposed which is start it's provided to the users to use the controller and try to hit the target using both controllers motion controller and eye gaze controller and conclude after experimenting and taking feedback from me a significant number of users can make a decision based on the collected data.

We identified a significant number of users' responses and checked how the usability for proposed control systems was scored. Taking SUS mean scores and comparison to make results meaningful and come to conclusion.

We also explore a proof of concept for using eye gaze-based controllers in comparison to handheld controllers in VRE control systems, which can be used in other VR DT simulation systems and enable disabled people's livelihood in the revolutionized 4.0 autonomous smart manufacturing industry.

For this work, the null hypothesis is that there is no significant difference in system usability and user experience while using hand control versus while using eye gaze control whether eye-gate is used or not [Hand Control with Eye Gate On/Off vs Eye Control with Eye Gate On/Off == no difference]. While the alternative hypothesis is that there's a significant difference in system usability and user experience either when using hand control with eye-gate on or off or when using eye gaze control with eye-gate on or off.

2.7 Outline of Thesis

This study article is divided into six separate chapters, with the following material stated for each of them:

First and foremost, Chapter One describes the aim of this research study, as well as all key technologies, their definitions, and their function in this research. Moving on to the Second Chapter, it contains earlier work connected to this study or comparable studies whose aims correspond to or are related to the methodology of this research paper in some way and are significant to this thesis research paper. The third chapter explains the methodology utilized in this research work, as well as data gathering approaches based on that basis analysis in Chapter Fifth, identifying all desired outcomes and extensive comparisons. The experiment design, implementation, and feedback procedure are all detailed in detail in the fourth chapter. Whereas in the sixth chapter, data visualization was described, graphically collecting all findings.

And in Chapter for conclusion for this master's thesis study paper has been elaborated with all results and future efforts. Document all material in the Appendix section.

3. State of the Art

3.1VR Control systems

The technology of virtual reality as it stands is an emerging technology on the consumer market that will provide a lab environment that is immersive and closely aligned with reality. VR’s close alignment with reality will offer many new opportunities for research[3].

These research papers indicate and build a strong case for this research paper aim to assess the control system interfaces in Virtual Reality. Making this assessment approach as first as has never been done before for 3D Crane Model simulation, provides a new dataset through conducted experimentations of overall 58 participants. This data set can be used by researchers for future work and make more stable, good quality user interfaces which can be a pivotal step in the future.

The potential, conditions, and prospects for using virtual environments as an empirical research instrument in the field of human-environment interaction are discussed in this article[14]. The paper showed, there were minimal statistically significant variations between assessments of the real and virtual building based on quantitative data; however, qualitative data analysis indicated variances about textures. User experience is critical in studies of human-environment interaction and should be considered when investigating 'naturalistic' human behavior in actual and virtual contexts. To understand and enhance the correlation between actual and virtual worlds, not only behavioral (e.g., navigation) and cognitive results, but also users' subjective experiences, must be evaluated. Then can virtual reality realize its full potential as a research tool, offering several opportunities for environmental and behavioral study, such as extremely accurate measurements and the examination of systematic building modifications in aspects of i.e., pre and post-evaluation?

The author of this[28] article researched to assess the cognitive burden encountered by an individual with ASD while driving in a VR-based driving system. In an investigation involving 12 teens with ASD, certain eye-gazing traits were observed that changed with cognitive load. Several machine learning algorithms were examined, and their ability to effectively measure cognitive load was confirmed using a therapist's subjective judgment. The findings will be utilized to develop models for an adaptive VR-based steering system that can detect an individual's real-time cognitive load and give driving activities at an acceptable degree of difficulty to optimize the user's protracted ability.

According to the findings, participants in this study had a greater blink rate and a larger pupil width when subjected to a high cognitive load. We found a comparable fluctuation in these characteristics among ASD youth. Other findings included lower focus time and increased saccade duration when driving for teens with ASD, as well as a higher cognitive burden. All six classification algorithms used in this article, Nave Bayes, SVM, Logistic Regression, KNN, Neural Network, and Decision Tree, have accuracies of more than 72 percent for binary cognitive load measurement utilizing eye gaze variables. With the KNN approach, the best accuracy was 78.38 percent.

This [16] paper denotes While the user is using a certain program, virtual–world coordinates, movements, and interactions are recorded and captured in real-time. Captured data is being examined to address the issue of augmenting VR apps with characteristics generated from real-time behavioral analysis. The results of the program aim to be utilized for real-time applications to adjust virtual environments depending on user experiences; however, more development efforts should be demonstrated to run models concurrently to feed the output of gathered information to the employed model. Furthermore, the provided approach may allow participants to be immersed in different scenarios and assist redirection strategies in VR-based applications analysis.

This article [17] discusses how to ensure that our approach is easily integrated into current industrial processes and enables quick outcomes, which facilitate rapid prototyping. Furthermore, virtual reality-based previews and assessments eliminate the requirement for physical workstation prototypes, allowing for faster feedback and evaluation as well as early customer integration. This illustrates how Using an Integrating Assembly Process Design and VR-based Evaluation with the Unreal Engine game engine gives an abstraction layer and flexibility in terms of features, graphic fidelity, and VR systems. The workflow was designed to lower the barrier of entry for industrial applications so that it may be utilized in existing production processes with minimal effort.

As for research [17], the argument for smart factory deployment of VR DES as a new platform for scenario testing and decision making is stated while highlighting future research objectives for this technology combo. It is stated that further research in the areas of lower latency image processing, DES delivery as a service, gesture recognition for VR DES interaction, and linkage of DES to real-time data streams and Big Data sets are required for VR DES to fully meet the visualization requirements of both Industry 4.0 and Industrial Internet visions of digital manufacturing. The autonomous enactment of scenarios on the manufacturing line is a feasible prospect shortly when combined with machine intelligence. It is clear from this research that interoperability with data providers and production line sensor streams will be critical in the future of VR DES; in this light, development in semantic technology and ontology with the goal of mapping sensor networks into a descriptive format readable by DES is critical. Standardization of communication protocols is still a work in progress in the field of semantics.

While this [18]work looks at VR as an experimental paradigm used to study gripping behavior in the real world. The researcher observed the participants' fingers and thumb as they picked up novel items displayed in different orientations and displayed the same physical behavior as their actual counterparts in a virtual reality environment. In VR and the analogous real-world setting, we compared grabbing behavior among and between individuals. This paper's research findings show the parallels and variations in grabbing behavior in actual and virtual worlds. The results of the Holding behavior, Portion of the object to be grasped, Amplitude of the grip, and Detection of the 3D finger position offered a deeper knowledge of human grasping behavior.

3.2Related Work

As in [4]research paper, the thorough findings revealed that, since the VR resurrection, the focus of VR locomotion research has been on VR technology and different technological elements, overshadowing the study of user experience. From the standpoint of interaction, the bulk of the used and researched VR locomotion systems was discovered to be focused on physical interaction, utilizing physical motion signals for navigation in VR settings. The suggested typology for VR locomotion, which introduces four unique VR locomotion types: motion-based, room scale-based, controller-based, and teleportation-based locomotion, is a significant addition to the literature review. As the outcomes were for the given literature analysis which offers researchers and developers a wealth of interaction-related information on the investigated VR locomotion systems, allowing them to base their future designs on sound theoretical knowledge. Apart from unifying the field and documenting the study on this important VR-related issue, this effort synthesizes new conceptual knowledge.

The proposed UI in [5] this work is based on eye-tracking. Eye-tracking-based UI not only reduces the amount of time the user spends directly manipulating the controller, but it also dramatically reduces the amount of time spent on simple operations, while also reducing the need for a dedicated controller by allowing multiple types of controllers to be used in combination. When compared to the standard VR controller-based technique, the suggested method significantly reduces consumption time. Furthermore, because fewer controllers are required than in previous approaches, it is conceivable to utilize a controller other than a dedicated controller in an environment such as a VR environment or an AR, allowing for the realization of a VR environment devoid of a typical controller-based VR UI. It is predicted that using the visual programming platform produced via this study, a non-professional programmer would be able to easily acquire the following effects through basic VR/AR experience.
The gap with this paper indicates further improvements needed for the usability and development of additional libraries.

For precision and accuracy of the timing systems of Unreal Engine, 4 and SteamVR in conjunction with the HTC Vive VR system were explored here [6]. Objective external assessments demonstrated that stimulus durations were extremely accurate in the first experiment. In contrast, a second experiment evaluating the precision of built-in timing techniques found very varied reaction time measurements and erroneous stimulus onset identification. With Unreal Engine and SteamVR, researchers built a novel software-based approach for precise and accurate response time measurements. Time collection was outsourced to a background program rather than using the regular timing techniques contained within Unreal Engine. Timing benchmarks indicated that the newly developed technology allowed for response time assessments with millisecond precision and accuracy. Aside from its amazing performance, it also allows you to make use of all the benefits that VR has to offer in terms of accuracy and precision. VR offers the ability to apply new paradigms including huge fields of view of realistic and three-dimensional settings when combined with precision research tools. It enables the integration of a wide range of behavioral reactions, boosting the ecological validity of neuroscientific investigations and perhaps leading to more generalizable and credible explanations for cognitive processes. Overall, the current findings show that the HTC Vive, in conjunction with Unreal Engine and SteamVR, can achieve high levels of precision and accuracy in both stimulus duration and crucial time measurements. The identified gap is discussed on the capacity to estimate the timing of stimulus events precisely and reliably, such as stimulus start and response periods, which is required not just for reaction time assessments. And for future is work it is deduced that instead, for the analysis of all time-sensitive functional measures, a precise recording of events inside an experiment is required.

Furthermore, studying from [7] elaborates that there is no single game engine that is superior for every purpose, and the comprehensive overview presented can assist users in selecting the best game engine for their purposes. Because it enables Visual Scripting and has a more complicated graphical interface, As a conclusion to this analysis and research as shown in this paper Unreal Engine 4 is better suited to experienced users; it has a steeper learning curve. This engine necessitates high-performance hardware, yet its visuals are outstanding.

According to research [8], Eye-tracking jobs need the detection of only one item - an iris in a certain location. As a result, it is rational to employ a neural network only for this purpose. However, there is a scarcity of relevant datasets for training the model. We offered a dataset appropriate for training bespoke models of convolutional neural networks for eye-tracking tasks in the study. Each user can separately pre-train the convolutional neural network models for eye-tracking tasks using data from the data set. This dataset comprises 10,000 annotated eye pictures with a 416 by 416-pixel extension. The locations and radius of the eye for each image are shown in the table with annotation information. This paper can be used as a guide for preparing datasets for eye-tracking devices. The output of this dataset was evaluated on its model of the convolutional neural network for training the neural network model's first layers. A personal dataset of 1000 photographs was utilized to train the final layers. As a consequence, the tracking error was three degrees, which is remarkable given that the tracking was done with a web camera.

Analyzing the paper [10], the terminology of Gamification in terms of Industrialization is understood by these statements as the use of game techniques and components into learning activities to boost motivation, learning rates, engagement, and so on. Models come to life and become active thanks to game engines. Various simulations become possible because of the features. The OpenXR standards lay the groundwork for communication across multiple providers to use a common language. Therefore, game engines and viewing devices may communicate with one another. Digital twins are manufactured more clearly to match the growth of Extended Reality. Extended Reality could bring divisive individuals together and allow them to work together by reducing miscommunication and preventing a lack of communication.

4. Methodology

4.1Experimental Data Collection

The practical method and stated aim serve as the foundation for this thesis. Analyzing both control methods for a 3D crane by conducting trials and analyzing the results from 58 participants.

Furthermore, a specific quantity of analysis for both control system usability has been incorporated and analyzed in this study report.

This study is centered on evaluating the control interfaces of a VR 3D crane simulation, as well as putting up an experiment environment for participants to test both control systems, one with a handheld controller and the other with an eye gaze interface controller. Finally, getting input from participants via SUS surveys. In addition, the findings of this testing technique will be analyzed.

As the experiment is immersive based on control systems on interfaces, using techniques such as SUS is very appealing to use for this research with strong credibility of System Scalability Scale as being cited over thousands of papers since 1986.

For finalizing the SUS results of some following research paper provided enough material to use and assessment of this paper experimentations as well:

The focus of this [29] thesis is to assist VR designers by evaluating the efficiency (job completion time and mistake rate) and usability of four different types of menus. This study reveals the findings of a study that used sus to calculate the usability of an interface. The usability of radial and linear menus of various widths was examined, and the findings from SUS indicate that the user experience is not much altered.

Another relevant SUS testing example can be found in this [30] publication, where researchers evaluated The VR system on students; usability ratings revealed student willingness, and in most cases, enthusiasm, to utilize this tool for skill practice. The findings of the user-reaction survey and observations indicated pupils who were thrilled and motivated to practice virtually via System Usability Scale. Hence, making choices to be used for this paper as well.

Our experiments required participants to express their usability of both control systems and provide feedback, to take usability data from participants some standardized technique is required which was fulfilled by using SUS to collect participants' feedback at the end of the experiment.

4.2Data analysis: the System Usability Scale approach

To evaluate the results of these experiments, SUS was chosen. Because it is simple to apply, it may also be a reliable and valid metric. With subjective to assessing visual usability, SUS utilizes to enhance data evaluation from other methods[24].

The System Usability Scale (SUS) is a popular standardized questionnaire for evaluating perceived usability. Google Scholar citations (as of March 13, 2018) revealed 5,664 citations for the publication that established the SUS (Brooke, 1996). The SUS includes ten five-point items with alternating positive and negative tones in its standard, most used version. It consists of 10 questions with a scale of 5 points from Strongly Agree to Strongly Disagree.

To determine the SUS score, add the score contributions from each item. The contribution of each item score will vary from 0 to 4. The score contribution for items 1,3,5,7, and 9 (all odd questions) equals the scale position minus 1. The contribution for items 2,4,6,8, and 10 (all even questions) is 5 minus the scale position. To get the total value of SU, multiply the sum of the scores by 2.5. SUS scores range from 0 to 100 [24] [25].

Using SUS only doesn't signify many results as to validate the outcomes from SUS scores for this paper, Sauro-Lewis CGS (Table 3) was used to give more guidance for interpreting and understanding SUS Means [24].

Table 1An detailed scale showing The Sauro-Lewis CGS

SUS Score Range	GRADE	Adjective Rating
84.1 - 100	A+	96 – 100
80.8 - 84.0	A	90 – 95
78.9 - 80.7	A-	85 – 89
77.2 - 78.8	B+	80 – 84
74.1 – 77.1	B	70 – 79
72.6 – 74.0	B-	65 – 69
71.1 – 72.5	C+	60 – 64
65.0 – 71.0	C	41 – 59
62.7 – 64.9	C-	35 – 40
51.7 – 62.6	D	15 – 34
0.0 – 51.6	F	0 - 14

The scale is set in comparison with SUS scores averages providing a scale and grade system on SUS means. Making A+ (84.1-100) most acceptable, anything below (51.7, D) is considered not acceptable.

The first factor to evaluate when evaluating a questionnaire is if it has valid content, that is, whether its items are relevant and reflective of what it is designed to measure. The extent to which questionnaire questions correspond with the underlying constructs of interest is important[24][25].

As a result, the questionnaire's phrasing was changed to match the experiment's language and to assist participants to relate to the questionnaire without jeopardizing the questionnaire's reliability[25].

5. Developed Experimental System

This section of the study will provide a full overview of experiment design and implementation, including how they were planned, carried out, and data was collected. As the research based on each topic has not been done before, so designing and conducting experiments for this research was a crucial factor in terms of data collection and analyzing the concept of the Eye-gaze-based interface for the 3D-Crane model. The experiment aimed to make participants involved in using 3d crane simulation and use its control systems with additional variable parameters mentioned below in Table [1].

5.1 Designing Experiment

The experiment is based on humans' participation based on the total number of 58 participants who took part in experiments and allowed their data to be used for this research.

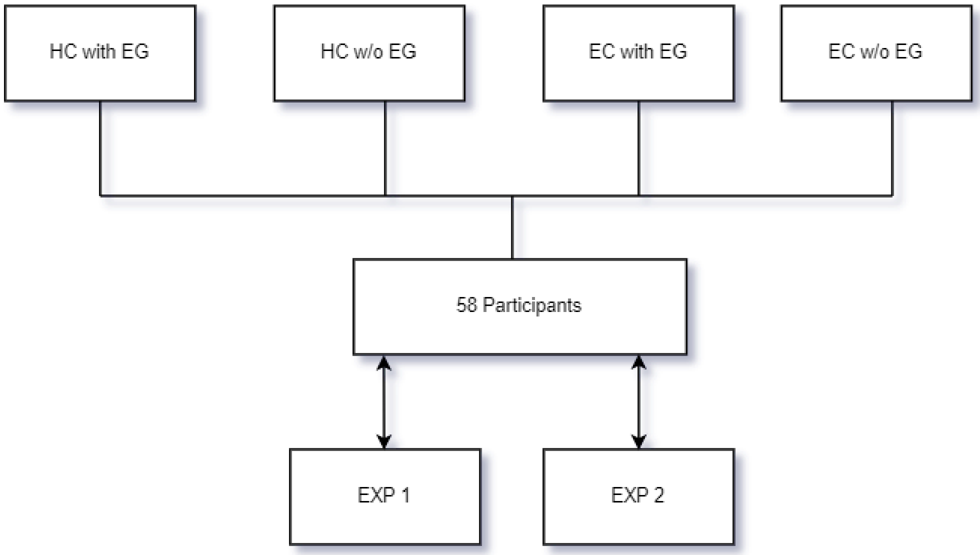


Figure 7 Flow Chart illustrating experimentation cases Hand Control (HC) with / without Eye Gate (EG); Eye Control (EC) with / without Eye Gate (EG) and groups for this research which was conducted and 58 participants

Participants were invited through advertising invite forms QR and sharing invitation links via emails and messages.

The experiment was based on two versions, where each participant took part in both. As 4 parameters were being used as listed in the table below:

Table 2 The Table lists all parameter variations (cases) used in the experiment and collecting data for this research paper.

Motion Controller	Eye Gaze Controller
Scale Normal	Scale Big
Sound On	Sound Off
Eye Gate Off	Eye Gate On

The parameters mentioned above were organized in the system according to the 4-bit description. The Detailed table can be seen in the Appendix section.

With these parameters being randomized according to a 4-bit systematic data truth table structure, each participant got a different experience making the data set to be as authentic as could be in each environment.

- Motion controller denotes crane controls benign used with handheld controllers for aiming and targeting for the load coordinates in workspace area in VR simulation environment. When key 7 [1.2] is squeezed slightly enabling guided aim beamline, on the desired position by squeezing furthermore key 7[1.2] to complete a click, will execute the trigger event and move the load to the aimed position.

- With the Eye gaze controller, the crane aim controls were being guided through eye-tracking control in VR headset and trigger to execute command for the load to move on aimed target was done by handheld controller key 7 [1.2] when pressed same as being used in motion controller.

- Eye Gate's purpose is to make an eye gaze controller more stable to reach the target point, its role was to assist with the eye gaze controller to make the accuracy more stable.

- Furthermore, the sounds, normal scale, and big scale represent the size of the crane which was a visual element of crane simulations. Which were part of the experiment and provided experience in 4 parameters as set to participants making their experience distinctive.

The experiment was set in two parts named Experiment 1 and Experiment 2 with a total number of 8 variables making 4 in each experiment which was being changed concerning participants' sequential increments.

To have a systematic flow of variable sets and making execution of controllers with identifiable UI, and python based interface was used as seen below in the picture. Enabling to keep track of all variables being used in the current experiment and tracking and keeping log files in the back end.

Beforehand to begin experiments, small introductory slides were shown to participants enabling them to have a better understanding of the experiment and getting them familiar with the device and VR experiment purpose.

The equipment being used for conducting these experiments is based on a VR headset by HTC named VIVE PRO 2. The whole interface is in a VR headset providing all necessary information to participants to execute the required task in the experiment.

5.2 Implementation

A specified room was set up with trackers and VR mapped experiment area for VR simulation to be used as required for the device to set up a usable and interactable environment once user in VR headset, allowing enough space to calibrate and map the virtual environment around the user.

Each trial protocol was followed. Well before participants arrived at the place where the studies were carried out, the following tasks had to be executed but rather ensured that they were fulfilled:

- Organize the experimentation work area
    1. Assure attendee protection by sanitizing and sterilizing surfaces encountered by the last participant.
    2. Prepared consent documents to be signed with the next ID number

- Prepare Introductory Presentation
    1. Reset Presentation to the first slide.

- Prepare VR Headset
    1. Set the headset and controllers to their initial positions
    2. Check that the trackers are properly placed and that there are no obstructions between the trackers and the experiment area.

- Run OBS
    1. Start OBS studio, and press record
    2. Confirm recording has been started /confirm directory files

- Execute MATLAB/Simulink program
    1. Run the control Simulink application.
    2. Press the run simulation key.

- Run the Python Program
    1. Set ID number for participant
    2. Confirm parameters for each experiment by pressing Ready
    3. Run Experiment One/ Experiment Two

- Prepare the Feedback Form (SUS)
    1. Set ID number for participant

Every participant was provided introductory slides as mentioned earlier after, VR default eye tracking calibration procedure was done for every user. After successful calibration according to Vive Pro Headset program experiments were initiated. Using a small interface form to track user id and experiments with their variable parameters, the procedure was held.

To begin set parameters once confirmed, MATLAB /Simulink was executed parallelly to an Unreal Engine 4 simulation file to enable controls in the simulation environment. Incrementing ID as the participant changed which by default according to the set program changed parameters to different states as mentioned above in Table 1.

From the beginning, once the participant initiates the eye calibration step, OBS was used to record participant screens to document the whole experiment from beginning till the end.

Each user experienced all possible parameters in shuffled in form of experiment 1 and experiment 2. Making participants chance to experience and interact in all possible controls set for experiments.

During the experiment if assistance needed was provided at participants’ request, otherwise, the main aim was to let participants experience themselves and explore the controls and environment as much as possible.

A maximum period of 10 minutes for each experiment was set. In total 20 minutes were designated for experiments and 10 minutes were set for the SUS questionnaire for feedback at the end, along with 5 minutes in beginning for introductory slides for participants making a total of 35 minutes time frame per participant. Total 58 participation occurred over a time of 3 weeks, making a total of 2,030 minutes of experimentation over 20 day period.

5.3 Feedback

After finishing the experiment some personal information was collected with consent, along with the SUS form at the end. Some general data and their descriptions are below:

The minimum participant's age was 18 and the oldest participant who took part in the experiment was above 50 years old.

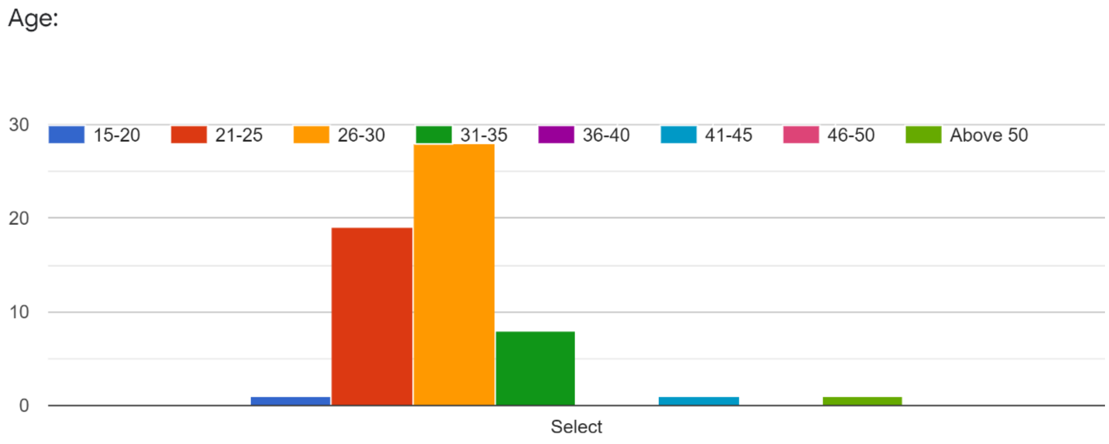


Figure 8 A depiction of the Age range for VR experiment participants for this paper, youngest (left) to eldest (right).

Participants came from various backgrounds, most of the participants were out of university, students, engineers from various backgrounds such as electrical engineers, geophysicists, mechanical engineers, finance specialists, and IT professionals took part in these experiments. The below table denotes specified number in each category and further description:

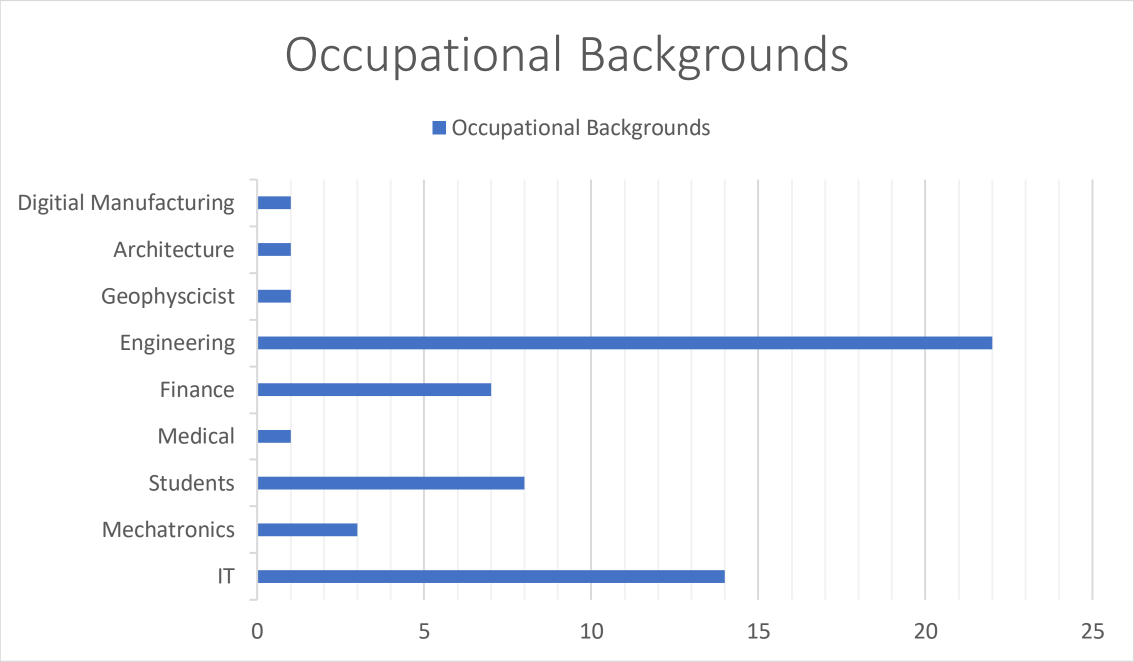


Figure 9 A depiction of occupation field backgrounds of participants from various industries.

Each participant was asked if they has prior experience with virtual reality before taking part in this experiments, which can be seen in [Figure 9] below:

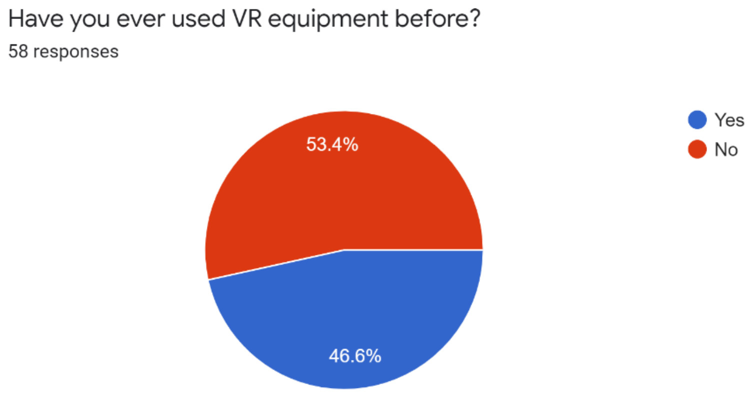


Figure 10 This is an illustration of the Virtual Reality ratio among participants who had previous experience with blue color and without with red color.

As for the pie-chart, more than half of the participants who came for experiments had no experience before making the result more distinct and the data set more valuable for researchers for future work.

After completion of both experiments, participants were asked to fill a SUS form to provide their experience. The collected data was used to do analyses in the next chapter

6. Analysis of Results

After processing all gathered data and implementation in MATLAB provided results at hand, which are further explained below:

6.1Assessment of Usability Scale Test

6.1.1 Hand Control VS Eye Control

Table 3The table shows SUS means score for Hand Controller and Eye Controller from 58 participant

Interface Controls	SUS Scores (0-100)
Hand Controller	77.53
Eye Controller	75.56

As it can be observed from the table 2 we can have our four cases, on which the experiments were based, and results denote SUS 0-100 scoring from participants feedback on the SUS questionnaire.

By comparing both HC and EC without eye gate the SUS scores are close with 1.97 difference. Overall Sauro-Lewis Grade of B+ for Hand Control (HC) and Grade B for Eye Control (EC) [Table 3].

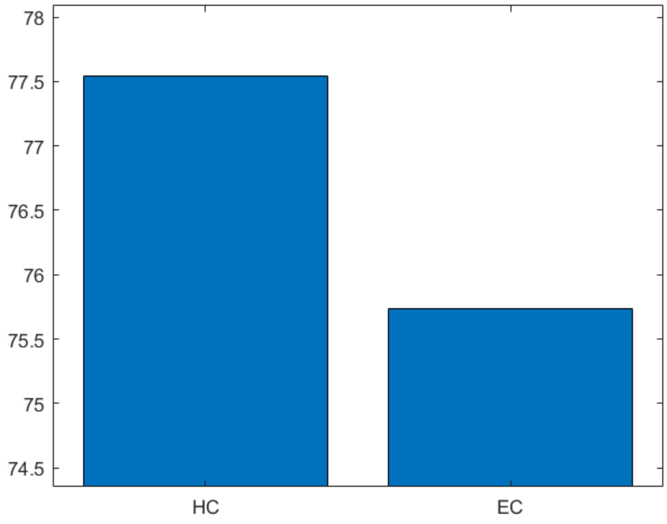


Figure 11 Figure 10 Motion Controller Graph comparing Hand Controller (HC) vs Eye Controller (EC)

So as conventionally using hand-based control systems is common as compared to Eye Controller, achieving such close results is interesting to see.

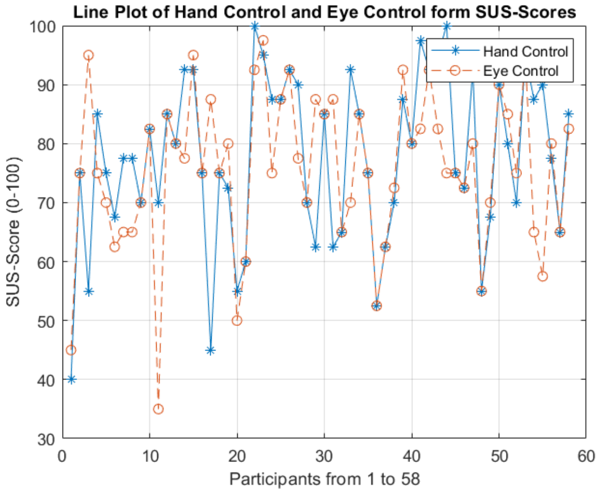


Figure 12 A depiction of Participants Experience in terms of scoring for Hand Control and Eye Control on bases on SUS- scoring.

Seeing this figure above shows interesting results, some participants had quite close experience with same control interfaces, but some really had an opposite experienced as can be seen the variance between red and blue nodes in figure respectively.

By comparing the participants results it can be seen it's almost the same. Near to half gave close results. But some are significantly different, this may be due to the fact that 53.4 percent of participants had never used Virtual Reality Head Mounted Display (VR HMD) before, so can be expected.

6.1.2Eye Control Interface with Eye Gate On VS Eye Gate Off

Table 4 The Table shows SUS Mean Scores for Eye Control Cases both On and off for Eye Gate

Eye Controller	SUS Scores (0-100)
Eye Gate On	78.44
Eye Gate Off	76.63

Moving ahead with Eye gaze-based control interface using Eye gate SUS score of 78.44 as without the Eye Gate assist earlier SUS-Score of 76.63, having a difference of 1.81; Sauro-Lewis Grade of B+ for Eye Control with Eye Gate On and B- for without Eye Gate (Eye Gate Off) .

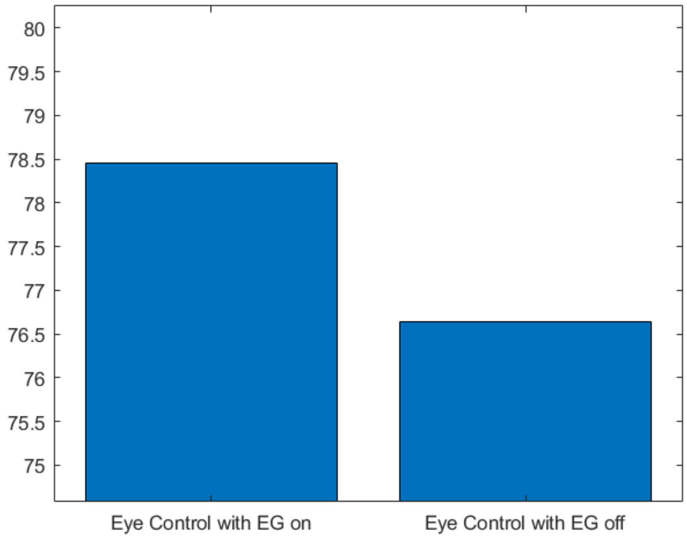


Figure 13A depiction of Eye Control Interface System SUS-Scores with Eye Gate On VS Eye Gate Off

These results shows having eye gate assist provided no significant difference. For eye gaze control, during experiments, participants raised questions with the aim being not stable, but this could be due to HMD weight or other factors as seeing the results we can conclude this with this reason . For future work this factor can be analyzed further.

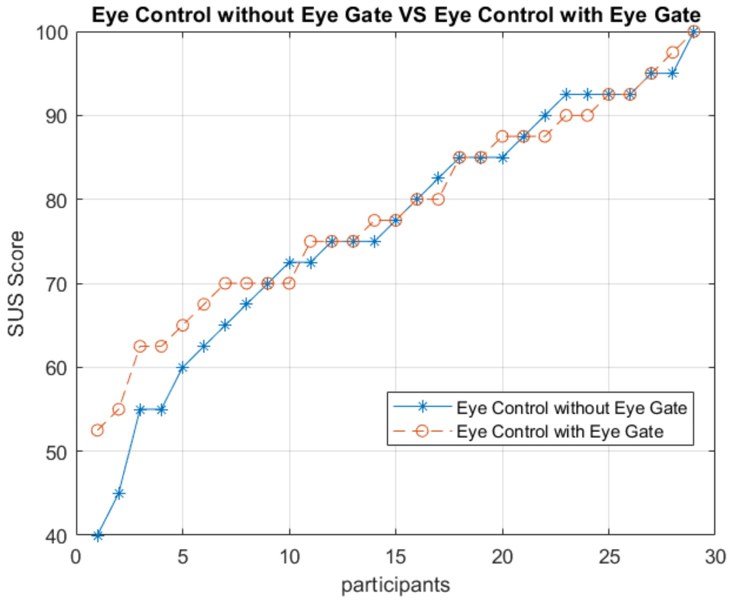


Figure 14 A depiction of Eye Control Interface system SUS-Mean Scores from 58 participants categorized for each Gate modes;(Eye Gate On VS Eye Gate Off) making 29 responses for each case.

From the figure above it can be seen the Eye Control results for both Eye Gate On and Eye Gate Off scenarios the participants had close experience.

Hence making the Eye Gaze interface-based controller an appealing choice for control systems for this research simulation, hereby, providing a concrete reason to be considered for usage by disabled people in industries.

6.1.3 Hand Control Interface with Eye Gate On VS Eye Gate Off

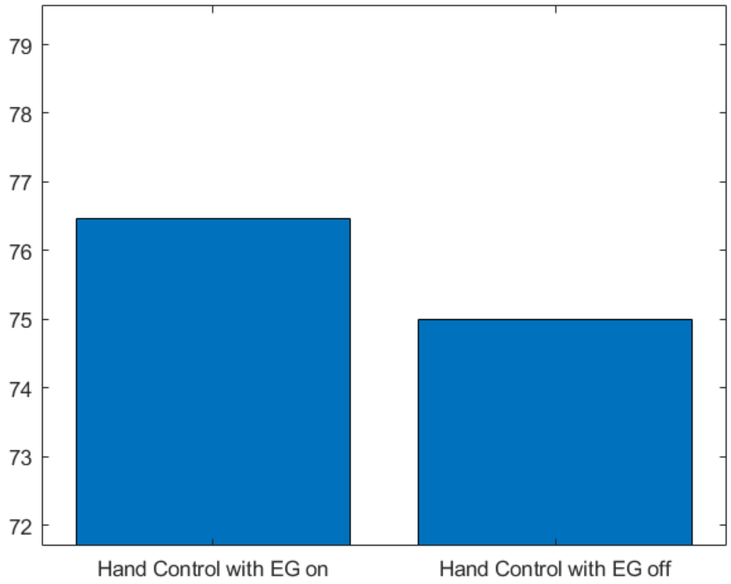


Figure 15A depiction of Hand Control Interface System SUS-Scores with Eye Gate On VS Eye Gate Off

The results show close sus-mean score for both case for this configuration.

Table 5 The Table shows SUS Mean Scores for Eye Control Cases both On and off for Eye Gate

Hand Controller	SUS Scores (0-100)
Eye Gate On	76.46
Eye Gate Off	75.00

For Hand Control interface using Eye gate SUS score of 76.46 as without the Eye Gate assist earlier SUS-Score of 75.00, having a difference of 1.46; Sauro-Lewis Grade of solid B for Hand Control with Eye Gate On and without Eye Gate (Eye Gate Off). Normalizing the outcome signifying the Eye Gate assist didn't matter much overall for Hand Control Interface for this experiment.

By which we can see similarity with Eye Control Interface as mentioned before for which score was close also. This also points that participants had high and low experience with all cases.

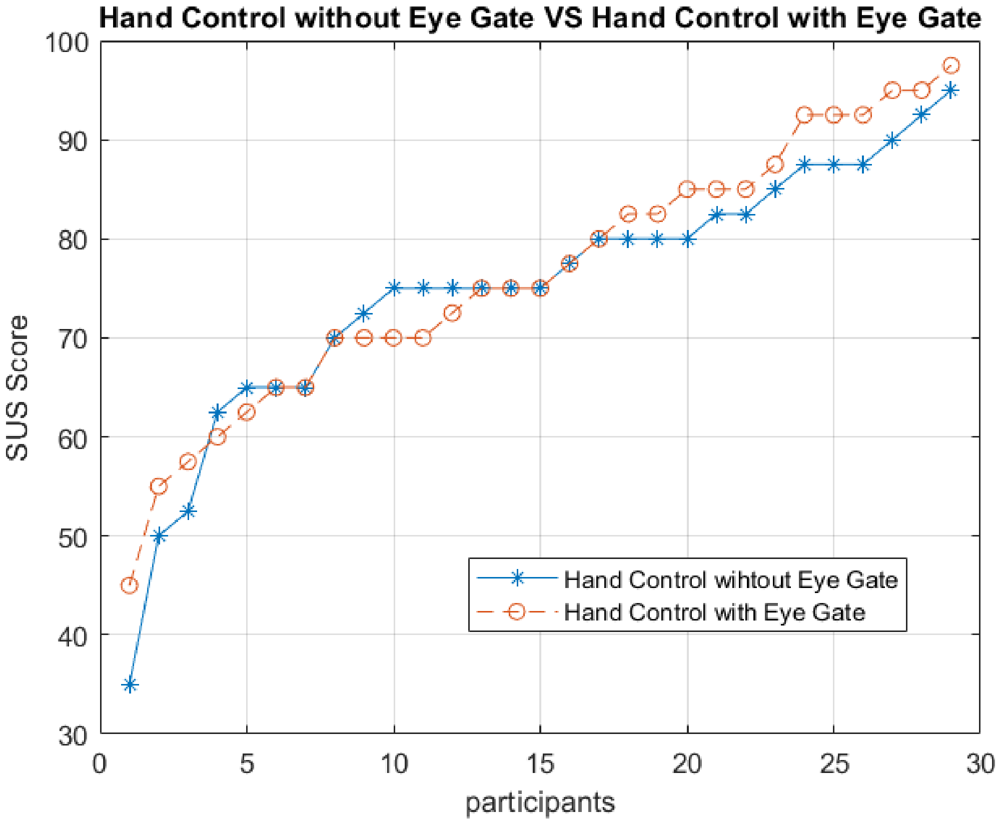


Figure 16 Figure 14 A depiction of Hand Control Interface system SUS-Mean Scores from 58 participants categorized for each Gate modes;(Eye Gate On VS Eye Gate Off) making 29 responses for each case.

The figure illustrates the pattern for both cases(Eye Gate On and Eye Gate Off). These results are close to Eye Control Interface as can be seen above. The Participants had quite close experience with both configurations. Except for small number of individuals which had significantly different experience using both case. Overall results are close and for this we can accept that eye gate mode did not affect much usability for this configuration.

Table 6 This table gives detailed results for Mean; Median; Standard deviation; Mode, and Variance results for VR Control Interfaces Cases used in experiments.

IC	Mean	Median	Mode	Std	Var
HC with EG	76.46	75	70	13.43	180.588
HC w/o EG	63.31	75	75	13.27	176.33
EC with EG	78.44	77.5	70	12.61	159.113
EC w/o EG	76.63	77.5	92.5	15.63	244.31

The table shows all mentioned results for SUS-Mean values for all cases. By seeing variance, we can see the eye control interface with eye gate has lowest value which makes it really interesting. Participants had overall smoother experience with Eye Gate interface using Eye Gate assist for simulation environment for control systems for 3d crane. This shows the natural usability for Eye Gaze Interface as control system for users keeping in mind that 53.4 percent out of 58 participants had never even used and VR HMD before (which includes Hand-Held Controller for VR device). These results indicate the possibility and use of Eye Gaze Control Interface systems very promising in regards to be applied for use for disabled individuals.

Table 7 This table shows data sample for SUS-Scores from Participants

Gender	Age:	Occupation	Used VR before Y/N	ECwithEGOff
Male	21-25	IT	No	95
Male	21-25	IT Student	No	92.5
Male	26-30	Engineering	No	92.5
Male	26-30	Product Manager	No	92.5
Male	26-30	finance	No	87.5
Male	26-30	Engineering	No	85
Female	31-35	Student	No	85
Male	26-30	mechatronics	No	77.5

By looking at this sample data set we can understand the mode from table above 50% participants who used eye control without eye gate have higher SUS-Mean score, half of this 50% had never used VR before which can be seen in this data. This further validates Eye Gaze Controller usability.

But overall, the EC without EG is unstable, even though it gave best experience to participant but its highly volatile, as it gave worst experience also by seeing the variance value with this configuration.

Gender	Age:	Occupation	Used VR before Y/N	ECwithEGOn
Male	26-30	IT	Yes	100
Male	26-30	Engineering Student	Yes	95
Male	26-30	IT	Yes	92.5
Male	26-30	IT	Yes	90
Male	15-20	Student	Yes	87.5
Male	26-30	Engineering	Yes	87.5
Male	21-25	Engineering	Yes	87.5
Male	26-30	IT	Yes	80
Male	31-35	Business Student	Yes	80
Male	26-30	Engineering student	Yes	77.5
Male	21-25	Finance	Yes	75
Male	31-35	Engineering	Yes	65
Male	31-35	Architecture	Yes	62.5

Majority of the participant who had prior experienced using VR and used Eye Control with Eye Gate On they have above average SUS-mean score, this maybe an indication that this configuration requires prior experience with VR to perform better with this configuration.

Table 8 The Table shows the Maximum and minimum SUS-Scores comparison for Hand Control Interface (HC) when Eye Gate (EG)is on or off.

Interface Controls	Maximum Score	Minimum Score
HC w/o EG	97.5	35
HC with EG	95	45

Table 5 shows the maximum score and the minimum score for both cases in comparison by 58 participants respectively. There is a 25% difference between the minimum score for both cases, while 2.5% difference for the maximum score.

6.4External Factors

The findings show that changes in gaze angle have the potential to affect visual function. It is critical to underline that we are not claiming that this is the sole possible source of visual stress within a head-mounted-display HMD. A variety of elements appear to have the capacity to combine and create a stressful atmosphere. On the other hand, reducing a known problem seems reasonable, and HMD makers may choose to allow for some modification of vertical screen position as well as some mechanism by which users may analyze the demands imposed on their visual system at various gaze angles[24].

Depending on the Person if he/she wears glasses can result in some different results entirely as while experimentations as the body of HMD and eye distances from optical lenses inside headset becomes more distant while using EYE Gaze Control System, this can be investigated more depth for future work and analyzing this dataset and results further.

7. Conclusion

Initially, smart systems are "systems of systems." Their components are smart systems in and of themselves, but the total functionality is the product of numerous systems functioning as one body[1].

As the hand controller is quite a generic way of control systems for VR Simulations, its minimum SUS scores are quite interesting (data from Table 5) but considering that 53.4%

of participants had never experienced and HMD / VR before (data from Figure 9), the results are making a strong case for Eye Gaze Controller.

This research paper purpose was to assess the control systems for this Virtual Reality Environment used for this experiments with Hand Control and Eye Gaze Control Interface Systems, and check if usability of eye control interfaces against hand control systems. This paper concluded that Eye Control Interface usability is close to hand control interfaces in VR.

Hence, making a great case to be considered for industrial implementation for Eye Gaze Control (EC) interface systems which can provide work opportunities for disabled individuals in the future, making more livelihood opportunities for them, and make control systems innovative for future use in machinery where this type of implementation can be used.

The Eye Gaze control interface system for this research can be worked on further in the future when considering a fully eye-based control system where the trigger function for aim is implemented by using eye-gaze with a timer or similar approach, which is a definite future work implementation on this system and may provide more appealing results.
Additionally exploring sensitivity for eye control interface system with variable configuration can be done for future work with this virtual reality environment simulation used for this paper.

8 References

[1] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, “Experimental Digital Twins-Streamlining Simulation-Based Systems Engineering for Industry 4.0,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1722–1731, Apr. 2018, doi: 10.1109/TII.2018.2804917.

[2] S. Jersov and A. Tepljakov, “Digital Twins in Extended Reality for Control System Applications,” 2020 43rd International Conference on Telecommunications and Signal Processing (TSP), 2020.

[3] E. Christopoulou and S. Xinogalos, “Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices,” *International Journal of Serious Games*, vol. 4, no. 4, Dec. 2017, doi: 10.17083/ijsg.v4i4.194.

[4] L. Delicado and A. A. Alcántara, “Synthesising character animation for real-time crowd simulation systems in Unreal Engine,” 2018.

[5] V. Clay, P. König, and S. König, “Eye tracking in virtual reality,” *Journal of Eye Movement Research*, vol. 12, no. 1, 2019, doi: 10.16910/jemr.12.1.3.

[6] C. Boletsis, “The new era of virtual reality locomotion: A systematic literature review of techniques and a proposed typology,” *Multimodal Technologies and Interaction*, vol. 1, no. 4. MDPI AG, Dec. 01, 2017. DOI: 10.3390/mti1040024.

[7] H. J. Joo and H. Y. Jeong, “A study on eye-tracking-based Interface for VR/AR education platform,” *Multimedia Tools and Applications*, vol. 79, no. 23–24, pp. 16719–16730, Jun. 2020, doi: 10.1007/s11042-019-08327-0.

[8] M. Wiesing, G. R. Fink, and R. Weidner, “Accuracy and precision of stimulus timing and reaction times with Unreal Engine and SteamVR,” *PLoS ONE*, vol. 15, no. 4, Apr. 2020, doi: 10.1371/journal.pone.0231152.

[9] I. Rakhmatulin, I. Dataset for Eye-Tracking Tasks. Preprints 2020, 2020120047 (DOI: 10.20944/preprints202012.0047.v1).

[10] M. Elstner, “Double Degree Program in Civil and Construction Engineering Use cases of Extended Reality in the construction industry.”

[11] S. F. Kuliga, T. Thrash, R. C. Dalton, and C. Hölscher, “Virtual reality as an empirical research tool - Exploring user experience in a real building and a corresponding virtual model,” *Computers, Environment and Urban Systems*, vol. 54, pp. 363–375, Nov. 2015, DOI: 10.1016/j.compenvurbsys.2015.09.006.

[12] A. Köse, A. Tepljakov, and E. Petlenkov, “Intelligent Virtual Environments with Assessment of User Experiences,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021, vol. 12854 LNAI, pp. 463–474. DOI: 10.1007/978-3-030-87986-0_41.

[13] S. Kloiber, C. Shinko, V. Settgast, M. Weinzierl, T. Schreck, and R. Preiner, “Integrating assembly process design and VR-based evaluation using the unreal engine,” in *VISIGRAPP 2020 - Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2020, vol. 1, pp. 271–278. DOI: 10.5220/0008965002710278.

[14] S. Chugh, “An Eye-Tracking System for a Virtual Reality Headset,” 2020.

[15] S. (Suzie) Kardong-Edgren, S. L. Farra, G. Alinier, and H. M. Young, “A Call to Unify Definitions of Virtual Reality,” *Clinical Simulation in Nursing*, vol. 31, pp. 28–34, Jun. 2019, doi: 10.1016/j.ecns.2019.02.006.

[16] M. Speicher, B. D. Hall, and M. Nebeling, “What is mixed reality?,” May 2019. doi: 10.1145/3290605.3300767.

[17] S. Kloiber, C. Shinko, V. Settgast, M. Weinzierl, T. Schreck, and R. Preiner, “Integrating assembly process design and VR-based evaluation using the unreal engine,” in *VISIGRAPP 2020 - Proceedings of the 15th International Joint*

[18] Christopher J. Turner, Windo Hutabarat, John Oyekan, and Ashutosh Tiwari, “DESandVRinIndustryTurneretal2016”.

[19] R. Teather, Y. Itoh, J. Gabbard, IEEE Computer Society. Technical Committee on Visualization and Graphics, and Institute of Electrical and Electronics Engineers, *Proceedings, 26th IEEE Conference on Virtual Reality and 3D User Interfaces : Osaka, Japan, 23-27 March 2019*.

[20] “SteamVR - Virtual Reality and Augmented Reality Wiki - VR AR & XR Wiki.” https://xinreality.com/wiki/SteamVR (accessed Dec. 14, 2021).

[21] “Welcome to Steam.” https://store.steampowered.com/ (accessed Dec. 14, 2021).

[22] “Simulink Documentation.” https://www.mathworks.com/help/simulink/ (accessed Dec. 14, 2021).

[23] “About the VIVE Pro 2 headset.” https://www.vive.com/eu/support/vive-pro2/category_howto/about-the-headset.html (accessed Dec. 14, 2021).

[24] KM On-w Illiam S, M. A., Nana Plooy, A., Obin Burgess-limerick, R., & Ann, J. W. (n.d.). *Gaze angle: a possible mechanism of visual stress in virtual reality headsets*.

[25] J. R. Lewis, “The System Usability Scale: Past, Present, and Future,” *International Journal of Human-Computer Interaction*, vol. 34, no. 7, pp. 577–590, Jul. 2018, doi: 10.1080/10447318.2018.1455307.

[26] J. Brooke, “SUS: A Retrospective,” Vol. 8, Issue 2, February 2013, Usability Professionals’ Association and the authors, http://www.usabilityprofessionals.org/.

[27] OBS Studio and Jim, “Open Broadcaster Software | OBS.” https://obsproject.com/ (accessed Dec. 16, 2021).

[28] L. Zhang, J. Wade, A. Swanson, A. Weitlauf, Z. Warren, and N. Sarkar, “Cognitive state measurement from eye gaze analysis in an intelligent virtual reality driving system for autism intervention,” in *2015 International Conference on Affective Computing and Intelligent Interaction, ASCII 2015*, Dec. 2015, pp. 532–538. doi: 10.1109/ACII.2015.7344621.

[29] A. Santos, I. Aedo, T. Zarraonandia, and P. Díaz, “A comparative study of menus in virtual reality environments,” in *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces, ISS 2017*, Oct. 2017, pp. 294–299. doi: 10.1145/3132272.3132277.

[30] A. L. Butt, S. Kardong-Edgren, and A. Ellertson, “Using Game-Based Virtual Reality with Haptics for Skill Acquisition,” *Clinical Simulation in Nursing*, vol. 16, pp. 25–32, Mar. 2018, doi: 10.1016/j.ecns.2017.09.010.

[31] Bertinetto, Carlo, Engel, Jasper, Jansen, Jeroen, “ANOVA simultaneous component analysis: A tutorial review”. Analytica Chimica Acta: X. 2020

[32] Lin, Shu Hui, Lee, Jack C.” Exact tests in simple growth curve models and one-way ANOVA with Equi correlation error structure”, Journal of Multivariate Analysis, 2003.

[33] Hutchinson, T.P.” On the generalized Friedman test”, Computational Statistics & Data Analysis, 1996.

[34] A. Tepljakov (ed.), “Virtual and Augmented Reality: State of the Art Technology Report”. VAM*Rs Project Consortium, 2021. [Online]. Available: https://vam-realities.eu/vam-realities-state-of-the-art-report/.

9Appendix 1 – **Non-exclusive licence for reproduction and publication of a graduation thesis**

MATLAB CODE:

```
clc, clear, close all;

n = 64; % because 58+6 == next whole number divisible by 16 [size of setup
rule binary array]
run dataImportAll.m;

% motionControlled = IThinkThatIWouldLikeToUseTheSystemFrequentlyExperiment1;
% eyeControlled = IThinkThatIWouldLikeToUseTheSystemFrequentlyExperiment2;
initialDataExpereiment1 = horzcat(A,C,E,G,I,K,M,O,Q,S);
initialDataExpereiment2 = horzcat(B,D,F,H,J,L,N,P,R,T);

reformingArray = zeros(n - length(initialDataExpereiment1), 10);

initialDataExpereiment1refined = [initialDataExpereiment1; reformingArray];
initialDataExpereiment2refined = [initialDataExpereiment2; reformingArray];

% setup rule for control
setupRule = zeros(8, 1);
setupRule1 = [setupRule; ones(8, 1)];
setupRule2 = [ones(8, 1); setupRule];

for i = 1:2 % length of for loop is size/16/2 == 19
    setupRule1 = [setupRule1; setupRule1];
    setupRule2 = [setupRule2; setupRule2];
end

% setup rule for eye gate
eyegateSetup1 = zeros(length(A), 1);
eyegateSetup2 = ones(length(A), 1);

clear reformingArray A B C D E F G H I J K L M N O P Q R S T;

for i = 1 : length(eyegateSetup1)
    if(mod(i,2) == 0)
        eyegateSetup1(i) = 1;
        eyegateSetup2(i) = 0;
    end
```

```
end

[~, cols] = size(initialDataExpereiment1refined);

% experiment 1 setup
HCwithEGOnSetup1 = setupRule1(1:58).*eyegateSetup1;
HCwithEGOffSetup1 = setupRule1(1:58).*eyegateSetup2;
ECwithEGOnSetup1 = setupRule2(1:58).*eyegateSetup1;
ECwithEGOffSetup1 = setupRule2(1:58).*eyegateSetup2;

% experiment 2 setup
HCwithEGOnSetup2 = setupRule2(1:58).*eyegateSetup2;
HCwithEGOffSetup2 = setupRule2(1:58).*eyegateSetup1;
ECwithEGOnSetup2 = setupRule1(1:58).*eyegateSetup2;
ECwithEGOffSetup2 = setupRule1(1:58).*eyegateSetup1;

% combining outcomes xor operation
HCwithEGOffSetup = HCwithEGOffSetup2 + HCwithEGOffSetup1;
HCwithEGOnSetup = HCwithEGOnSetup1 + HCwithEGOnSetup2;
ECwithEGOnSetup = ECwithEGOnSetup2 + ECwithEGOnSetup1;
ECwithEGOffSetup = ECwithEGOffSetup2 + ECwithEGOffSetup1;

% make sure HCwithEGOffSetup HCwithEGOnSetup ECwithEGOnSetup ECwithEGOffSetup
% are strictly binary arrays by normalizing setup arrays; check output of
% this for loop
for i = 1 : length(HCwithEGOffSetup)
    if (HCwithEGOnSetup(i)) > 1
        HCwithEGOnErrorPoints = [HCwithEGOnErrorPoints; HCwithEGOnSetup(i)];
    end
    if (HCwithEGOffSetup(i)) > 1
        HCwithEGOffErrorPoints = [HCwithEGOffErrorPoints;
HCwithEGOffSetup(i)];
    end
    if (ECwithEGOnSetup(i)) > 1
        ECwithEGOnErrorPoints = [ECwithEGOnErrorPoints; ECwithEGOnSetup(i)];
    end
    if (ECwithEGOffSetup(i)) > 1
        ECwithEGOffErrorPoints = [ECwithEGOffErrorPoints;
ECwithEGOffSetup(i)];
    end
end

for i = 1 : cols
    motionControlled1(:, i) = initialDataExpereiment1refined(:,
i).*setupRule1;
    eyeControlled1(:, i) = initialDataExpereiment2refined(:, i).*setupRule2;
    eyegateOn2(:, i) = initialDataExpereiment1(:, i).*eyegateSetup1;
    eyegateOff2(:, i) = initialDataExpereiment2(:, i).*eyegateSetup2;
    HCwithEGOn1(:, i) = initialDataExpereiment1refined(1:58,
i).*HCwithEGOnSetup1;
    HCwithEGOff1(:, i) = initialDataExpereiment1refined(1:58,
i).*HCwithEGOffSetup1;
    ECwithEGOn1(:, i) = initialDataExpereiment1refined(1:58,
i).*ECwithEGOnSetup1;
    ECwithEGOff1(:, i) = initialDataExpereiment1refined(1:58,
i).*ECwithEGOffSetup1;
end
```

```
expereiment2Data = motionControlled1 + eyeControlled1; %%% cleaned eye
control data
eyegate2Data = eyegateOn2 + eyegateOff2; %%% cleaned eye gate data

for i = 1 : cols
    motionControlled3 = initialDataExpereiment1refined.*setupRule2;
    eyeControlled3 = initialDataExpereiment2refined.*setupRule1;
    eyegateOn1(:, i) = initialDataExpereiment1(:, i).*eyegateSetup2;
    eyegateOff1(:, i) = initialDataExpereiment2(:, i).*eyegateSetup1;
    HCwithEGOn2(:, i) = initialDataExpereiment2refined(1:58,
i).*HCwithEGOnSetup2;
    HCwithEGOff2(:, i) = initialDataExpereiment2refined(1:58,
i).*HCwithEGOffSetup2;
    ECwithEGOn2(:, i) = initialDataExpereiment2refined(1:58,
i).*ECwithEGOnSetup2;
    ECwithEGOff2(:, i) = initialDataExpereiment2refined(1:58,
i).*ECwithEGOffSetup2;
end

% clear HCwithEGOffSetup HCwithEGOnSetup ECwithEGOnSetup ECwithEGOffSetup;

clear ECwithEGOffSetup1 ECwithEGOffSetup2 ECwithEGOnSetup1 ECwithEGOnSetup2;
clear HCwithEGOffSetup2 HCwithEGOffSetup1 HCwithEGOnSetup2 HCwithEGOnSetup1;

clear eyegateSetup2 eyegateSetup1 setupRule1 setupRule2 setupRule;
clear initialDataExpereiment1 initialDataExpereiment2
initialDataExpereiment1refined initialDataExpereiment2refined;

expereiment1Data = motionControlled3 + eyeControlled3; %%% cleaned motion
control data
eyegate1Data = eyegateOn1 + eyegateOff1; %%% cleaned eye gate data

HCwithEGOff = HCwithEGOff2 + HCwithEGOff1;
HCwithEGOn = HCwithEGOn2 + HCwithEGOn1;
ECwithEGOff = ECwithEGOff2 + ECwithEGOff1;
ECwithEGOn = ECwithEGOn1 + ECwithEGOn2;

clear motionControlled1 motionControlled3 eyeControlled1 eyeControlled3;
clear eyegateOff1 eyegateOff2 eyegateOn1 eyegateOn2;

clear HCwithEGOff2 HCwithEGOff1 HCwithEGOn2 HCwithEGOn1;
clear ECwithEGOff2 ECwithEGOff1 ECwithEGOn1  ECwithEGOn2;

% final outputs
DataVariables.handControl = expereiment1Data(1:58,1:10);
DataVariables.eyegazeControl = expereiment2Data(1:58,1:10);
DataVariables.eyeGateOn = eyegate1Data;
DataVariables.eyeGateOff = eyegate2Data;

DataVariables.HCwithEGOff = HCwithEGOff;
DataVariables.HCwithEGOn = HCwithEGOn;
DataVariables.ECwithEGOff = ECwithEGOff;
DataVariables.ECwithEGOn = ECwithEGOn;

clear HCwithEGOff HCwithEGOn ECwithEGOff ECwithEGOn;
clear expereiment1Data expereiment2Data eyegate1Data eyegate2Data rows cols;
clear initialDataExpereiment1 initialDataExpereiment2
initialDataExpereiment1refined initialDataExpereiment2refined;
```

```matlab
%% finding raw score

[rowsDV, colsDV] = size(DataVariables.handControl);
% processing1 = zeros(rowsDV, colsDV);

for i = 1 : colsDV
    if(mod(i, 2) == 0)
        processing1(:, i) = 5 - DataVariables.handControl(:, i);
        processing2(:, i) = 5 - DataVariables.eyegazeControl(:, i);
        processing3(:, i) = 5 - DataVariables.eyeGateOn(:, i);
        processing4(:, i) = 5 - DataVariables.eyeGateOff(:, i);
        processing5(:, i) = 5 - DataVariables.HCwithEGOff(:, i);
        processing6(:, i) = 5 - DataVariables.HCwithEGOn(:, i);
        processing7(:, i) = 5 - DataVariables.ECwithEGOff(:, i);
        processing8(:, i) = 5 - DataVariables.ECwithEGOn(:, i);
    else
        processing1(:, i) = DataVariables.handControl(:, i) - 1;
        processing2(:, i) = DataVariables.eyegazeControl(:, i) - 1;
        processing3(:, i) = DataVariables.eyeGateOn(:, i) - 1;
        processing4(:, i) = DataVariables.eyeGateOff(:, i) - 1;
        processing5(:, i) = DataVariables.HCwithEGOff(:, i) - 1;
        processing6(:, i) = DataVariables.HCwithEGOn(:, i) - 1;
        processing7(:, i) = DataVariables.ECwithEGOff(:, i) - 1;
        processing8(:, i) = DataVariables.ECwithEGOn(:, i) - 1;
    end
end

% Here sum is calculated row-wise, giving a more accurate sum data in 11th
% col
for i = 1 : rowsDV
    DataVariables.handControl(i, colsDV + 1) = sum(processing1(i, :));
    DataVariables.eyegazeControl(i, colsDV + 1) = sum(processing2(i, :));
    DataVariables.eyeGateOn(i, colsDV + 1) = sum(processing3(i, :));
    DataVariables.eyeGateOff(i, colsDV + 1) = sum(processing4(i, :));
    DataVariables.HCwithEGOff(i, colsDV + 1) = sum(processing5(i, :));
    DataVariables.HCwithEGOn(i, colsDV + 1) = sum(processing6(i, :));
    DataVariables.ECwithEGOff(i, colsDV + 1) = sum(processing7(i, :));
    DataVariables.ECwithEGOn(i, colsDV + 1) = sum(processing8(i, :));
end

% making sure there are no errors
DataVariables.HCwithEGOff = DataVariables.HCwithEGOff.*HCwithEGOffSetup;
DataVariables.HCwithEGOn = DataVariables.HCwithEGOn.*HCwithEGOnSetup;
DataVariables.ECwithEGOff = DataVariables.ECwithEGOff.*ECwithEGOffSetup;
DataVariables.ECwithEGOn = DataVariables.ECwithEGOn.*ECwithEGOnSetup;

clear HCwithEGOffSetup HCwithEGOnSetup ECwithEGOnSetup ECwithEGOffSetup;

clear processing1 processing2 processing3 processing4;
clear processing5 processing6 processing7 processing8;

%% finding sus score

[rowsDV, colsDV] = size(DataVariables.handControl);

DataVariables.handControl(:, colsDV + 1) = DataVariables.handControl(:,
colsDV).*2.5;
DataVariables.eyegazeControl(:, colsDV + 1) = DataVariables.eyegazeControl(:,
colsDV).*2.5;
```

```
DataVariables.eyeGateOn(:, colsDV + 1) = DataVariables.eyeGateOn(:, 
colsDV).*2.5;
DataVariables.eyeGateOff(:, colsDV + 1) = DataVariables.eyeGateOff(:, 
colsDV).*2.5;
DataVariables.HCwithEGOff(:, colsDV + 1) = DataVariables.HCwithEGOff(:, 
colsDV).*2.5;
DataVariables.HCwithEGOn(:, colsDV + 1) = DataVariables.HCwithEGOn(:, 
colsDV).*2.5;
DataVariables.ECwithEGOff(:, colsDV + 1) = DataVariables.ECwithEGOff(:, 
colsDV).*2.5;
DataVariables.ECwithEGOn(:, colsDV + 1) = DataVariables.ECwithEGOn(:, 
colsDV).*2.5;

% Averages
DataVariables.handControl(rowsDV + 1, :) = mean(DataVariables.handControl);
DataVariables.eyegazeControl(rowsDV + 1, :) = 
mean(DataVariables.eyegazeControl);
DataVariables.eyeGateOn(rowsDV + 1, :) = mean(DataVariables.eyeGateOn);
DataVariables.eyeGateOff(rowsDV + 1, :) = mean(DataVariables.eyeGateOff);


% raw score avg
rawScore.handControlAvg = DataVariables.handControl(rowsDV + 1, colsDV);
rawScore.eyegazeControlAvg = DataVariables.eyegazeControl(rowsDV + 1, 
colsDV);
rawScore.eyeGateOnAvg = DataVariables.eyeGateOn(rowsDV + 1, colsDV);
rawScore.eyeGateOffAvg = DataVariables.eyeGateOff(rowsDV + 1, colsDV);

% sus score avg
susScore.handControlAvg = DataVariables.handControl(rowsDV + 1, colsDV + 1);
susScore.eyegazeControlAvg = DataVariables.eyegazeControl(rowsDV + 1, colsDV 
+ 1);
susScore.eyeGateOnAvg = DataVariables.eyeGateOn(rowsDV + 1, colsDV + 1);
susScore.eyeGateOffAvg = DataVariables.eyeGateOff(rowsDV + 1, colsDV + 1);

%% generating output table

handcontrol = sort(DataVariables.handControl(1:58, 12));
eyegazeControl = sort(DataVariables.eyegazeControl(1:58, 12));
eyeGateOn = sort(DataVariables.eyeGateOn(1:58, 12));
eyeGateOff = sort(DataVariables.eyeGateOff(1:58, 12));

susTable = table(handcontrol, eyegazeControl, eyeGateOn, eyeGateOff);
writetable(susTable, 'susTable.xlsx');

ECwithEGOff = DataVariables.ECwithEGOff(:, 12);
ECwithEGOn = DataVariables.ECwithEGOn(:, 12);
HCwithEGOff = DataVariables.HCwithEGOff(:, 12);
HCwithEGOn = DataVariables.HCwithEGOn(:, 12);

ECwithEGOnFinal = [];
ECwithEGOffFinal = [];
HCwithEGOnFinal = [];
HCwithEGOffFinal = [];

for i = 1:58
    if ECwithEGOn(i) == 0
        ECwithEGOn(i) = NaN;
    else
```

```
        ECwithEGOnFinal = [ECwithEGOnFinal; ECwithEGOn(i)];
    end
    if ECwithEGOff(i) == 0
        ECwithEGOff(i) = NaN;
    else
        ECwithEGOffFinal = [ECwithEGOffFinal; ECwithEGOff(i)];
    end
    if HCwithEGOn(i) == 0
        HCwithEGOn(i) = NaN;
    else
        HCwithEGOnFinal = [HCwithEGOnFinal; HCwithEGOn(i)];
    end
    if HCwithEGOff(i) == 0
        HCwithEGOff(i) = NaN;
    else
        HCwithEGOffFinal = [HCwithEGOffFinal; HCwithEGOff(i)];
    end
end

DVTable = table(ECwithEGOff, ECwithEGOn, HCwithEGOff, HCwithEGOn);
writetable(DVTable, 'datavaraibles.xlsx');

%% comparisons

% with averaging
xData = categorical({'HC', 'EC'});
xData = reordercats(xData, {'HC', 'EC'});
yData = [susScore.handControlAvg; susScore.eyegazeControlAvg];

figure(1)
b1 = bar(xData, yData);

figure(2)
% HC w/o EG vs EC w/o EG sus data comparison
% plot(DataVariables.eyeGateOff(1:58, 12), DataVariables.handControl(1:58,
12))
gcf = plot(1:58, DataVariables.handControl(1:58, 12),...
    1:58, DataVariables.eyegazeControl(1:58, 12));
title('Line Plot of Hand Control and Eye Control form SUS-Scores')
xlabel('Participants from 1 to 58')
ylabel('SUS-Score (0-100)')
gcf(1).Marker = '*'; gcf(1).LineStyle = '-';
gcf(2).Marker = 'o'; gcf(2).LineStyle = '--';
gcf(1).DisplayName = 'Hand Control';
gcf(2).DisplayName = 'Eye Control';
legend; grid on;

% Eye control sus score average comparison between eyegate on or off
xData = categorical({'EG on', 'EG off'});
xData = reordercats(xData, {'EG on', 'EG off'});
yData = [mean(ECwithEGOnFinal); mean(ECwithEGOffFinal)];

figure(35)
b2 = bar(xData, yData);

figure(25)
% EC w/o EG vs EC with EG sus data comparison
gcf = plot(1:58, sort(ECwithEGOff), 1:58, sort(ECwithEGOn));
gcf(1).Marker = '*';gcf(1).DisplayName = 'EC w/o EG'; gcf(1).LineStyle = '-';
gcf(2).Marker = 'o';gcf(2).DisplayName = 'EC with EG'; gcf(2).LineStyle = '--
';
legend; grid on;xlabel('participants'); ylabel('SUS Score');
title('EC w/o EG vs EC with EG');

% Hand control sus score average comparison between eyegate on or off
xData = categorical({'EG on', 'EG off'});
xData = reordercats(xData, {'EG on', 'EG off'});
yData = [mean(HCwithEGOnFinal); mean(HCwithEGOffFinal)];

figure(36)
b2 = bar(xData, yData);

figure(26)
% HC w/o EG vs HC with EG sus data comparison
gcf = plot(1:58, sort(HCwithEGOff), 1:58, sort(HCwithEGOn));
gcf(1).Marker = '*';gcf(1).DisplayName = 'HC w/o EG'; gcf(1).LineStyle = '-';
gcf(2).Marker = 'o';gcf(2).DisplayName = 'HC with EG'; gcf(2).LineStyle = '--
';
legend; grid on;xlabel('participants'); ylabel('SUS Score');
title('HC w/o EG vs HC with EG');

clear handcontrol eyegazeControl eyeGateOn eyeGateOff
clear HCwithEGOff HCwithEGOn ECwithEGOff ECwithEGOn

HCwithEGOffData.mean = mean(HCwithEGOffFinal);
HCwithEGOffData.std = std(HCwithEGOffFinal);
HCwithEGOffData.var = var(HCwithEGOffFinal);
HCwithEGOffData.median = median(HCwithEGOffFinal);
HCwithEGOffData.mode = mode(HCwithEGOffFinal);

HCwithEGOnData.mean = mean(HCwithEGOnFinal);
HCwithEGOnData.std = std(HCwithEGOnFinal);
HCwithEGOnData.var = var(HCwithEGOnFinal);
HCwithEGOnData.median = median(HCwithEGOnFinal);
HCwithEGOnData.mode = mode(HCwithEGOnFinal);

ECwithEGOffData.mean = mean(ECwithEGOffFinal);
ECwithEGOffData.std = std(ECwithEGOffFinal);
ECwithEGOffData.var = var(ECwithEGOffFinal);
ECwithEGOffData.median = median(ECwithEGOffFinal);
ECwithEGOffData.mode = mode(ECwithEGOffFinal);

ECwithEGOnData.mean = mean(ECwithEGOnFinal);
ECwithEGOnData.std = std(ECwithEGOnFinal);
ECwithEGOnData.var = var(ECwithEGOnFinal);
ECwithEGOnData.median = median(ECwithEGOnFinal);
ECwithEGOnData.mode = mode(ECwithEGOnFinal);
```

I, ZAIN ALI KHAN

Grant Tallinn University of Technology free license (non-exclusive license) for my thesis " **Assessment of User Interfaces for Control System in an Immersive Virtual Reality Environment** ", supervised by Aleksei Tepljakov, PhD;Department of Computer Systems, TalTech and Saleh Alsaleh, MSc; Department of Computer Systems, TalTech.

   1. to be reproduced for preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until the expiry of the term of copyright;
   2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until the expiry of the term of copyright.

2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive license.
3. I confirm that granting the non-exclusive license does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act, or rights arising from other legislation.

[03.01.2022]

TALLINN 2022