2024-05-20

Manual Assembly in Virtual Reality facilitated by subtle visual guidance

Marcus Bohm **Nils Wikström**

DEGREE PROJECT
Computer Engineering
Bachelor level G2E, 15 hec

Department of Engineering Science, University West, Sweden

EXAMENSARBETE

Manual Assembly in Virtual Reality facilitated by subtle visual guidance

Sammanfattning

Framstegen inom blandad verklighetsteknik har öppnat nya möjligheter för att förbättra användarupplevelse och prestanda i olika tillämpningar. En sådan tillämpning skulle kunna vara manuella monteringsuppgifter, där visuell vägledning kan spela en avgörande roll för att förbättra effektivitet och noggrannhet. Denna studie syftar till att utvärdera effektiviteten av ett Discrete Guidance System (DGS) i en virtuell verklighet (VR)-miljö. DGS använder subtila visuella signaler för att vägleda användare genom monteringen av ett pussel, med målet att förbättra den övergripande arbetsutförandet och minska fel. En blandad metod användes, som kombinerade kvantitativa prestationsmått och kvalitativ användarfeedback. Deltagarna ombads att slutföra två olika pussel i en VR-miljö, ett utan vägledning och sedan med DGS aktiverat. DGS använde en blinkande kontur som utlöses med tangentbordsinmatning för att subtilt vägleda deltagarna. Implementeringen av DGS förbättrade uppgiftens slutförandetider och noggrannhet. Deltagarna rapporterade en positiv upplevelse med vägledningssystemet, och betydligt färre fel gjordes och användare som hade problem med pusslen fick mer jämna resultat jämfört med de som hade lättare för pusslen. Det Discreta Vägledningssystemet (DGS) är ett effektivt verktyg för att förbättra manuella monteringsuppgifter i en virtuell verklighetsmiljö. Den subtila visuella vägledningen som tillhandahålls av DGS hjälper användare att utföra uppgifter mer jämnt och noggrant, vilket belyser potentialen för bredare tillämpningar av denna teknik inom olika områden.

Datum:	2024-05-20
Författare:	Marcus Bohm, Nils Wikström
Examinator:	Rashid Ali
Handledare:	Thomas Pederson / Open Labs
Program:	Datateknik, högskoleingenjör – programmering och nätverksteknik, 180 hp
Huvudområde:	Datateknik
Utbildningsnivå:	Grundnivå
Kurskod:	EHD500, 15 hp
Nyckelord:	*VR, Nudging, Unreal engine 5, subconscious, Varjo, Varjo rx-3*
Utgivare:	Institutionen för ingenjörsvetenskap, Högskolan Väst 461 86 Trollhättan

Manual Assembly in Virtual Reality facilitated by subtle visual guidance

Manual Assembly in Virtual Reality facilitated by subtle visual guidance

Summary

The advancement of mixed reality technologies has opened new avenues for enhancing user experience and performance in various applications. One such application could be manual assembly tasks, where visual guidance can play a crucial role in improving efficiency and accuracy. This study aims to evaluate the effectiveness of a Discrete Guidance System (DGS) in a virtual reality (VR) environment. The DGS employs subtle visual cues to guide users through assembling a puzzle, with the goal of enhancing overall task performance and reducing errors. A mixed-methods approach was used, combining quantitative performance metrics and qualitative user feedback. Participants were asked to complete two different puzzles in a VR environment, one without guidance and then with the DGS enabled. The DGS utilized a blinking outline triggered by keyboard input to subtly guide the participants. The implementation of the DGS improved task completion times and accuracy. Participants reported a positive experience with the guidance system, and a significantly less errors were made and users that trouble with the puzzles results were more even compared to those who had an easier time with the puzzles. The DGS is an effective tool for enhancing manual assembly tasks in a virtual reality environment. The subtle visual guidance provided by the DGS helps users perform tasks more evenly and accurately, highlighting the potential for broader applications of this technology in various fields.

Date:	2024-05-20
Author:	Marcus Bohm, Nils Wikström
Examiner:	Rashid Ali
Advisor:	Thomas Pederson / Open Labs
Programme:	Computer Engineering – programming and network technology 180 hp
Main field of study:	Computer Engineering
Education level:	First cycle
Course code:	EHD500, 15 HE credits
Keywords:	VR*, Nudging, Unreal engine 5, subconscious, Varjo, Varjo rx-3*
Publisher:	Department of Engineering Science, University West SE-461 86 Trollhättan, Sweden

Preface

This thesis will end our engineering bachelor's degree within computer engineering at University West, Trollhättan. This thesis is equal to 15 university credits.

Since this project would not have been possible to execute without help and support from our teachers, supervisors, examiners and classmates, we would like to give a special thanks to:

- Thomas Pederson, our supervisor
- Open Labs for equipment and lab space
- Rashid Ali, our examiner
- Thanks to all our classmates that helped us with testing
- Thomas Lundquist for giving us a chance

Contents

1 Introduction

In an era where technology seamlessly integrated with our daily lives, a new era of innovation could be the jump to virtual- or augmented reality. Augmented reality (AR) and virtual reality (VR) technologies have rapidly emerged as powerful tools, aiming to transform the various aspects of being human. **VR** has a multitude of benefits like treatment of phobias where in 1995 Nathanial Durlach and Kalman [1] Glantz used VR to try and treat three conditions: acrophobia the fear of heights, agoraphobia fear of being in public spaces and the fear of flying, their work understood the potential of VR and its potential to address conditions like acrophobia. By creating immersive yet controlled scenarios, VR enabled the ability to simulate situations that could challenge maladaptive behaviours.

Consider a scenario where a novice employee encounters an intricate machine like a computer numerical control (CNC) machine and is not fully capable of using it, a discrete guidance system (DGS) embedded in an VR headset could be used to practice how to use the machine. This could reduce time to train employees and remove the need for a more senior employee to spend their time teaching new hires.

1.1 Research questions

The project aimed to explore and compare the differences in human performance while being subtly guided [2, 3] during a task. The task in this project was to solve two different puzzles, one where the participant is subtly guided with a DGS and one where they are not guided. By doing this we wanted to see if we can help the participants complete the puzzle faster by guiding them with the DGS

Research question 1: "Will the average completion time for each puzzle when using the DGS be faster than when not using it?"

To test if the participants are following the DGS, we wanted to see if they make less mistakes in their puzzle piece picking. This was done by counting the number of times they put a piece in the wrong position.

Research question 2: "Will the participants make less errors while solving the puzzle if DGS is used?"

1.2 Goals

The goal of the project was to successfully guide each participant to complete one puzzle with the DGS and one without the DGS to compare how the participants performed in the different scenarios. To measure the difference in performance there were two metrics considered: time to complete puzzles and number of errors made while attempting to complete the puzzles. Another important metric was how subtle the DGS was which was measured by asking the participants if they noticed the guidance. The goal was to make it subtle enough that no one noticed.

2 Method

To get a better understanding of how the DGS works this chapter explains and discusses how the nudging process works and how the implementation of the DGS has been done.

2.1 Nudging

To make the user pick the right puzzle piece, nudge theory is going to be implemented. Nudge theory [4] is based upon shaping the environment around an individual, this could be called the choice architecture. One can influence the likelihood that an individual picks either one or two. A key factor to this is the ability for the individual to keep its freedom of that choice and feel like they are in control of their own decisions.

Some studies have been made but using food by the world health organization [4] but instead of food a blinking light, making it so the user subtly get directions on what to do next without noticing. How the implementation works is by having a small filter over a piece or an aura around it for a brief period. This is where the eye tracking is important because as soon as the eye sees the piece the filter or aura disappears. The human eye can detect things in the peripheral vision that the brain does not recognize right away, and this is the phenomena that is being exploited, so the user finds the right piece without being immediately told what piece to pick up. As for this project a blinking outline is chosen and is triggered by the testers using keyboard inputs.

2.1.1 Implementation

The implementation was done in unreal engine 5 using their blueprints system, as seen in figure 1 we can see how the flow of the test. The tester started by putting on the headset and entered the VR environment, in front of them was a puzzle with 1x7 pieces witch they had to grab and place in a correct order. If a piece was not touched or been interacted with for 5 seconds a feint blinking effect started on a piece that the operators picked, and the effect would instantly stop if the participants gaze hit the piece. The exploited phenomena here is called Inattentional blindness [5] which happens when an individual fails to perceive objects or stimuli that are fully visible but not the focus of their attention.

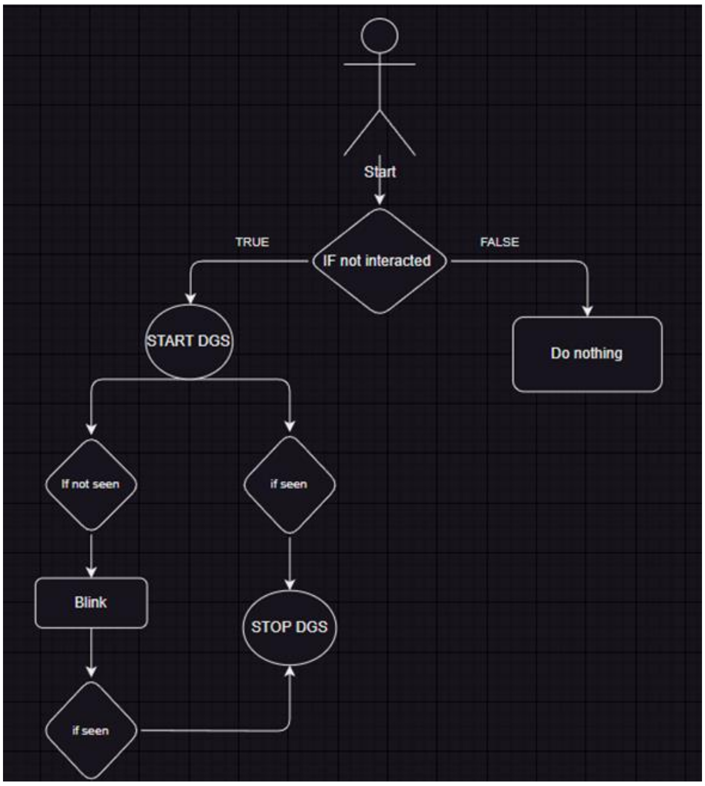

Figure 1 Flowchart how a run is going to look like

In the appendix figure B is what is going to trigger the DGS and triggers the phenomena inattentional blindness, it is a custom material that adds an outline around a static mesh that can be customized with thickness, glow intensity and colour.

When the DGS was triggered, the chosen piece would change an attribute called "custom render depth". This attribute is a bool-type and when set to "True" the custom material would be displayed on the piece to show the flashing outline.

Here in figure 2 is example of how the puzzles can look being split into 7 pieces and later textures that is appearing on the pieces in figure 3. In the appendix under D the second harder puzzle can be viewed.

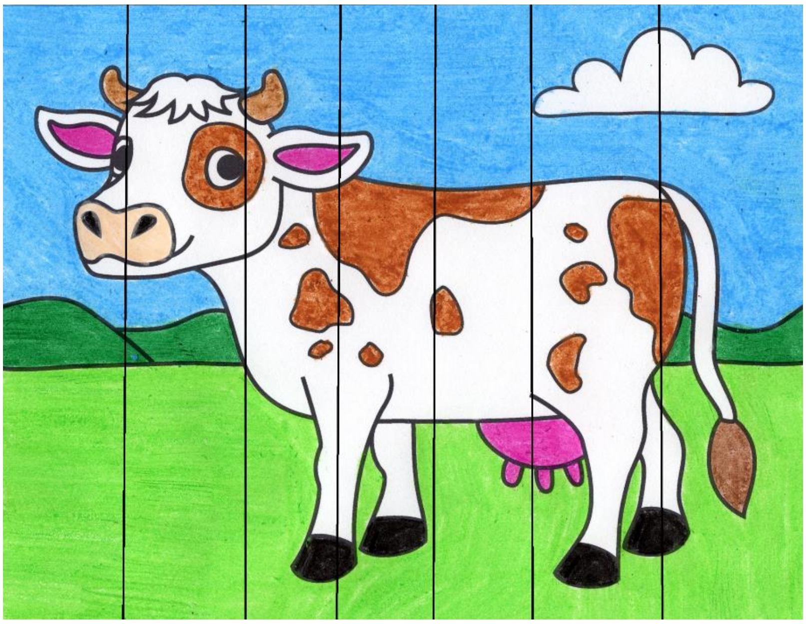

Figure 2 an example of one of the puzzles

The custom outline can be seen in Figure 3, and it was triggered by a keypress from the test handler to initiate the flashing effect. The participant would then be nudged to that piece for a similar effect that Weiquan Lu et al. [5] used, but in this way, having a more vivid and bright outline flashing in the corner of the eye of the tester. As soon as the eye tracks over this cube, the flashing would stop.

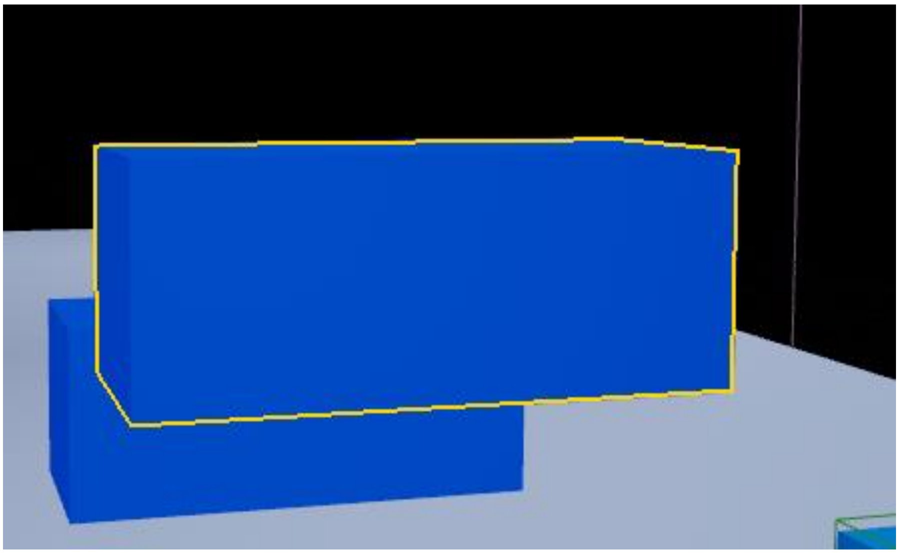

Figure 3 piece triggered by the DGS

2.1.2 Eye Tracking

The eye tracking was done by the Varjo headset itself and Varjo [6] has a solution in their development portal that was implemented in this project as shown in appendix C. The eye tracking data was used to control an invisible dot in the virtual world which would hit the trigger boxes attached to the puzzle pieces and cancel the DGS.

2.1.3 Hand tracking

Varjo has an implementation of hand tracking that they have created as an installable plugin in Unreal Engine. This works by dragging in a "ultraleap hands actor" into the worlds and setting the main camera to be that new actor. This is marketed by Varjo [7] as allowing the user to reach a new level of immersion.

2.1.4 DGS implementation

The DGS was started at the same time a participant was placed into a virtual world. It kept running in the background to check for a keypress from the test operators and would not trigger any visible effect until it got an input. When the operator pressed a key between the numbers 1-7 the corresponding puzzle piece would activate the DGS to show the flashing effect. The effect kept going until the participant looked at the piece or if the operator pressed a new key. As a failsafe, the keypress 0 would stop any effects in case of bugs or issues with the test. In the image in the next subchapter, you can see how the blueprint looks for this type of implementation for the eye tracking.

This blueprint shown in figure 4 are two event triggers, one that activates on the beginning of an overlap and another on the that activates on the end of an overlap. In this case it is the eye gaze causes the overlap and turn an eye tracking bool to false and turn of the DGS, and when the eye tracking leaves the trigger box area it turns on after a slight delay so it's not obvious that that specific piece is lit.

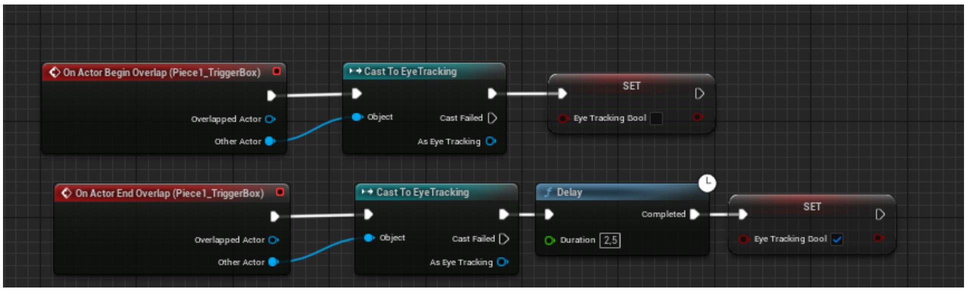

Figure 4 blueprint to set bool to true or false depending on eye gaze

The pieces were activated by a button press as seen in figure 5, this sets the eye tracking bool to true and then activate the DGS. The one shown in figure 5 only shows the trigger for piece 1 but in total there were 7 similar blueprints with the only difference being the keypress and the corresponding "nudging target". The nudging target is the piece that gets the DGS effect and start flashing.

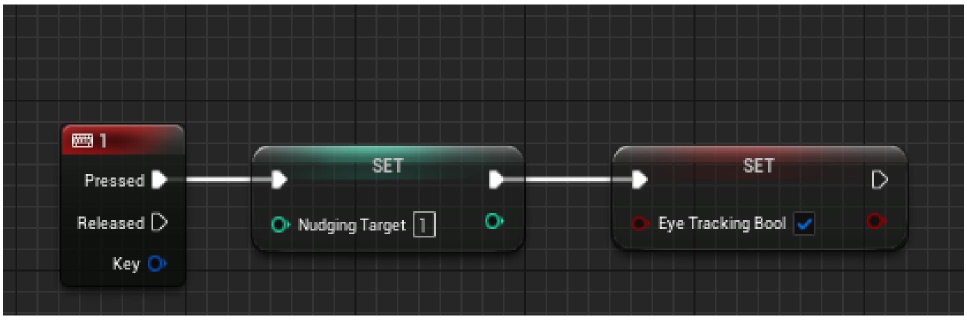

Figure 5 Blueprint for setting the nudging value

As mentioned earlier in the chapter, a failsafe was implemented to manage any bugs or issues with software. This is shown in figure 6 and it works by pressing the 0-key and not picking any of the seven pieces as a nudging target thus there is no visible effect of the DGS.

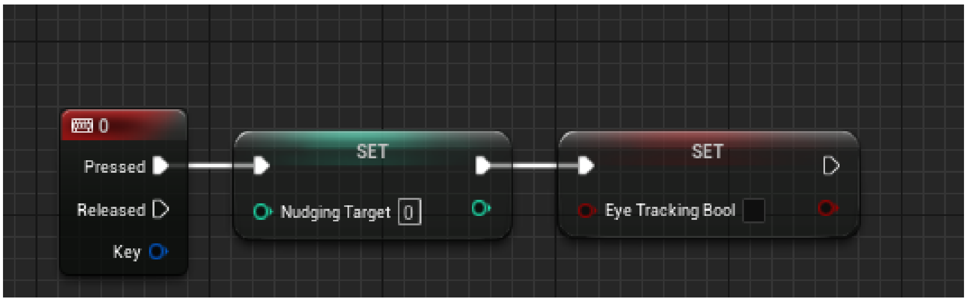

Figure 6 Blueprint for resetting all the pieces

The DGS blueprint worked by being reactive to the value of the "nudging target"-integer. After the participant was placed in one of the virtual worlds it starts to get the value of said integer, see appendix B:part 1. The value of the integer changes with the operators inputting the value through a keypress. The valid values contain 0 or 1-7. The value gets gathered from the "nudging target"-integer and gets sent through a "Switch on Int"-function that works the same as a switch case in programming. This essentially picks what should happen based on the value of the integer, if the value is 1, for example, it triggers the flashing effect on puzzle piece 1 as shown in appendix B:part 2. How it triggers the flashing effect is by using a "flip flop" which goes back and forth between the value 1 and 0. This sets the "custom render depth", as mentioned earlier in the chapter, to true and false respectively. Within this part of the blueprint is a "delay"-timer to edit the speed at which the outline flashes, in this case it sets the "custom render depth" to "True" and then delays the effect for 0,2 seconds before setting it to "False" for 0,2 seconds.

The outline material that is yellow outline on the pieces are a bunch cumulative multiplier that add together to give this glow look that get set to activate when the render custom depth is turned to true. This material later gets put onto a post process volume that surround the pieces. The blueprint for this material can be seen in the appendix under A.

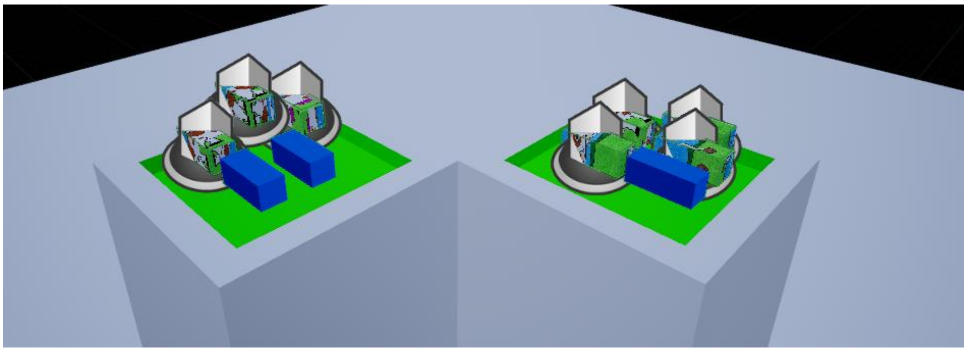

Figure 8 Image of the test area in VR.

2.2 Mixed method approach

To determine the effectiveness of the DGS a mixed method is being used. A qualitative approach is done to determine if the participants notice the use of the DGS and see if they felt like their choices were forced or not. Questions that were asked was:

"Did you feel like anything was off or something that felt like it should not be there?" and "Did you notice any difference between the puzzles besides the picture?".

With the collection of their times a quantitative approach was used to assemble data to determine if the DGS was effective or not. Recording the time, it took for participants to complete a puzzle and the number of errors made.

The mixed method is used to draw a conclusion if participants noticed if the use of the DGS and to see how effective the DGS is.

2.3 Tutorial

To help the participants get comfortable with the virtual environment and to practice the gripping with the hand-tracking they were placed in a tutorial environment. This environment was an exact copy of the puzzle environments with the difference that the puzzle pieces where blank. This helped the participants get used to moving the pieces and gave the test operators a chance to give further explanation in a more interactive manner.

3 Background

3.1 Nudging

To guide the user wearing the headset a technique called nudging is utilized. The nudging should be done as discreetly as possible, and the goal is that the user hardly notices that they are being affected by it at all.

3.1.1 What is nudging

There are different types of nudging both in the real world as well as digital. In this case the focus will be on digital nudging which exploits different design elements in the software which guides the user's attention toward a specific spot or tries to make the user make a desired decision. There are many ways to do this and there are also different scales of how subtle or not it can be. The aim here is to be as subtle as possible.

3.1.2 How it can be used

Nudging can be used differently depending on the situation. In the real world, a restaurant might put a certain dish at the top of the menu which means that more people are likely to see and order it. In a digital context, nudging could be the use of standard settings which most people are likely to ignore meaning that the developers can get a preferable outcome. For example, putting tipping as pre-picked when the user orders food online. For this thesis, nudging is used to subtly guide the user's attention towards specific items in front of them. For example, a missing puzzle piece the user is looking for.

3.2 Extended Reality

A big part of this thesis involves working with XR [8], or extended reality. XR is a broad term that includes VR, AR and MR. VR, virtual reality, uses a display mounted on the head of the user to give a more immersive experience of the virtual world. AR, augmented reality, is technology that aims to combine the virtual world with the real-world surroundings of the user. MR, mixed reality, describes the merging of the real-world the user is in and virtual objects. This gives an immersive experience since the virtual and real can interact with each other in real time. The experiment in this project was done in VR while using hand tracking and no controllers to add some immersion.

3.3 Game engine

Using a game engine can greatly help with the development of a game. It comes with built in functions, animations and graphical elements that removers the need to make everything from scratch. As mentioned above, the two game engines that have been considered are Unity [9] and Unreal Engine [10]. The role of the game engine in this thesis work is to create the virtual components that is being used for the nudging. Using a game engine greatly help the efficiency of the work being done. Since the virtual graphics part of this project is quite small there was not a big emphasis placed on the game engines maximum capability, rather there have been focus on which game engine is easier to work with and provide sufficient support to help the work get done quickly.

3.4 Hardware and Software

3.4.1 Meta quest and Varjo

To see what the participants are seeing a VR/AR headset is being used with passthrough technology and with a video transmitting API. Looking what kinds of glasses on the market that has these requirements, the headsets chosen to test are Meta quest pro [11] and the varjo Rx-3 [12] glasses. Some tests include the screen quality, how easy they are to develop for, eye tracking and passthrough, if they are mobile or not, battery time or need of computer.

3.4.2 Meta quest

Metas meta quest pro is Metas premium headset giving the user a lot of mobility by not being required a direct connection to a computer. The meta quest pro gives full colour passthrough using 5 cameras [13], this headset features stereoscopic MR passthrough which means that a combination of virtual objects and the real world is possible. Both eyes of the meta quest pro have an advanced VR LCD technology that can deliver 1800 x 1920 pixels per eye, this resolution could be of use when using the set when looking at the real world through the glasses. The use of controllers is optional with the quest pro having hand tracking.

Figure 9. Picture of the Meta quest pro

One more requirement for the glasses is eye tracking witch the quest pro can do [13]. The quest pro comes with inward facing sensors which makes the eye tracking possible. This eye tracking can be used as an input as well [13] making it so it would fit great with this project.

Meta also has a development hub that offers tools, accelerate debugging letting the user to capture videos, screenshot or even cast directly to a computer

The meta quest pro gives the following features that would be needed.

- Colour passthrough
- Eye tracking
- Mobility, no need to be directly connected to a computer
- Good screens
- Developer center

3.4.3 Varjo

The varjo Xr-3 is a virtual and mixed reality headset made by Varjo. Varjo markets this headset as tailor made for professionals [12], working with professionals in different industries to make most advanced headset at the time now having the new Xr-4 being their new product. Varjo is known for having human eye resolution running at 1920 x 1920 resolution and a refresh rate of 90 hz. This resolution being marginally better than the quest pro. The rx-3 also comes with integrated eye tracking that is marketed with a speed of 200hz. Varjo also mentions their "ultraleap" which is their hand tracking which they claim are the fastest and most accurate tracking with superior performance.

Figure 10 picture of Varjo headset

The varjo Xr-3 being so powerful would make it not so mobile making it so a computer connection always would be necessary. For mobile applications this would become a con for being restricting.

Varjo has a lot of documentation for both unity and UE5 even giving examples to make it easier to understand and give an easier hand on experience.

The varjo Rx-3 provides these features that was needed:

- Colour passthrough
- Eye tracking
- Good screens
- Developer portal with documentation with different software like unity and UE5

3.4.4 Unreal Engine and Unity

To create the virtual objects for this project there are two game engines that have been considered. Unreal Engine and Unity are the two most popular and free to use softwares for game development. While they both do similar things there are some differences that needed to be thought over.

3.4.5 Unity

Unity comes with a more user-friendly interface compared to Unreal Engine which is its biggest argument for using it for this project. Its graphics can be perceived slightly worse than what Unreal Engine, but it uses C# for development which is an easy programming language to learn.

3.4.6 Unreal Engine

Unreal Engine is arguably the most advanced out of the two options which is likely why it is popular for game development and is used regularly for AAA games because of the high-quality visuals it can produce. With this comes a learning curve that can feel very steep and arguably a waste of time if the project calls for simpler animations. Unreal Engine uses C++ for development and to make this easier for the user there are blueprints to make the development process more visual. This is ultimately what ended up being used for the project, largely because of the bigger developer community as well as existing experience with the software within the team.

4 Experimental setup

No personal information was collected from the participants, the only data collected was the recorded times to finish the puzzle and the number of errors made. The participants in the experiment consisted of people between the ages of 22-40, all of whom are students at University West and studying computer engineering. All of the participants had at least tried virtual reality before, mostly for gaming.

Figure 11 picture of the test area

The experiment would follow the flow shown in figure 1, a user was seated in the test area as seen in figure 11 and were then instructed to put on the VR headset. Participants were first put in a test world where they get used to being in VR and also get use to grabbing the individual pieces, they were kept in the test world until they felt comfortable enough to do the experiment. They were then moved to the next VR world where the real test will take place, testers check if they are in the right place in the test area and also that they are placed right in the VR world. When they were ready the participants would start the puzzle and the test handlers started a stopwatch and counted the number of errors that they made during the duration of their puzzle. If the participant was part of group one, they did the cloud puzzle first, see appendix D, whilst being assisted by the testers. If they were part of group two, they started with the cow puzzle with DGS activated. If the participants did not touch a piece of the puzzle for 5 seconds, the test handlers would input the which piece was next in line to the DGS and nudge the participant to a certain piece. If they looked at the piece the DGS would stop.

After the first puzzle was completed and times and errors were noted the next puzzle world was started and the same as above would happen again but without help from the DGS.

As shown in figure 12 the puzzle is going to be placed in the air in front of the participant, where the participant picks what piece they wanted to use from the two pedestals seen in figure 8 and place the pieces in front of them in the air. More of the puzzle and videos of a run through can be seen on the public github for this project [14] and all the files required for recreation.

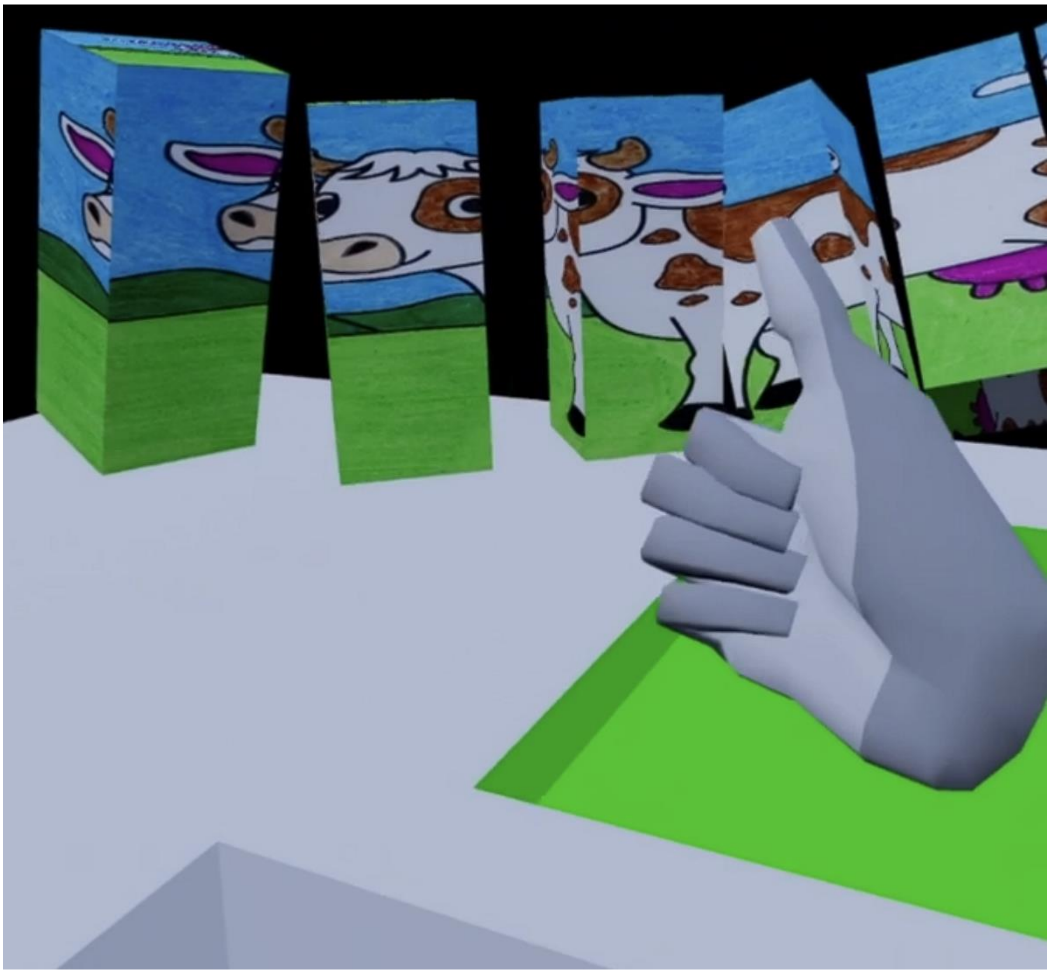

Figure 12 View of what it looks like for the participants

5 Results

The results presented below are of both quantitative and qualitative nature. The quantitative part of these results are the recorded times being shown in graphs along with average times. The qualitative part of the results is about the DGS efficiency where the observations during the tests are discussed as well as if the system was subtle enough.

5.1 Time results

The graphs in this chapter shows the duration of completion for the participants. The participants were split into two groups of six, one group is doing a puzzle with the DGS and the other group is not using the DGS,

5.1.1 Completion time - Cloud puzzle

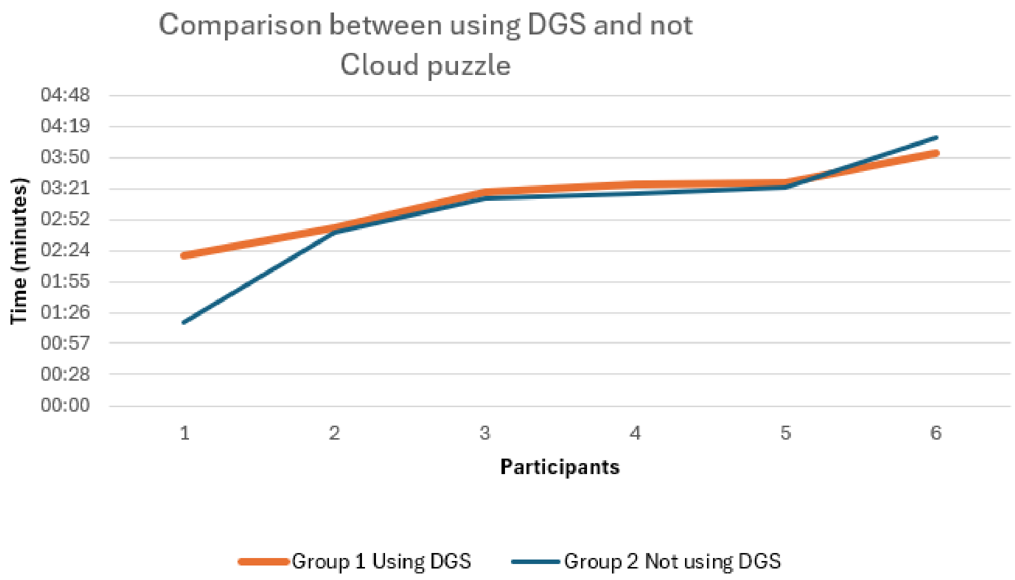

Figure 13 Graph depicting the difference in time between not using and using DGS.

Average time with DGS was 3:11 minutes
Average time without DGS was 2:59 minutes
This puzzle can be viewed in the appendix D.

5.1.2 Completion time – Cow puzzle

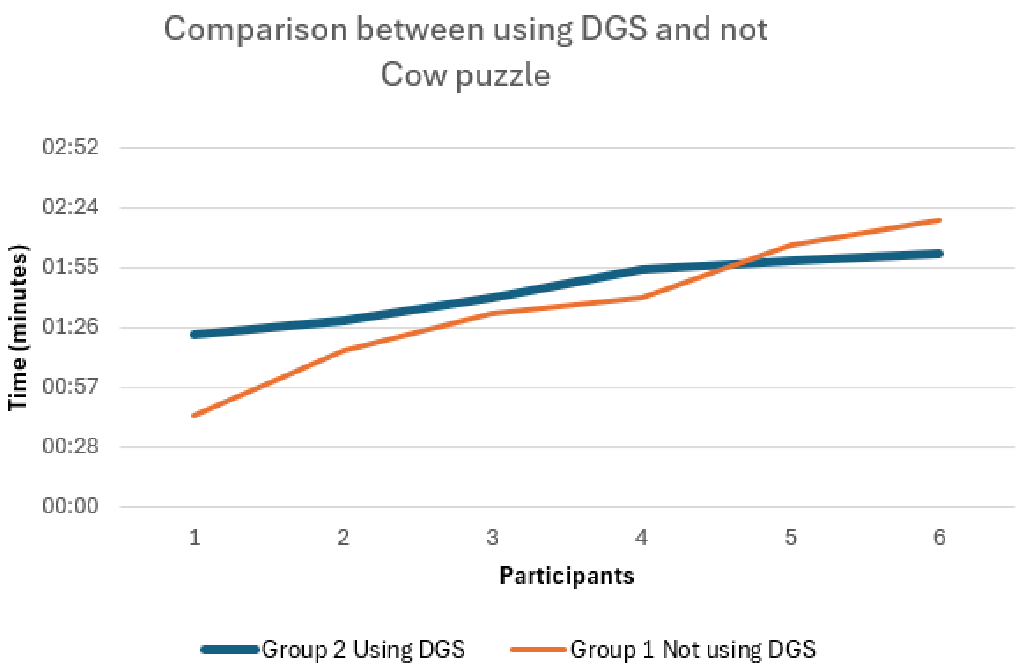

Figure 14 Graph depicting the difference in time between not using and using DGS.

Average time with DGS was 1:44 minutes
Average time without DGS was 1:36 minutes
This puzzle can be viewed in chapter 2 and is figure 2.

5.2 DGS Efficiency

A key part of the DGS efficiency lies in the recorded times, another important aspect is if it was subtle enough and if it helped the user pick the right piece more often than not. To try to quantify this, every user was asked if they noticed the DGS being activated and how often they put a piece in the wrong place.

5.2.1 Subtlety

Only one out of the 12 people who participated in the experiment noticed the DGS. During the tests a clear guidance was observed on participants, when pieces was activated with the DGS, they would go straight for that piece and ignore the piece they were going for initially. This would not be consistent result with some participants not noticing at all and some being very responsive and still not noticing. There are different factors that could explain the mixed responsiveness to the DGS. Some of the factors include the colour used in the flashing, the colour could be too similar to the colour of the puzzle piece for the user to notice it properly, and also the speed at which it flashed could be problematic if it is too fast or too slow for the user to properly pick up on it. It was also not possible for the test users to wear glasses while wearing the VR glasses, this impacted the results in some cases both in terms of noticing the DGS but also the puzzle itself being more difficult because of it.

5.2.2 Errors

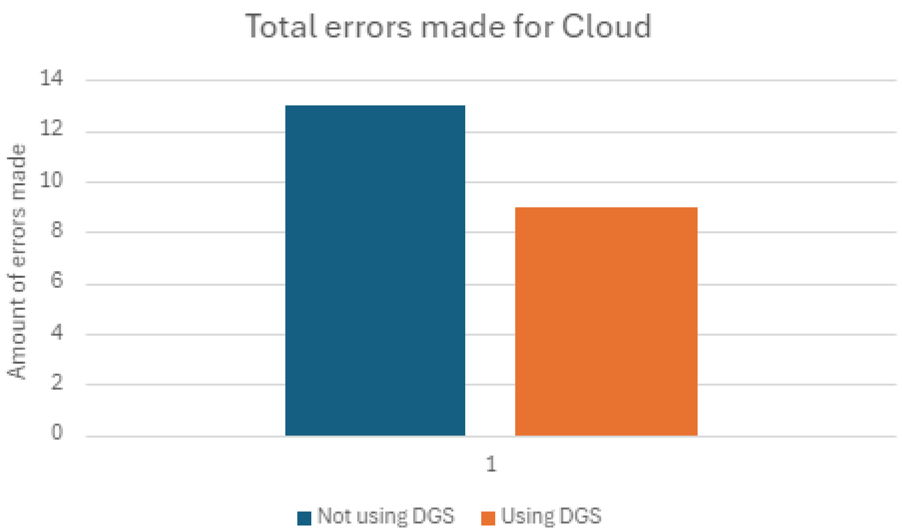

Figure 15 Depicting total number of errors for the Cloud puzzle

The total amount of errors done when the DGS was not active was 13 and when it was activated 9 errors were made in total. A 31% decrease in total errors.

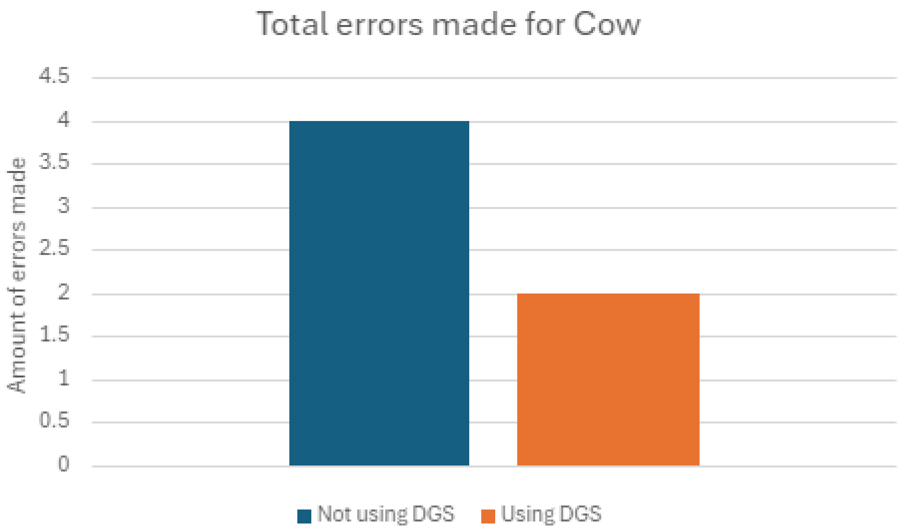

Figure 16 Depicting total number of errors for the cow puzzle

The total amount of errors done when the DGS was not active was 4 and when it was activated 2 errors were made in total. A 50% decrease in total errors.

6 Analysis/Discussion

An important part of the project was making sure that the DGS was subtle and not noticed by the participants. This was measured by asking the participants about their experience during the experiment. The results of the question indicate that only one person noticed anything at all which shows that the DGS was subtle enough in almost all cases.

6.1 Research question 1

While analysing the results it is clear to see that the average recorded time with DGS was slightly longer on both puzzles. The difference is not huge but it is clear that the DGS did not have a big noticeable impact on this metric. So the answer to research question 1 with these results is no.

6.2 Research question 2

For research question 2 we were looking at the errors made by the participant. Here we can see a noticeable difference that indicates that the DGS helped the participant complete the puzzles with fewer errors being made. So the answer to research question 2 with these results is yes.

6.3 Additional observations

When analysing the results there was an interesting observation that was not initially part of the test. The observation was that the line in the graphs looked flatter when using the DGS. This led to a quick comparison of the time spread on all of the puzzles with and without DGS, meaning the difference between the slowest and the fastest time. On the cloud puzzle, a time spread of 1:35 minutes with DGS and 2:51 minutes without. On the cow puzzle, a time spread of 39 seconds with DGS and 1:34 minutes without. This is a considerable difference and could indicate that the DGS helps level out the difference between the puzzle-solving skills of each individual participant to produce a more even playing field.

6.4 Future improvements

For this project several improvements can be made, and different angels of approach can be attempted. A machine learning model was attempted to provide targets for the DGS but due to the complexity of the model and time this ide was scraped. Other forms of nudging could be implemented to test and see if different kinds of stimuli or visual stimuli could be implemented. Sound was an idea by using infrasound [15] frequencies that would make the user uncomfortable or feel weird but again due to time and complexity this was scraped. Complexity of the pieces could also be an improvement that could be implemented, not having rectangular pieces but having triangular shapes. Having a larger participation pool would give a larger dataset and give more of whole picture of this implementation. Most importantly the testing environment would need to be more controlled to have less disturbances on the results. The user might have benefitted from a video demonstration of exactly how the test would be carried out and the hand tracking would need to be either improved or replaced with physical hand controllers. With more time to develop the system, it could also have been carried out in a more mixed reality environment and used physical pieces to reduce this as a factor completely.

6.4.1 Problem with headsets

In the beginning of this project a choice needed to be made regarding both the software and the hardware that was going to be used. The varjo rx-3 and unreal engine 5 was chosen but as mentioned in chapter 3 there was another contender in hardware that could have been a change in both result and development. When testing first started to determine what headset to use some tests were made like testing the screens and the tracking of eyes and hands. Since the main issue with the Varjo during the official test was the hand tracking with not letting pieces go or not grabbing them at all. A controller of some sort would have been a massive improvement to this but the Varjo headset lack controllers, but the meta quest pro does have controllers, but when trying to get access to the developer portal and developer features where not accessible, so none of the meta developer menus in the meta-app that is needed to develop with the headset where available to use during this project.

7 Conclusions

The conclusions from this project are that the DGS has had a considerable effect on the participants proneness to error while remaining subtle enough that it was unnoticeable to everyone but one person. The results also showed that while it did not have any noticeable effect on the average time to complete the puzzles it helped reduce the differences in the fastest and slowest times. This indicates that the DGS can help mitigate the individual puzzle-solving skills for each participant and make the playing field more level.

References

[1] K. Glantz, N. I. Durlach, R. C. Barnett, and W. A. Aviles, "Virtual Reality (VR) and Psychotherapy: Opportunities and Challenges," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 1, pp. 87–105, Feb. 1997, doi: https://doi.org/10.1162/pres.1997.6.1.87.

[2] N. Norouzi et al., "Virtual Animals as Diegetic Attention Guidance Mechanisms in 360-Degree Experiences," in IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 11, pp. 4321-4331, Nov. 2021, doi: 10.1109/TVCG.2021.3106490. (accessed 2024-05-20)

[3] E. Nonino, J. Gisler, V. Holzwarth, C. Hirt and A. Kunz, "Subtle Attention Guidance for Real Walking in Virtual Environments," 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Bari, Italy, 2021, pp. 310-315, doi: 10.1109/ISMAR-Adjunct54149.2021.00070. (accessed 2024-05-20)

[4] Imperial College London, "What is Nudge Theory?," Imperial College London, 2023. https://www.imperial.ac.uk/nudgeomics/about/what-is-nudge-theory/ (accessed 2024-05-14)

[5] W. Lu, D. Feng, S. Feiner, Q. Zhao and H. B. -L. Duh, "Subtle cueing for visual search in head-tracked head worn displays," 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, SA, Australia, 2013, pp. 271-272, doi: 10.1109/ISMAR.2013.6671800.

[6] "Eye tracking," *Varjo for developers*, (accessed 20-04-2024) https://developer.varjo.com/docs/unreal/ue5/eye-tracking-with-unreal5 (accessed 2024-04-15).

[7] "Hand Tracking," *Varjo for developers*, May 03, 2024. https://developer.varjo.com/docs/get-started/hand-tracking (accessed 06-05-2024).

[8] T. Gerencer, "What Is Extended Reality (XR) and How Is it Changing the Future?," Hp.com, Apr. 03, 2021. https://www.hp.com/us-en/shop/tech-takes/what-is-xr-changing-world (accessed 2024-05-17)

[9] Unity Technologies, "Unity," Unity, 2019. https://unity.com/ (accessed 2024-05-14)

[10] Epic Games, "Unreal engine" https://www.unrealengine.com/ (accessed 2024-05-14)

[11] "Meta Quest Pro: Premium mixed reality | Meta Store," www.meta.com. https://www.meta.com/se/en/quest/quest-pro/ (accessed 2024-05-14).

[12] Urho Knottori, "Virtual & Mixed Reality Headsets for Professionals – Varjo XR-3 & VR-3," *Varjo.com*, Jan. 27, 2021. https://varjo.com/blog/twice-the-performance-half-the-price-%E2%80%AFsee-how-the-new-varjo-xr-3-and-vr-3-improve-from-earlier-varjo-headsets/ (accessed 2024-03-03)

[13] "Eye tracking on Meta Quest Pro," *www.meta.com*. https://www.meta.com/en-gb/help/quest/articles/getting-started/getting-started-with-quest-pro/eye-tracking/ (accessed 2024-03-01)

[14] NilsWik, (2024) Final_thesis_manual_asembly_in_virtual_assembly_facilitated_by_sublte_visual_guidance [source code] https://github.com/Nilswik/Final_thesis_manual_asembly_in_virtual_assembly_facilitated_by_sublte_visual_guidance

[15] G. Leventhall, “What is infrasound?” *Progress in Biophysics and Molecular Biology*, vol. 93, no. 1–3, pp. 130–137, Jan. 2007, doi: https://doi.org/10.1016/j.pbiomolbio.2006.07.006 (accessed 2024-05-17)

A. Outline material

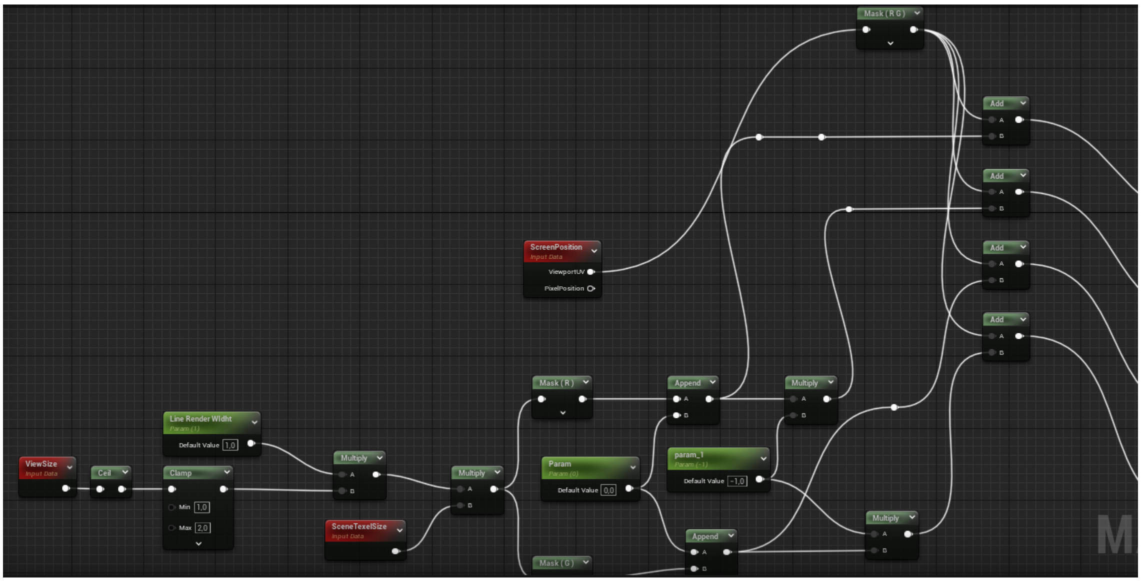

Figure A part 1 of the outline material blueprint

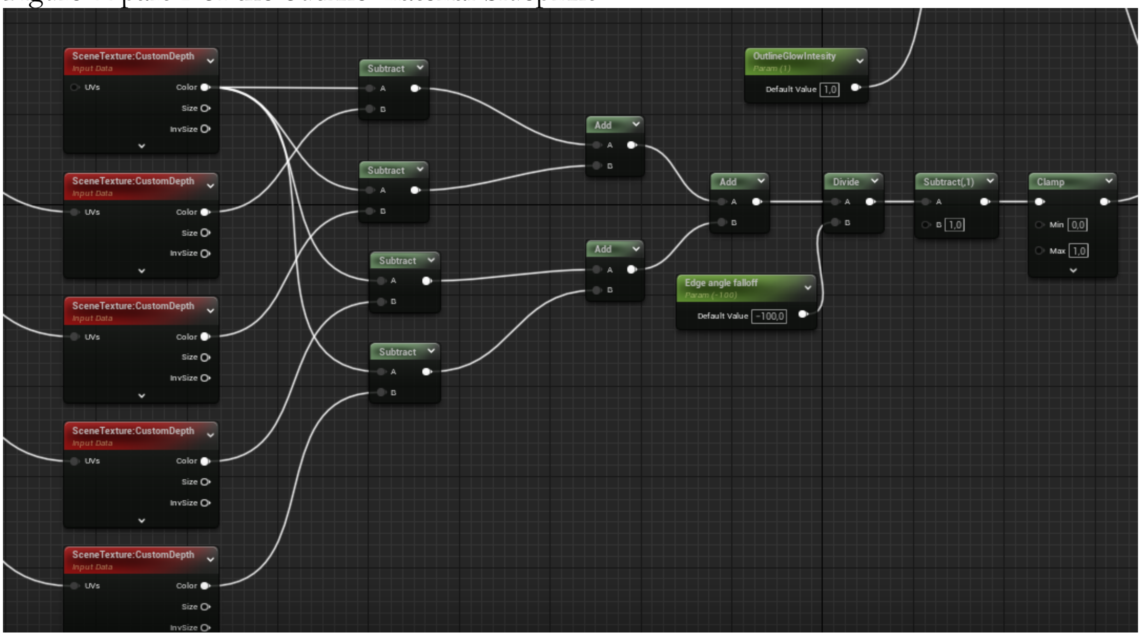

Figure A part 2 of material blueprint

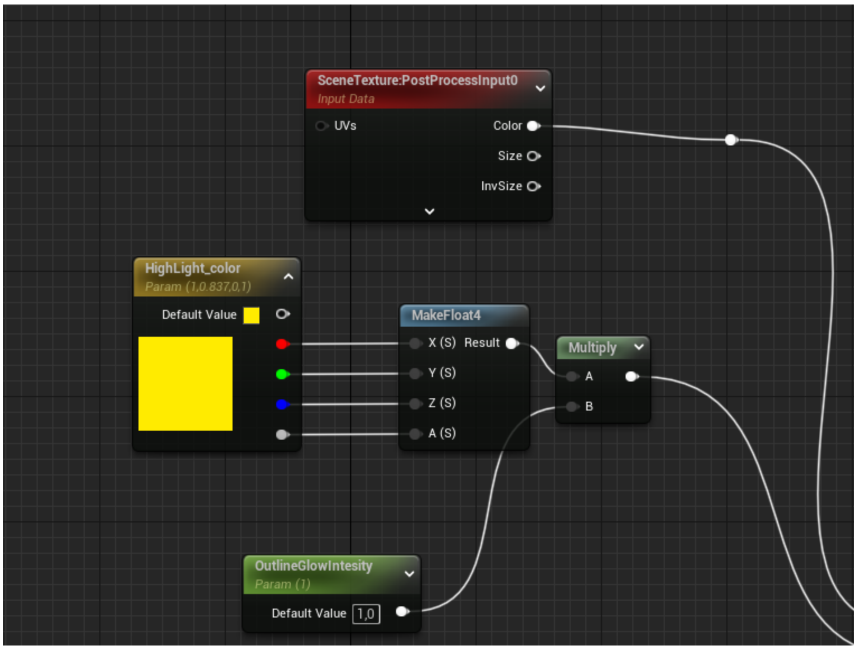

Figure A part 3 of the control canter of the material blueprint

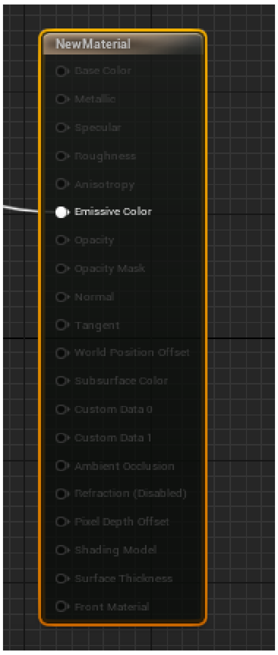

Figure A part 4 of the material getting initialized.

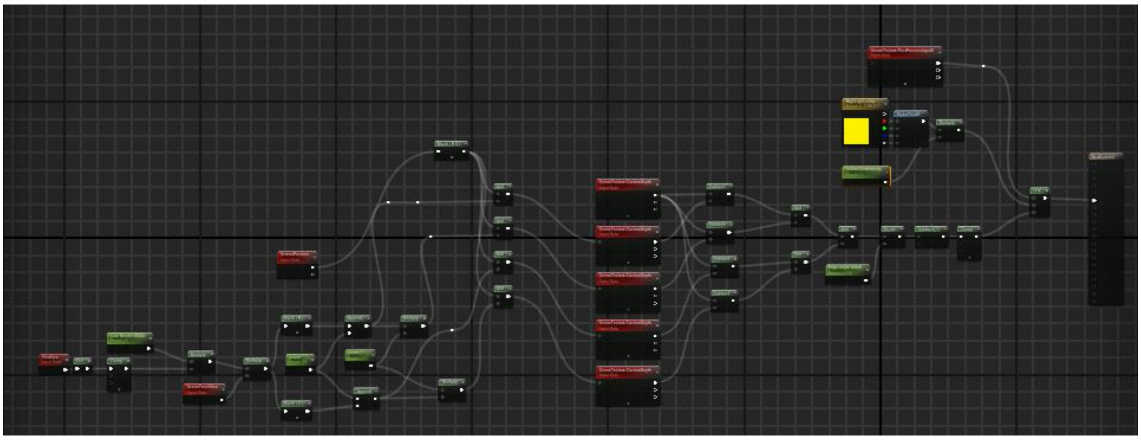

Figure A part 5 whole blueprint in one whole image

B. Blueprint of the DGS

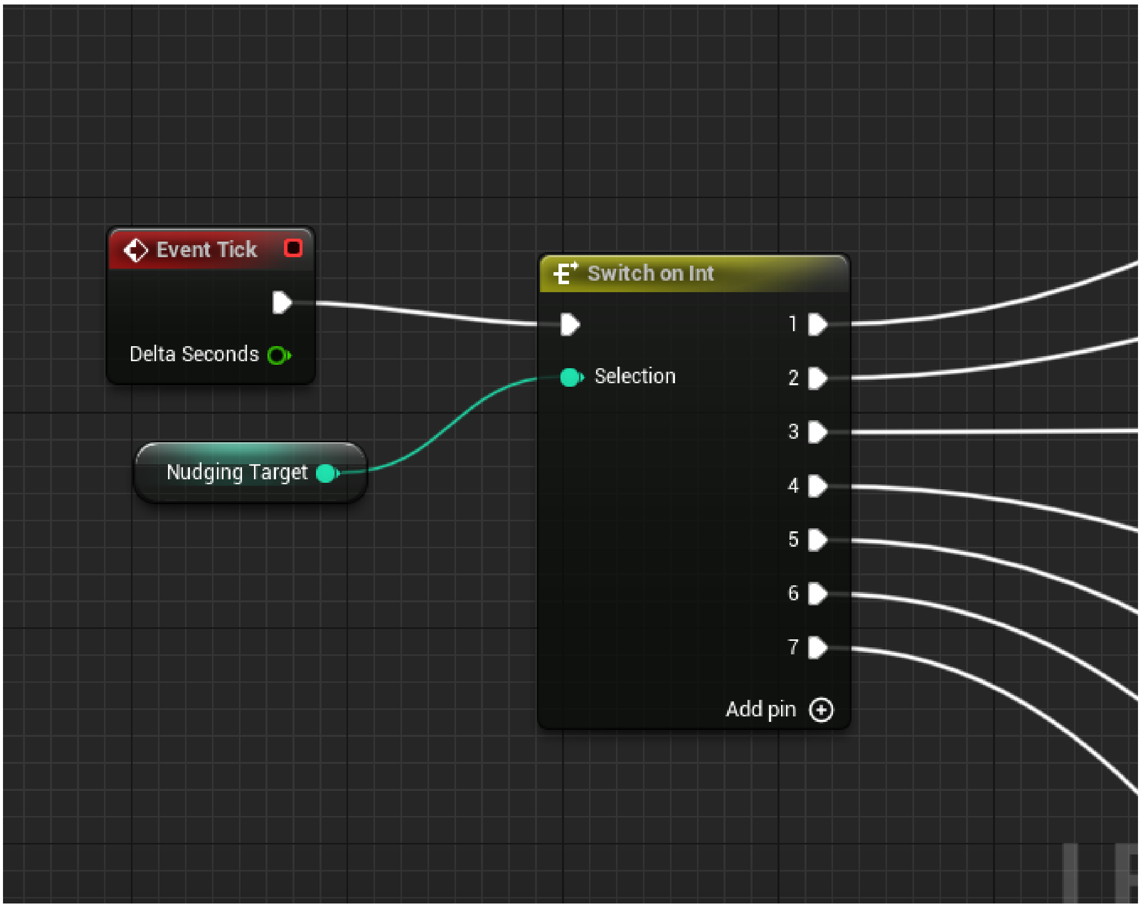

Figure B part 1 of the DGS section where the nudging target get picked

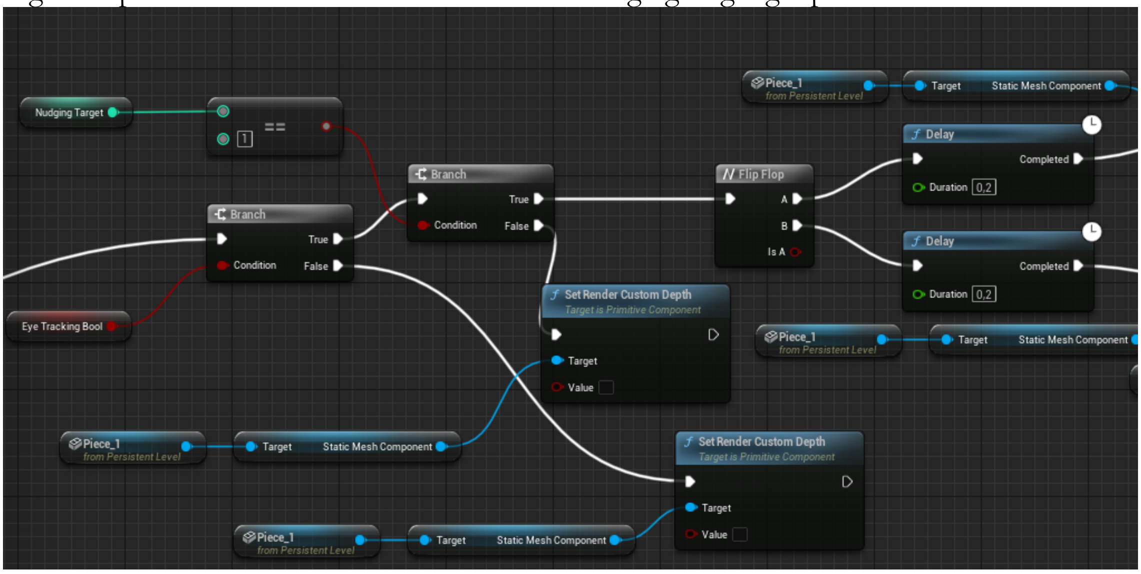

Figure B part 2 of the DGS

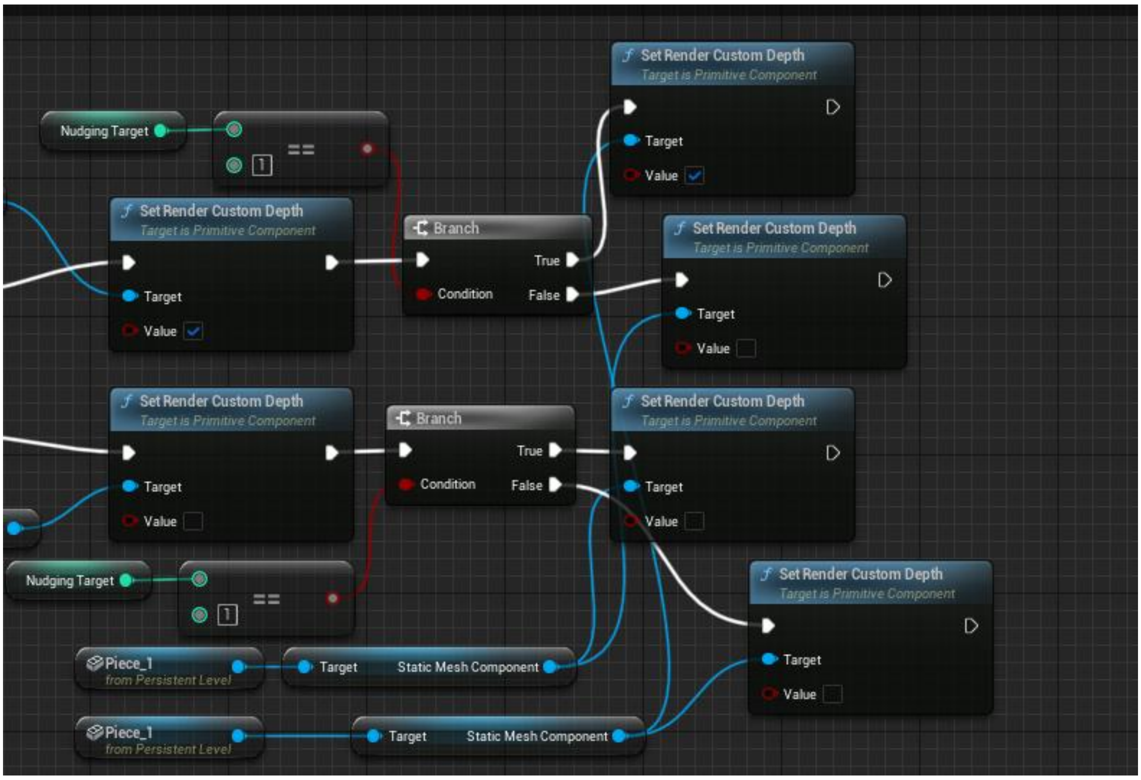

Figure B part 3 of the DGS

C. Eye tracking

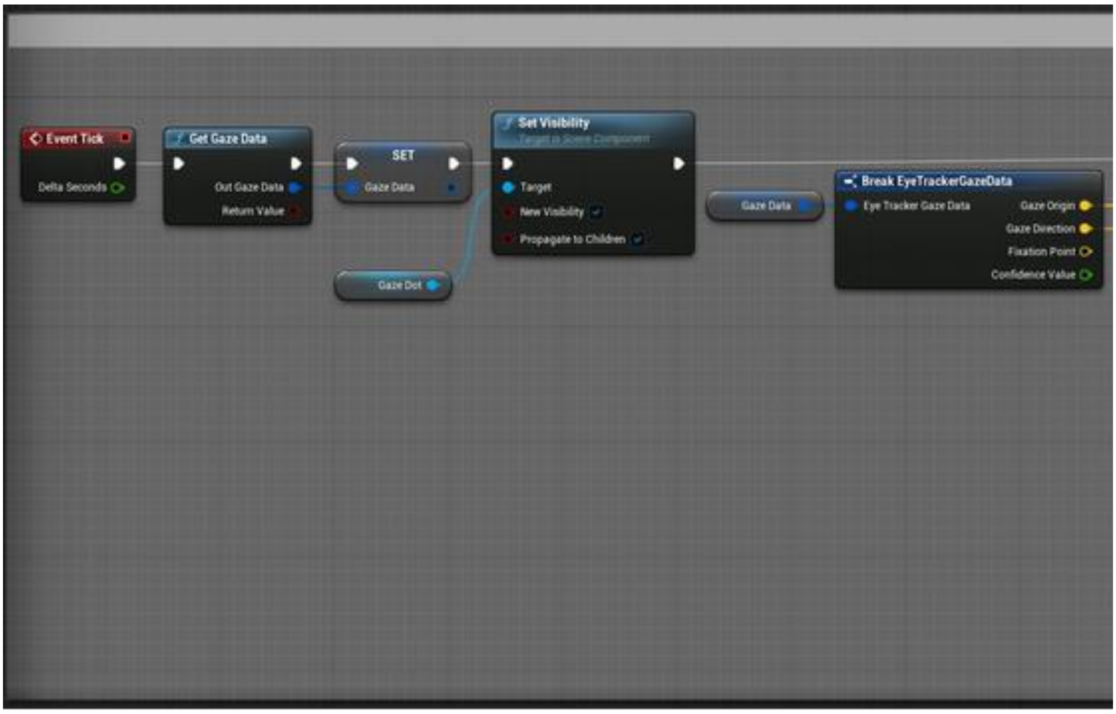

Figure C part 1 part of the eye tracking blueprint

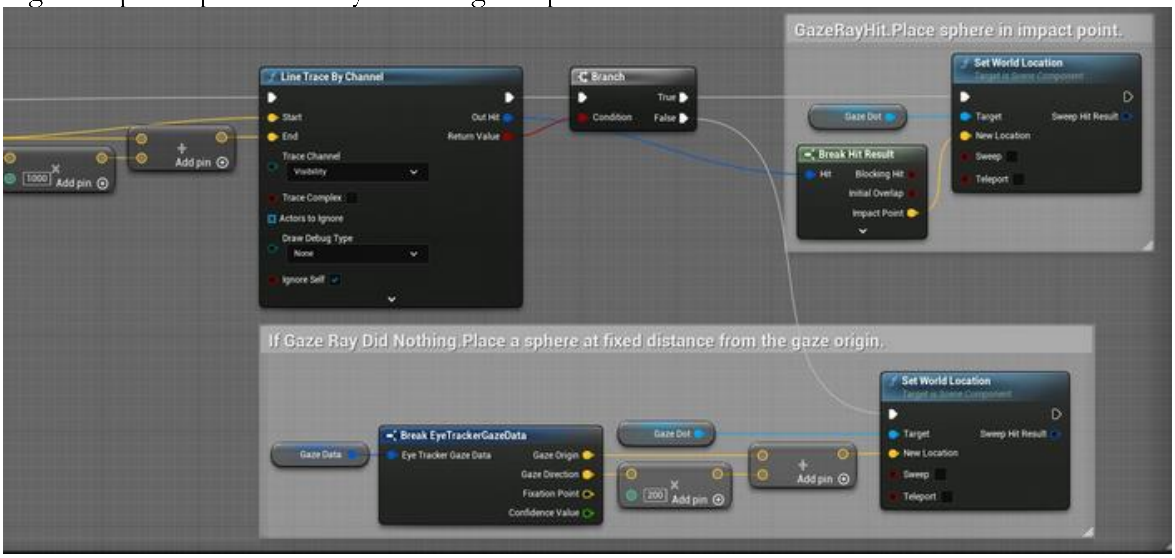

Figure C part 2 part of the eye tracking blueprint

D. Picture of the second cloud puzzle

Figure D the harder cloud puzzle used in this project