Group 306

BIRDOWATCH

Visuel Computing

Exploring the use of head tilt to navigate in a game setting

3. Semester | Medialogy | AAU CPH

Ahsan Masood Iqbal
Frederik Rasmussen
Halipha Ndikumana
Joshua Tobias Bussink
Niels Christian Malte Felix Iuel
Simon Ingemann Axelsen

Department of Architecture, Design & Media Technology
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Birdowatch - Exploring the use of head tilt to navigate in a game setting

Theme:
Visual Computing

Project Period:
3 SEMESTER

Project Group:
306

Participant(s):
Ahsan Masood Iqbal
Frederik Rasmussen
Halipha Ndikumana
Joshua Tobias Bussink
Niels Christian Malte Felix Iuel
Simon Ingemann Axelsen

Supervisor(s):
Niels Erik Raursø

Copies: 1

Page Numbers: 87

Date of Completion:
December 19, 2023

Abstract:

Visual computing's growing environment has permeated several technological domains, including its incorporation into the area of video games. This research investigates and compares two significant technology advancements: eye-tracking and face-tracking. The goal of the iterative research method was to determine the best technology for performance and user pleasure. The central question of the project centers around the utilization of face-tracking in controlling the in-game camera for a puzzle-solving game, with the goal of striking a balance between user satisfaction and usability.
Participants in the SUS-test provided useful insights, resulting to the elimination of the blink mechanic in favor of a head mechanic. Despite the established success of eye-tracking in narrative-driven games, participants rated face-tracking as more useful in the setting of a puzzle-solving game, akin to titles like 'Monument Valley.' The SUS test, which was conducted early on, raised user concerns, specifically the lack of customizable thresholds.
Our pursuit of an enhanced design highlighted the underestimated impact of feedback as a design principle. Incorporating user suggestions and addressing concerns helped that. While input from questionnaires and interviews indicates a positive user satisfaction rate, there remains a recognition of the need for continued refinement to align with user expectations. This project underscores the dynamic nature of design, emphasizing the importance of ongoing iteration and user-centric considerations for the success of interactive technologies in gaming.

Contents

1 Introduction

In an era where gaming technology continually pushes the boundaries of interactivity and immersion. Virtual reality, augmented reality and visual computing add ways to play video games and adds alternative spins on interaction methods, and challenges the traditional ways of playing. The evolution in gaming technology integrates with various genres, and one such genre is the puzzles, which is synonymous with strategic thinking and challenging environments, as seen in games such as the Portal and the Q.U.B.E franchise.

Visual computing is a broad, multidisciplinary field that includes computer vision, computer graphics, image analysis, and visualization. It emphasizes effective visual data management, investigation, and interpretation through the use of insights from perceptual psychology and human-computer interaction [30]. The field has evolved significantly over time, moving away from the early emphasis on low-level techniques and the need for expensive supercomputers to display datasets. The advancement of visual computing demonstrates its dynamic adaptability to the changing scene of technology and information processing by addressing contemporary issues such as the effective representation of massive datasets and the integration of graphics hardware with displays [3]. This broad topic involves a wide range of techniques and technologies used in the creation, analysis, and presentation of visual data, including the critical aspect of image processing, such as face tracking [27]. Face tracking is a technological advancement within the world of visual computing that through a module collects data of the facial landmark. The collection of the data may serve different purposes like for emotion detection or to identify perpetrators through CCTV cameras. There are many possibilities in the context of face tracking [26].

Puzzle games open doors of the exploration for the intersections of face tracking terminology in the captivating world of their genre. While face tracking offers a new game interaction that create an alternative and diverse gaming experience. This led into the first problem statement, which state the following:

Initial problem statement

How can a game through face tracking affect the user experience in puzzle games?

The project aims to integrate intuitive game control mechanics that challenges the current trends of video gaming. The core objective of this venture was to develop a game that utilizes a non-traditional input method, especially, head-tilt to alter the camera angle within the game environment. This endeavor aimed not only to enhance the gaming experience, but also to explore new ways of player engagement.

At the start of the project, the potential of eye tracking was recognized as a game mechanic, envisioning a gaming experience that was more intuitive and immersive. Preliminary A/B usability testing revealed critical insights: while the concept of eye tracking, specifically a blinking mechanic, was compelling in theory, its practical implementation posed significant challenges. This discovery led us to pivot our focus solely on the head-tilt mechanism. As the project progressed, the head-tilt mechanic emerged not only as a viable alternative but as a robust method of game control.

In the culmination of the project, it was discovered that while the head-tilt mechanic was effective, there remained opportunities for refinement. Adjustments in user interface feedback and the fine-tuning of threshold levels were identified as key areas that could further elevate the game's playability and user engagement.

2 Analysis

Continuing from our initial problem statement, the key to creating a puzzle gaming experience through face tracking lies in in-depth research. The research of game design is important to understand for the flow of the game. It is essential to investigate the processes behind eye and face tracking technology with possible uses in puzzle games, and embodied cognition is key to understand how the users acts. Lastly, state of the art puzzle games and alternative interaction methods will be analyzed.

2.1 Verification of Problem Area

The verification of the problem area in this context involves establishing the significance and implications of the project. This subsection examines eye and face tracking in image analysis and the elements that constitute engaging puzzle games. The goal of studying these areas is to identify the problems and opportunities that can be accomplished.

2.1.1 Face interaction

In the field of computer vision and image processing, there are many sub-fields that each contribute to the advancement of technology. Among them is face tracking, which serves as an important mechanism used to identify and then monitor the changes in the movement and location of an individual's face inside any given video stream or image. This complex process is mainly dependent on the use of advanced face recognition and detection algorithms, which are specifically designed to detect and constantly monitor the movement and track of facial attributes, including the eyes, nose, and mouth, in real-time [8].

2.1.2 Game Design of puzzle games

Engaging in gaming is a source of entertainment for many, and puzzle games, are no exception. The essence of puzzle games is rooted in the pursuit of enjoyment. This chapter delves into the strategies and considerations that contribute to creating a truly enjoyable and entertaining experience within the realm of puzzle games.

Something that differentiates puzzle games from most other games, is the requirement for a solution. According to Kim, S. (2008) in "*What is a puzzle*". In *Game Design Workshop: A*

Playcentric Approach to Creating Innovative Games (pp. 35-39) it is stated that to make a fun puzzle you need 3 elements: [21]

- Novelty is the first element. Puzzle games that play around with altered everyday rules, like symbols and shapes bent of viewed from new angles, and create new rules. This helps the player solve the puzzle due to the bridge to real life [21].
- Perspective in puzzle games is an interesting and enticing way to solve puzzles, due to needing to change the perspective of how you interpret the big picture, and can be a useful skill outside of puzzle solving. This has also proven to help learning in educational games [22].
- Additionally, finding a good level of difficulty is key to making a good puzzle. A puzzle solved too easy will feel disappointing, and a puzzle too hard will feel discouraging [11].

When diving further into difficulty and the importance of 'appropriate challenge' of a puzzle game, it has been discovered in two studies, that challenge in educational games affect motivation [25], [24]. The studies show that educational games with less challenge greatly increased motivation in players than more challenging games. These tests are not interested in the player's perception of challenge, however is interested in games that are considered successful. The following games are reverse engineered to learn about their structure and difficulty. *Portal* [35], *Portal 2 co-operative mode* [36], *Braid* [19] and *Lemmings* [28]

2.2 Related Research

This chapter aims to provide insight into the field of human-computer interaction (HCI). By delving into the various facets of HCI, we explore related research regarding the functionalities associated with eye and face tracking technology. Additionally, it will delve into the dimensions of user-centered design, shedding light on the significance of design principles in interactive interfaces. Moreover, the chapter will help understand how the principles of game design can be integrated with the utmost efficiency.

2.2.1 Face-tracking

Through the use of face-tracking, hands-free computer interaction can ultimately be achieved by using different head gestures for control. There are many types of gestures that a user can use for interaction with a device, and the effectiveness of such interaction hinges on the intuitiveness of gestures, ensuring a natural and user-friendly experience. In their work on hand-free Head Mounted Display (HMD) devices, Yan et al. [39] present different approaches for hand-free input based on head movement and made a table, showcasing the range of interactions to be had.

Features	Explanations for the Features
Movement Tracking	Map the different head orientations to the different positions of the cursors, by ray casting metaphors. (Nouse [22], HeadTilting [15])
Rotation	Rotate the head to the left or the right, with a range threshold to avoid the false positives. (HeadTurn [45], HeadPager [54] and [29])
Leaning	Leaning the Head to the shoulders, with the head facing forward. (HeadTilting [15] and [33])
Drawing Shapes	Drawing special shapes (e.g., circle, triangle) using the head. (GlassGesture [64])
Existing Gestures	Leveraging the existing head gestures, e.g., nodding and shaking. (HeadNod [40])

Figure 1: Table of head-gesture interactions [39]

The hand-free head movement gestures not only work in the realm of HMD but can also be utilized for other types of interactions, such as in a mobile context, as presented in the work by Crossan et al.[10] on "Head Tilting for Interaction in Mobile Contexts" with a focus on how head-tilt can be used as an interaction element in the landscape of mobile interaction and comparing the use of head-tilt in a static and in a walking environment, where it concluded that a static environment is better suited for the stability of the interaction.

Another use of hands-free head interaction was discussed by Liu et al. in their paper "Head-Pager: Page Turning with Computer Vision Based Head Interaction" [33], where the purpose was to investigate the use of a head-paging system that would allow the users to navigate through electronic documents when both hands are occupied. It does that by having a camera to monitor the user's head movement and control the document using that information. Some of the advantages and disadvantages of using Headpager over traditional methods is, by default, it's an interaction which can be used in situations where hands are not available. It also uses a calibration utility that adapts the criteria for each specific user, so the system works most efficiently for a wide range of users, but the standard boundaries for the tilting were 30 percent of the screen for both left and right. When Liu et al. evaluated the system through its usability, it was found out that the system was useful in situations where the hands were occupied, but it was not more preferred because of its lack of flexibility, and it still needed refining.

Facial Landmark detection Facial landmark detection is a computer vision task that can locate and identify key points called landmarks on a face, these key points could be mouth, nose, chin or the eyes, all being strategic anatomical locations. The facial tasks that can be achieved though the use of analyzing the key landmarks can serve as an identifier for providing information about the face. The use of facial landmarks' detection can be used beyond point identification, like to track movement and extract information about expressions or geometry [26]

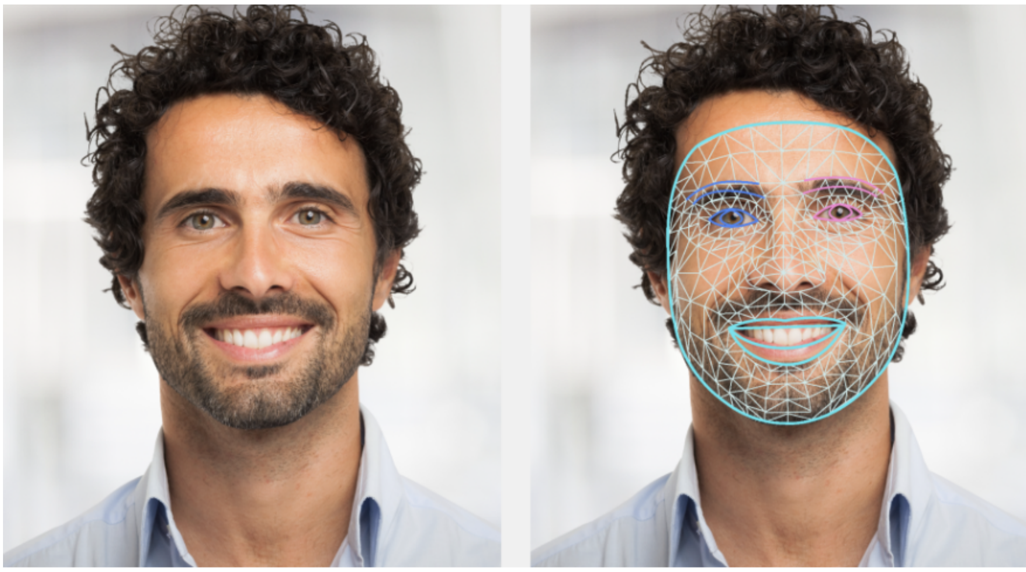


Figure 2: Showcase of Googles Mediapipe facial landmarks [26]

2.2.2 Eye-interaction

Eye tracking is a sensor technology used to follow and record the movement of a person's eye on a monitor in real-time. The technology of tracking then converts the eye movement information into data that can be processed. Some of the data that gets recorded is eye movement, positions, and points of gaze [15].

Anatomical aspects of the eye:

The way that the human eye processes visual information is through a complex system, where the retina is the center of this process. The retina contains light-sensing cells called photoreceptors; these consist of cones and rods. The cones are for detecting colors and detail in

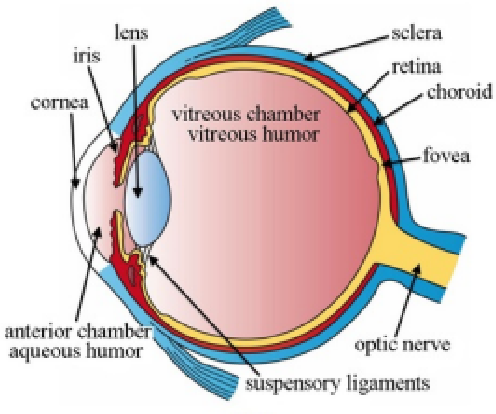


Figure 3: Anatomy of the eye [18]

well-lit conditions, and the rods are more sensitive to light levels and make the eye see in low-light conditions.

In the front of the eye are the cornea and the lens, which both have the use of focusing the incoming light and ensuring the light rays converge onto the retina. The iris is the colored part of the eye, and it is what helps adjust the pupil size, by either constricting or dilating it, and then controlling the amount of light entering the eye. The pupil is the dark spot in the middle of the iris [6].

Some of the movements of the eye that the eye-tracking technology tracks are[13]:

- Fixations: This is a brief pause in eye movement where the eye remains still.
- Saccades: Rapid movement that shift the gaze from one point to another.
- Smooth pursuit: Continuous tracking, usually when the eye is tracking a moving object.

There are many types of eye-tracking technologies on the market, each with their own benefits and disadvantages. There are two widely adopted options: being specialized glasses and screen based, where the screen-based category can be further divided into a hardware and software solution. The screen based software solution is the most cost-effective since it only requires a webcam and will be the focus for the rest of this chapter.

When monitoring the eye, some of the common quantitative metrics that you get (the type of data you get is dependent on the specific technology used) are:

- Gaze point and plot: The point shows where the person is looking, and the plot shows the scan path, which is the order of fixations and saccades.

- Areas of Interest (AOI): Regions of the screen that are of interest to be analyzed
- Different fixations measures

There are different fixation measures, such as:

- Fixation time: The duration of the user's gaze on a specific point
- Time to First Fixation (TTFF): Time for the person to fixate on a visual stimulus
- Average fixation duration
- Fixation count: Inside the AOI

For eye-tracking technology to be an effective tool to understand gaze behavior, there need to be ways to visualize it. Heat maps are a way to show and analyze an individual's focus and attention. It conveys this data it gets from the tracking into color intensity in a grid and gives more intense color for areas with high a fixation count or fixation duration. The way that the intensity of the color is generated is by giving each cell in the grid a value based on where and how long the gaze is. It then has a gray-scale heat map, which is then converted into colors, with green being the lightest and red the strongest. [14]

In the paper "The Emergence of EyePlay: A Survey of Eye Interaction in Games [37] it discusses the mechanics of eye interaction that can be utilized in a gaming context and the various effects these different eye movements can have. Fixations can, together with a threshold that is longer than the normal fixation length, be used to detect a possible attention pattern. Saccades can be used in a sequence, called eye gestures, to match a predefined pattern that activates a function within the game. This could trigger an attack or healing spell. Eye gestures can also be used in a user interface, as a way to navigate in the options' menu. Smooth pursuit in a gaming context can be particularly useful for identifying if the user is following a moving target by comparing the user's gaze point and the trajectory of the target.

Some common problems with using eye gestures as a way of interaction are both fatigue from the more unnatural saccades and the fact that they can also overlap with the user's natural search pattern.

Eye detection algorithms / methods

When it comes to algorithms for eye tracking, they can generally be divided into two main categories: those using machine-learning-based detection and those using traditional eye detection.

A practical example of one such pipeline of traditional eye detection, where you go from a photo or still frame to detecting the eye with the use of different algorithms, is presented in the work by Swirski et al., outlined in their paper "Robust real-time pupil tracking in highly off-axis images." [32]

In their study, Świrski et al. combine three key techniques for their pupil detection pipeline: a Haar-like feature, together with k-means segmentation, and a Random Sample Consensus (RANSAC) algorithm to estimate pupil detection and tracking. This is just one example of how a pipeline can be implemented, and for real-time applications, different approaches can be made, with other techniques, but still achieve the same results.

RANSAC is a method that can be used in the field of computer vision and image processing that can be applied to an eye detection pipeline and is suited because of its robustness in handling out-liners. In general, the RANSAC algorithm is a parameter estimation approach that, given a set of data points, can be used to fit a model to them [12]. And in the case of eye tracking, this is an elliptic model of the edge pixels of the pupil. Another similar method used for detecting and tracking the pupil is Binary Large Object (BLOB) analysis. This is primarily used for identifying connected regions of interest in an image. It does that by grouping pixels based on their same similarities.

The first step in their pipeline is to use the Haar-like center-surround feature for a given radius to approximately find a square around the pupil. They can do this because an eye can be approximated to a darker spot, the pupil, surrounded by a whiter area. The picture of the

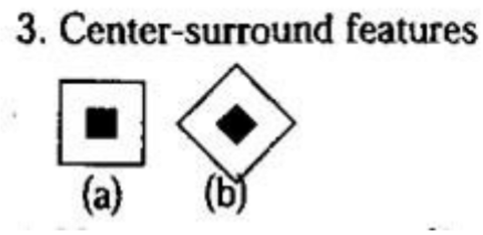


Figure 4: Haar-like centre-surround feature

area of the pupil is then segmented using k-means clustering of its histogram of pixel values. The purpose of this is to separate the foreground (the pupil) and the background (the rest of the eye) by using a dynamic threshold. The reason they use k-means clustering rather than a fixed manual threshold is that it does not give errors with different illumination or lightning because it clusters groups of pixels based on their similar pixel values.

The subsequent step in the pipeline involves using an edge detection method on the segmented pupil, followed by the use of the RANSAC algorithm to fit the elliptic model to these edge pixels, ultimately achieving eye detection [32].

When stable tracking of the eye and the area around the eye is achieved, it can be used as another element for the possibilities of interaction between the user and computer. For a game perspective it can be used as another way of player interaction. It can also be expanded into further mechanics, such as a blink or wink mechanic.

2.2.3 Eye interaction in a gaming context

A promising application in the context of eye-interaction is its use in a game setting, since the implementation of eye interaction technology have become widely available through software (e.g., Google MediaPipe [26]) and can be used as a resource for game design.

Understanding eye movements is useful information for creating a human-computer interaction (HCI) and can be extended to video games. The paper "The Emergence of EyePlay: A Survey of Eye Interaction in Games [37]" discusses the potential for the incorporation of eye tracking in game design. It comes up with different ways that the eye can interact with a game, with two main distinctions being discrete and continuous movements. The discrete-only actions are actions that use the eye as a trigger for an action; this can both be incorporated with a blink or wink mechanic that can be used for e.g. selecting options or as an eye gesture that is a sequence of saccades that correspond to different options. Continuous control is where the game's environment is controlled based on the user's gaze and can be implemented in two ways. Firstly, by having a central viewport that ensures that the user's point of focus is at the center, the game's environment is dynamically responding to the user's gaze. The other way a continuous control can be implemented is by adjusting and rotating the viewport based on the user's gaze, where if the user is looking to the side, it rotates the camera in that direction, so the perspective of the game corresponds to the gaze. There are many different ways to use these interactions inside a game, and the paper comes up with five core mechanics: navigation, aiming and shooting, selection and commands, implicit interaction, and visual effects, a comprehensive overview of the different game mechanics that can be implemented using eye interaction

2.2.4 Embodied Cognition and Interaction

Embodied cognition places a strong emphasis on the role that the body's interactions with its surroundings and the relationship between the mind and body play in the creation, growth, and comprehension of knowledge. The idea that the body not only has connections to the brain but also influences the mind. Embodied interaction is the term used to characterize how the brain and body interact and how this influences how meaningful interactions with technology are shared, created, and manipulated [23]. Which is important when it comes to visual computing. In the article: *Contesting control: Journeys through surrender, self-awareness and looseness of control in embodied interaction* [5] Their second case study, the 24 test participants are wearing a head-worn electroencephalogram (EEG) sensor for control. While watching the film "The Disadvantages of Time Travel," the EEG harnesses the electrical activity of the test participant's brain. The film has an interactive narrative that lasts 17 minutes. The film is controlled by the test participant's blinking and EEG readings. Through this, the test participants become "co-creators" since each time they blinked, it shifts the scene between distinct layers of video: the protagonist's objective reality and their dream-like internal perspective. Participants reported using blinking deliberately, with an average of 5 blinks per minute. However, challenges arose as participants struggled with fully utilizing the product due to factors such as eye fatigue, dryness, reactions to external stimuli, or just forgetting about their control over the product through blinking. Through this, the test participants lose control of the experience.

2.2.5 Design Principles

The purpose of design principles is to aid interaction designers in crafting experiences with the user in mind. This field consists of generalizable abstractions intended to guide designers in enhancing various aspects of their designs. Helen Sharp, Jennifer Preece, and Yvonne Rogers delve into the essence of these principles in the book 'Beyond Human-Computer Interaction.'[31] The principles encompass feedback, which involves providing an immediate response to user actions; visibility, ensuring the findability of relevant information; and consistency, maintaining coherence in patterns, among other key principles. This chapter aims to provide a concise summary of these principles.

Feedback and Visibility

The visibility principle is important when it comes to facilitating user anticipation and comprehension of both their current and forthcoming tasks. This includes placing elements strategically, ensuring that they're easily accessible, and by a layout that encompasses a visual flow.

Visual flow is achieved by integrating visual hierarchy. Visual hierarchy is instrumental in directing the user's gaze effectively. Scanning patterns, such as the Z and F patterns, are common practices in designing for the user's visual engagement.

Related to the concept of visibility is feedback. Feedback involves sending cues to the user in response to their action. These cues may vary visually, auditory, or as a combination of the two. The purpose of feedback lies in enhancing user navigability by providing insights into their completed actions.

Consistency

Consistency is a fundamental principle in interaction design, emphasizing the application of similar operations and design elements throughout the system. The benefits of maintaining consistent interfaces are profound, as they ensure users can familiarize themselves with the structure, making the overall experience easily learnable.

Consistency extends to elements such as buttons, placement, and color schemes. When users encounter elements with which they are familiar, they can anticipate their functionality, thereby reducing cognitive load and enhancing the overall intuitive experience.

Affordance and Constraints

Affordance and constraints are also principles in interaction design that shape the user experience in distinct ways. Affordance involves designing elements, so their functionality is intuitively suggested.

For instance, a clickable button looks press-able, creating a visual language users can easily understand.

On the flip side, constraints act as intentional limitations, preventing unintended actions. Constraints help maintain users on the right track, reducing confusion and conserving a clear path for the user.

Balancing the nature of affordances and constraints is key to creating an interface that feels guided.

2.3 State of the art

This section delves into the innovative aspects of two notable video games, Monument Valley and Before Your Eyes, both of which have significantly influenced the contemporary landscape of game design and technology.

2.3.1 MonumentValley

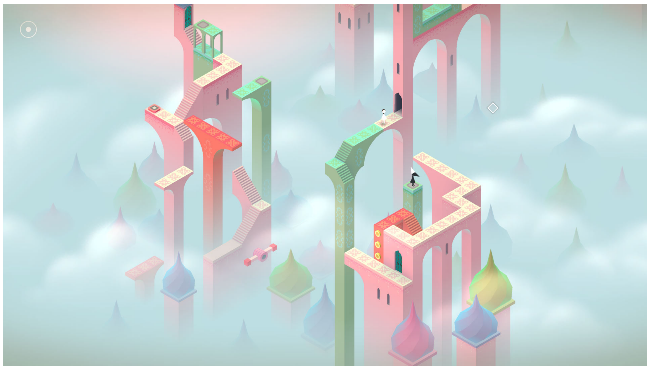


Figure 5: Monument Valley level with the isometric camera angle[34]

The game Monument Valley is a puzzle adventure game that was developed by Ustwo Games and released in April 2014 for smartphones. The game has an isometric camera angle, where you move around the avatar, Princess Ida. The art style is inspired by a minimalistic approach, which in turn helps with simplicity.

Gameplay of Monument Valley

The gameplay mechanics of Monument Valley focus on manipulating the environment on the different platforms by being able to move and rotate different elements for the avatar to cross. The games blend optical illusions, architectural design, and geometry to create puzzles for the player to solve to be able to progress in the game. Slowly, through playing the game, it gets increasingly harder per level [ref. 2.1.2]. The gameplay is easy to understand, and makes it enjoyable for a wide audience [17]. However, this is also criticized in a Steam Review, this issue can cause apathy due to the lack of challenge [ref. 2.1.2].

Inspiration

The rotational aspect to solve puzzles in monument valley is used as inspiration for the project, as well as the level-design aspect that progress the game though platforms. This was quickly inspired to be used as an element in combination with facial tracking to further develop the rotational mechanic. The camera angle used in the game was also inspired to be used in the project due to it creating an overview and could easily be used in combination with. Because Monument Valley uses a more abstract and minimalistic art-style, it would not be preferable to use design the product with to include of facial tracking to have the same art-style, but instead having a more clear and composed platform, where the player can explore the platform, and not having limiting path options. Its consistency in its levels is also something that the product will take inspiration from, so that there is a connection between the levels. The perspective change that the game incorporate in the solving is also something that the facial tracking would control.

2.3.2 Before Your Eyes

Figure 6: Before your eyes [4]

Before Your Eyes is a video game developed by GoodbyeWorld Games that was released in 2021. The game describes itself as an emotional first-person narrative adventure, where the player is in control of the story and is able to affect its outcomes. The player controls the game by using their eyes.

In the video game 'Before Your Eyes,' the player's experience begins with the calibration of the webcam. The game's core method of interaction involves monitoring for blinks through the webcam, providing a rather unconventional way of interacting. Before the player can proceed, the system must ensure that the method of interaction performs seamlessly. Initially, the system checks if the webcam is operational. Following that, the player must answer yes or no to whether or not they are wearing glasses or anything else that might complicate the experience. Lastly, the player gets to test the overall blinking experience, calibrate the sensitivity, and optimize the threshold. The game also lets the player know that they can return to the calibration screen at any time by pressing the easily accessible Space bar button, in case the system stops detecting blinks. However, even with all of these precautions, the most complained about feature is the eye-tracking not working as intended. In the store-page, Negative Steam Reviews can be looked at for more in depth. This points towards lack of development of web cameras and eye-tracking combined.

Gameplay of Before Your Eyes

Every time the game registers a blink from the player, the game advances the story. Through this, it allows the player to explore their avatar Benjamin's life and make decisions, either intentionally or unintentionally, that shape the story of the game [ref. 2.2.4]. Eye-tracking is the most essential part of the game. Through this visual computing, you control the game, with this the player is able to reflect on and experience significant time in the avatar Benjamin's life, and making choices that interact with the story. The art style uses a low poly approach, which helps production time [16].

Inspiration

The use of eye-tracking as one of the main game play mechanics was a big inspiration towards the final product for this project. It gives good merit towards games and eye-tracking being combined due to the overwhelmingly Positive Steam Reviews. However, this also makes the development team of this project take cautioned actions towards eye-tracking system due to aforementioned complains. This in mind, it inspired a different approach to eye-tracking, but still within the limits of visual computing. The art style of Before Your Eyes is low poly and is helps low-cost computers with rendering costs and should provide higher Frames Per Second for the player. The in-game camera's field of view is not inherent from the Before Your Eyes, and will not be used as inspiration. This is also the case for the story telling, and will not be a core object of this project's purpose.

3 Delimitation

The goal of this section is to describe the important conclusions from the research so that a prototype for this project may be developed. A final problem statement (FPS) will also be created to summarize the delimitation.

The research on hands-free computer interaction, such as eye and face tracking, should be utilized as game controls. In order to operate the game, the main focus should rely on either head or eye tracking. Examples such as "Before Your Eyes," which uses blinking to regulate gameplay, demonstrate the possibilities of this technology. These findings point to an approach to a different game control that makes use of head and face interactions, specifically employed as a method for camera control within the game. This innovative use of face tracking technology for manipulating the in-game camera view can offer a unique interactive experience, that could have an effect on the player.

The implementation should be using a cost-effective approach for eye and face tracking for the application. This involves running the program on a low-cost webcam that is frequently used in desktop setups.

The study explores the complexities involved in creating puzzle games. New ideas are emphasized, along with choosing suitable levels of difficulty and creative approaches to problem-solving. The source of inspiration is "Monument Valley," where the fundamental element of solving puzzles is switching between platforms and viewpoints. The player's interaction and engagement with the game world are improved by this creative approach to game creation.

The research also reveals the importance of progressively increasing game difficulty. This approach ensures continuous challenge and provides a rewarding experience for the player. Such a design philosophy is crucial in maintaining player engagement and satisfaction throughout the game.

3.1 Final problem statement

How can a game with an emphasis on puzzle-solving, utilize face-tracking as a method to control the in-game camera view, and simultaneously meet user satisfaction and usability?

3.2 Design requirement

No.	Functional	Non-Functional
1	The user should be able to control aspects of the game through a camera	The program should be able to perform through a standard webcam on a laptop.
2	Game design that showcase novelty	The system should operate smoothly without lag or performance issues.
3	The game's level progression must demonstrate an incremental increase in difficulty, as the player advances.	The game should reinforce a feeling of being rewarded
4	The game should take place on a platform	The face interaction through the camera should be used to alter the platform, to help the player completing the level
5	There should be an intuitive user interface, that guide the player though the game	A tutorial and proper user onboarding that helps the player understand the basic mechanics of the game.
6	The face tracking process should be iteratively improved	The mechanics should reflect the best options from the user testing

1. Through the analysis of face-interaction (2.2.1), which goes into the many different sub-field, where the focus it put on face tracking. Through this, the user is able to control different by using different head-gestures. As seen in the video game before your eyes (2.3.2). Here it is done through the webcam on your computer, where it tracks if you blink to control the game. Here through face-interaction you control the game. This implements a novel way of controlling the game, through face-interaction

2. Through the analysis of Game Design in Puzzle Games (2.1.2), goes into the aspect such as novel, in how to form entertainment with plays around every day rules such as letter or shapes through different alteration. In the video game Monument Valley (2.3.1), we see platform and structures being able to move around with to change the different perspective, to get through the puzzle. Throughout this novelty, it adds a fun way of interacting with the world and its puzzle solving

3. Through the analysis in the state of the art on Monument Valley (2.3.1), we see that each time a level in the game is solved. We see the game get progressively harder with each level. This make the game challenging all the way throughout it, and adds reason to keep playing it, and finding it rewarding to play the game.

4. Through the analysis state of the art, we see that in Monument Valley (2.3.1), that the game takes place on platforms, where the goal is to alter different objectives to pass through the level, with focus on isometric design.

5. Through the analysis section about Design Principles (2.2.5). With the goal of aiding crafting the experiencing with user in mind. It is important to ensure a clear feed and visibility with a layout that encompasses the visual flow. Visual flow is achieved by integrating visual hierarchy to directs the user's gaze effectively. Consistency show the importance of making sure that things look the same, to ensure complete understanding throughout the different elements. Affordance is important for the intuitive functionality and constraints helps maintain user on the right track, for reducing and conserving a clear, user-friendly path.

6. The iterative process that will take place during the implementation ensures that the system that would be used as the mechanic would be the most optimal, and that ensures that the tracking will be the most approved by the users, and the most robust solution. This way, the overall game-play experience will be based on feedback and preferences.

4 Methods

This section outlines the methods that were employed in the study, with a focus on the design and evaluation of a game with a unique visual interaction element. These methods were purposefully created to ensure thorough testing and reliable results.

4.1 Pilot Test

Pilot-testing is an integral part of the iterative design process in the development of the game, that offer insight into the functionality, and potential areas of improvements before the final testing. For the pilot-testing of the product, it was play tested within the development team, where a whole play-through was made, to see if all the functionalities of the game works as intended.

4.2 The iterative process

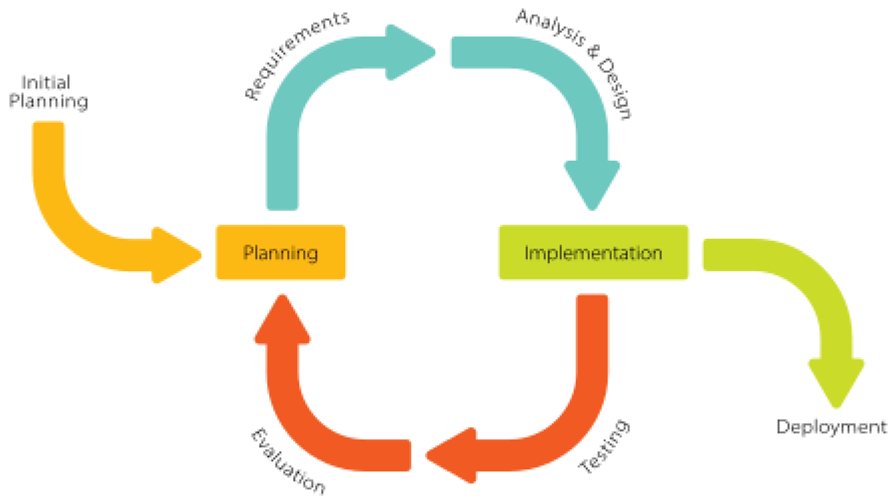


Figure 7: Iterative development model [38]

For the design and implementation of our game together with the visual interaction element, we want to do it in an iterative process, where we do the first usability test in a A/B testing method to determine which course of action we proceed with, being either the head-tilt or the blinking mechanic. Before the testing was done, a hypothesis was made based on our assumptions. The head tilt function is anticipated to be a more consistent and user-friendly approach because of its less involuntary movements when compared to the blinking function, as it was discussed in embodied interaction [section: 2.2.4]. It would have a better correspondence between the input and the response from the output, that being the Gulf of Execution is smaller, due to the players having a more clear way of executing their intentions.

4.2.1 Hypothesis

The Null Hypothesis (H0): There is no significant difference in usability between head-tilt and the blinking mechanic based on the SUS score Alternative Hypothesis (H1): The head-tilt mechanic will result in a higher usability compared to the blinking mechanic based on the SUS score.

4.3 Usability test

4.3.1 What is a Usability test

The process of assessing a product or service by having representative users evaluate it is known as a usability test. Most typically, the test participants will see if they are able to finish the given tasks, while the observers watch and listen to feedback and take notes on the user experience. The usability test will give both qualitative and quantitative data with a questionnaire and feedback. This will allow you to ascertain whether the test subject is satisfied with the product under test and whether any changes or removals are necessary [2].

4.3.2 The Questionnaire S.U.S scale

The System Usability Scale (SUS), is a tool used for evaluating the usability of a certain system, service or a product. The SUS scale has a 10-item questionnaire, where the user rate their agreement on the statement thought the likert scale. Through this, it aims to measure the user's subjective perceptions of usability, efficiency and learn-ability. The test participant's answer is used to provide a usability score ranging from 0 at it lowest, and 100 at it highest. The SUS scale is important because it provides a fast and dependable way to get overall usability and user satisfaction of a system. The SUS test is heavily tested and considered very reliable. Through these insights into the different area of improvement can be discovered, and be improved upon. The S.U.S scores the results from the usability on a scale from 0 to 100, where 68 and above is considered above average, and meets the general usability for most users [1].

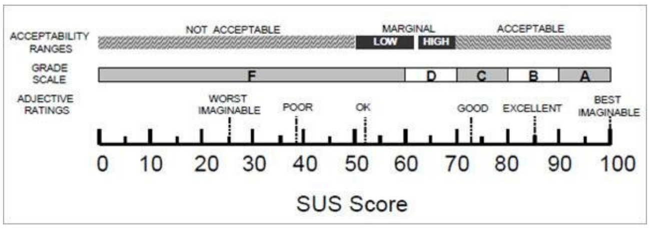


Figure 8: S.U.S scale[7]

4.3.3 The testing environment

The significance of getting the testing environment optimized for the visual systems is a critical role, the accuracy and reliability of the testing done. For the controlled environmental conditions, there are some key factors that need to be optimal, such as lighting conditions and background considerations. Unfavorable conditions introduce noise and can hinder the detection algorithms, being undesired or unintended signals that may interfere with the accuracy of the detection, therefore compromise the precision of the systems.

4.3.4 Test-participants

The test participants selected for this project consist of mostly users from the gaming community. The objective is to design a game that resonates with a broad spectrum of gamers, catering to individuals with varying preferences and levels of gaming experience.

4.3.5 This project intended way of using the usability test

Usability testing is a quick way to discover problems, and test if the participants understand the product and able to use it. The point of using the usability test in this case is to see which of the two types of mechanics that the most users prefer.

In order to determine whether a test participant understands how to use the product, they will be led to the testing room and given a brief explanation of what will happen during the test. Following this, they will try the product with the aim of completing three easy stages. We will take extensive notes when they play the game to make sure everything is recorded in case a mistake or error is found later on. Finally, in order to gather the remaining data, they will need to respond to the questionnaire.

4.3.6 Expectations

We anticipate that users will submit comprehensive feedback and expect the test to provide valuable insights into what type of mechanic that the product should continuous with, and area of improvements to it. The examination of user retention will guide our understanding of what captivates the audience, and with the information gathered, further enhance elements present in the game. The usability test will assist us in optimizing the gaming experience as a result of the iterative nature. In the event of negative responses towards elements in the game, a review of these elements will be conducted. This iterative process allows for the exploration of alternative approaches to the identified shortcomings, ensuring that the final product aligns more closely with what the users may expect.

4.4 Final test

The final test will be conducted in a specific order. Initially, participants will be read a statement explaining the purpose of the test. Subsequently, they will be asked to sign a consent form, ensuring their agreement for the collection of data for the project. After this, participants will answer brief demographic questions about themselves, including their age, gender, and field of study. This process ensures the establishment of a comprehensive demographic profile of the individuals being tested.

Participants will then have 10 minutes to progress as far as possible or complete the game. While playing, their facial expressions will be recorded, and a script will log the system's placement of their heads, providing data for later analysis. Once the game is completed, participants will proceed to a questionnaire designed with statements about the game on a Likert scale. The scale ranges from 1 being the best score in all the uneven numbered question and 5 being the highest score in the even numbered questions, encouraging participants to take their time reading and reflecting on the questions.

Upon completing the questionnaire, most participants will be thanked for their participation. However, a random selection will be asked to provide an in-depth interview, intended to gather qualitative data to complement the quantitative information obtained from the questionnaire. This comprehensive approach aims to collect diverse data to find out the answer to if the alternative interaction method have a positive user satisfactions, and usability.

For a detailed description of the entire test procedure, please refer to the test paper in the appendix A.8.

5 Design

This chapter will provide concise coverage of the design of our puzzle-like game that we named '*Birdowatch*'. In the game *Birdowatch*, the overall gaming experience is designed by its pivotal camera system. Head tilt is the method of interaction used to shift the perspective of the camera view. By shifting the perspective of the camera view, the player may find new clues that will help them reach new areas and help them make way to the bird objective in order to progress to the next level. The fixed bird's eye perspective provides the player a clear view of their surroundings, enabling them to plan their routes strategically. Our game draws inspiration from the aforementioned game Monument Valley, where an isometric diorama-style platform takes center stage in the camera view. While Monument Valley incorporates puzzle elements involving the rotation of the game environment, *Birdowatch* differs in that the player may rotate the camera view for perspectives that will favor them.

5.1 First iteration of game idea

The general idea of the game is a level-based puzzle game, where the player have to complete the level based on a visual game mechanic. There were many different ideas of how of a unique game mechanic could be implemented, with the main part of ideas being around the head or eyes. One of the ideas that were brought up, but was not used for its complexly and being too difficult to accurately detect, was emotion interaction to control the environment in the game. The idea behind using emotion tracking were translating the shown emotions of the player onto the playable character, that would then be used to solve the puzzle. In the first version of the game, the idea of the game mechanic was to be able to control the camera by having it move according to where your eye gaze was. For example, if you look on the right side of the screen, the camera would move in that direction; the same would apply to the left. The game should take place on a platform with the aesthetic of a magic forest. To advance through each level, you must solve a unique puzzle that combines traditional puzzle elements with perspective manipulation. It will become more difficult to solve over time, and you'll gradually be introduced to new game-play mechanics that add more puzzle components to the game.

5.1.1 Mood-board

For the creation of the game, the process started by creating sketches of how the game would feel and look like. We made a mood-board to begin the idea process, where the theme and concept was agreed upon, without explicitly creating anything yet. Through the use of A.I. image generator, it was able to create game environments closer to that we were aiming for.

Here are some examples of the direction for the game's environment:

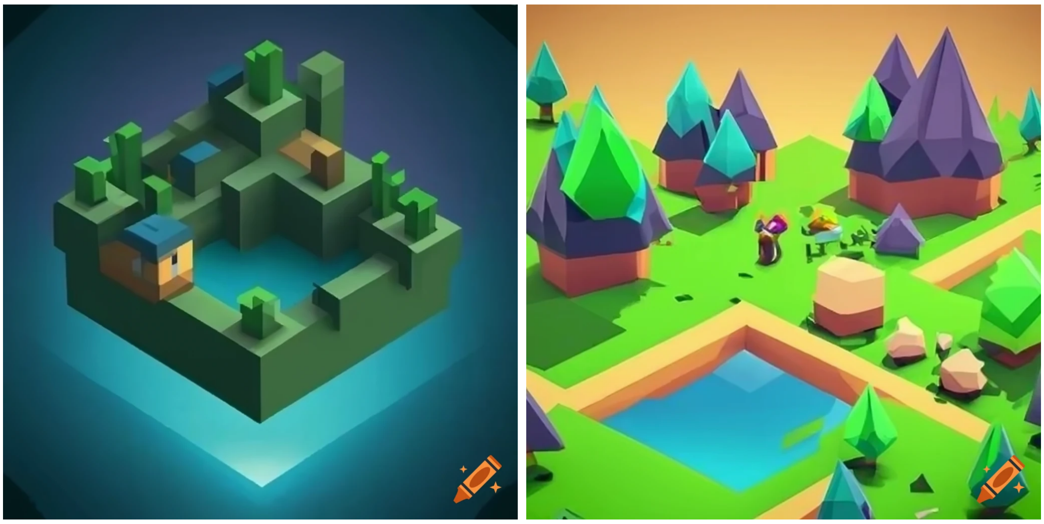


Figure 9: Examples of AI generated mood-board images [9]

5.1.2 User Onboarding

We employ a similar approach as 'Before Your Eyes' in our game, where before the player engages with the experience, they must first ensure that the hardware works as intended and feels natural for the player. Following this approach, coupled with a tutorial that covers the controls in broad terms, allows a justifiable final evaluation, since the user has a clear understanding of the mechanics present in the game.

5.1.3 UX/UI design

The UI design mirrors the vibrant aesthetics in our game. The color palette, fonts, and iconography exude a cartoonish appeal. Our theme revolves around vibrant colors that har-

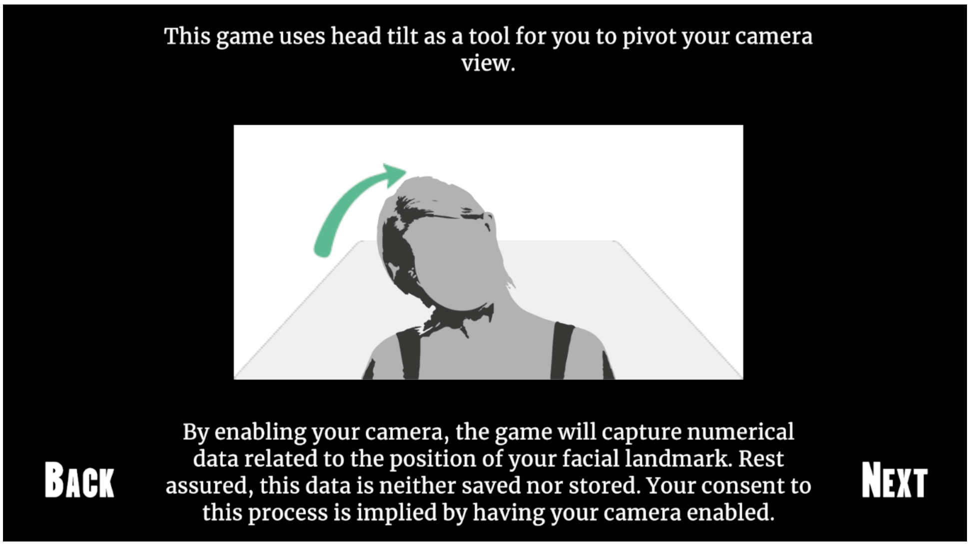


Figure 10: Illustrates how the player may tilt their head.

moniously complement a dark blue-gray hue. The use of complementary colors not only maintains vibrancy but also enhances contrast.

We also employ both sans-serif and serif fonts within the UI. We mainly use sans-serif to enrich the characteristics of the game, However, for more mature sections, such as when users need to review the terms and conditions for webcam usage, we make use of serif fonts thanks to its enhanced readability.

We have minimized the Heads-Up Display (HUD) to feature the retry button, ensuring that players remain undistracted during the puzzle-solving experience. However, during the tutorial, a prompt placed at the bottom of the HUD, guides the users though the first level. Placing the retry button in the top-left corner aligns with established visual flow principles, considering the natural tendency of individuals to direct their gaze from left to right.

In the main menu do we also establish visual flow principles, as we position graphics on the left side of the screen, while buttons reside on the right. This arrangement aligns with the broader scheme of visual flow, capitalizing on the tendency of individuals to naturally rest their eyes on the right side.

By considering these design elements, we aim to elevate the player experience, providing an intuitive interface that ensures player retention.

Figure 11: Visual flow, visualized. Graphics on the left side, logo in the top right and buttons in the bottom right where the gaze rests.

5.2 Visual Clarity

Visibility involves presenting elements in a manner that guides the player's eyes and attention (See:2.2.5). This principle is implemented throughout various aspects of *Birdowatch*. A notable instance involves the projection of particles at the location where the player is clicking, offering a visual representation of the player character's intended move.

The particles will change color based on the element the player is interacting with, providing a visual cue [Section 2.2.5] about what the player can expect from said element. For instance, when the player left-clicks on an axe, yellow particles are projected, indicating the player's ability to pick up the axe.

Right-clicking anywhere on the game environment will project lime green particles to indicate that the player character will make an attempt to move there.

A bird, that serves as the objective across all levels, projects blue particles upon left-clicking, distinguishing it from all the other particle colors. By having the bird as the objective for all levels, do we manage to establish functional consistency [Section 2.2.5]. Consistency is something we see a lot of in the game.

For instance, do we maintain visual consistency by having the retry level in the top left corner on every level. We also have consistency by having the player in the camera view at the start of every level, so the player always knows where to find their character. Even in the main menu, interactive consistency is upheld through a size increase when hovering the mouse over buttons.

Feedback is the responses generated by the game in response to the player's actions. As

Figure 12: On the left: Lime green particles projected, Player moves to the new location. In the middle: Yellow particles projected, player may interact with object. On the right: Blue particles projected, the objective for the level.

previously mentioned, players can anticipate visual feedback upon interacting with the game environment [Section 2.2.5]. The player may also anticipate animations and audio cues as additional modalities of feedback.

The game manages to make a great balance between visual, animated and auditory cues and by doing so reinforce the player's sense of accomplishment.

For instance, when a player encounters a point of no return, the game employs visual feedback by causing the retry button to pulsate with colors. This serves as a subtle hint, signaling to the player that the player must retry and consider a different approach for navigating through the level. Another example, regarding animated cues, upon interacting with a ladder, the player character will transition to a climbing animation to correspond the player successful interaction with the ladder.

Furthermore, some objects necessitate certain items. In the event of not having the correct item, the game will display a prompt on the screen that serves as a hint that what they're holding is not compatible.

Conversely, possessing the correct item triggers an uplifting auditory cue. A rewarding player experience is maintained through the provision of immediate feedback for every player action.

5.3 Game Design

Kim S., in his book, introduced three key elements that characterize puzzle game of high-quality: Novelty, Difficulty, and Perspective [ref. 2.1.2]. In our game, we have incorporated these three aspects.

Novelty, representing the level of uniqueness and innovation, is already exemplified by our utilization of head-tilt as an interactive method. The head-tilt mechanic distinguishes our game from traditional game norms, introducing a novel dimension to the gaming experience.

Additionally, each level presents new challenges, testing the player's problem-solving abilities. One level the player is familiarizing themselves with the controls and the next level their teachings are put to the test. As new elements are introduced, so does the difficulty increase.

The game's learning curve is crafted to provide players with a progressively satisfying challenge. Starting with the first level where players master the controls and the mechanic, subsequent levels introduce new elements and mechanics, raising the overall difficulty. As players advance, they are, in some instances, required to apply their accumulated knowledge to navigate through puzzles.

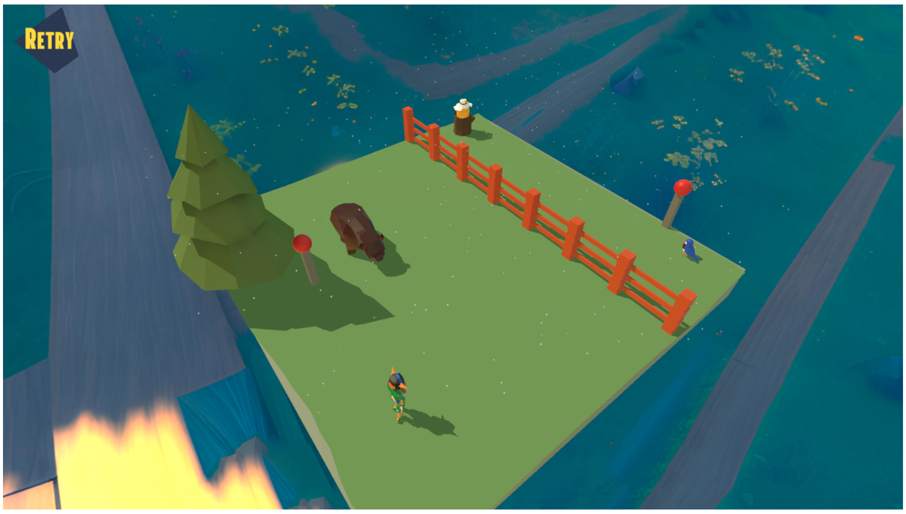


Figure 13: Showcase of the level 3 map

For example, in the third level, players learn that picking up honey attracts bears. Armed with this knowledge, players must lock gates before picking up honey, highlighting the necessity for strategic thinking. This level of back-and-forth thinking only increases as the player progresses to later levels. In contrast with the very first level, where the idea of dying was far from a possibility. In order to solve these levels, the player must see the wider picture.

Perspective remains a pivotal component for problem-solving across levels. By tilting their head, players can unveil hidden cues and perspectives crucial for completing tasks, like changing the perspective in monument valley [Section 2.3.1]. In one of the levels, players must insert three gems into color-coded pipes, requiring them to tilt their heads to observe the assigned colors on the other side of each pipe. This approach encourages players to strategically use the mechanics that they're given in order to make their way thought the evolving nature of challenges.

6 Implementation

This section delves into the technical aspects of the game's implementation, offering a view into the components and their iterations. From character control to the implementation of the different types of face mechanics and the interaction between Python and Unity with the use of sockets.

6.1 Implementation of game

The game is implemented in Unity where the world-building of the game is focused designing on the platform of each level, so the level would need to be completed by the use of navigating around the IT, with the in-game camera.

6.1.1 Music and sound effects

For the music and sound effects used in the game, the music was implemented as simple background music created in FL Studio. While the sound effects were implemented as feedback for when you interact with an element inside the game, The sound effects used have been taken from the Mario franchise by Nintendo and the BBC.

6.1.2 Camera Control

The camera control class orchestrates the dynamic camera behavior in the game by having the 'SocketMessage' base class as its' parent. Thanks to the inheritance structure, the SocketMessage class and all its components and core functionalities can be inherited by the CameraControl class, enabling the head-tilt functionality. The management of the GetCurrentMessage() integer from the SocketManagement class accounts for one of these functionalities. The integer is streamlined within an if-else statement. The if-else statement consists of virtual methods that the 'CameraControl' class overrides. With the SocketMessage class, we can seamlessly implement head-tilt interaction in various logics throughout the game by simply overriding these methods, similar to what has been done with the control of the camera. The CameraControl class inherits from SocketMessage and not from Monobehaviour. By doing so, can the methods RightTilt(), LeftTilt(), and NoSocket() be overridden with ease. The methods are overridden by a coroutine that handles camera rotation. With the use of a coroutine, can the rotation happen in a burst-like manner instead of a continuous manner. The coroutine initiates an IEnumerator RotateCamera() that interpolates the camera's current yaw to an adjusted yaw over time, ensuring smooth camera motion. The parameters define left and right movements, corresponding to negative and positive yaw speeds within the context of the system.

```
public override void TiltRight()
{
    base.TiltRight();
    if (!isRotating)
    StartCoroutine(RotateCamera(yawSpeed));
}

public override void TiltLeft()
{
    base.TiltLeft();
    if (!isRotating)
    StartCoroutine(RotateCamera(-yawSpeed));
}

public override void NoSocket()
{
    base.NoSocket();
    currentYaw -= Input.GetAxis("Horizontal") * yawSpeed * Time.deltaTime;
}
```

```
IEnumerator RotateCamera(float rotationAmount)
{
    isRotating = true;  // Checks if the coroutine is in motion

    float targetYaw = currentYaw + rotationAmount; //Current Yaw added with the yaw speed.
    float elapsedTime = 0f;

    while (elapsedTime < 1f)  // Duration before the coroutine may start again
    {
        currentYaw = Mathf.Lerp(currentYaw, targetYaw, elapsedTime); //Moves yaw from currentyaw to the current yaw added with yawSpeed
        elapsedTime += Time.deltaTime;  //Time progress
        yield return null;  //Yields control to the rest of the update method, (for frame friendlyness)
    }

    currentYaw = targetYaw;  // Ensure we reach the exact target yaw
    isRotating = false;  // Coroutine is no more in motion
}
```

Figure 14: On the left: The overridden SocketMessage methods. On the right: IEnumerator for camera motion

6.1.3 Character control iterations

Initially, the player movement used GetAxisRaw, so the player would move the character using either the W.A.S.D. or the arrow keys.

```
float horizontal = Input.GetAxisRaw("Horizontal");
float vertical = Input.GetAxisRaw("Vertical");
Vector3 direction = new Vector3(0,0,0);
```

Figure 15: The first attempt for a movement of the character

Because of the game-play mechanics and the world-building of the game, the use of keyboard keys to control player movement did not feel very intuitive based on our own play testing. The reason why the use of the W.A.S.D. or arrow keys did not achieve intuitiveness is because when the platform is rotating, the movement associated with the keys does not follow, so the convention and the widely understood action are not followed. When rotating the platform, the point of origin for the player moves as well, but the movement of the axis does not, so for example, if the platform is rotated 90 degrees to the left, now the A key is the forward key instead of the widely understood W key. Because of this issue, the movement was changed to a mouse-based movement that followed the mouse click. You would move using the right button and interact with elements of the level with a left click.

6.1.4 Final Character Control iteration

The Interactable script, similar to the SocketMessage script, serves as the parent script for various child scripts, each inheriting the core functionality of interaction upon a mouse click. These child classes override specific methods to achieve different results. For instance, some objects, upon interaction, may be picked up, while others may necessitate certain items before you can interact with the object. The interaction through mouse input is made possible by using Ray casting, a commonly used technique involving the projection of an imaginary line, or ray, from a point of origin. This ray provides information about any object that intersects its path. In the context of our game, the ray cast from the position of the mouse nested in the camera viewpoint.

By checking for collisions detected by the ray, we can not only enable the player to interact with objects in the game environment but also make the player move to the point where the ray makes contact.

```
//Ray hits something?
if (Physics.Raycast(ray, out hit, 10000))
{

    //Check if ray hits interactable
    Interactable interactable = hit.collider.GetComponent<Interactable>();
    //Checks if it's interactable
    if (interactable != null)
    {
        Debug.Log("<color=lime>" + "Interacting with " + interactable.name + "</color>");

        // Check if the interactable has the ObjectiveInteraction component
        if (interactable.GetComponent<ObjectiveInteraction>() != null)
        {
            PlayParticles(hit.point, "Objective");
        }
        else
        {
            PlayParticles(hit.point, "Interactable");
        }

        SetFocus(interactable);
    }
}
```

```
//Checks input: Right Mouse button
if (Input.GetMouseButtonDown(1))
{
    //Casts ray to camera
    Ray ray = cam.ScreenPointToRay(Input.mousePosition);
    RaycastHit hit;

    if (Physics.Raycast(ray, out hit, 10000, movementmask))
    {
        PlayParticles(hit.point, null);
        //Debug.Log("We hit " + hit.collider.name + " " + hit.point);
        motor.MoveToPoint(hit.point);

        RemoveFocus();
    }
}
```

Figure 16: Update method that covers Ray Casting for left mouse click (on the left) and right mouse click (on the right)

To move the player, we implement Unity's Navigation Mesh System, which facilitates pathfinding and navigation in the game. The SetDestination() method allows an agent object to navigate to points in the environment using the shortest available path. However, this is restricted to points the agent considers walkable. By baking the game environment, Unity calculates which static objects are walkable and which are not. This system is not only utilized by the player character but also by the enemies present in the game.

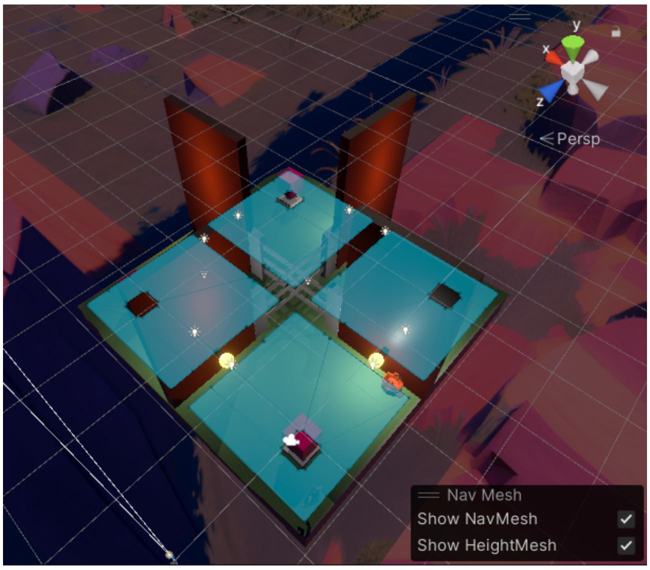


Figure 17: Visualization of the Navigation Mesh for one of the levels, areas entities may move

6.2 Implementation of Face interaction

The way that the face interaction was implemented were in an iterative process, where the core game and game-idea would not be change, being that the in-game camera rotation based on real-world expression. The different facial interactions were the ones changing and were tried out and tested, to see what mechanic best suits the solution. The first solution that were being implemented was an eye gaze tracker, that would rotate based on if the user is looking on the sides. As the eye gaze implementation had unsuitable issues that is described in its section, two other variances were implemented, where the usability testing was able to give insight into which solution were improved further on. The two facial interactions, were a blinking and a head-tilt mechanic. The reason for these, was that they still could be used as a simple way to rotate the in-game camera by expressing yourself in the real world. Both these mechanics were in the planing stage before the eye gaze was implemented, but initially chosen off because of personal preference. The two mechanics both have benefits and disadvantages, the blinking mechanic's is more complex and, if working with a stable version, allows for more discreet and subtle interaction. Head-tilt have a simpler and more robust approach, but it is a trade-off for its more excessive movement

For the implementation of the facial interactions, we used the programming language Python because of its libraries and frameworks, such as openCV, Dlib, numpy and Mediapipe, that can be leveraged to make the implementation more seamless. Its simplicity in its syntax also makes it preferable. The different iterations of the implementation, have had different variance of success, but still followed the python language.

6.2.1 OpenCv

For the webcam to be able to capture the facial interaction, it first needs to be initialized, and that is done though the OpenCv library. OpenCv is also used to control the video-capture window. This includes reading the frame, flip it horizontally and converting the frames from BGR (Blue, green, red) to RGB (Red, green, blue). OpenCv is also used for displaying and exiting the capture window.

```
while True:
    ret, frame = cap.read()
    frame = cv2.flip(frame, 1)
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    # Perform operations on the frame
    # ...

    cv2.imshow('Test4', frame)

    # Press 'Esc' to exit
    if cv2.waitKey(1) & 0xFF == 27:
        break

cam.release()
cv2.destroyAllWindows()
```

Figure 18: The use of OpenCv for video capture and display

6.2.2 First iteration of eye interaction

The first iteration for the implementation of the eye-tracking and further into pupil tracking made use of a simple pipeline, going from capturing the webcam to tracking the pupil of the person in frame. The first step is getting a object detection of Haar-like features is using cascade classifiers that is a machine learning technique that are models pre-trained to detect various views of faces and upper body. The two used for this first iteration are `haarcascade-eye.xml` and `haarcascade-frontalface-default.xml` from OpenCV's GitHub.

The next step is detecting the face and drawing a rectangle around it, which is done with the face-cascade, that have some parameters that specifies it.

```
faces = face_cascade.detectMultiScale(gray,
scaleFactor=1.3, minNeighbors=5, minSize=(30, 30))
```

Figure 19: Face cascade detection

The next step in the pipeline is getting the eye detection inside the face region, which is done with the use of the eye-cascade xml file.

```
eyes = eye_cascade.detectMultiScale(face_roi)
```

Figure 20: Eye cascade detection

Now we have a small eye region that can be used to detect the pupil inside, with the use of a simple blob detector. It is done by setting a threshold in the eye region of interest (ROI) and iterating over these detected blobs (keypoints) to get their coordinates. The issue with the implementation of this eye-tracking pipeline is its stability and consistency in real time use, while it can find and track the pupil of the eye in a controlled environment rather well, when it comes to the dynamic nature of eye movements of the user, such as saccades and blinks the precision is not stable and robust enough to achieve the goal that was intended.

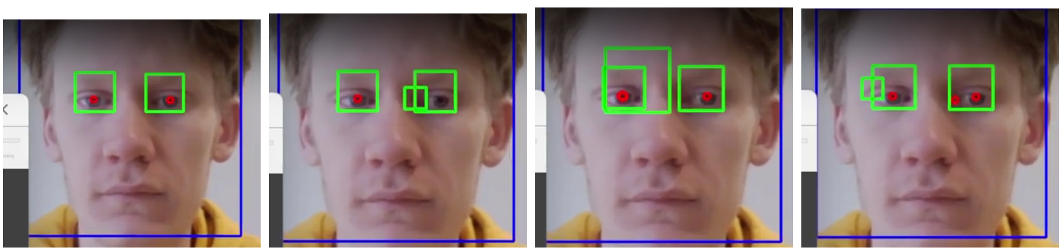


Figure 21: Four still-frames right after each other of the pilot-test of eye-tracking with threshold adjusted

A video showcase of the implementation of the eye-tracking is linked in the appendix 1.1.

Another issue that became prominent when trying to implement this pipeline, was the translation from tracking the pupil, to getting information on where the user would look on the screen. As a test before implementing the solution into the game, we would set the mouse-cursor as the screen element that should move in correlation with where the pupil is looking. Because the robustness of the pupil-tracking is not relatively strong enough to track fine adjustments on eye movements, it does have a harder time following the pupil when looking to the sides, when translation to the location of the mouse cursor would stay in the middle of the screen.

6.2.3 First iteration for the usability test

To make a more stable and consistent solution, we make use of an already existing python library that manage the most basic functionality, and therefore lets us focus and expand on the specific features that our game would implement. The solution chosen was the media-pipe library ([26]) that include a face-mesh solution, that would track facial features in real-time, and for the first implementation of this we only focused on the key landmarks around the eyes 2.2.1.

The media-pipe implementation work with the two solutions that are used in the first usability test, being the head-tilt mechanic and a blink mechanic. The basis for these solutions lay in the use of face landmark detection, which tracks key facial features. For both solutions the script starts by initializing the face-mesh from the Media-pipe library, and then extracting the detected facial landmark points, and lastly extracting the height and width of the frame:

```
face_mesh = mp.solutions.face_mesh.FaceMesh(refine_landmarks=True)
```

Figure 22: Initializing the face-mesh

```
landmark_points = output.multi_face_landmarks
frame_h, frame_w, _ = frame.shape
```

Figure 23: Extracting facial landmarks, height and width

Now that the landmarks are extracted, some of them can be chosen as a baseline for the tracking of the face, and for simplicity's sake the same landmarks were chosen to be the baseline for both the blink and the head-tilt implementation, that being the landmarks around the eyes. For the blinking implementation, both the right and left eye needed its key landmarks to be tracked, because it needs to be distinctable what eye is blinking. The way that the blinking is registered is by having two landmarks points, one on the upper eyelid and one on the bottom eyelid, and then when they are colliding within a threshold, an input is registered.

For the head-tilt, the procedure for implementation use a simple threshold for detecting the

```
if (left[0].y - left[1].y) < 0.008:
    client_socket.send(struct.pack("<I", 2))
else:
    client_socket.send(struct.pack("<I", 0))
```

Figure 24: How the blink mechanic works

tilting for either side, so when the landmark point reach over the threshold, the input gets registered. For the tilting in the first iteration of head-tilt use the screen width as a parameter for the threshold, based on Liu et al. "HeadPager: Page Turning with Computer Vision Based Head Interaction" ([33]) the boundaries is 1/3 of the screen.

```
if screen_x < screen_w / 3:
    current_state = "Left"
elif screen_x > 2 * screen_w / 3:
    current_state = "Right"
else:
    current_state = "Center"
```

Figure 25: First iteration of Head-Tilt code

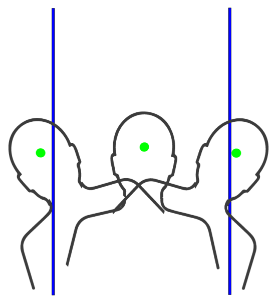


Figure 26: Illustration of how the head-tilt works

6.3 Usability test of the system

A system usability test was made with the focus of exploring of different variations of how the final product would align closest to the user's expectations and preference. The testing consisted of an A/B testing approach and was conducted to determine the usability of head tilt and blinking mechanic to rotate the in-game camera as a game-play mechanic. As these two approaches were designed to have the same feature, but we wanted to ensure that only the mechanic with the best usability gets improved. The 2 tests had the same physical conditions, ensuring that no external variables would interfere with the results. We split 12 total test participants into two groups of 6 at random and just changed the script used for the system, since that was the only variable to change, while the game-play remained the same. The main goal of testing the usability of the applications is to determine the better solution for the final product.

6.3.1 The testing environment

The testing environment for the first usability testing was carefully selected to have the most favorable key factors, being the lighting and the background. A location was chosen where the participant would sit with a white wall right behind them, this way there were limited noise and disturbances in the background that could interfere with the detection algorithm. Conveniently, the place chosen had a window to the right side of the participants, and a lamp above them, to ensure the best lighting conditions. For further detail about testing environment, see appendix (A.6.2).

6.3.2 The SUS test results

Before the SUS test was made, some simple democratic data was collected from the participants, and the results are as follows:

- **Age Groups:** The participants were categorized into distinct age groups, and were split up into two groups, 7 participants in 18 to 25, and 5 being in 26-35.
- **Genders:** The gender distribution among the participants was recorded as: 4 women and 7 males
- **Educational Backgrounds:** Information regarding the participants' fields of study was collected: 12 test participants who are studying Medialogi

The SUS scores is then calculated to find which of the two alternative approaches for the in-game camera rotation that would have the best usability. First the mean result for the head-tilt approach, where the raw data is in appendix:

Mean $= \frac{57.5+75+70+72.5+62.5+97.5}{6}$

Mean ≈ 72.5 According to the S.U.S scores guideline, it can be seen that the mean result of 72.5 is above the mean value of 68. The results from the usability test using the blinking mechanic instead of head-tilt is shown here:

Mean $= \frac{47.5+72.5+45+75+72.5+50}{6}$

Mean ≈ 60.42

Here the SUS score is considered below the mean according to the guidelines, and it can be seen that the mechanic may be difficult to use for users.

The next few statements are a reference to figure 26. The observation of the processed data is; blinking mechanic has a lower mean. Additionally, it also has a lower Q1 and Q3 compared to head tilt. The range of head tilt whiskers is lower compared to the range of blinking mechanic whiskers. For more processed data and more in depth look at test methods, look appendix.

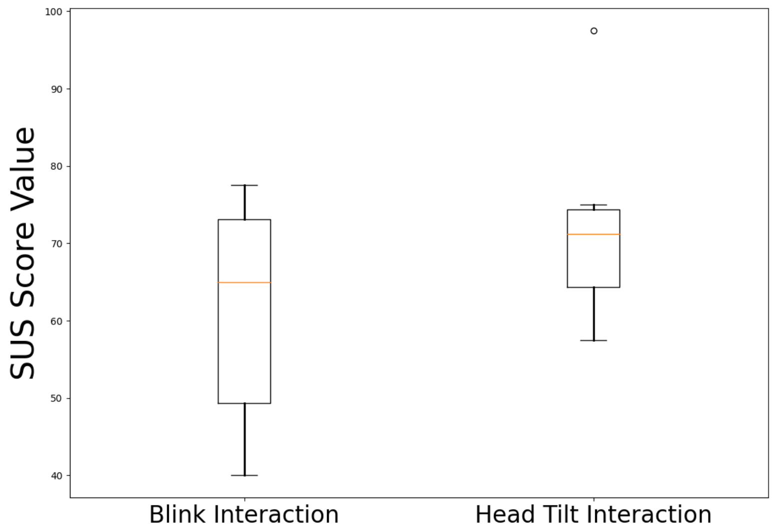


Figure 27: Boxplot of S.U.S score evaluation

Some of the qualitative data collected during the game-play of the SUS test due to Think-aloud for head tilt was verbal expression of confusion about how the game mechanic worked, generalized questions about game mechanics.

These results do not give a solid conclusion as to what type of mechanic is superior due to [20] (pp.26 and pp.161) stating you can cancel a usability test if 3 out of 5 test participants experience the same issues, it was decided to stop the usability test, in our case it was stopped when a total of 12 participants were met. However, based on the S.U.S score and the verbal feedback, it was decided to continue with head tilt interaction to further improve on, and use for further testing. Since the S.U.S scores tells that there is a clear difference in the usability of the two mechanics, being 72.5 for head-tilt and 60.42 for blinking. With the average SUS score being 68, a score above is considered above average and below, considered below average. Therefore, the head-tilt is preferable for the continue work of the project [1]. The alternative hypothesis (H1) from methods 4.2.1 can then be concluded.

6.3.3 Improvements to head-tilt

Based on the initial usability test and the results from the S.U.S. score, it was chosen to work and improve further on the head-tilt mechanic. The first improvement that was made to the head tilt was changing the location of the point of interest for the key landmark to be in the center of the forehead instead of on the one eye, so that there is an equal distance for the left and right sides for the boundaries. This way, the amount of tilt is more constant for both sides. Most of the other improvements to the head-tilt code were made for tracking the data for testing, so the data could be collected in a way that required the least amount of manual input.

Firstly, the printing of the conditions 'Center', 'Left', and 'Right' has been made to only trigger once and not continuously printed out into the console, as it was before. It was made with two simple variables—the current state and the previous state—that are used as the controlling mechanism. This way, it was easier to manage the input from the user.

The next improvement done for the script was to log the data that the users would give; this way, if their movements are tracked and stored for further analysis, and together with a video recording of the webcam, it can be used for testing of errors when the participants do not reach the boundary of the head tilt. The way that their data is stored is in a csv file format, where their input of their movement improved on before gets logged together with the timestamp of when the state changed.

```
with open(csv_file_path, 'w', newline='') as csvfile:
    fieldnames = ["Timestamp", "State"]
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
```

Figure 28: The initialization of the csv file

And the writing of the data onto the now made csv file

```
writer.writerow({"Timestamp": datetime.now().strftime('%Y-%m-%d %H:%M:%S')
, "State": current_state})
```

Figure 29: The writing of the data onto the file

6.4 Python and unity connection

Since the game is made in Unity and the implementation of the facial interactions is made in Python, they need to communicate. There are multiple ways to make the connection between them possible, and the one that has been used for this product is the use of a socket. The way it works is by creating a local server in Python that is responsible for sending out the messages, and then a client in Unity that can pick up the messages that the Python script sends out. A socket is an endpoint that can be used across a network, and in this case, a local network, and is used for sending or receiving data. On the Python side, it starts by initializing the connection with sockets by binding it to a specific address and port. Then it sets its state to be listening for incoming connections that can receive the message the Python server sends.

6.5 Sockets

Sockets helps us establish network communication between two programs, facilitating data exchange. In the context of our game, Unity and Python communicate through sockets, making it possible to conduct the data transfer for the head-tilt of the player. Although there are various types of sockets, In this implementation, Transmission Control Protocol (TCP) was chosen for being way more reliable in data transfer. For the reason being that TCP ensures a stable connection before initiating data transmission, unlike other types of sockets. The process begins with the creation of a socket using Python's socket module. This socket is then configured and bound with a host and a port. In our case, the host is set to 'localhost,' indicating that both the Python server and the Unity client are operated on the same machine. The port is where the communication channel takes place.

The server socket listens for incoming Unity clients. In order to send a connection request, you must start up the game. Once a connection request is received, the server accepts the connection. With the connection set in stone, values are processed through a while-loop, translating the location of the face landmark into integer values. When a head tilt is detected, whether to the left or right, an integer value is transmitted to the Unity application in a binary data format. Additionally, an integer value 0 is also transmitted if the head is not tilting. To address potential problematics in byte order between Python and Unity, we make use of the struct pack module. How the game manages to send a connection request is all defined in

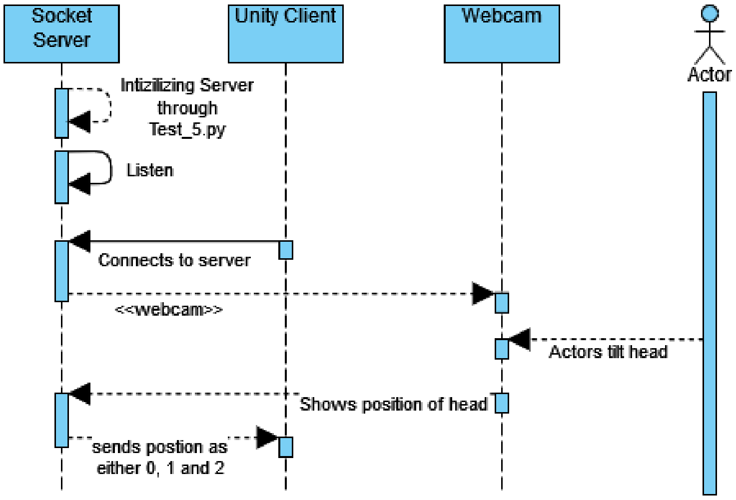


Figure 30: Sequence diagram, on how the connection between the camera, socket server and unity client

the SocketManagement-class.

```python
# Creation of socket server
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
host = 'localhost'
port = 12345
server_socket.bind((host, port))
server_socket.listen(1)

print("Listening for Unity client on " + host + ":" + str(port))

# Accept the connection outside the loop
client_socket, addr = server_socket.accept()
```

Figure 31: The python side of the socket connection that create the local server

In Unity, The SocketManagement-class facilitates data transfer between Python and Unity. By deploying the System.Net.Sockets namespace, we gain access to properties inherent to the TcpClient and NetworkStream classes. The TcpClient acts as the server client, while the NetworkStream manages the exchange of data. In the Start method, we instantiate a new TcpClient, configuring it with the specified host and port parameters. Here, it is important that these parameters align with what we have set in the Python script. Deploying the Read()-method, we retrieve bytes of data from the NetworkStream, storing them in the buffer. This method allows us to extract the initial value in the byte array and assign it to the currentMessage variable. in the Update method, We extract and assign the initial number from the byte array to the currentMessage, given that the update method updates for every frame. This process ensures that currentMessage remains synchronized with the player's current head tilt.

6.6 Activity diagram

Through this activity diagram, it shows how the game is intended to be played through the game from the start to the end. You follow it through black dot, and then throughout the different choices it navigates down to the end.

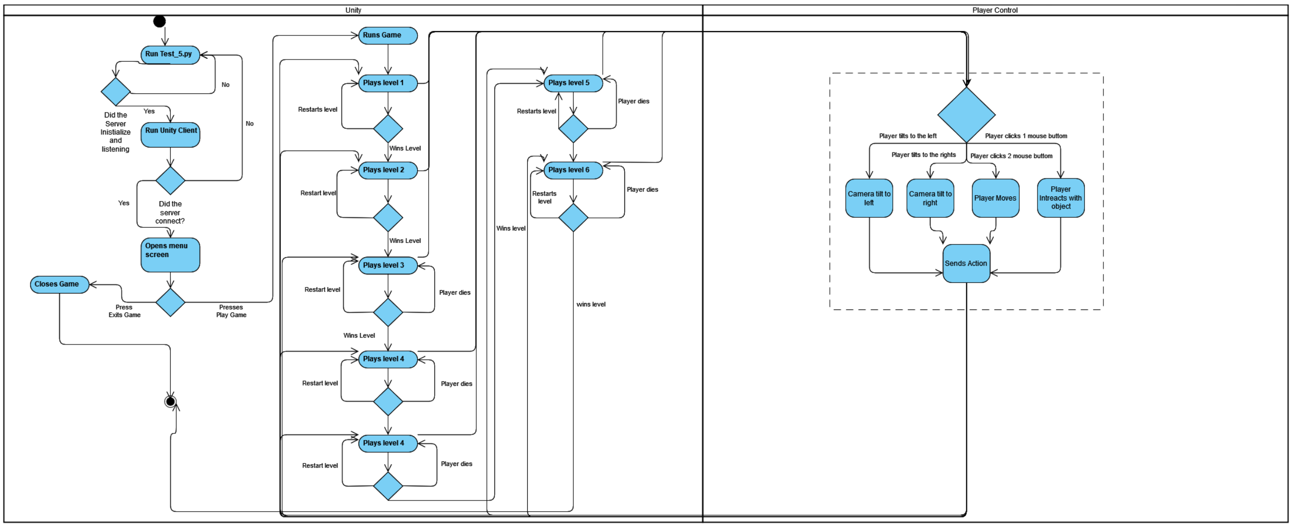


Figure 32: Activity Diagram of the Game

7 Evaluation

The final evaluation combines quantitative data from input logging, completion times, and a questionnaire, along with qualitative insights obtained through interviews.

7.1 Final test

The final test was conducted in the same setting as the initial system usability test due to its ability to adapt in managing the noise from the webcam feed. , and the logging of left and right inputs from the webcam.

7.1.1 The pilot for final testing

When the pilot test was made, some errors in the game and logging system were found. The biggest error found was that when you complete a level in Unity, the Python script that was sending the sockets to Unity, which dictate the visual element, did not follow along, and therefore the rotation of the platform did not work. To fix this issue, the game object that the socket script was on needed to be a prefab and include a DontDestroyOnLoad method.

7.1.2 Method for the final test

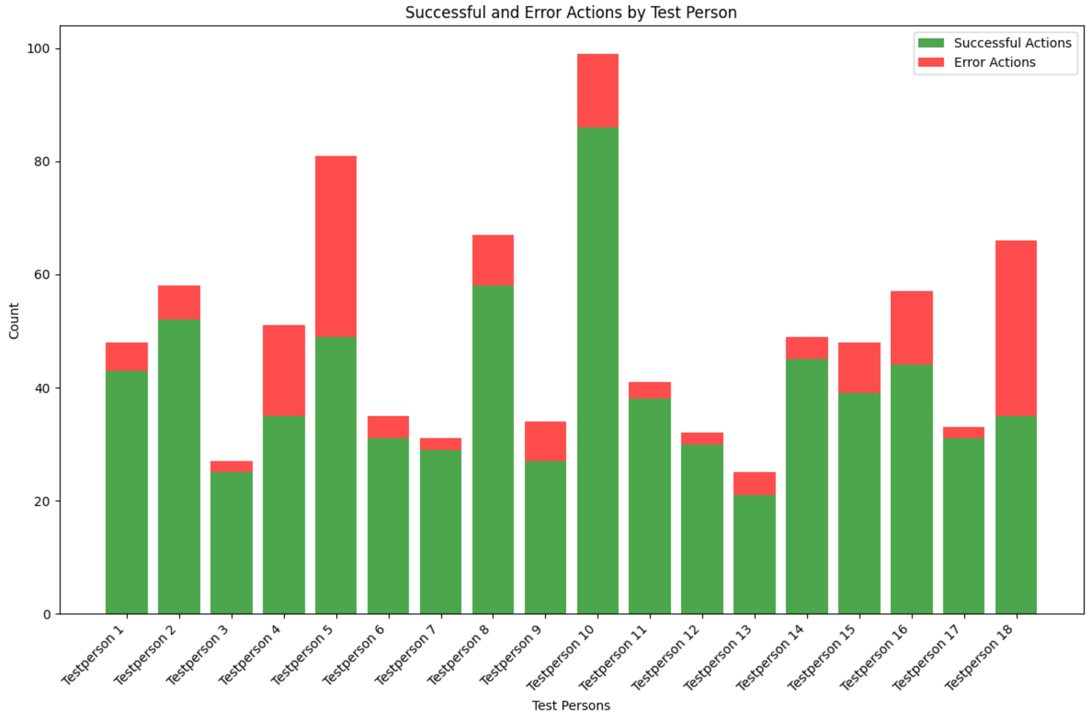
The methods used in our final test were a mix of quantitative and qualitative data gathering, where a logging system was set up to capture relevant quantitative data, such as the states of the test participants heads, and through a webcam feed tracking their errors and successful attempts. The data collection also includes a questionnaire and for randomly chosen test participants an in-depth interview. The validity of the testing is strengthened by the triangulation of the multiple methods. The validity of the testing instruments used is important to eliminate that as a changing factor; therefore, the same setup will be used each time, with the same procedure for each time, such as a mouse, webcam, and computer. Some other variables that can hinder the testing participants are factors such as outside influence, people walking by thereby distracting the test participant, or they're the level of knowledge and skill to be able to clear the levels.

7.1.3 Results from input logging

As explained in the improvement of head-tilt chapter in implementation [ref. 6.3.3], when the system was tested, the inputs from the users were logged onto a CSV file. Due to the CSV and the webcam feed getting the exact timestamp on them, they could be manually compared to measure the errors that had occurred during the testing. These errors can both be from the system's side, being that it did not recognize the input from the user's or recognize false inputs, but there can also be errors from the user's side, being that they were not able to use the system to its fullest extent.

When collecting the data and translating it from raw inputs into workable data, there were measures that were looked for in the recordings; they are explained together with all the data in Appendix A.9, but the four measures that were most useful were the amount of successful actions on both the left and right sides and the amount of unsuccessful actions on both sides.

The first statistical visualization to look at based on the collected data is how well the users are able to use the head-tilt system, and it can be seen that they have a generally low error rate, but there are still some outliers. On this plot, the total count of non-continuous right and left actions compared to total not-successful actions per person can be seen.



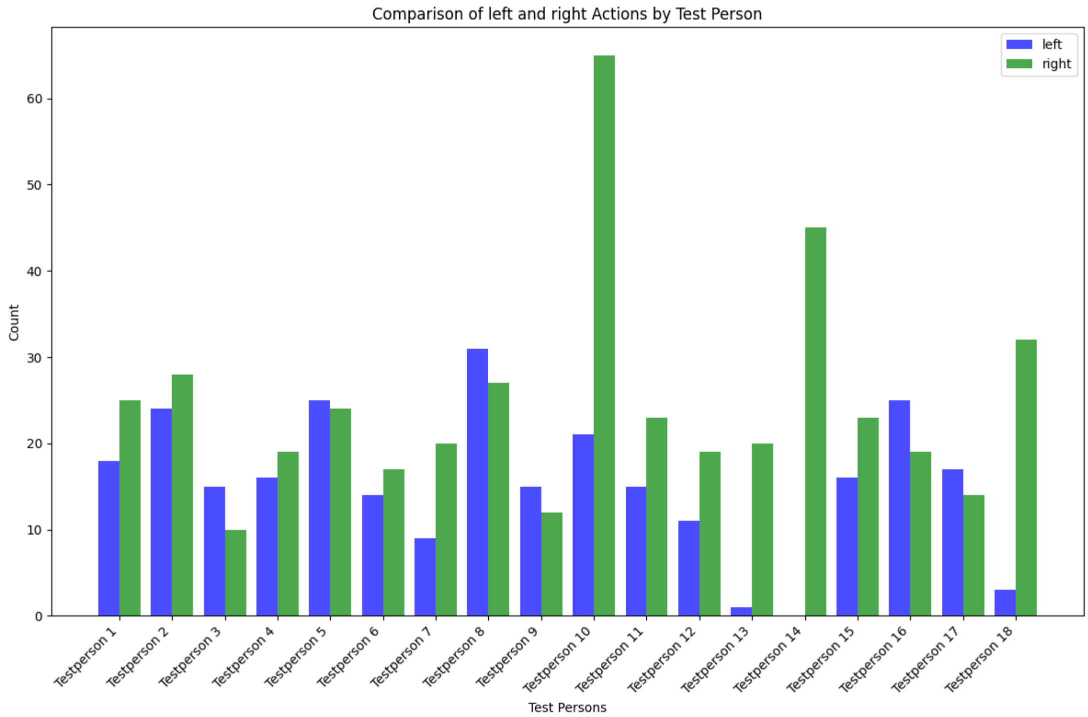

```
Error Rates for Each Test Person:
Testperson 1: 10.42%
Testperson 2: 10.34%
Testperson 3: 7.41%
Testperson 4: 31.37%
Testperson 5: 39.51%
Testperson 6: 11.43%
Testperson 7: 6.45%
Testperson 8: 13.43%
Testperson 9: 20.59%
Testperson 10: 13.13%
Testperson 11: 7.32%
Testperson 12: 6.25%
Testperson 13: 16.00%
Testperson 14: 8.16%
Testperson 15: 18.75%
Testperson 16: 22.81%
Testperson 17: 6.06%
Testperson 18: 46.97%

Average Error Rate Across All Test Persons: 16.47%
```

Figure 33: Count of Successful and Error Actions by Test Person

It can be seen that the mean error rate was 16.47 percent in total. This percentage reflects the average rate of inaccuracies observed across the entire dataset. These error-rate is only the errors that occurred when the test-participants did not reach far enough to activate the tilting action, and not errors of the system not recognizing a successful action from the user.

Another observation found was the comparison between right inputs and left inputs given, that looks at the total right / left inputs from each user, it can be seen that the right side is more preferred in total, and for some participants heavily preferred, or being the only input used. This can be for different reasons, but the two most likely being user dominance or preference, and control imbalance, because most people are right-handed, it could feel more natural to tilt to that side when giving the input, and since there is no benefits of tilting one way or another, users might be inclined to tilt to the side that feel most natural.



Test Person	Percentage of 'right'	Percentage of 'left'
Testperson 1	58.14%	41.86%
Testperson 2	53.85%	46.15%
Testperson 3	34.78%	65.22%
Testperson 4	48.39%	51.61%
Testperson 5	39.02%	60.98%
Testperson 6	41.67%	58.33%
Testperson 7	65.38%	34.62%
Testperson 8	29.55%	70.45%
Testperson 9	11.76%	88.24%
Testperson 10	70.42%	29.58%
Testperson 11	58.33%	41.67%
Testperson 12	42.11%	57.89%
Testperson 13	90.00%	10.00%
Testperson 14	100.00%	0.00%
Testperson 15	40.74%	59.26%
Testperson 16	41.86%	58.14%
Testperson 17	43.33%	56.67%
Testperson 18	88.46%	11.54%

Mean Percentages
Mean Percentage of 'right' across all test persons: 53.21%
Mean Percentage of 'left' across all test persons: 46.79%

Figure 34: Amount of right and left actions, and it's mean

The data indicates that the mean right tilt is only 6.5 percent higher, which is not a very good balance. However, when we examine specific test takers, such as test persons 13, 14, 18, and 10, we find that some of them almost exclusively used their right side, with no one using their left side at the same rate.

When looking at the error-rate individually for left and right side, some of the explanation can be found, it can be seen that the success-rate of the right is slightly higher.

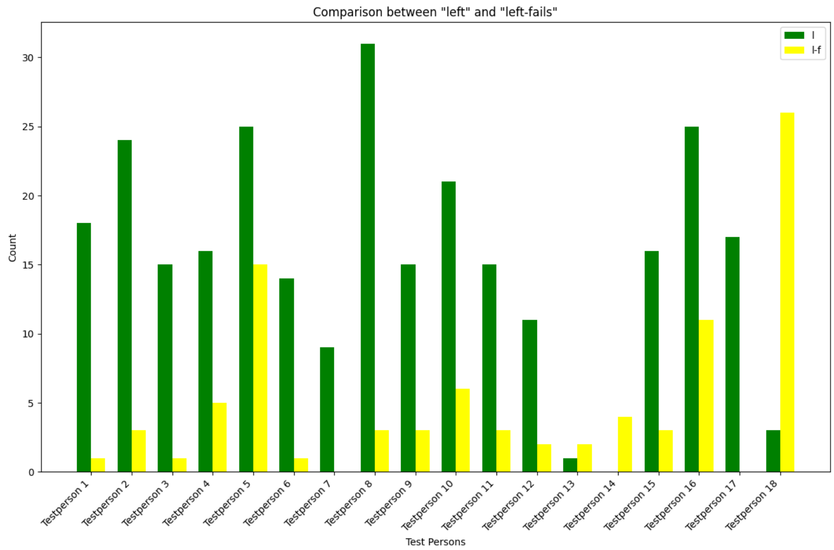


Figure 35: Success-rate of the individual left and left-fails tilt

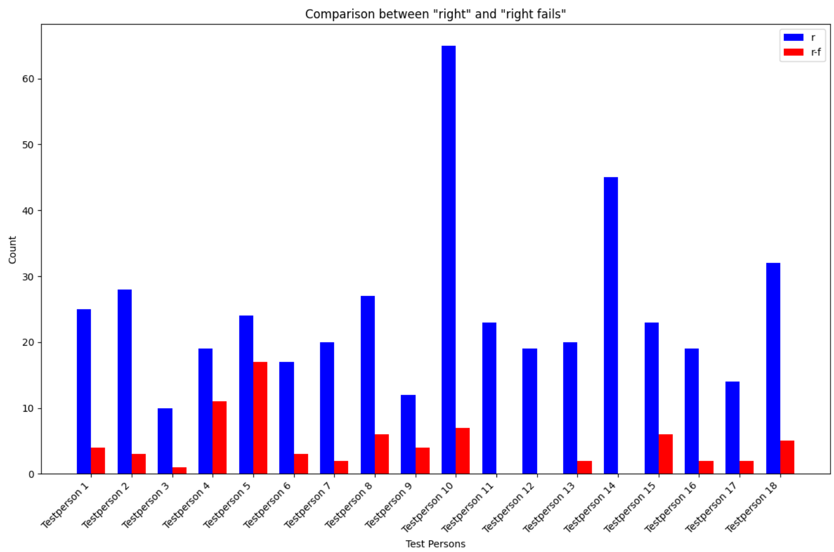


Figure 36: Success-rate of the individual right and right-fails tilt

The mean success rate for the left tilt is 73.72 percent, where it's 85.94 percent for the right tilt, and the answers to why some of the test-participants preferred like right tilt, like 13, 14 or 18, that have a more fails for the left side than successful actions, and therefore could be more likely to use the right side, because it felt better for them to use that side, and the imbalance can be seen that the right side have higher rates.

There is also an occurrence of the right side have a shorter boundary line, and it has been observed that the action on the right side can be triggered before the boundaries are met,

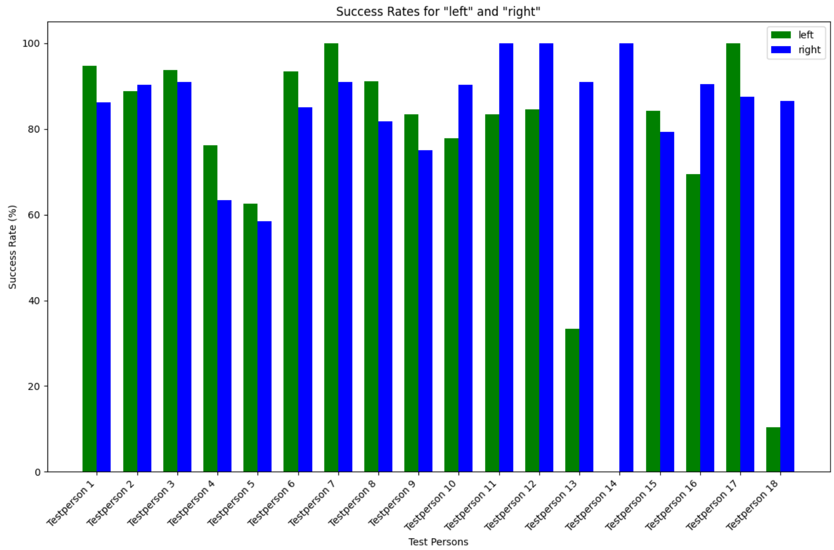


Figure 37: Success-rate of both in percent

and that is not occurring on the left side, and therefore easier to tilt the camera rotation of the platforms.

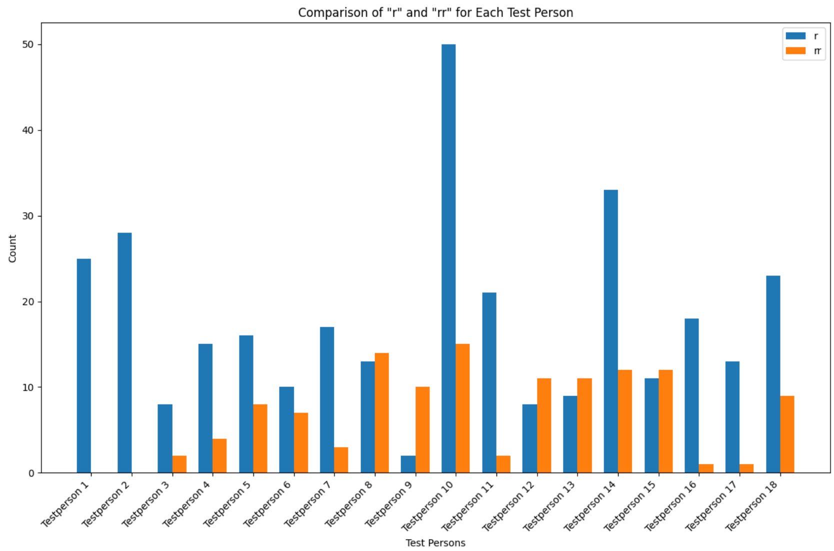
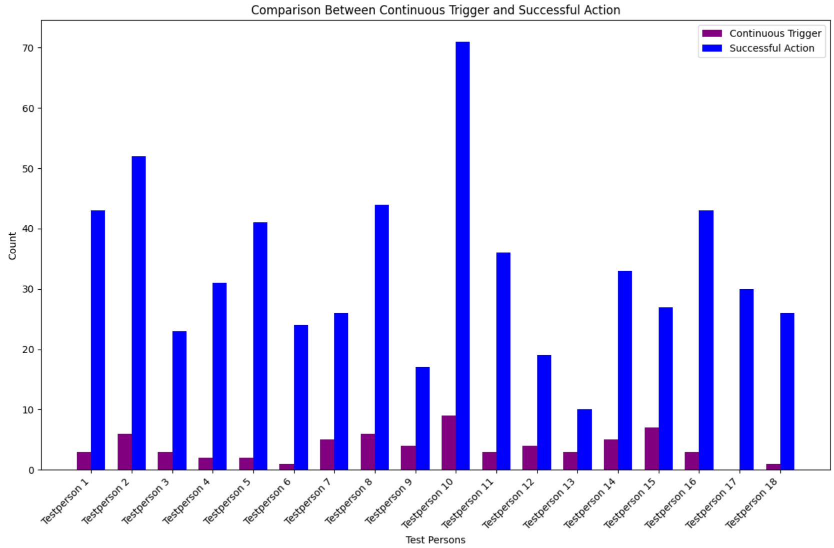


Figure 38: ’r’ being right side over the boundaries and ’rr’ being below it

There is no triggering of the left side below the threshold boundary, so either the line that is set to indicate the boundary on the webcam is set slightly wrong or the dimensions of the webcam are different from the computer that it’s implemented on.

Another interesting observation that happened during the testing was that sometimes the input given by the user would count double or more times, with it only being one action. It could be seen on the CSV file that it registered two inputs on the same timestamp, while the user only did one tilt. This could happen for multiple reasons, such as noise in the webcam feed or the recording of rapid, successive actions that could not be distinguishable from the user's side.




```
Mean Continuous Trigger Rate Across All Test Persons: 8.47%
```

Figure 39: Continuous actions of the users, and the mean

For a visualization of all the actions, a graph has been made of every user action, grouped together accordingly. The grouping has composited the left and right sides into one group, so the different types of actions are distinguished based on the actions.

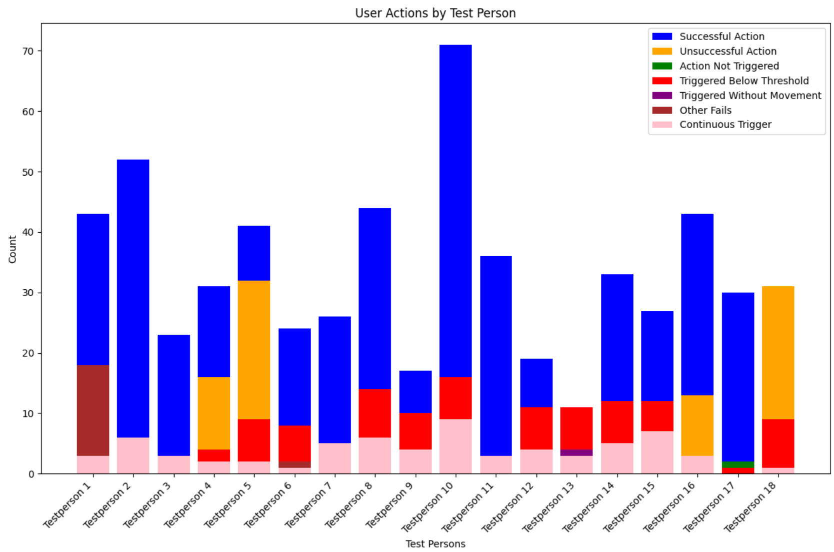


Figure 40: All the different actions grouped together accordingly

It can be seen that the grouped data of "Action Not Triggered" only shows up rarely, which is the parameter that looks when the test participants move past the threshold but the system does not trigger. And another action that does not show up often is "Triggered Without

Movement," which looks at when the system gets triggered with no attempt made by the participant. This shows that the system does not give false positives, such as incorrectly identifying an action, or false negatives, such as failing to identify an action, very often. If the mean of the total number of actions and the mean of both false positives and false negatives are taken, it comes to approximately 2.09 percent of actions, which are these errors from the systems side.

```
Testperson 1: 0/69
Testperson 2: 1/69
Testperson 3: 0/30
Testperson 4: 2/57
Testperson 5: 0/84
Testperson 6: 1/40
Testperson 7: 0/37
Testperson 8: 0/73
Testperson 9: 1/41
Testperson 10: 3/113
Testperson 11: 1/45
Testperson 12: 1/37
Testperson 13: 4/32
Testperson 14: 3/57
Testperson 15: 0/55
Testperson 16: 2/63
Testperson 17: 2/35
Testperson 18: 0/67

Mean Total Actions: 55.78

Mean Combined Type 1 or 2 Errors: 1.17/55.78 or 2.09%
```

Figure 41: Type 1 and 2 errors for each test-person and the mean for total

7.1.4 Relevant observations/findings

The aim of the final testing was to have at least 20 test participants, but there were only 18 recordings we're able to extract data from. For one of the participants, the CSV file that is used to log and confirm the data inputs they gave did not record the data for unexplained reasons, since it automatically creates and starts the file when the Unity program starts, but for this one test participant, it only created the file but did not log any data. It can be seen that the date of the webcam recording's starting timestamp and the timestamp for the creation of the file match. Since the CSV file was not recorded, it is not possible to check their error rate or other observations; that can not be confirmed just by interpreting the webcam recording.

For the second test participant, whose data was not able to be used, the reason lies in the hardware qualifications of our laptops that were used to test on. The way the system is set up, the data needs to be collected from one computer that runs both the Unity program and the Python head-tilt script, together with a feed from the webcam that needs to be recorded. Therefore, when the computer runs Unity and the recording is not on the screen and being rendered, the computer gives its thread less priority and reduces its performance to a minimum, leading to slower updates. Because of this, there was only one laptop computer that was meant to handle higher workloads and could run the whole system without losing a substantial amount of frames per second from the recording. Because our pilot test on one less powerful laptop had both Unity and the webcam feed open when conducted, it was not taken into consideration that the test participant needed to have the webcam feed minimized, leading to unusable data.

7.1.5 Completion time

Completion times for the various levels are measured using the in-game engine. As participants progress to the next scene, the duration is recorded and logged into a text file. The chart below visualizes the data, showing the minutes taken by players to complete each level[42. The level that the participants found the most challenging was level 3, with a total completion time of approximately 50 minutes.

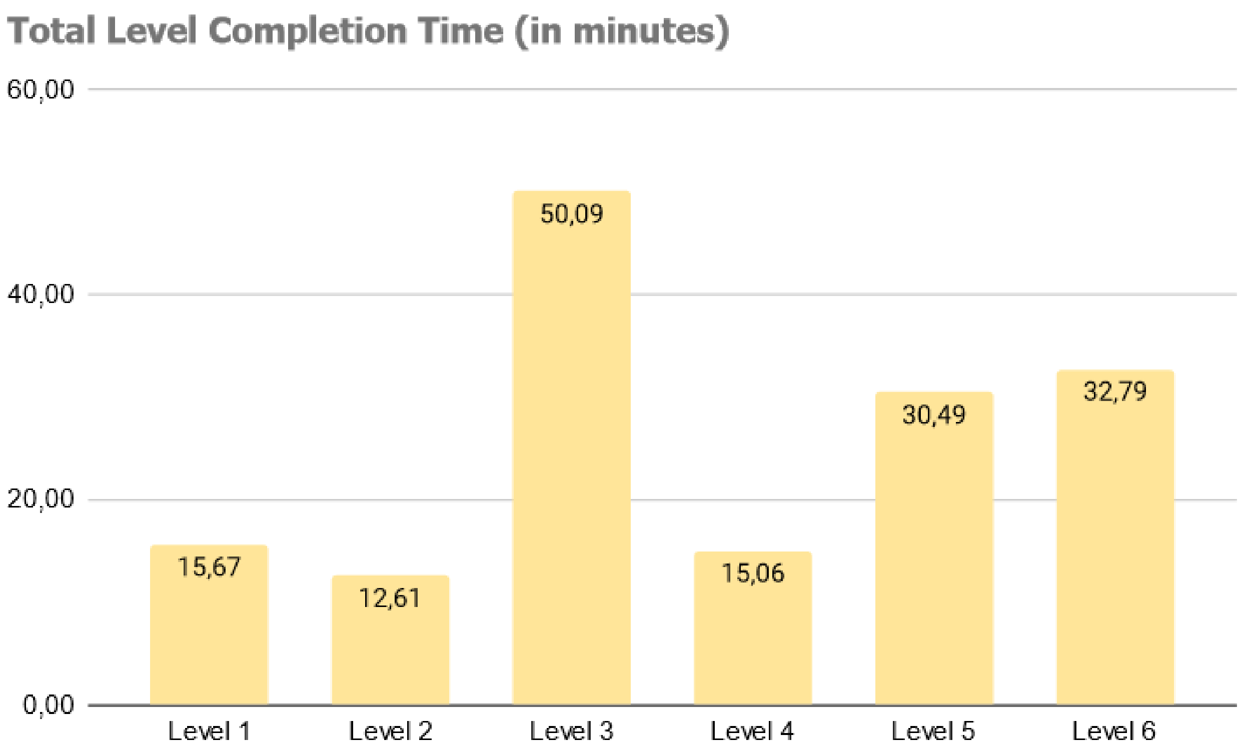


Figure 42: Approximate completion time in minutes over the various levels.

7.1.6 Results from questionnaire

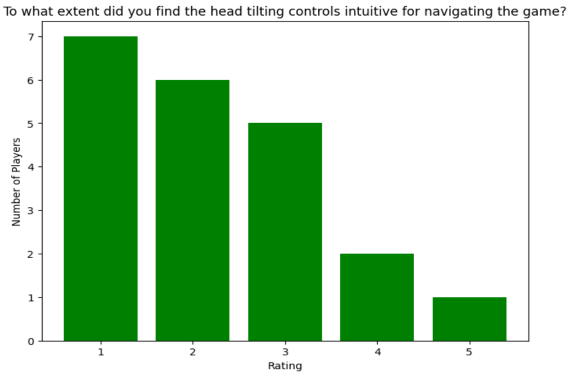
The questionnaire mentioned in the final test (4.4), conducted after the participants played the game, serves as a valuable tool for assessing various aspects of the participants' experiences. In addition to evaluating their overall gaming experience, the questionnaire gathers crucial demographic data that contributes to a comprehensive analysis of the study participants.

The collected demographic data includes information about the distribution of participants across different age groups, genders, and educational backgrounds. This data is essential for understanding the perspectives and preferences of the study participants. A breakdown of the demographic details is summarized below:

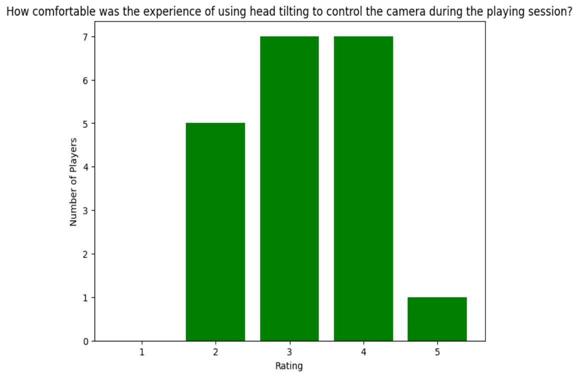
- **Age Groups:** The participants were categorized into distinct age groups, there were 13 test participants in the age group: 18–25 years old, and 6 test participants in the age group: 26–35 years old and lastly 1 in 36–50 years old.
- **Genders:** The gender distribution among the participants was recorded as: 17 men, 2 Woman and 1 non-binary
- **Educational Backgrounds:** Information regarding the participants' fields of study was collected: 19 test participants who are studying medialogi, and 1 test participant who were studying odontology.

How did the test participants feel about using head tilt

According to the questionnaire, the majority of respondents thought using the head-tilt mechanic was intuitive. However, while user satisfaction was still high, respondents didn't feel it to be as comfortable as they thought using it was intuitive.



(a) See Appendix (A.8.4) 1 (Very intuitive) - 5 (Not very intuitive)



(b) See Appendix (A.8.4) 1 (Very uncomfortable) - 5 (Very comfortable)

The questionnaire yielded favorable responses regarding participants ease of learning. Many of the respondents said it was simple to learn how to use the head-tilt system. This favorable opinion shows that users were able to quickly understand the capabilities and functions of the system, indicating that the learning materials and user interface were successful in promoting a seamless learning process. These results demonstrate a positive element of the participants' interaction with the facial tracking system, which is an important component of the user experience: simplicity of learning.

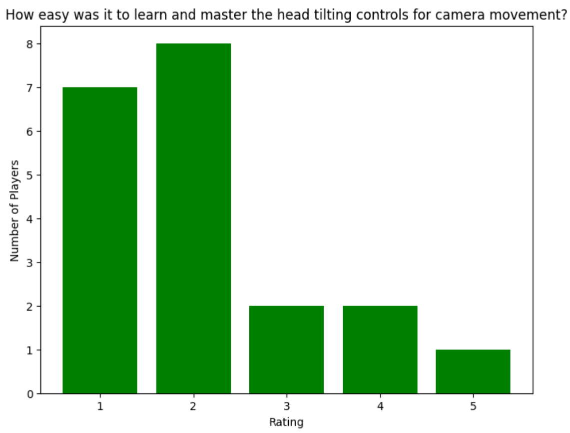


Figure 44: See Appendix (A.8.4 1 (Very easy) - 5 (Not very easy)

To sum up, the responses to the questionnaire show that test subjects were generally satisfied with the game's head-tilt mechanic. The majority of responses were pleased with the game's user interface, learning curve, and natural facial tracking application. The general feedback suggests that the participants found the game to be accessible and user-friendly, notwithstanding a few subtle reservations about comfort. This positive feedback indicates that, overall, players saw the technology as an enjoyable tool for their gaming experience, which is encouraging for the game's wider usage. These revelations offer insightful viewpoints that will be helpful in improving the game's usability in subsequent versions.

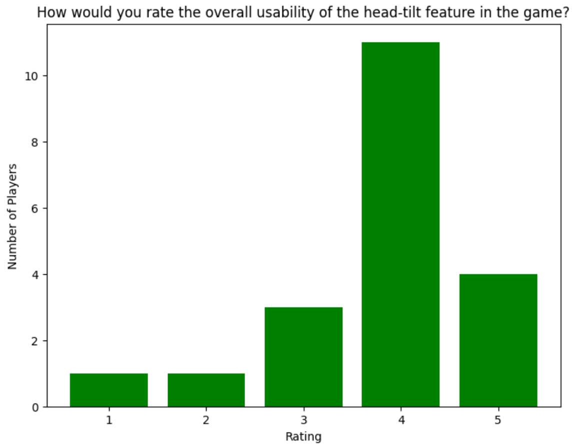


Figure 45: See Appendix (A.8.4 1 (Very Poor) - 5 (Very Good)

How the test participants experience the game with head tilt

Beyond just evaluating facial tracking, the study investigated the participants' head-tilt gaming experiences. The results showed an interesting pattern: a sizable majority of participants reported that they thought the head tilt interactions made the games especially engaging.

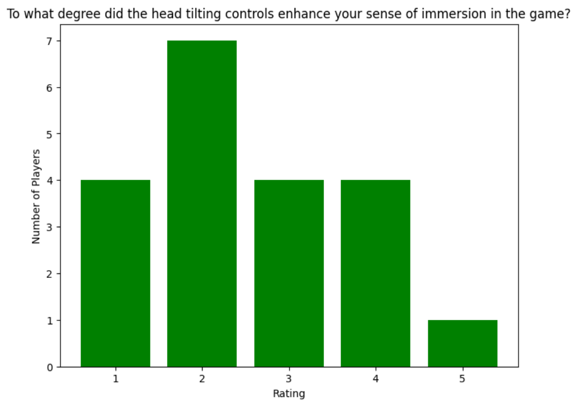


Figure 46: See Appendix (A.8.4 1 (Highly immersive) - 5 (Not Immersive at all)

The test participants' responses tended to fall on the middle-to-positive end of the scale when assessing the effect of head tilting on the gaming experience. Most participants said that head tilting did not significantly interfere with their ability to play. The general sentiment was neutral to good, despite a few participants expressing slight worries or modifications in adjusting to this engagement approach. This implies that, overall, players were able to smoothly integrate head tilting into their gaming without experiencing significant difficulty.

Figure 47: See Appendix (A.8.4 1 (Significantly hindered) - 5 (Significantly enhanced)

Participants' answers about how head tilting was incorporated into the gameplay were largely favorable. A sizable fraction of test subjects were in the middle or pleased with how head tilting smoothly enhanced the game experience. The results show that head tilting has the potential to be a useful and well-liked engagement technique that improves users' gaming experiences.

Test Participants felt about controlling the game with head tilt

Participants' responses to the questionnaire yielded insightful information about their experiences with the controls, especially with regard to how accurately the controls reflected their intended camera movement. The responses gathered show a consistent pattern where participants perceived the control system to be generally correct, with opinions ranging from indifferent to favorable. An essential component of determining the overall efficacy and player satisfaction with the game's control system is evaluating how successfully the controls translate user input into desired camera motions.

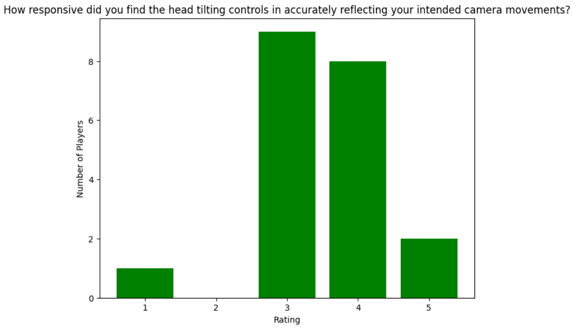


Figure 48: See Appendix (A.8.4 1 (Not Responsive at All) - 5 (Very Responsive)

In addition, the answers to the questionnaire provided insight into how the players felt about using the head tilt feature to aim or concentrate on particular game features. The results show that most participants felt that head tilting worked well in various parts of gameplay, and this is a generally positive impression. Its smooth integration of head tilting to improve focus and accuracy is consistent with the control system's generally favorable feedback.

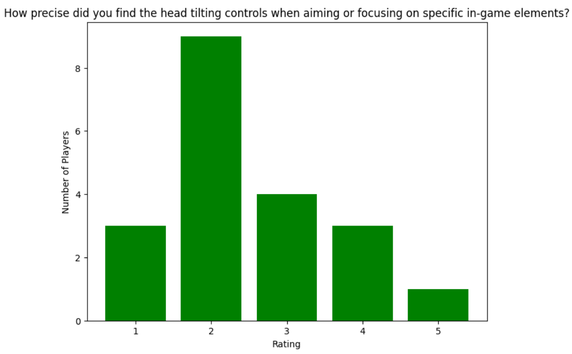


Figure 49: See Appendix (A.8.4 1 (Not Responsive at All) - 5 (Very Responsive)

7.1.7 Results from interview

The objective of the interview was to gather additional qualitative insights beyond what could be gathered from the non-standardized questionnaire alone. Only a selected few were chosen for the interview process to insert nuance into their quantitative responses.

1 Interview with test person:

Participant 1 discussed their ideas on the first level during the interview, characterizing it as "very intuitive." They discovered that using the mouse was generally significantly simpler and proposed that the totem pole might be smaller. They weren't expecting much, therefore they were pleasantly delighted by the goods. Participant 1 pointed out how important the positive components are and how well it suits the genre. They also praised the general quality of the game's nature and noted that making the game more responsive would be a wonderful enhancement.

2 Interview with test person:

The second interview provided insightful commentary on the product's functioning and usefulness. The test person expressed worries about the difficulties of zooming in on particular things, although they also mentioned that the system was typically usable for back and forth movements.

In terms of responsiveness, the test participants said that they saw performance increase during the session and at first thought the system was rather responsive. They made a comment on how much head movement was necessary and suggested that a lower threshold could improve the user experience in general. They still thought the head-tilt feature was fun and easy to use, but it did highlight the need for sensitivity changes. They also acknowledged the product's overall effectiveness and indicated that they were originally just as efficient. They recommended some modifications, one of which was to potentially lower the threshold for head movement.

3 Interview with test person:

Participant 3 gave positive feedback regarding their use of the product during the interview. After understanding the threshold concept, they found head tilt to be a really pleasurable feature and characterized it as simple and easy to use. Though they had anticipated a game more along the lines of an infinite runner, they were pleased with the puzzle element and emphasized how well it worked. Participant 3 emphasized how helpful head tilt is at improving the game's intuitiveness, particularly at the tutorial phase, when it greatly aided in improving comprehension of the gameplay. They also mentioned how the head-tilt feature made exploring easier and offered a more convenient method to gaze around. Participant 3 recommended changing the threshold in line with the suggestions made by other participants, suggesting a pattern in the input.

4 Interview with test person:

Participant 4 made thoughtful comments on their use of the product. They thought the head-tilt control was a novel and interesting method, although they encountered difficulties while using the mouse to multitask. Participant 4 noted that there was occasionally instability in the head tilt and that a significant tilt was required to obtain the correct rotation. After overcoming the learning curve, they discovered that the product performed as planned and lived up to expectations. While Participant 4 agreed that the product usually functioned well, she also noted that any deviations from the intended functionality were more evident than expected. Participant 4 described the control scheme as extremely intuitive. They recommended a minor modification to the camera's rotation, making it more progressive rather than a quarter turn every time.

5 Interview with test person:
Participant 5 offered insights into several facets of the product during the interview. They pointed out that the requirement for constant rotation could provide a gameplay restriction, especially with relation to the rear. This finding, which implies that some movements could feel constrictive during gameplay, shows attention to the user experience. Another participant brought attention to the game's reliance on lighting, saying that in order for the camera to detect motions in the game, well-lit areas must be used. Furthermore, Participant 5 expressed a concern about the pacing of the game, mentioning that it moves a bit too slowly.

8 Discussion

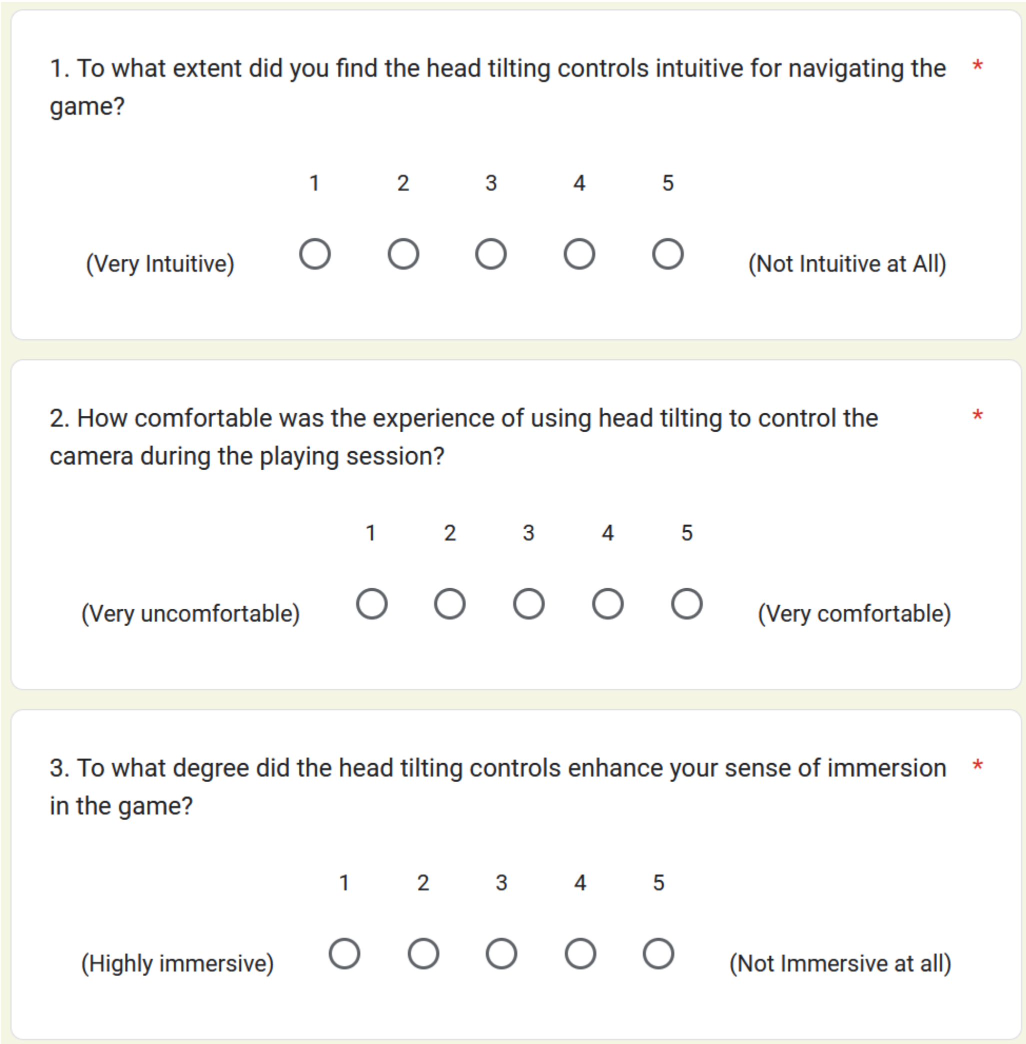
This section delves into the rationale and implications of adopting a non-standardized questionnaire in the present study. The choice of such an instrument hinges on its suitability for the specific aims and requirements of this research. The ensuing discussion will illuminate both the strengths and limitations inherent to the use of a non-standardized questionnaire in this context.

8.1 The Non-standardized questionnaire and why we chose it

In terms of selecting a non-standard questionnaire for this undertaking, It can have benefits and drawbacks, so the choice should ideally be made in light of the particular objectives and specifications of the research for this article. In this segment, we will go over the advantages and disadvantages of our questionnaire.

8.1.1 Being able to tailor it to once need

We can design questions specifically related to our research issue by using non-standardized questionnaires. This can be especially helpful when investigating a particular or specialized field, where standardized instruments might not be able to collect the kind of data we need for the final assessment.


1. To what extent did you find the head tilting controls intuitive for navigating the game? *

	1	2	3	4	5	
(Very Intuitive)	○	○	○	○	○	(Not Intuitive at All)

2. How comfortable was the experience of using head tilting to control the camera during the playing session? *

	1	2	3	4	5	
(Very uncomfortable)	○	○	○	○	○	(Very comfortable)

3. To what degree did the head tilting controls enhance your sense of immersion in the game? *

	1	2	3	4	5	
(Highly immersive)	○	○	○	○	○	(Not Immersive at all)

Figure 50: The first 3 question of the questionnaire

A downside when it comes to using a non-standardized questionnaire is that standardized questionnaires are rigorously tested for reliability and validity. Non-standardized questionnaires may lack these psychometric properties, raising concerns about the accuracy and consistency of your findings.

8.1.2 Flexibility

There is flexibility in the question structure, phrasing, and response possibilities available with non-standardized questionnaires. You can modify your questionnaire as necessary during the research process, thanks to its adaptability. A downside to this is that when using non-standardized questionnaires, it could be difficult to compare your results to those of other research projects. Comparing several research initiatives might be easier by using standardized instruments.

8.1.3 Interview questions

Since only a random selection would be chosen for a post-interview, it was essential that we chose questions that expanded on what we wanted to research with the final test. With the questions, it was our intent to give the test participants the possibility of giving an in-depth insight and concrete answer to their experience with playing the game. This is done with the hope of getting a nuanced understanding of their experience and perspective. Through this, it is hoped to gather more qualitative information, which will be better than getting an understanding from quantitative data alone. Which is important when it comes to how it felt to use head-tilt in the game. With them being so open, it can be hard to be able to standardize them, especially with the quantitative data we have also gathered that is easy to compare to all participants.

With the question, focus on the usability of the product. The inquiries pertain to the head-tilt experience of users, encompassing its intuitiveness and effect on productivity. This is crucial for assessing how users engage with and perceive these features. With the questions being so open-ended, the respondents may overlook certain details and concentrate more on others. This could generate bias in favor of the subjects you brought up and leave out other crucial facets of their experience.

The final questions specifically target any areas that participants believe need improvement. This can offer specific information that could be helpful for the head-tilt feature's continued evolution or iterations. However, the results can be less transferable to other contexts or comparable products because the questions are so tailored to our particular context and product. Which is important for comparing how people compare it to other games or products in general.

8.2 Significance of Results

It is important to evaluate throughout a project. It serves as a tool to assess the effectiveness and impact of the project. If used correctly, evaluation provides a systematic approach to gathering data and measuring performance, leading to informed decisions about the direction the project should go, what should be improved, and what should be removed completely. This is important for an iterative process of development. Cycling through development and evaluation.

8.2.1 Reliability

Reliability refers to the extent to which the test consistently produces similar results for the outcome. A reliable test should produce results that are comparable when given in a consistent environment. Maintaining consistency is crucial in order to prevent random influences from influencing any changes in test scores and to make sure they are accurately reflective of the measured construct.[29]

By doing all the tests in the same manner, we ensured reliability by using the same location, the same computer mouse, and the same webcam. By doing them at the same time, we also ensured the lighting would be as consistent as possible. A total of 20 participants went through the testing, and 18 of them were able to get reliable data.

8.2.2 Testing Conditions

Since the test is done in the same room, it allows for control over the testing environment, which ensures that the test is done as closely as possible for every test while also ensuring observational insights. This can offer valuable insights (such as non-verbal cues, expressions, and behaviors) that may not be captured from the questionnaire, such as the video capture of their face playing the game. Another important factor is that it is done on one of the research computers, so it can be run.

The downside of the testing conditions is that since the researcher is there, it may influence the test participants by looking at them for help or hints. Also, when it's done in a certain place, it ensures a limited diversity of participants since it is done at the university. Another problem is that they are taking place at a university in a specific place, and there may be distractions from people walking through and talking. Lastly, the problem is that the game is quite heavy to run, so it needs a quite specific computer.

8.2.3 The Game's Ethical Dilemmas

It's crucial to follow industry standards, regulatory requirements, and ethical guidelines while creating and deploying facial tracking in games in order to give players a courteous and enjoyable experience. Evaluate the ethical consequences on a regular basis as social norms and technology advance.

Even though a game that uses face tracking to control an aspect of it may have good intentions, there are ethical issues to take into account. A significant factor is privacy because it may be used to collect information about the player and their facial features, which many people may find personally identifiable. It is crucial to remember that the game just saves the location of the face using a script; no video of the face is stored after the game is finished. Therefore, to make sure the date isn't stored anywhere, it's important to not use any kind of third-party access where they should be able to store the data.

It is important to ensure user awareness through clear communication. This approach makes sure that players are fully aware of the ways in which the game uses facial tracking. Having open lines of communication is crucial to getting informed consent. This was done with the consent form (A.8.1)

8.2.4 Results

From the usability testing, the results of the SUS score showed that the head tilt was the more preferred mechanic to use for the users, and there are a couple of reasons, based on the testing, it has gotten above average while the blinking got below average. The first reason is the unintentional rotation of the in-game camera that the blinking can create, because the human eye has to blink once in a while, and it can cause inconvenience for the users when they do not have full control of the main game-play mechanic that is needed to clear the levels [Section]. There is also a level of robustness that the head-tilt mechanic have over the blinking, since the amount of control points that the blinking mechanic need to look at is four, two for each eye, they have a higher sensitivity to noise than the one control-point of the head-tilt. With fewer control points, the system is less affected and provide a more stable user experience.

For the final test, the participants helped us identify issues, regarding the rotation of the camera view. Specifically, the issue regarding the threshold of the head-tilting mechanic. The threshold not being wide enough, when it comes to quality assurance process, it allows us to find the problems where there are need for improvements. Anticipating and resolving problems before they worsen or have an adverse effect on the user is known as proactive problem-solving. When it comes to game creation, this means that rather than waiting for users to run into issues with the final release, we actively sought out and fixed any potential problems throughout the testing phase.

It is important for the player to understand what is going on, how their actions affect the game's world. If the player is ever in doubt about what their actions are doing, it will create a disconnect between the player and the game, causing confusion and/or frustration.

By systematically logging input results, we are able to analyze the different preferences and error rates, through this we are able to find individual success rates for left and right tilting, this offers valuable insights on how the product functioned. This thorough analyzing of the system errors, such as continuous inputs, add to the transparency of our evaluation.

8.2.5 Suggestions for Improvement

Because of the requirements the whole system has for a computer, we could only run one test at a time, therefore not running the testing as effectively as we could with some improvements to the system. The reason that the testing runs slower than acceptable on average and low-end computers is because the webcam feed needed to be recorded and could not be minimized in the background, and that makes its frames per second too slow to get any answers from. An improvement was that instead of running the whole system on one computer via localhost, it was using a LAN connection, so the processing was split between two computers. This means that two instances of the testing system could run in parallel and would have more time to get more results.

8.3 Future-work

The findings of this study have led to advancements in the application's functionality and efficiency. Through research and testing, several improvements have been identified that can further enhance the application's performance. The forthcoming sections will delve into potential areas for future work aimed at leveraging these improvements. Importance will be placed on both the modification of existing features and the integration of innovative features. This exploration will serve as a roadmap for continued advancement in this field.

- **Implementation of Visual Feedback Bar.**

  To address the lack of feedback during head tilting, a visual indicator is something to be worked on in the future. This bar will dynamically display the player's head tilt relative to the threshold required for camera movement, offering real-time, intuitive feedback.

- **Lowering Head-Tilt Thresholds**

  To combat the feeling of unresponsiveness, the sensitivity thresholds for head tilting will be lowered, allowing for more fluid and responsive camera movement. The threshold were 1/3 of the screen width, because of it was the found balance of not giving accidental inputs, but still not relatively low.

- **Customizable Settings**

  Instead of just lowering the threshold. Another solution, for a future development, could be a settings option will be added to allow players to adjust the head-tilt threshold to their comfort, ensuring a more personalized gaming experience.

- **Enhanced Hitbox Responsiveness.**

  There have been some issues with unresponsive hitboxes. A future development will focus on refining hitbox detection, ensuring consistent responsiveness across various gameplay scenarios. This includes maintaining hitbox stability regardless of zoom levels.

- **Zoom Functionality.**

  When zooming, it focused on the origin. For future development, a zoom feature that allowed players to get a closer view of important objects or areas in the game. This feature will not only enhance visual engagement but also aid in understanding and interacting with the game environment.

- **Comprehensive Tutorial.**

  Acknowledging the steep learning curve, a more in-depth tutorial will be developed.

This tutorial will guide players through the nuances of head-tilt control, zoom features, and optimized gameplay mechanics, ensuring a smoother onboarding experience.

9 Conclusion

The multidisciplinary field of visual computing encompasses a wide variety of technological advancements, all converging to redefine how we view and interact with computers. The field has even found its home in the world of video games. In the context of this project, we delved into the research and exploration of two technological advancements: eye-tracking and face-tracking. Through the iterative process could we determine what technology would have the utmost optimized performance. The selected technology would help answer another pivotal question, does the integration of said technology contribute to user satisfaction This inquiry was better summarized in our final problem statement:

"How can a game with an emphasis on puzzle-solving, utilize face-tracking as a method to control the in-game camera view, and simultaneously meet user satisfaction and usability?"

The participants of the SUS-test helped us eliminate blink-mechanic as a method of interaction, favoring head mechanic over blink-mechanic. While we acknowledge the established history of eye-tracking as a reliable method of interaction in media, as seen in the award-winning game 'Before your eyes', the participants perceived face-tracking to offer greater usability within the context of our game. Given that the game follows a puzzle-solving game design, similar to the popular title 'Monument Valley,' rather than a narrative-focused game-design, it can be argued that face-tracking suits better for the overall game experience.

The SUS test was conducted in the early stages of the iteration process, during which many crucial elements were yet to be implemented. These included fundamental design principles and a fully optimized head-tilt mechanic.

Several user concerns were related to the absence of adjustable threshold, encompassing both mouse input and webcam head-tilt data. Even regarding the usability test (6.3), particularly in relation to the blink-test, the test could potentially benefit from a model, that proficiently integrated concise eye-integration theories like the Blob-analysis. In our pursuit of a better design, it's apparent that we may have initially undervalued the profound impact of the design principle of feedback. The iterative nature of design processes demands not only creativity and technical finesse but a genuine appreciation for user input. Addressing these concerns through the implementation of these features holds the potential to enhance overall user satisfaction.

From the feedback we accumulated from questionnaires and interviews, it tells us that the products do have a positive user satisfaction rate, but there is still a need for improvement to align with user expectations. We can therefore conclude that our final problem statement has been met.

References

[1] Assistant Secretary for Public Affairs. *System usability scale (SUS)*. Sept. 2013. URL: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[2] Assistant Secretary for Public Affairs. *Usability testing*. Nov. 2013. URL: https://www.usability.gov/how-to-and-tools/methods/usability-testing.html.

[3] Dale Beermann and Greg Humphreys. "Visual computing in the future: Computer graphics as a remote service". In: *University of Virginia, Computer Science Department, University of Virginia Technical Report CS-2003-16* 25 (2003).

[4] *Before your eyes - steam page*. URL: https://store.steampowered.com/app/1082430/Before_Your_Eyes/.

[5] Steve Benford et al. "Contesting control: Journeys through surrender, self-awareness and looseness of control in embodied interaction". In: *Human–Computer Interaction* 36.5–6 (2020), pp. 361–389. DOI: 10.1080/07370024.2020.1754214.

[6] Kierstan Boyd and David Turbert. *Eye anatomy: Parts of the eye and how we see*. May 2023. URL: https://www.aao.org/eye-health/anatomy/parts-of-eye.

[7] John Brooke. "SUS: a retrospective". In: *Journal of Usability Studies* 8 (Jan. 2013), pp. 29–40.

[8] Yong Kok Ching, Anton Satria Prabuwono, and Riza Sulaiman. "Visitor face tracking system using OpenCV library". In: *2009 IEEE Student Conference on Research and Development (SCOReD)*. IEEE. 2009, pp. 196–199.

[9] Craiyon. "Craiyon V3". In: (2023). URL: https://www.craiyon.com/ (visited on 10/23/2023).

[10] Andrew Crossan et al. "Head tilting for interaction in mobile contexts". In: *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 2009, pp. 1–10.

[11] Mihaly Csikszentmihalyi. "Flow: The Psychology of Optimal Experience". In: Jan. 1990.

[12] Konstantinos G Derpanis. "Overview of the RANSAC Algorithm". In: *Image Rochester NY* 4.1 (2010), pp. 2–3.

[13] eyeware. "Understanding Eye Tracking How it Can Work for You: Definitions, Metrics, and Applications". In: (). URL: https://eyeware.tech/blog/what-is-eye-tracking/ (visited on 10/13/2023).

[14] Bryn Farnsworth. "How to Analyze and Interpret Heat Maps". In: (). URL: https://imotions.com/blog/learning/analyze-heat-maps/ (visited on 10/18/2023).

[15] Bryn Farnsworth. "What is Eye Tracking and How Does it Work?" In: (). URL: https://imotions.com/blog/learning/best-practice/eye-tracking-work/ (visited on 10/13/2023).

[16] GoodbyeWorld Games. *Before your eyes on steam*. Apr. 2021. URL: https://store.steampowered.com/app/1082430/Before_Your_Eyes/.

[17] ustwo games. *Monument valley: Panoramic edition on steam*. July 2022. URL: https://store.steampowered.com/app/1927720/Monument_Valley_Panoramic_Edition/.

[18] Anjith George. "IMAGE BASED EYE GAZE TRACKING AND ITS APPLICATIONS". In: (Aug. 2017). DOI: 10.13140/RG.2.2.33992.88329.

[19] Number None Inc. *Braid*. 2009.

[20] Sauro J and Lewis J R. *Quantifying the user experience: Practical statistics for user research. Morgan Kaufmann. (2nd ed.)* 2nd. Elsevier, 2016.

[21] Scott Kim. "What is a puzzle". In: *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. 2008, pp. 35–39.

[22] *Learning curves | Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play — dl.acm.org*. https://dl.acm.org/doi/abs/10.1145/2658537.2658695?casa_token=xgK4vHMHX_UAAAAA:f17eT01nmNB36pP303IMGNy0F69byjzHkSp9no jwuCj8#sec-ref. [Accessed 05-12-2023].

[23] Serena Lee-Cultura and Michail Giannakos. "Embodied Interaction and Spatial Skills: A Systematic Review of Empirical Studies". In: *Interacting with Computers* 32.4 (Nov. 2020), pp. 331–366. ISSN: 1873-7951. DOI: 10.1093/iwcomp/iwaa023. eprint: https://academic.oup.com/iwc/article-pdf/32/4/331/38855595/iwaa023.pdf. URL: https://doi.org/10.1093/iwcomp/iwaa023.

[24] Conor Linehan et al. "Learning Curves: Analysing Pace and Challenge in Four Successful Puzzle Games". In: *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '14. <conf-loc>, <city>Toronto</city>, <state>Ontario</state>, <country>Canada</country>, </conf-loc>: Association for Computing Machinery, 2014, pp. 181–190. ISBN: 9781450330145. DOI: 10.1145/2658537.2658695. URL: https://doi.org/10.1145/2658537.2658695.

[25] Derek Lomas et al. "Optimizing challenge in an educational game using large-scale design experiments". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Apr. 2013), pp. 89–98. DOI: 10.1145/2470654.2470668.

[26] *mediapipe 0.10.8*. URL: https://pypi.org/project/mediapipe/.

[27] Frank Nielsen. *Visual computing: Geometry, graphics, and vision (graphics series)*. Charles River Media, Inc., 2005.

[28] Amiga. Psygnosis. *Lemmings*. 1991.

[29] Paula Roberts and Helena Priest. "Reliability and validity in research". In: *Nursing standard* 20.44 (2006), pp. 41–46.

[30] Falk Schreiber and Daniel Weiskopf. *Quantitative visual computing*. 2022.

[31] Helen Sharp, Yvonne Rogers, and Jenny Preece. *Beyond Human-Computer Interaction*. 5th. Cambridge, United Kingdom: Cambridge University Press, 2019. ISBN: 978-1107184804.

[32] Lech Swirski, Andreas Bulling, and Neil Dodgson. "Robust real-time pupil tracking in highly off-axis images". In: *Eye Tracking Research and Applications Symposium (ETRA)* (Mar. 2012), pp. 173–176. DOI: 10.1145/2168556.2168585.

[33] Zhenyu Tang et al. "HeadPager: page turning with computer vision based head interaction". In: *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III 13*. Springer. 2017, pp. 249–257.

[34] Ustwo. *The Monument Valley game series from Ustwo Games*. URL: https://www.monumentvalleygame.com/.

[35] Valve. *Portal*. 2007.

[36] Valve. *Portal 2*. 2011.

[37] Eduardo Velloso and Marcus Carter. "The emergence of eyeplay: a survey of eye interaction in games". In: *Proceedings of the 2016 annual symposium on computer-human interaction in play*. 2016, pp. 171–185.

[38] *Wikipedia: Iterative and incremental development*. Nov. 2023. URL: https://en.wikipedia.org/wiki/Iterative_and_incremental_development.

[39] Yukang Yan et al. "Headgesture: Hands-free input approach leveraging head movements for hmd devices". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.4 (2018), pp. 1–23.

A Appendix

A.1 Eye-tracking pilot test showcase

https://youtu.be/hGf2vbdj4TA

A.2 Link to gameplay of Birdowatch

https://youtu.be/niamGKjEXlw

A.3 Link to github

https://github.com/gulfurs/P3$_{p}roject$

A.4 Link to AV Video

A.5 Usability test results

A.5.1 Results from head-tilt

2, 2, 4, 2, 3, 3, 4, 2, 3, 4 - SUS Score: 57.5 3, 2, 4, 1, 4, 2, 5, 3, 4, 2 - SUS Score: 75 2, 2, 4, 1, 4, 1, 3, 2, 4, 3 - SUS Score: 70 3, 2, 4, 2, 4, 1, 5, 2, 2, 2 - SUS Score: 72.5 4, 3, 3, 1, 4, 2, 4, 3, 3, 4 - SUS Score: 62.5 4, 1, 5, 1, 5, 1, 5, 1, 5, 1 - SUS Score: 97.5

A.5.2 Results from Blinking mechanic

A.6 Final Test Protocol

A.6.1 Expectations

We anticipate that users will submit comprehensive feedback and expect the test to provide valuable insights into the areas for improvement. The examination of user retention will guide our understanding of what captivates the audience, and with the information gathered, further enhance elements present in the game. The test will assist us in optimizing the gaming experience. In the event of negative responses towards elements in the game, a review of these elements will be conducted. This iterative process allows for the exploration of alternative approaches to the identified shortcomings, ensuring that the final product aligns more closely with what the users may expect.

A.6.2 Testing environment

The testing environment takes place in a well lit room, where there is a preferable lamp right above the test participants, so the detection algorithm has the best conditions. The background would also be white, and have the least amount of noise as possible, by not having any other elements in it.

A.6.3 Testing Participants

Convenience sampling

Minimum 20 participants

A.6.4 Execution: Participant

The participant enters the room, where we read them the info document about the game. The participant starts playthrough of the game, while the game is running, we are recording their face and logging the python scripts actions. The participant is free to think-out-loud, and would be noted as well. While the game is running the participants are not allowed hints or clues from the testers to help game completion. The participants have until completion or 15-minutes from game start. After the game is finished they will have a questionnaire to answer and lastly some interview questions from the testers.

A.6.5 Execution: Tester

While the testing of the game is going on, another computer is used to note down key points that the participants come up with during their testing. There will need to be at least two test runners during the whole test, one to control setup of the game play and asking the interview questions. One for notetaking the think-out-loud and interview, and have the computer ready for the survey, when the game is done. All level-completions are also timed.

A.6.6 Data Collection

There will be a recording of the test participants' faces, together with boundaries on the webcam that detect when the participants move to the left, center or right side of the screen. There will also be a record of the head tilt position for each participant, that logs in python the position each time it changes, together with a timestamp of when. Speak-aloud to get qualitative data during gameplay.

The way that the data collected would be used, would be by comparing the recording of the gameplay with the recording of the facecam, and checking if the movements line up, and if there are false-positives and false-negatives.

A.6.7 Statement

Welcome to this test session! The test involves having 10 minutes to navigate through a game using head movements. After the game, please fill out a brief questionnaire, and there may be a subsequent interview session where you are encouraged to share your thoughts aloud. Your feedback is crucial for improving our understanding of this innovative gaming experience. Enjoy it, and thank you for your participation!

Before the test begins, please grant permission for us to collect your data, including your age group, gender, field of study, and allow us to film your face during the gaming session.

A.6.8 Questionnaire

https://docs.google.com/forms/d/e/1FAIpQLSecSTAkVWHh79apuHtUG0Hpby9zY7nFW0GGXrWjR0o$_S$

A.6.9 Post Interview Questions

Can you elaborate on your responses on the scale and provide specific examples of situations where head-tilt was either particularly useful or problematic?

How would you describe your learning process regarding using head-tilt in the game? Were there specific challenges or moments that were especially memorable?

Can you provide more insight into your experiences with the game's response to your head-tilt inputs? Were there times when the response did not align with your expectations, or when it worked particularly well?

When discussing the integration of head-tilt into the gameplay mechanics, what are some specific elements or moments that made it either intuitive or less intuitive for you?

How has head-tilt affected your overall efficiency in the game? Can you provide examples of situations where it has made you more or less effective?

How do your expectations regarding the user-friendliness and performance of head-tilt align with your actual experiences in the game? Are there areas where it surprised you positively or negatively?

How do you assess the role of head-tilt in your overall gaming experience? How has it contributed to or influenced your enjoyment of the game as a whole?

Are there aspects of head-tilt that you feel could be improved or changed to make it more user-friendly?

A.7 Data from usability test

A.8 Questionnaire for final test

A.8.1 Consent form

Introduction: This questionnaire is part of a study, and we aim to gather your views and opinions. Your participation is voluntary, and your responses will be treated confidentially.

Purpose: The purpose of this questionnaire is to collect information about basic details of individuals participating in the test, and your responses will be used to create a demographic dataset of those we have tested.

Handling of Personal Information: We collect personal information, including age, gender, field of study, and occupation. This information will be treated confidentially and will only be used in connection with this study.

Face Recording: In addition to the questionnaire responses, please be informed that your face will be recorded during the study for the purpose of enhancing our understanding of non-verbal cues in participant reactions. This facial recording will also be treated confidentially and will only be used in connection with this study.

Participation and Consent: Your participation in this study is entirely voluntary. By answering the questions in the questionnaire and consenting to the face recording, you agree to the use of the collected information as described above.

Confidentiality: All collected information, including questionnaire responses and facial recordings, will be treated confidentially, and your responses will be anonymous in the reporting of results. We will not disclose your personal information to third parties without your explicit consent.

Right to Withdraw Consent: You have the right to withdraw your consent at any time without affecting the lawfulness of the processing based on consent before withdrawal.

Contact Information: If you have any questions regarding the processing of your personal information or wish to withdraw your consent, please contact Niels Christian Malte Felix Iuel at niuel22@student.aau.dk.

I hereby agree to have my data collected for the purpose of the study

- Yes

- No

A.8.2 Section 2: Collection of basic infomation

What is your age:

- 18 - 25 year
- 26 - 35 year
- 36 - 50 year
- 51 - 65 year
- 65+ year

What is your gender

- Man
- Woman
- Prefer not to answer
- other

What do you study "Short text field"

A.8.3 Section 3: Play the game!

I have played the game?

- Yes

A.8.4 Questionnaire

1. To what extent did you find the head tilting controls intuitive for navigating the game? Likert Scale: 1 (Very intuitive) - 5 (Not intuitive at all)

2. How comfortable was the experience of using head tilting to control the camera during the playing session? Likert scale: 1 (Very uncomfortable) - 5 (Very comfortable)

3. To what degree did the head tilting controls enhance your sense of immersion in the game? Likert Scale: 1 (Highly immersive) - 5 (Not Immersive at all)

4. How responsive did you find the head tilting controls in accurately reflecting your intended camera movements? Likert scale: 1 (Not Responsive at All) - 5 (Very Responsive)

5. How easy was it to learn and master the head tilting controls for camera movement? Likert scale: 1 (Very Easy) - 5 (Very difficult)

6. To what extent did the head tilting controls enhance or hinder your overall enjoyment of the game? Likert scale: 1 (Significantly Hindered) - 5 (Significantly enhanced)

7. How precise did you find the head tilting controls when aiming or focusing on specific in-game elements? Likert scale: 1 (Very precise) - 5 (Not precise at all)

8. Considering the game's design, how well do you think the head tilting controls complemented the overall gameplay experience? Likert scale: 1 (Poor Complement) - 5 (Excellent complement)

9. Did you experience any discomfort or motion sickness while using head tilting controls for an extended period? Likert Scale: 1 (No discomfort/motion sickness) - 5 (Frequent discomfort/motion sickness)

10. How likely are you to recommend a game with head tilting controls for the camera to a friend or fellow gamer? Likert Scale: 1 (Very Unlikely) - 5 (Very Likely)

11. How did the usability and performance of head-tilt in this game compare to your expectations? Likert Scale: 1 (Expectations meet) - 5 (Expectations not meet)

12. How would you rate the overall usability of the head-tilt feature in the game? Likert scale: 1 (Very Poor) - 5 (Very Good)

A.9 Data from final test

A.9.1 Ruleset

The measures taken is to check for false negatives, where the system fails to detect a change when the testperson is moving

r = one right side successful, and match the timestamp from the csv file l = one left successful, and match the timestamp from the csv file

rf = one right side fail (they do not reach far enough) lf = one left side fail (they do not reach far enough)

rd = right does not trigger (they move past threshold but system not triggered) ld = left does not trigger (they move past threshold but system not triggered)

rr = they attempt, but did not extend over the threshold, but it triggered lr = they attempt, but did not extend over the threshold, but it triggered

rn = There is no movement attempt, but system got triggered ln = There is no movement attempt, but system got triggered

rfo = right fail other (other reason that they looked right) lfo = left fail other (other reason that they looked left) fo = fail other (eg looks up)

In parentheses (...) = continuous triggering of side

A.9.2 Testperson 1

Data did not record

Testperson 1	**Testperson 2**	Testperson 3	Testperson 4	Testperson 5
r = 25	r = 28	r = 8	r = 15	r = 16
l = 18	l = 24	l = 15	l = 16	l = 25
rf = 4	rf = 3	rf = 1	rf = 11	rf = 17
lf = 1	lf = 3	lf = 1	lf = 5	lf = 15
rr = 0	rr = 0	rr = 2	rr = 4	rr = 8
lfo = 15	lfo = 4	lfo = 0	lfo = 2	lfo = 0
(r, r) = 1	(r, r) = 3	(r, r) = 1	(r, r) = 1	(r, r) = 0
(l, l) = 2	(l ,l) = 3	(l, l) = 2	(l, l) = 1	(l ,l) = 0
(lfo, lfo) = 2	(lfo, lfo) = 0	(lfo, lfo) = 0	(lfo, lfo) = 0	(lfo, lfo) = 0
ln = 0	ln = 1	ln = 0	ln = 2	ln = 0
rd = 0	rd = 0	rd = 0	rd = 0	rd = 0
ld = 0	ld = 0	ld = 0	ld = 0	ld = 0
fo = 0	fo = 0	fo = 0	fo = 0	fo = 0
rn = 0	rn = 0	rn = 0	rn = 0	rn = 0
lr = 0	lr = 0	lr = 0	lr = 0	lr = 1
(rr, rr) = 0	(rr, rr) = 0	(rr, rr) = 0	(rr, rr) = 0	(rr, rr) = 2

Testperson 6	**Testperson 7**	Testperson 8	Testperson 9	Testperson 10
r = 10	r = 17	r = 13	r = 2	r = 50
l = 14	l = 9	l = 31	l = 15	l = 21
rf = 3	rf = 2	rf = 6	rf = 4	rf = 7
lf = 1	lf = 0	lf = 3	lf = 3	lf = 6
rr = 7	rr = 3	rr = 14	rr = 10	rr = 15
lfo = 2	lfo = 0	lfo = 0	lfo = 2	lfo = 1
(r, r) = 0	(r, r) = 4	(r, r) = 0	(r, r) = 0	(r, r) = 6
(l ,l) = 1	(l ,l) = 0	(l ,l) = 4	(l ,l) = 2	(l ,l) = 2
(lfo, lfo) = 0	(lfo, lfo) = 0	(lfo, lfo) = 0	(lfo, lfo) = 0	(lfo, lfo) = 0
ln = 0	ln = 0	ln = 0	ln = 0	ln = 0
rd = 0	rd = 0	rd = 0	rd = 0	rd = 0
ld = 1	ld = 0	ld = 0	ld = 1	ld = 3
fo = 0	fo = 1	fo = 0	fo = 0	fo = 0
rn = 0	rn = 0	rn = 0	rn = 0	rn = 0
lr = 1	lr = 0	lr = 0	lr = 0	lr = 1
(rr, rr) = 0	(rr, rr) = 1	(rr, rr) = 2	(rr, rr) = 2	(rr, rr) = 1

Testperson 11	**Testperson 12**	Testperson 13	Testperson 14	Testperson 15
r = 21	r = 8	r = 9	r = 33	r = 11
l = 15	l = 11	l = 1	l = 0	l = 16
rf = 0	rf = 0	rf = 2	rf = 0	rf = 6
lf = 3	lf = 2	lf = 2	lf = 4	lf = 3
rr = 2	rr = 11	rr = 11	rr = 12	rr = 12
lfo = 0	lfo = 0	lfo = 0	lfo = 0	lfo = 0
(r, r) = 3	(r, r) = 2	(r, r) = 1	(r, r) = 5	(r, r) = 3
(l ,l) = 0	(l ,l) = 1	(l ,l) = 1	(l ,l) = 0	(l ,l) = 4
(lfo, lfo) = 0	(lfo, lfo) = 0	(lfo, lfo) = 0	(lfo, lfo) = 0	(lfo, lfo) = 0
ln = 1	ln = 0	ln = 0	ln = 0	ln = 0
rd = 0	rd = 0	rd = 0	rd = 0	rd = 0
ld = 0	ld = 1	ld = 0	ld = 0	ld = 0
fo = 0	fo = 0	fo = 0	fo = 0	fo = 0
rn = 0	rn = 0	rn = 4	rn = 3	rn = 0
lr = 0	lr = 0	lr = 0	lr = 0	lr = 0
(rr, rr) = 0	(rr, rr) = 1	(rr, rr) = 1	(rr, rr) = 0	(rr, rr) = 0

Testperson 16	**Testperson 17**	Testperson 18
r = 18	r = 13	r = 23
l = 25	l = 17	l = 3
rf = 2	rf = 2	rf = 5
lf = 11	lf = 0	lf = 26
rr = 1	rr = 1	rr = 9
lfo = 0	lfo = 0	lfo = 0
(r, r) = 2	(r, r) = 0	(r, r) = 1
(l ,l) = 0	(l ,l) = 0	(l ,l) = 0
(lfo, lfo) = 0	(lfo, lfo) = 0	(lfo, lfo) = 0
ln = 0	ln = 0	ln = 0
rd = 1	rd = 0	rd = 0
ld = 1	ld = 2	ld = 0
fo = 0	fo = 0	fo = 0
rn = 0	rn = 0	rn = 0
lr = 1	lr = 0	lr = 0
(rr, rr) = 1	(rr, rr) = 0	(rr, rr) = 0