Aalborg University Copenhagen

Semester:
MED10 – 2022 Spring

Title:
A Continuous Evaluation of Narrative Intelligibility and a Photogrammetric Approach to Enhancing the Narrative Game Design in 'Mind Diver'

Aalborg University Copenhagen
A.C. Frederikskaj 12,
DK-2450, Copenhagen SV

Semester Coordinator: Stefania Serafin
Secretary: Lisbeth Nykjær
Phone: 9940 2470
Mail: Lny@create.aau.dk

Project Period:
February 1st – May 25th

Semester Theme:
Master Thesis

Supervisor(s):
Olga Timcenko

Project group no.:
N/A

Members:
Andreas Havmøller Johnsen
Gabriel Henriksen Gaspar

Abstract:
This thesis explores the use of continuous evaluation methods to provide usable player insights into the narrative intelligibility of a narrative-focused video game, Mind Diver, a video game production co-created with other students from various universities and various backgrounds.

With 'Mind Diver' as the case study, this thesis proposes a new method of qualitative data collection based on interpretive analysis of player comprehension. Using in-game questioning, we gather data on player comprehension of narratives in real time, along with other qualitative and quantitative data such as 'Narrative Engagement' and 'Continuation Desire'. Furthermore, we explore an artistic use of photogrammetry and how it affected the gameplay and narrative by using the imperfections that come with it and looking at the production process to understand how this was accomplished.

The study found that using an in-game feedback tool helped identify areas where the player would have difficulty understanding the connection between the story and the puzzles. The study also suggests that the in-game questions generally did not negatively impact the narrative engagement. However, the constant requirement for players to answer questions while playing the game seems to indicate that some felt that the constant in-game questions were intrusive and annoying, while others found it helpful to pause and reflect on their choices.

The artistic use of photogrammetry was found to have a positive impact on the game's narrative experience. Additionally, the study also found that using an iPad Pro 12 with the software application PolyCam, was a smart decision for the purposes of our case study, due to the convenience of cloud processing.

Even so, the thesis is mainly exploratory and the findings should be interpreted with caution since reliability and validity are limited.

A Continuous Evaluation of Narrative Intelligibility and a Photogrammetric Approach to Enhancing the Narrative Game Design in 'Mind Diver'

Andreas H. Johnsen & Gabriel H. Gaspar

Supervisor:
Olga Timcenko

Master of Science in Medialogy, Digital Games
Aalborg University of Copenhagen
Denmark
May 25, 2022

Contents

1 Introduction

Once upon a time, narratives were purely passive experiences where the viewer would experience the author's chosen structure. Today, we see stories play out in multiple different mediums. As a result, video games, in particular, have given rise to the notion of interactive narratives, which have transformed the narrative experience from a passive to an active one. It presents a variety of narrative structures, from linear and branching plotlines to giving players agency in influencing the narrative. As a result of these many narrative structures and direct gameplay that can influence how a story is told (or completely change it), a hypothetical infinite amount of ways to explore a narrative makes it even more difficult to ensure a coherent intended narrative.

The ability to automatically evaluate a player's understanding of a narrative based on each player's own journey through the narrative is valuable knowledge for any game developer or game researcher. Flow[Csikszentmihalyi 2014], engagement, and immersion are common aspects of a player's experience in a videogame that you would look into when conducting game research[Schønau-Fog and Bjørner 2012][Przybylski et al. 2010][Abbasi et al. 2017]. However, little has been said about specifically looking into the comprehension of the narrative in which the player is actively involved. Further, qualitative metrics for exploring immersion, flow theory, and engagement are often based on interviews or questionnaires given after a finished experience, so it is impossible to get players' thoughts at the moment. *'Continuation Desire'* is one approach trying to solve this by interrupting the experience at certain points in the game, asking questions to evaluate desire to continue playing a video game providing a representation of the players' thoughts at the exact moment in time[Schoenau-Fog 2011]. Despite this approach's advantages, however, it can negatively impact the pacing of the game, which in turn can cause a misrepresentation of the authentic experience. Furthermore, another problem, which appears to be unsolved in academic video game research, is the issue of storing continuous data effectively. Most often, either a facilitator would ask and record the answers as they were given, or they would leave the game completely to answer an external survey, or they would have the questions in the game, but they would have to deliver the data themselves. We believe the latter is the best option, but to extend this further we want to have the same convenience of the data being sent directly online. Providing us with more data and less risk of participants doing things wrong as well as not getting the data we need, as well as allowing the testing to run completely independently without a test facilitator present, enhancing the experience of playing a video game as a whole.

The aim of this system is to provide a method for reliably gathering continuous qualitative data in an interactive narrative, in our use case we explore a term referred to as *'Narrative Intelligibility'*[Bruni and Baceviciute 2013], being the notion of an authors narrative being told as intended to the audience. The broader objective is to contribute an easy-to-use method to any form of continuous qualitative data collection within interactive digital media.

With that being said, we will also highlight a specific element of the game we are creating for the evaluation, photogrammetry.
The photogrammetric process is a versatile and widely employed method of extracting 3D geometries from 2D images. The use of photogrammetry in video games has only recently started as the technology has been disregarded due to the technical limitations of video game engines. In recent years, however, many video games have started using 3D photogrammetry for creating game assets. Photogrammetry enables the recreation of pure natural details as well as creating 3D game assets in a much faster manner than traditional techniques[Statham 2020]. Currently,

most games that utilize photogrammetry use it as a non-interactive environment part of the game and then add 'handmade' 3D assets on top of the photogrammetry assets, as well as relying solely on it for photorealism rather than for artistic purposes. For this project, we will implement sections of the game that use purely Photogrammetry and are directly intended to complement an interactive narrative. We can thus contribute to the challenges of implementing photogrammetry into Unity, while also showing how our case study tied photogrammetry directly to gameplay and story as well as providing other artistic ways to use photogrammetry in video games.

1.1 Motivation

This thesis is accompanied by collaboration with other students from a several universities and a variety of academic backgrounds on the video game 'Mind Diver', a narrative video game supported by the National Film School of Denmark. At the time of joining 'Mind Diver', some key requirements for the game have already been established. These include the artistic approach to photogrammetry, the style of gameplay, as well as the story and theme. Our main objective in joining the project is to provide an in-game feedback system that can cater to both quantitative and qualitative data collection within the game. The thesis evaluation will, therefore, focus on the narrative intelligibility of the game. We also worked on designing and developing the game from a prototype to a fully functional and engaging video game experience using photogrammetry.

In the big production of 'Mind Diver', Andreas H. Johnsen has been a programmer and Gabriel H. Gaspar has been a game- and level designer. In light of these two distinct but core disciplines, we will provide both design and technical perspectives when developing a video game. For our academic endeavours within video game research, our thesis explores continuous evaluation methods to provide usable player insights within narrative intelligibility, along with how it relates to narrative game design, as well as how photogrammetry impacts the narrative experience both from a production and an artistic standpoint.

2 Analysis

In this section, we will focus on photogrammetry and interactive narratives, two of our major topics. We will discuss photogrammetry primarily as a technology, along with a similar method, Light Detection and Ranging (LiDAR). Next, we will explore specific applications of photogrammetry in video game development, including examples of industry workflows, and the software available to support them. Afterward, we demonstrate the state-of-the-art for the type of game we are creating, information games, leading to narrative theory. In order to accomplish this, we must look at narratology, specifically interactive narratives, as the core of our research. To support our methodology, we will examine scientific proposals related to narrative measurement, intrusiveness measures, and industry proposals related to in-game feedback tools.

2.1 Photogrammetry

"It is relatively easy to get into photogrammetry, but it is really difficult to master it."[Poznanski 2014]
A foundation will be laid for our prototype by defining photogrammetry from a technical point of view, followed by an explanation of how videogames use photogrammetry. In this way, we can gain insight into the basis of our design choices and how we utilize photogrammetry.

2.1.1 Defining Photogrammetry

The photogrammetric technique is a versatile and widely used method for obtaining 3D geometry from 2D images, or as written by the International Society for Photogrammetry and Remote Sensing:
"Photogrammetry is the science and technology of extracting reliable three-dimensional geometric and thematic information, often over time, of objects and scenes from image and range data."[ISPRS-Foundation 2016].
Photogrammetry offers the versatility of not requiring expensive and specialized equipment to achieve generally decent 3D model results. With only an average smartphone to capture the imagery, which is used by the photogrammetry algorithms, the algorithms can create 3D models of at least acceptable quality. Various methods of photogrammetry exist, however all are aimed at obtaining information about the shape and location of objects present in images taken of a particular object or area. The data is then typically used for representation either in three dimensions as digital models, or in two dimensions as drawings or maps[Luhmann et al. 2019].
A photogrammetric process begins by first acquiring imagery of the object or area to be scanned. Once this imagery is collected, it should be given to a photogrammetry software program for processing. The program will then provide the desired output after processing the images. A few of these software applications are described in detail in Photogrammetry Software 2.1.4, while a general overview of the process is provided in Figure 1.

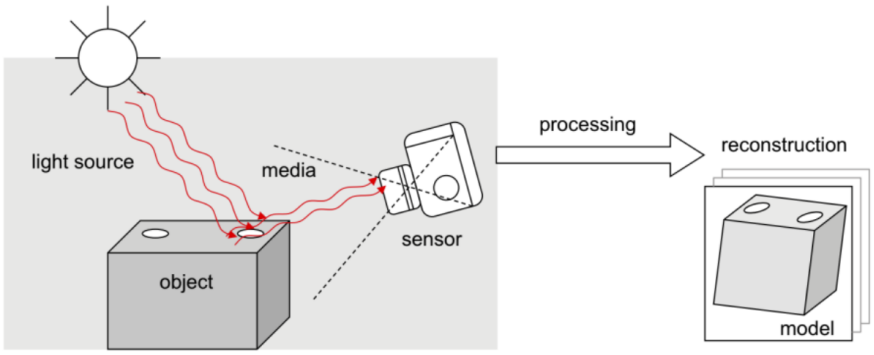


Figure 1: A simple overview of factors to think of when taking pictures[Luhmann et al. 2019]

In particular, when obtaining imagery, one should take into account a number of factors, including the light source, the camera setup, and attributes of the imagery. This can be seen on Figure 2.

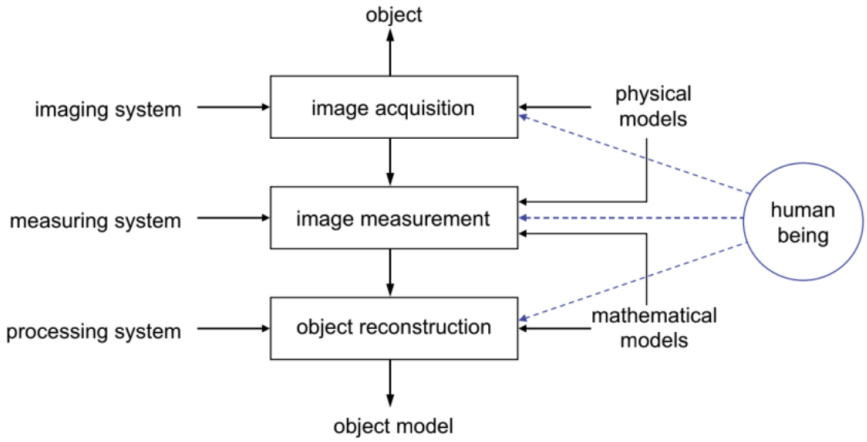


Figure 2: An overview of the process from imagery to model[Luhmann et al. 2019]

It is then necessary to process the imagery in order to determine the points in space for output. In stereo-photogrammetry, this is commonly achieved through shooting rays from camera locations towards points on the object. By using this information, it is possible to determine the coordinates of the point in space by determining intersections between rays from different images[Luhmann et al. 2019]. Thus, it is also necessary to determine the positions of the various perspectives. This is why in most photogrammetry software, the first step after importing imagery is to align the images. Structure-from-motion (SfM) is a method that allows for automatic alignment of images, without providing any additional information other than the imagery itself[Luhmann et al. 2019]. When aligning the images, the limited number of points in space that are utilized for the process results in what is known as a sparse point cloud. To be able to properly represent the structure as a 3D model, more points in space will be required, so a dense point cloud will have to be generated by processing many more coordinates. By forming triangles between sets of three points, the dense point cloud can then be used to create a mesh for a 3D model[Luhmann et al. 2019]. In addition to utilizing mathematical

models to acquire the required information, it is also possible to acquire the information in other ways with the correct equipment. An example of this is the use of Light Detection and Ranging(LiDAR)[NOAA 2012] equipment, which will be described in Defining LiDAR 2.1.2.

2.1.2 Defining LiDAR

A LiDAR[NOAA 2012] sensor allows for the simultaneous capture of images and measurement of distances, allowing for the creation of depth maps without requiring the use of mathematical models to calculate them. In the case of photogrammetry, depth maps can be created without the necessity of processing the images with mathematical models. For the measurement of distances, LiDAR requires specialized equipment, as the measurements rely on the reflection of light beams. As a result, the equipment must be able to send laser pulses and receive their reflections. For the collection of aerial data, other equipment is also required to ensure accurate measurements, as can be seen on Figure 3[NOAA 2012].

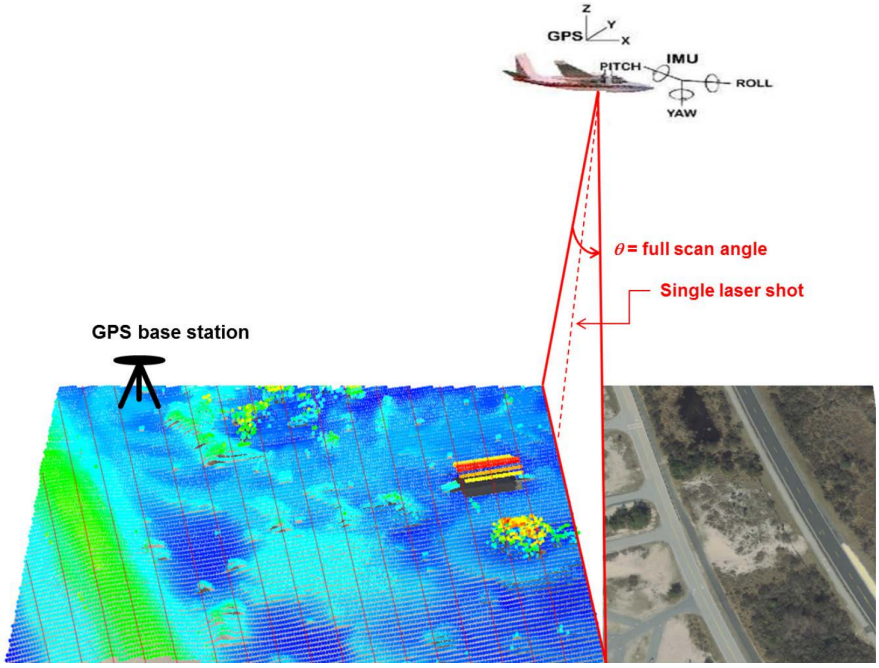


Figure 3: A schematic diagram of airborne LiDAR[NOAA 2012]

In terms of specialized equipment, at the time this thesis was written, Apple's iPhone 12 Pro, iPhone 13 Pro, and iPad Pro models from 2020 are capable of performing LiDAR scanning. Using this method, you can easily, quickly, and conveniently create 3D assets.

Now that we have explored the technical aspects of photogrammetry and LiDAR, we will turn our attention to our perspective on utilizing photogrammetry, which is video games.

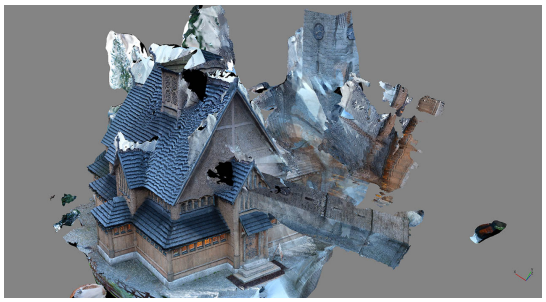
2.1.3 Photogrammetry in video games

Until recently, video games have not been investing a lot of resources into photogrammetry. For 3D assets, the real world is used only as a visual reference. Because of real-time rendering, game engines have been limited to meshes with lower resolutions while photogrammetry is able to output extremely dense meshes with millions of polygons[Foster and Halbstein 2014]. As of 2021, Unity and Unreal, the two most popular commercial game engines accept 3D assets with up to 65534 vertices for a 16bit index buffer or upwards of 4 billion vertices for a 32bit index buffer. The difference between the two is that, first and foremost, older GPUs does not support 32bit, meaning that if a model is used with a 32bit index buffer, it might simply not render in your game on certain devices. Another difference is the bandwith and resources taken up by a model, since the amount of bits taken by a model will differ between the two[Unity 2022]. Therefore most games would go with a 16 bit index buffer, Nataska Statham has done research within photogrammetry in videogames, and she lays the groundwork for commercial video games that really started benefiting of on photogrammetry, both in terms of production and creating quality 3D assets[Statham 2020].
The 2014 video game "The Vanishing of Ethan Carter" was the first to use photogrammetry extensively in video games[Statham 2020]. The developers behind 'The Vanishing of Ethan Carter' have shared their experience with creating their 3D assets using photogrammetry. One of their 3D artists directly working with photogrammetry for the game, Andrzej Poznanski shares some valuable insights on what to be aware of when taking photographs for photogrammetry[Poznanski 2014] .

- Do not use Shallow depth of field (DOF)
- On a pixel level, camera shake, scene/background movement and ISO noise are all things can mess up the scan.
- Avoid high contrast lighting
- Avoid highlights on photos
- Everything in a photo should be static, including background and lighting

A core benefit of using photogrammetry for video game assets is the ability to create more natural and immersive environments as the assets can get the more realistic minor details, such as ageing and dirt on the environment, which can be hard and much more time consuming to achieve by making assets from scratch[Statham 2020]. Figure 4 shows two examples of photogrammetry scans of the game 'The Vanishing of Ethan Carter', one when it is done correctly and another one example when not following some of the guidelines when producing photogrammetry 3D content.

(a) Photogrammetry done good
[Poznanski 2014]



(b) Photogrammetry done bad

Figure 4: Photogrammetry examples of 'The Vanishing of Ethan Carter
[Poznanski 2014]

Ponznanski explains how they set up for photographing for photogrammetry. In order to take a photo of a static object, a free-standing camera would be used, but for moving objects and faces it becomes much more complicated, since they need to use multiple cameras from different angles to trigger the photo simultaneously. As part of the photo processing process, the developers used Photoscan from Agisoft and then manually optimized the raw scan output for the low-poly specification needed by the game engine[Poznanski 2014].
Video-game company DICE, the people behind the Battlefield franchise, has also invested a lot in Photogrammetry in its games. Their first major game that used Photogrammetry extensively was Star Wars Battlefront (2015).

- Cover all angles to reduce cleanup
- avoid cast shadows
- Most reliable assets was made by covering the full scanned object was visible in most of the photos
- Avoid scanning objects with too intricate details
- Ensure that all capture methods are consistent and clean
- Weather conditions can destroy consistent results.
- Having lighting equipment is ideal.

After all photos had been captured managing the data was a tedious manual workflow which they believe can be mostly automated going forward.

- Over 100.00 photos had to be manually sorted
- Over 2000 assets had to be manually reviewed
- Over 14 hours of video footage had to be categorized

Most assets need to be masked, which can be very time-consuming if not automated. For terrain, baking at high gave the most usable and manageable results, while baking at ultra gave the most effective results, though at the cost of a large texture memory. In our project, photogrammetry is used more for artistic purposes. In both of these examples, it is critical that

the 3D assets produced by photogrammetry are high-fidelity, visually accurate, and aligned with the hand-made 3D assets. Our approach embraces photogrammetric errors and decimates the 3D assets', decimation allows us to reduce the vertex count of meshes while changing little about their shape, providing the look with a low-polygonal style.

2.1.4 Photogrammetry Software

Various software applications are available for conducting photogrammetry, some of which are free, some of which are open source, and many of which are paid[Übel 2021]. In the previous sections, we discussed photogrammetry, LiDAR, and its uses in video games. In this section, four software applications will be examined, namely Polycam[Polycam 2020], Multi-View Environment[Fuhrmann et al. n.d.], Metashape[Agisoft n.d.] and Reality Capture[Reality n.d.].

2.1.4.1 Polycam

In the beginning, Polycam was developed as a LiDAR scanning tool for Apple devices that had LiDAR sensors, but it has since been further developed by adding a photogrammetry mode, which allows most Apple products to utilize this software and generate 3D models. It is developed by Polycam Inc.[Polycam 2020]. The free version of the software allows unlimited LiDAR scans, but only five photo mode scans, and only exports to GLTF or iMessage. On the other hand, the paid subscription allows 150 photo mode captures and export to any file format. Polycam uses cloud processing, meaning the data is processed in the cloud instead of on the device. It is possible to see the results of a scan quickly, allowing for redoing a scan if necessary on site. As a result, the user can feel more secure by knowing that the scan was successful, and avoid the situation where a scan is performed, but the results are of poor quality, meaning that a new day has to be scheduled and performed for a new scan. Additionally, Polycam allows users to edit the output before it is exported to a 3D model format.

2.1.4.2 Multi-View Environment

Multi-View Environment(MVE) is an open source solution[Fuhrmann et al. n.d.]. Developed at the Technical University of Darmstadt in Germany, its source code is available on GitHub[Fuhrmann et al. 2022]. In addition to being able to use MVE as part of a Graphical User Interface(GUI), it can be utilized entirely through a command line interface to make it feasible to implement it within batching processes. This program uses its own file format, .mve, to handle the setup of scenes and related data. The basic pipeline consists of utilizing six MVE binaries, starting with makescene to create a .mve scene, followed by sfmrecon to perform SfM on the input images in the scene. With the base setup, dmrecon is used to create a depthmap, using Multi-View Stereo, for all the images. Then, scene2pset is used to create a dense point cloud using the depth maps generated by dmrecon, and when the dense point cloud has been created, fssrecon will construct a mesh from the points, which should then be cleaned with meshclean.
The attempts to use MVE indicate that even with the developers' recommendations regarding how to use this tool, there is still a great deal of learning to be done. The reason for this is that none of the attempted scans produced any suitable output, in most cases completely missing the central element of the scans. MVE would also very quickly begin taking a very long time to process. With just over 400 images of 1920x1080 resolution as input, it could take over four hours to obtain an output, while with around 520 images, it took over eight hours.

2.1.4.3 Metashape

Metashape is a commercial product developed by Agisoft[Agisoft n.d.]. There are two versions of this product, a standard version and a much more expensive professional version with a free 30-day trial of the standard version. Metashape is a simple to use software that features a GUI and the ability to generate batching processes that can be saved as XML files. The standard version does not allow for command line integration, and as a result, it is difficult to automate a full batch process pipeline from image input through video extraction to model output. In the professional version, python scripting is supported, which enables one to create a python script that hooks into Metashape. This script can then be called from the command line as part of a batching process. Metashape consists of 4 main steps after you provide it with the input images. The first step is to have these images aligned, followed by the creation of a dense point cloud, which also creates depth maps. From this basis, it is possible to create a mesh and then texturize this mesh. On the subject of performance, Metashape performed well, as opposed to MVE, which would take over four hours to process a scene with over 400 input images, Metashape was able to handle a little over 1000 images in approximately three hours[Agisoft n.d.].

2.1.4.4 Reality Capture

RealityCapture is a commercial product developed by CapturingReality[Reality n.d.]. This product, like Metashape, offers two ways of paying for it. There is only one full version, which is free of charge. Payment may be made in the form of a license for an unlimited number of inputs or a payment model known as Pay Per Input. Pay Per Input is a process in which you pay based on the amount of input you provide to the program. By this, it means that you are free to test the software and utilize it from input to modeling. Only when you are satisfied with the output will you have to pay for the 3D model, where the application will inform you of the exact cost. Using the software is similar to using Metashape, add the photos, align them and then construct the model. The RealityCapture software also features color correction capabilities for the texturing of the model, as well as the ability to simplify the model. This is a great feature since a model created using photogrammetry, in most cases, will have a high polygon count and would therefore be too expensive to be used outside of pre-renders or single model displays. This note is of interest due to the fact that photogrammetry can also be used in video games. One indication of this can be found in the fact that CapturingReality has entered into a partnership with Epic Games, the creators of Unreal Engine. Furthermore, there are already examples of high profile games made with assets created through the use of photogrammetry, as noted in Photogrammetry in video games 2.1.3.

We will determine which software tools will be beneficial to our production as part of our design process. In order to gain a deeper understanding of what type of game we are creating, how we should approach design and what we want to evaluate, we will examine proven works within the genre in which we are developing in order to incorporate the photogrammetry aesthetics of our video game design into the prototype.

2.2 State-of-the-art within Information Games

We are collaborating on a detective game called Mind Diver. A vision and inspiration for 'Mind Diver' had been formed prior to our joining. In essence, it can be described as a surreal detective game inside the mind of a person and would be considered an 'information game', which is not genre-specific as it may fall into other genre-categories as well. Game Designer, Tom Francis describes 'information games' as a class of games, meaning that it has a place in many different kinds of games[Francis 2019]. Essentially, the goal of information games is to acquire information. This is achieved by using what you know already and deducing things from it. Then, you can use those theories to search for more information[Francis 2019]. It should be noted that information games constitute a very niche type of game, and as such are not scientifically documented, and other terms regarding educational entertainment (edutainment) could also be used to describe information games, but in the context of this paper we will refer to them in the manner that Tom Francis describes them above. 'Mind Diver' is highly influenced by three information games, whom constitutes examples of proven work: *Her Story*, *Outer Wilds*, and *Return of the Obra Dinn*. In spite of the fact that these three video games do not directly market themselves as 'information games', Tom Francis takes a look at them and explains why each of them could be considered such[Francis 2019].

2.2.1 Her Story

Her story is an interactive film/video game in which players watch live-action interrogation video footage from a police database. The suspect being interviewed is the wife of the missing person, her husband. The game is about finding inconsistency in her story based on seven different interviews. The gameplay is simply typing in words in a search database which then will return video clips from these interviews based on the word you searched for[Barlow 2015]. For example, you watch a video clip, the suspect mentions some specific location, you then search for that location in the database and you then get the amount of video clips that relate to that location. The information you are then gaining, is giving you new leads, hence more information. Finally, the game ends when you, the player feel like you have come to a conclusion to the case. Figure 5 illustrates how 'Her story' looks like, Figure 5a shows how the police database looks like in-game. Figure 5b is simply an example of one of the live-action video clips of the suspect with the only written information being the time and date of the video clip.

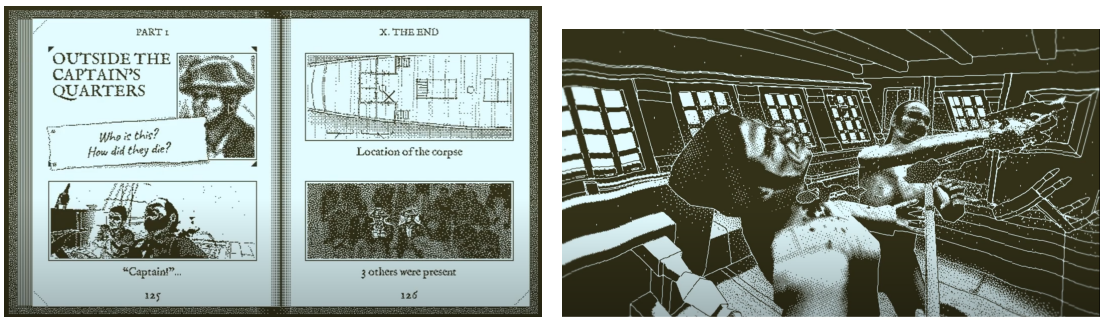
(a) Police database example

(b) Video clip example

Figure 5: Her Story - Videogame[Barlow 2015]

2.2.2 Return of the Obra Dinn

Return of the Obra Dinn[Pope 2018] is an adventure puzzle video game in which players have to report the fate of everyone who was on board on a ship that had a tragic end. The player has a journal that contains of 10 chapters, each chapter being a flashback from a certain time and space on the ship. The goal is to then fill out a short sentence describing how a person died or disappeared, including what and who killed that person, see Figure 6b for an example of how that looks like. Visually, the flashbacks starts out by only hearing the audio from the flashback which then transitions over to a frozen in-time still image in 3D of the final moments of the flashback, allowing the player to move freely around and look closely to what happened. The game features over 50 characters which the player has to solve the faith on. To help the player on their way, a journal contains information that can help the player as-well as the whole list of people that was on the ship, and their function on the ship, see Figure 6a. In terms of the faiths, a long list of ways a person could have been killed or fled to is also on the list, meaning that you can theoretically guess your way to success. However, the game does make it almost impossible to guess your way through due to the huge amount of different possibilities it could be. Furthermore, to give the player some sense of progression and reward for their effort, the game will verify if you have been successful with a character's complete faith, but only when you have three different faiths correct, giving a reward to the player but still making it unlikely to guess your way through.



(a) Journal example

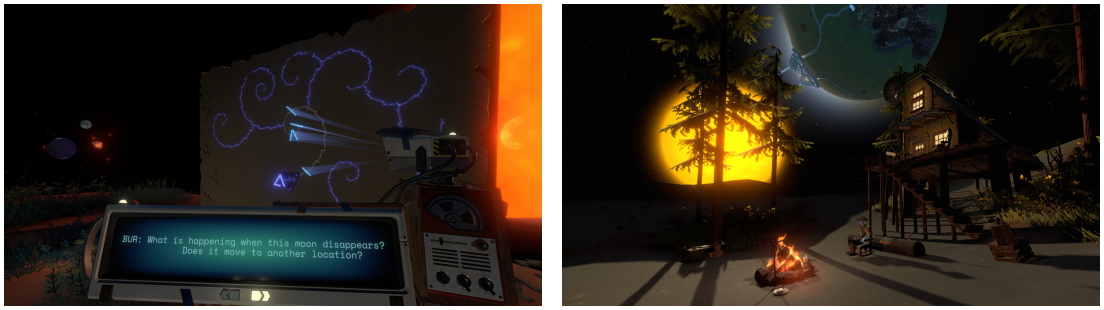
(b) Still image example

Figure 6: Return of the Obra Dinn - Videogame[Pope 2018]

2.2.3 Outer Wilds

Finally, Outer Wilds is an exploration mystery video-game[Annapurna-Interactive 2019], in which the goal is to figure out what has happened to a fictional solar system. The goal is to stop the end of this particular solar system, the most unique part of this game is that you are playing the same exact 22 minutes each time, meaning every 22 minutes the solar system is destroyed and you die. You, the player then wake up at the same point that you started and the only progression that carries over is the information you learned from your previous "life". The player experience is therefore very much about gaining some knowledge by exploring a particular planet, then in your next "life" you use your 22 minutes on exploring other parts of the solar system, other planets or other parts of the same planet. In this game loop, the player gradually builds a connection to why the solar system is collapsing, using one of the "lives" to complete the game within the 22 minutes you have, interacting with the environment in the correct order and following their logical reasoning to save the solar system and end the game loop. Figure 7 exemplify the first person perspective on a planet, Figure 7b and how you

explore the world to gather information about the story, Figure 7a.



(a) Written information example (b) A Planet example

Figure 7: Outer Wilds - Videogame[Annapurna-Interactive 2019]

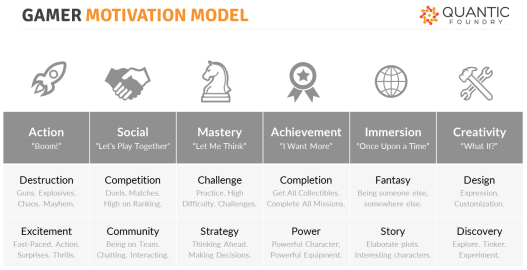
When looking at all three games from the perspective of being an information game, *Her Story* is arguably closest to being purely an information game. We believe this is mainly due to how the game does not give much of an *extrinsic motivation* to play. Extrinsic motivation is being motivated to do something based on external causes, such as rewards[Course-Hero 2018]. What we mean by this is that *Her Story* does not verify any information or tell the player in any way shape or form that they have successfully understood or completed the game. We therefore argue that *Her Story* is all about the intrinsic motivation, meaning doing a task for the sake of doing it[Course-Hero 2018]. We are well aware that video games in general is based on intrinsic motivation, as you play the game for the sake of playing it. However, most games do give virtual rewards for completing challenges in a video game which is an incentive to do something in a certain way. In Her Story, a player has to gain information for the sake of gaining even more information. The only interference the game system has to the player is when the player has watched half of the clips, the game will ask a simple yes or no question to the player if they believe they have acquired enough information. If yes, the game ends with a final video clip, if no, the game continues and the player can keep on interpreting the story until the player them-self believe they are done. In contrast to both *Return of the Obra Dinn* and *Outer Wilds*, both game has a specific ending and a specific way to complete it, as well as verifying, hence rewarding the player when they have done something correct. In Return of the Obra Dinn we could essentially call a feature in the game a verification mechanic, in that you get verified if the faith you have puzzled together is correct or not, and to successfully complete the game you need a certain amount of faiths correct. In Outer Wilds it is a bit more indirect. Here it is more about exploration and finding information, all the key information you find will be stored for you, and the game will automatically link certain information pieces that the player has found throughout their play-through. In this case, the game provides the player with a reminder of all the information and helps the player organize it, but the player them self still needs to interpret the information their own to then complete the game successfully.

All in all, *Her Story* is purely based on intrinsic motivation, while, *Return of the Obra Dinn* and *Outer Wilds* have extrinsic motivations linked to them. However, the core of all three games are based on intrinsic motivation to gain information for the sake of gaining more information, which is what they have in common and is the core reason to why they are all information games in their own right.
For 'Mind Diver', the video game we are co-creating. Just like in the three games mentioned,

the narrative is at focus and to give the player a reason for being intrinsically motivated we need a narrative that the player wants to explore. Since the narrative of the game is established, our contribution to the game will be to implement within the game design, evaluate it and iterate further. To fulfill this, we need to validate our implementation by getting actionable data. For our game, evaluating the narrative understanding is one of the most important factors we need to get right. Since, if the player does not understand the narrative, the player will not be able to proceed further in the game. We therefore need to evaluate players comprehension of the game's story. First, however, since different types of people like different types of games and stories, we will first get an overview of "Player Types" a research topic that is of huge relevance from an academic target group perspective as-well as designing a video-game. As with everything, we design something for specific people and with more different types of video games there are, the more types of players will emerge.

2.3 Player Types

Researchers have been studying patterns of player type since at least 1980, when Glenn Blacow[Blacow 1980] examined the behaviors of players in Tabletop Role Playing Games (TRPGs) that were being developed at the time. There are four categories that he identifies here that both players and dungeon masters (one person who controls the game world) will fall under. Further, in 1996 Richard Bartle[Bartle 1996] studied the player behaviors in Multi User Dungeons and came up with four groupings, which have some similarities to the schools of Blacow[Blacow 1980]. Later, in 2002, Robin Laws[Laws 2002], who also had a focus on TRPGs, was able to identify seven types in his book: "Robin's Laws Of Good Game Mastering". Older proposals of player types were primarily derived from personal experience and observations of forum interactions. Nick Yee[Yee 2005] published the results of a study in 2005, based on Bartle's model, in which he identified three limitations of. Results from the study were based on a combination of qualitative and quantitative data, gathered from players of Massive Multiplayer Online Role-Playing Games, who were willing to participate in surveys, as part of his 'Daedalus Project,' which ended with 3200 respondents. As part of his research into player types in 2015, he established Quantic Foundry[Yee and Ducheneaut 2019a] with Nicolas Ducheneaut. Quantic Foundry has an extensive dataset of over 850 thousand responses to its questionnaire, which is continually growing as new responses from all over the world are given, and Quantic Foundry continues to provide yearly reports on the findings. The early work of Quantic Foundry led to the identification of twelve player motivators, which enabled a better descriptor of player behavior, focusing on the motivation behind the behavior rather than a categorization based on the behavior, and these twelve motivators can be seen on Figure 8a.



(a) Figure showing the twelve motivations Quantic Foundry identified and their categories[Yee and Ducheneaut 2019b].

(b) Figure showing the three clusters[Yee and Ducheneaut 2020]

It is important to understand that these motivators are continuous in structure, which means that a low score of a motivator is equally valuable as a high score, in terms of indicating the person's preferences. Figure 8a shows how the motivators were able to be categorized into six pairs, each within a particular category. Additionally, these categories allowed for the definition of three clusters; 'Action-Social', 'Mastery-Achievement', and 'Immersion-Creativity', which can be seen in Figure 8b.

It is important to consider both the genre and type of player we are designing for when we are constructing a narrative for our game. We need our players to understand the narrative as intended to complete our prototype, as the goal of the game is to build the narrative together as we as designers intended. Accordingly, the following sections will cover a broad notion of narratives in digital media in order to incorporate these topics into the design of the prototype as-well as designing our in-game evaluation questions.

2.4 Narratology

Throughout history, narratives have been an integral part of constructing individual and collective meanings. As a result, narrative communication has been studied since the classical age. Considering its age, "Narratology" is relatively new to the academic scene. Nonetheless, the term is applied retrospectively to works that can be traced back to Aristotle being defined as a series of connected events[Ensslin 2015][Toolan 2006].
We can understand narratives as acts of communication between a sender and a receiver which is common in fiction and documentary. [Herman 2013], sets stories as a target of interpretation versus stories as a resource for sense-making.
There are thousands of ways to define a narrative, one in which describes it as perceived sequence of nonrandomly connected events[Toolan 2006].

There are a few basic terms you will hear when discussing narratives, such as *story* also known as *fabula*. A story/fabula is the fundamental events of a narrative in its natural chronological order[Toolan 2006]. A similar term is *discourse*, also known as *sjuzhet*, which describes how the sender presents the real cinematic experience to the receiver[Toolan 2006]. Thus, while a story is a complete narrative constructed chronologically, a narrative is what we tell and discourse is how we tell it, with the purpose that the receiver will understand the story as a whole.
A video game in itself is not a story, however, you can use video games to convey a story. The beauty and risk when designing narratives for video games is the interactivity that comes

with it. In film, conveying the story into a narrative is from a receiver perspective a passive experience. The receiver is passively participating in the authors discourse. In video games, however, the interactive element inherently ties the receiver into the narrative and by that becomes an active participant in the narrative, giving the receiver (the player) a role in the story's discourse, illustrating that stories are not media-specific, but rather that the medium itself is part of the discourse.[Larsen and Schoenau-Fog 2016].
This leads to complexity of how interactive narrative are being evaluated since the receiver (the player) can theoretically change the discourse entirely. One study proposes a framework, Author-Audience-Distance (AAD), being a function of narrative intelligibility. We will get back to AAD, as we first have to establish what narrative intelligibility is and how it relates to narrative closure.

2.4.1 Narrative closure and intelligibility

[Bruni and Baceviciute 2013] establish narratives not only as a communication act but also a human cognitive mode for organizing experience. Narrative intelligibility is the understanding of a narrative very closely to how the sender intended it, while narrative closure is the experience of coherence and completeness of understanding after having experienced a narrative, which works both with high level of intelligibility or not, since the receiver can find closure(coherence and completeness) in a narrative without it being the narrative understanding that was intended[Bruni and Baceviciute 2013].

A successful narrative communication would ideally entail both narrative intelligibility and closure. However this ideal may vary according to the intentions of the author-designer[Bruni and Baceviciute 2013]

[Bruni and Baceviciute 2013] goes over the relationship between narrative intelligibility and closure and how it would theoretically work if one was without another. Narrative intelligibility without narrative closure is definitely not ideal, but theoretically can happen in the case of the sender's intended for a specific narrative substance, but the construction of the narrative lacked coherence and completeness and is then faithfully transmitted to the receiver who then experience the original lack of closure[Bruni and Baceviciute 2013]. As the authors point out, this is a very theoretical argument since we believe that in practice it is very difficult not to have narrative closure but still have a high degree of narrative intelligibility since it means you have understood its intentions, but to understand a narrative in one specific way, we believe at-least coherence is needed since it illustrates how coherent a story is. In this case, the story should not make sense, but in a specific intended way, which we can ask - Does that not mean that it does in fact make sense?
We will therefore acknowledge the notion of this type of relationship in a theoretical sense but, practically at-least not replicable.
On the other hand, the relationship of narrative closure without narrative intelligibility is extremely likely and a narrative can definitely be intended to be such an experience. This would be when the receiver experiences coherence and completeness to a narrative without necessarily understanding the narrative that was intended by the sender, which can be designed for when the sender particularly does not intend to communicate a particular narrative substance but is in fact interested in providing an experience with narrative closure.
Narrative intelligibility and narrative closure are both relevant for our design, we will go deeper into narrative intelligibility in the form of the author-audience distance, see Author-Audience Distance 2.5.1.1.

2.5 Measurement Methods

As mentioned in Motivation 1.1, we set out to design and develop a continuous in-game tool to evaluate narrative intelligibility as well as in general evaluating the narrative engagement for our prototype. We are motivated by developing a tool that can both see relevance in academia but also for the game industry itself. Therefore, the following sections will analyze academic and industry approaches for us to develop a hybrid method incorporating strengths from both areas.

2.5.1 Academic approaches

Starting off with academic approaches we will detail measurements within relevance to our topic. We have previously shortly mentioned the Author-Audience Distance Framework, but we will now put forward the details, this is followed by measurements within engagement, narrative engagement and intrusiveness all to gather additional data of our prototype and tool we are developing.

2.5.1.1 Author-Audience Distance

[Bruni and Baceviciute 2013] proposes a framework for narrative evaluation. The Author-Audience-Distance (AAD) is essentially a high level illustration of the relationship between two opposite within narration, abstract and didascalic, and the author and audience of the narrative, see Figure 9. Abstract narration being the notion of the narrative being up to different interpretation and didascalic being that there is a limited amount or one specific interpretation to the narrative, which we believe also could be referred to as being a descriptive narrative, hence giving a very detailed account of an experience. The authors go on explaining that the more abstract a narrative is, the greater distance there is between the audience and the author, and vice-versa, the more didascalic/descriptive the narrative is, the shorter the distance is between the author and the audience. 'AAD' is essentially a visualization of narrative intelligibility, since intelligibility is all about the understanding of a narrative between the author and the audience. However, the framework is not directly related to any other narrative related elements, such as closure, since narrative closure can be reached in both an abstract or didascalic narrative. With that said, in order to obtain any form of intelligibility the audience would arguably need closure since that is all about getting a meaning out of a narrative.

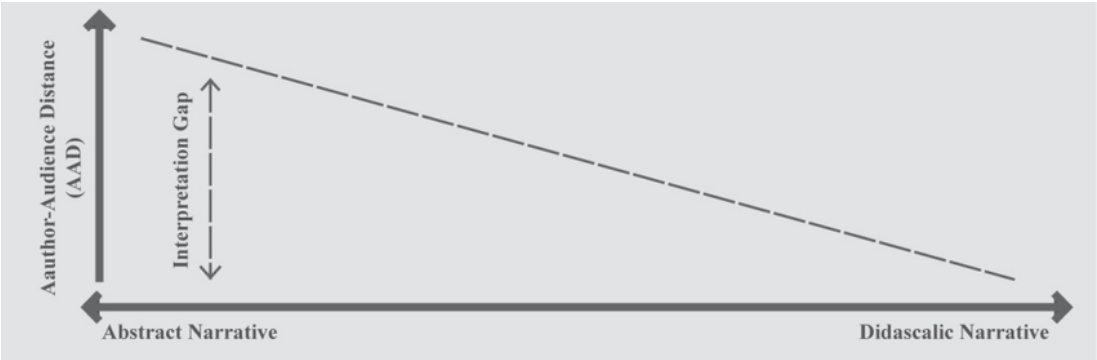


Figure 9: Author-Audience Distance[Bruni and Baceviciute 2013]

2.5.1.2 Measuring Narrative Engagement

Engagement, flow theory, and immersion are all important concepts when evaluating video games. Each one has its own unique perspective on how people interact with their environment.

Generally speaking, engagement is defined as the extent to which an individual is interested and involved in a particular activity[Abbasi et al. 2017][Przybylski et al. 2010]. In flow theory, people are most content when they are fully engaged in an activity and are able to lose sense of time and self-consciousness[Csikszentmihalyi 2014]. The concept of immersion involves a complete sense of absorption in an activity[Ryan et al. 2001].

Each of these concepts all relates to a game being fun and engagement itself is arguably the more general concept of fun. By understanding the engagement within the video game we create, we can more effectively create experiences that are enjoyable and fulfilling.

[Busselle and Bilandzic 2009] proposes a Narrative Engagement Scale, which is developed and validated through exploratory and confirmatory factor analysis with data from viewers of film and television. It seeks to explore four dimensions of experiential engagement in narratives.

- Narrative Understanding
- Narrative Presence
- Attentional Focus
- Emotional Engagement

The Likert scale consists of 12-items, 3 items per dimensions. it is originally developed for television and film, but other studies have already used it for video games, such as [Nielsen and Rafferty 2020][Christy and Fox 2016]. In addition, [Nielsen and Rafferty 2020] argues that the measurement of narrative engagement would hold true across mediums and that video games may actually be more engaging, as is shown in comparison to audiobooks and film[Richardson et al. 2018].

Narrative engagement is an interesting area to see whether the in-game feedback tool we want to create actually affects that, but we do believe that some form of general engagement would also be necessary to triangulate the data with. [Brockmyer et al. 2009] has developed a Game Engagement Questionnaire (GEQ), this questionnaire could prove to be a good alternative for the narrative engagement questionnaire. However, we do prioritize narrative engagement for this particular study and we do suspect for the game we are creating you will not find engagement in the game itself without any narrative engagement since the gameplay and narrative are inherently linked in our game, aswell as not wanting to exhaust the participants with too many post-questions leading to other potentially negativity towards the testing as a whole. With that said, what we see more fitting for the purpose of our continuous evaluation is the so-called continuation desire[Schoenau-Fog 2011] as 'GEQ' is not intended to investigate the triggers of engagement or the willingness to continue in-game[Schoenau-Fog 2011], which the willingness to continue playing is very relevant for our case, since we will continuously interrupt the player with our in-game questions. In [Schoenau-Fog 2011] an Engagement Sample Questionnaire (ESQ) was developed for their case study. This questionnaire contains a mixture between self written qualitative answers and Likert items regarding engagement. Most relevant for our study is the questions during the experience. These questions focus on the area of whether a person still wants to play the experience and the reasoning for why the participant might or might not

want to continue. We see potential in adapting some of these *during the experience* questions for our in-game questionnaire as we can seemingly correlate the notion of understanding the intended narrative and whether the participants wants to continue playing.

2.5.1.3 The Narrative Experience Measurement Tool

NEM (Narrative Experience Measurement) combines a combination of methods to assess player engagement, internal coherency, and external consistency within a narrative game. This tool is designed for measuring narratives within procedurally generated embedded narrative videogames[Nielsen and Rafferty 2020]. In this section, we will describe what the tool does and evaluate the design of such frameworks in terms of our proposed in-game feedback system.

For narrative engagement the authors coin the definition as *"a measure of how each individual player was engaged in the narrative"*, here they pick certain Likert items from [Busselle and Bilandzic 2009] for evaluating narrative engagement in their players.

Internal Coherency is defined as *a measure of how much each individual player perceived the narrative as being coherent and making sense.*, here the authors use a specific part of [Busselle and Bilandzic 2009] narrative engagement questionnaire that points at narrative coherency. Finally, external consistency is *A measure of the extent to which multiple players experience and understand the narrative in the same way.* In order to do so, [Bruni and Baceviciute 2013] is combined with [Larsen, Bruni, et al. 2019] definitions of an /afterstory], allowing the player to choose from several predefined answers to the story or to write down their own interpretations to determine if there is narrative comprehension within the player themselves rather than narrative comprehension aimed at the author's intention, hence narrative intelligibility.

2.5.1.4 Measuring Intrusiveness

To gain some insight into whether repeated interruptions in-game will be annoying, we plan to assess the intrusiveness of our developed in-game questionnaire. Because an intrusive system interferes with gameplay, it is important to consider how it makes the participant feel, since a system that feels unwelcome and something the participant does not like could negatively affect the overall experience.
The medium of commercials has been extensively studied when it comes to perceived intrusion[Li et al. 2002][Wehmeyer 2007][Ying et al. 2009]. According to [Bauer and Greyser 1968], intrusive advertising is one of the top reasons for annoyance with advertising. We do not want the in-game questionnaire to be annoying to answer, and as intrusive advertising can be annoying, we would assume if ours is intrusive, it must also be annoying. In addition, by using measures of advertisement intrusion, we can really determine whether or not our participants find it annoying. [Li et al. 2002] developed and validated a scale that measures the degree to which ads are perceived as intrusive. The scale consists of seven items that all ask the same question: *When the ad was shown, I thought it was...*, followed by words referring to intrusiveness[Li et al. 2002], all statements ranging from 1-7 from strongly disagreeing to strongly agreeing. In changing 'ad' to 'game', we hope to get a better perspective on how intrusive our in-game questionnaire might be.

2.5.2 Industry approaches

The previous section detailed academic measurements within the topic of narratives and intrusion. Below we describe two examples of feedback tools that have been implemented in the video game industry. Although these tools are mostly general, it is their design and implementation within the context of evaluating video games that are of interest. These cases represent continuous in-game evaluation systems, which are similar to what we intend to implement. In the first scenario, there is the feedback system for the video-game Subnautica[UnknownWorlds 2014], which is a solution that is specialized to Subnautica and integrated into the backend of the video-game itself. The second system is STOMT[STOMT n.d.], which is a commercial product available through their website. Unlike the Subnautica system, this one is much more generalized with plugins available for game engines such as Unity and Unreal Engine, allowing a button to display a user-interface (UI) when clicked. A short form of text feedback may be submitted, and this feedback will be uploaded to the STOMT website.

2.5.2.1 Subnautica's Feedback Tool

The Subnautica feedback system can be operated by the user without difficulty, by pressing a button the game pauses, goes greyscale, and a UI component appears on the left side. The UI piece allows users to enter a text message of 190 characters, select whether the feedback is general feedback or a bug, and whether a screenshot should also be included. Lastly, an option to choose from four smileys indicating whether the user likes or dislikes the situation can be found on Figure 10.

Figure 10: Figure showing the feedback system in Subnautica[Jeremy 2015]

As stated in the earlier section Industry approaches 2.5.2, the feedback system hooks into the backend, and the public presentation of this data can be viewed on their website[UnkownWorlds 2014]. It is noteworthy that given the integrated nature of this system, despite the fact that it appears quite simple from the earlier Figure 10, it actually sends much more information, including a player's position in the game world, as well as some information about the system running the game, such as the operating system, GPU, and CPU. Due to the fact that the player position is noted, a map overlay can be created as presented in the GDC talk[Jeremy and GDC 2015]. In this example, we can see that clustering is present outside of the center, where one begins the game, as seen on Figure 11.

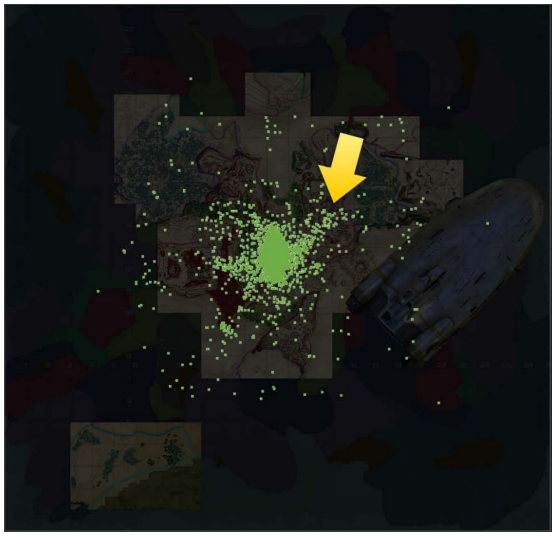


Figure 11: Figure showing Subnautica feedback player positions[Jeremy 2015]

This method is effective because it is non-intrusive and enables players to determine whether they wish to provide feedback as the UI piece only appears when a player presses a button. However, this system requires the setup of a server and programming of a backend system in order to receive data.
Subnautica has enjoyed tremendous success both in terms of sales and positive reviews on Steam, demonstrating that the game is of high quality. From the developer's own words, they credit the success of the game to the feedback they received through this tool. This illustrates just how powerful feedback can be, especially when it is accompanied by additional data from within the game.

2.5.2.2 STOMT

STOMT is a commercial in-game feedback solution, which has a free version that allows for the use of the website and integrations, but does not provide analytics or labeling capabilities. Labeling and filters are available for a monthly subscription fee of 14.99€ per active seat, and advanced analytics and a management dashboard are available for a fee of 69.99€. STOMT can be utilized in multiple ways, it can be accessed directly from the website without an additional application. As long as the users receives a link to one's webpage they will be able to provide feedback in a small text box as shown on Figure 12.

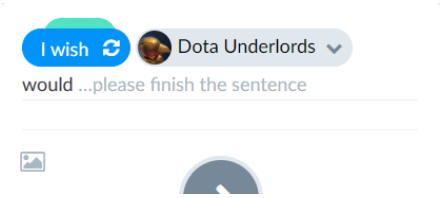


Figure 12: Figure showing the STOMT feedback textbox[STOMT n.d.]

The post may be accompanied by an image. It is also possible to integrate STOMT with a wide

variety of game engines, applications, and popular websites, including Unreal Engine, Unity, Discord, Trello, and Twitch. Through the use of this potential for integration, and the API developed by STOMT, it is able to provide a more specialized feedback, incorporating other pieces of data in addition to just text. This can be accomplished by using the labeling system. For example, a feedback post can be labeled with a versioning number, the screen resolution, or the operating system the user is using. For the user, the integration of STOMT in Unity for example could involve a small button that the user can click to open up the small feedback text box as seen on Figure 13.

Figure 13: Figure showing the in game STOMT button in All Walls Must Fall[Zentner 2018]

As stated on the STOMT website for Unity[Schukies 2015], it is possible to attach data outside of the labeling system by adding key-value pairs to the API gameobject. It is not clear how this data is to be viewed on the website, but it implies that it could be possible not to upload a file, but to at least send the contents of the file, if it is a text file.
Given how limited the free version of STOMT is, compared to the paid options, and given how long STOMT has been operating, it would be fair to conclude that STOMT shows that there is a market for in game feedback tools.

Ultimately, we believe that by combining both scientific and industrial approaches, a methodological approach may be developed that combines the best of both perspectives. Academic methods will be applied to ensure that an unbiased approach is taken in conjunction with industry methods that are more focused on gathering data efficiently and enhancing the user experience as a whole.

2.6 Analysis Conclusion

Overall, photogrammetry is a versatile tool that can be used to create 3D models from images using relatively low-quality equipment. Additionally, it is relatively easy to learn and use, making it a good option for those who wish to generate 3D models without investing a great deal of money in specific tools or software. With that said, for Mind Diver, we are not attempting photo-realism, but rather are embracing the photogrammetric errors. Therefore we might have the ability to utilize Polycam on the iPad pro 12 which affords the versatility to check the results of a scan on site with cloud processing instead of using a less versatile and expensive software such as Metashape.

To understand the focus points of what makes these games work, we examined similar games for Mind Diver, and with the three games Her Story, Return of the Obra Dinn and Outer Wilds, it becomes evident that the narrative of the game is an important factor in the player's engagement. Moreover, this is also where intrinsic motivation comes into play, since it is the driving force behind the user's desire to progress in the game and learn more about the narrative and world of the game.

With respect to the motivational forces required for engagement with the game. The study of player types and motivations facilitated a better understanding of the users we wished to focus on. Using the player motivators defined by Quantic Foundry, we were able to identify which of these motivators makes the most sense for users who could find Mind Diver an engaging experience.

We have mentioned that the narrative is important; therefore, we have investigated narratology, so we can understand what the difference between a story and a narrative is. Narratology also allows us to understand various ways of conveying a narrative. This is important for video games, as they are able to convey content in multiple ways. For example, the user could be a spectator to the story or an integral part of it and actively interact with it.

Considering the notion of a user participating actively in a narrative, we also looked at narrative intelligibility and closure, since these aspects enable us to understand the relationship between the user and the way a narrative might be presented. It is therefore important to consider the relationship between narrative intelligibility and closure when designing narratives. Although it is difficult to have one without the other, it is possible to create a narrative that is intelligible without being coherent or complete, or a narrative that is coherent and complete without being intelligible. In general, each type of narrative has its own purpose and can be used to create different types of experiences. In order for us to ensure that the user experiences the narrative the way we intend, narrative intelligibility is an important aspect to consider, especially since Mind Diver is an information game that incorporates narrative into the gameplay.

In order to measure these, we researched academic approaches to the measurement and industry approaches to provide feedback from an in-game environment. The combination of these approaches forms the basis of the in-game feedback tool that we would like to develop.

2.7 Research question

The research question is the following:

Will using our implemented in-game evaluation tool provide usable player insights within narrative intelligibility, without being intrusive and negatively impacting narrative engagement?

To answer our research question we will be doing an exploratory analytical design using our own methodology to evaluate continuous narrative intelligibility, while using established methodologies within narrative engagement and intrusiveness.

Furthermore, since we are using photogrammetry in an innovative way for games and photogrammetry is still an underused method to create 3D assets, we will provide new perspectives on photogrammetry's use in video games by detailing the design and technical challenges associated with developing for photogrammetry, in conjunction with the research question. Furthermore, we will want to formulate a sub-research question that focuses on the use of photogrammetry in our video game and ponders whether or not it has provided value to the video game from both a production perspective and from an artistic standpoint. Therefore, the sub-research question is as followed.

What impact does photogrammetry have on the narrative experience both from a production and an artistic perspective?

3 Methods

In our project, we will assess whether the in-game questionnaire provides practical insights into continuous narrative intelligibility and whether the system's intrusiveness could affect a player's narrative engagement. Additionally, we are working on a video game using photogrammetry to produce 3D assets and also want to explore if this technology was beneficial.

Our work is using a convergent parallel mixed method that will result in both qualitative and quantitative data[Bjørner 2015]. We first need to evaluate the full gameplay and narrative experience, which will be refined over many iterations. The game will then be evaluated to determine whether it achieves its goals.

In order to guarantee the highest level of quality within the time-frame of the project, an iterative process is necessary from design through implementation and evaluation. Study of topics beneficial to our design and development, such as informational, narrative and puzzle video games. In addition, evaluation tools and frameworks within the field of narrative intelligibility, closure, and engagement. We will continuously iterate on our user-centered design approach to get valuable feedback on how to proceed with adjusting the gameplay and narrative, as well as constantly iterating on the in-game questionnaire itself, both from a design and technical perspective, as well as comparing the questions we ask to determine whether or not we can convert our findings into actionable data.
Since we are developing a video-game that is intended as being distributed as entertainment and will have an average gameplay time around 2-3 hours, we will shorten the experience to specific sections to adapt to the research we are doing. For our research we will be using our developed continuous in-game questionnaire with self formulated questions based on narrative intelligibility, while using parts of the narrative engagement questionnaire[Busselle and Bilandzic 2009] to quantify the narrative experience compared with intrusiveness to the system itself. The following will detail all the methodical steps.

3.1 Sampling & Target Audience

The target audience for this study is people who are at least 18 years old, have some experience playing video games, and are comfortable using a mouse and keyboard to control a game. We are not looking for any specific type of player, but we are open to considering any player motivation that might come up.

- At least 18 years old
- At least a Casual Gamer (play video games, but in short sessions or infrequently)
- Experience with the mouse and keyboard computer game control scheme ('WASD' for movement and the mouse for direction)
- Experience playing first-person perspective video games.
- We do not require any specific player motivator but will account for it if needed.

All participants will be recruited online, using quota-, convenience- and snowball sampling[Bjørner 2015]. By using these methods and by targeting a broader audience within the target group, we can ensure a good range of participants while still maintaining some degree of homogeneity in terms of video game competence and knowledge.

3.2 Iterative Narrative Intelligibility and Gameplay Evaluation

Prior to conducting our main research we first need to design and implement the video game experience we want the participants to engage in. As mentioned in Motivation 1.1 the authors of this thesis joined the 'Mind Diver' video-game at an early greybox state of play with initial iterations on the game design. Our related task from 'Mind Diver' to our research is the presence of narrative intelligibility within the the game's design, as understanding the narrative is the key element for a satisfying experience with the gameplay.
This will be done through an iterative design approach going from paper prototypes to the playable video game being released. To evaluate the goals of narrative intelligibility within the game design we will be doing in-house user experience testing using methods as 'think out loud' and continuously asking specific qualitative questions regarding narrative intelligibility in the same way as we would want to implement the in-game questionnaire. This approach both provide insight to improve the inherent link between the gameplay and narrative, as well as iterating on the narrative intelligibility questions leading up to the final in-game questionnaire to ensure better actionable data for our research question.

3.3 Research design

The research question will be answered through interpretative analysis on both quantitative and qualitative data. Participants will all play the first couple of sections of the final video game. This is solely due to the time it would take for the participants to complete the test. The participants will first go through a Pre-Questionnaire (Pre-Q), this would contain questions regarding the demographics and video game experience. This is followed by them playing the video-game. First, they will play an introduction to the gameplay and story, which intend to introduce the opening of the story, but more importantly providing a tutorial to how to play the game. When the player has completed this first part of the game, the first in-game questions will prompt (In-Game-Q1) evaluating the players desire to continuing playing and their understanding of the story so far. This is to have a starting baseline to how much the player understands before diving into the actual challenge of the game. This is followed by the next section of the game where the actual evaluation of the players' continuous narrative intelligibility within the gameplay will take place. Each time the player makes a choice by trying to correct a memory in the game an in-game questionnaire will prompt (In-Game-Q2). This loop will continue until the player has corrected all memories which will prompt the last in-game questionnaire (In-Game-Q3). Finally, the player will quit the game and answer the Post-Questionnaire (Post-Q), see Figure 14 for a visual illustration of the testing process.

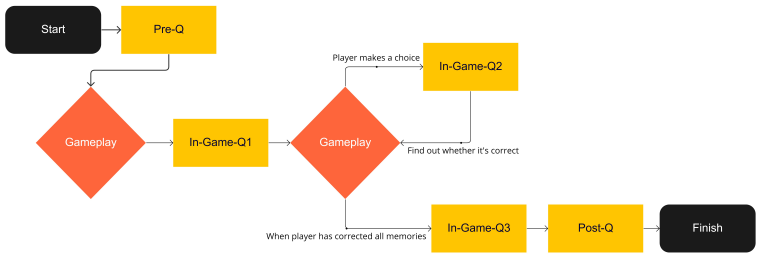


Figure 14: Flowchart of the research design

3.3.1 Pre-questionnaire (Pre-Q) / Demographics

The pre-questionnaire (Pre-Q) consisted of a series of questions related to general demographics regarding the participants (age, gender, gameplay experience). Additionally, quantifying the participants continuation desire in the form of a Likert item as-well as obtaining any information the participant might have on the video game prior to this test.

3.3.2 In-Game-Q1 & In-Game-Q3: Continuation Desire and Retelling

For the 'In-Game-Q1' and 'In-Game-Q3' we will prompt the same questions at two different points in the game. One qualitative question regarding narrative intelligibility, inspired by explicitly retelling a narrative as in [Larsen, Bruni, et al. 2019] and [Nielsen and Rafferty 2020] and adapt one Likert item from [Schoenau-Fog 2011], that will be used to correlate with their initial continuation desire with their continuous in-game continuation desire, as-well as identifying any relation between the understanding of the narrative with their willingness to continue playing the game. We do not see it necessary to utilize the the complete continuation desire questionnaire as its purpose is only to supply additional data. All the answers from the questionnaire will be directly transferred to a google form, see Google Form Request 5.2 for details.

3.3.3 In-Game-Q2: Continuous evaluation of Narrative Intelligibility

For the 'In-Game-Q2' we will continuously prompt the same questions while the player tries to solve the story of the game. One item and two qualitative questions that should provide insight into a video-games narrative intelligibility.

3.3.4 Post-questionnaire (Post-Q)

The post-questionnaire (Post-Q) will consist of 5 sections

Narrative Engagement Questionnaire
[Busselle and Bilandzic 2009] 12-item narrative engagement scale will be used. As a result, we will be able to determine if the play-through of the game contained any narrative engagement, which we will use to determine if this is related to the intended narrative understanding and annoyance towards the in-game questionnaire, see Measuring Narrative Engagement 2.5.1.2 for more details.

Narrative Tension Questionnaire
This series of questions is based on a 6-point Likert scale regarding narrative tension. There is an assessment of progression, attention, and suspense associated with a particular narrative. Please note that this Likert Scale is derived from [Nielsen and Rafferty 2020] which uses it as a method to ensure narrative coherence. In this case, we see it tied to narrative closure, and as we argue in Narrative closure and intelligibility 2.4.1, evaluating narrative intelligibility requires some form of narrative closure. The scale, however, is not properly cited from [Nielsen and Rafferty 2020], and we are therefore unable to verify whether it is a valid scale. However, based on the results of [Nielsen and Rafferty 2020] evaluation, we know it provided valuable results and are therefore confident that we can use it in such a manner whilst examining each item individually as well.

Photogrammetry related questions
Our sub-research question will be answered by asking them a series of questions about how static photogrammetry scenes compliment or ruin the gameplay experience when conveying the narrative. Using self-made items allows us to quantify the experience while also providing qualitative answers to determine the reasoning behind their answers as well as determine if the questions are asking what we intended.

Intrusiveness towards the In-Game Questionnaire
We will adapt [Li et al. 2002] 7-item AD Intrusiveness Scale. This scale is intended for how intrusive ads are when unwillingly shown on screen. We argue that we can translate that into our in-game questionnaire as it will interrupt in the same way as an ad by being shown at a forced point in time. This will help us evaluate on whether or not our in-game questionnaire has been intrusive and potentially indicate whether or not that has influenced the narrative engagement and intelligibility of the participants. To triangulate with, we also provide self-made items and qualitative responses regarding their annoyance with the system.

Technical & Usability Evaluation
This will have a series of usability questions of the gameplay experience itself, as well as questions indicating if they encountered any technical errors. This information is crucial to locate any form of direct gameplay which could have negatively impacted the overall experience and locate some confounding variables with the testing in general.

3.4 Design Requirements

The first step will be to establish a list of requirements for the prototype before we proceed with the design. These requirements are both functional and non-functional requirements divided into three categories, 'Photogrammetry, 'Narrative and Gameplay', and 'In-Game Questionnaire', some of which are requirements from the Mind Diver project itself, while others are based on our test and research specifics. This will then serve as the basis for describing the specific needs for our three major design areas in the following section.

Photogrammetry Requirements

- The photogrammetry graphics should be readable in the sense of understanding what the players are actually looking at in the prototype.

Based on the 'Mind Diver' requirements, we need the player to visually read the photogrammetric 3D assets, as the player is required to interact with them to proceed in the game.

- The photogrammetry graphics should compliment the narrative design.

Based on the 'Mind Diver' requirements the photogrammetric assets should give the sense of being a part of the narrative through the broken memories.

Narrative and Gameplay Requirements

- The video game (including the in-game questionnaire) should on average take 40 minutes to complete.

Based on our evaluation, this length was chosen to ensure that we were able to get the player sufficiently invested in a narrative. This would enable us to accurately evaluate narrative intelligibility and continuation desire. That said, it should not be longer than specified to limit the possibility for participants to become fatigued or prematurely unwilling to commit so much of their time to a voluntary test.

- The puzzle challenges should be completeable by understanding the narrative as intended by the author.

In order to meet the Mind Diver requirements, we would expect puzzles in the game to be solved by understanding the narrative, which is also why we are interested in evaluating the video game's narrative intelligibility.

- The video game must be playable on a mid-range Windows desktop or laptop with mouse and keyboard.

Since photogrammetry contains a high number of vertices and there will not be much time to fix performance issues, we anticipate that an average computer of today's standard (2022) is required to play the game smoothly.

- The control scheme should be easy to learn and understand for the target audience.
- The game should be manageable to complete for the target audience.

In order to be able to evaluate the system, we need to understand the narrative, but the player also has to know how to play the game. This should be as simple as possible so that we can ensure that the level of game literacy required is as low as possible for our target audience.

In-Game Questionnaire Requirements

- The In-Game Questionnaire should automatically store player data online
- The In-Game Questionnaire should provide answers that can relate to narrative intelligibility.
- The In-Game Questionnaire should be easily readable and understandable.

Our goal is to collect data that is seamlessly sent to us online without us or participants having to do additional steps, as is the case with the video game Subnautica, see paragraph 2.5.2.1. We are evaluating continuous narrative intelligibility in accordance with our research question, so the questions we ask should provide answers that can be effectively analyzed to provide insights in this manner, and by that, be easily readable and understandable.

4 Design

Based on the design requirements, this section will discuss and display the iterative approach that has been done from when we joined 'Mind Diver' to where it ends off to answer our research question. With that said, we have developed a video game in conjunction with this thesis called Mind Diver on which we will base our case-study on to answer our research question. Mind Diver is a project we joined at an early alpha stage of the production, specifically at a playable proof-of-concept stage. Hence, to understand the scope of our contribution, we will first detail the state of the project before we joined it.

4.1 Initial design

Mind Diver is a game in which players swim through memories of a woman whose boyfriend has gone missing.
A player encounters what we call 'clusters', which are three memories within a time-span regarding the same overall memory, hence a sub-story. Since the general story itself had already been shaped before we joined, we will not go into great detail about it now. More importantly, we want to discuss the gameplay and how it relates to the narrative.
The player would come across a memory at this stage, the memory would be a 3D still image, accompanied by a voice recording from approximately 30 seconds before that moment occurred. When encountering a memory for the first time, the memory would intentionally provide false information, including what had been explicitly said between the characters and how the scene was explicitly constructed. Afterwards, the player must reconstruct the memories to their true form. In order to do this, the player moves between different false memories and literally picks up objects or characters from one and places them at the memory they believe it belongs to.
As a guide, the game would visually indicate what object in a memory was incorrect, and it would then be the players' challenge to figure out where the incorrect object should go. A simple example of the initial design was the main character, Lina, coming home and expecting her boyfriend to be there. After concluding he is not home, she picks up the 'toothbrush' and states, "Hmmm, he forgot his toothbrush.". At another place, a mobile phone is highlighted in red, indicating that it should not be there. It is simply a logical assumption that her boyfriend has not forgotten his toothbrush, but rather his phone. By picking up the phone and replacing it with the toothbrush, the player would succeed. This is a simple example, where the goal later on is to actually discover inconsistencies in the story instead of just simple logical errors.

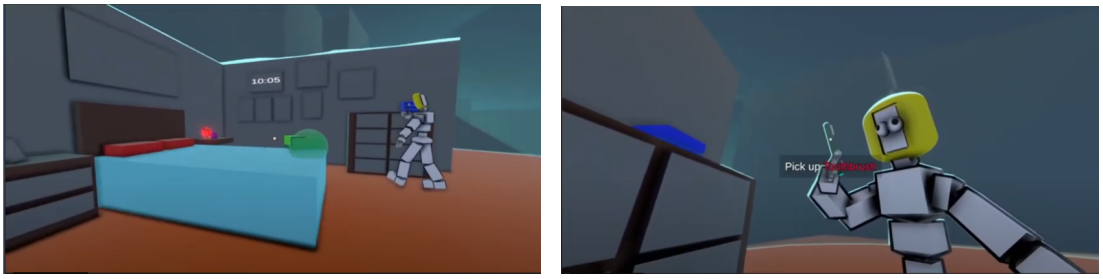


Figure 15: Mind Diver - Initial State

All in all, at this point in time 'Mind Diver' could be played through start to finish, though still missing a lot of features, design and audio-visual feedback to become a ship-able game and a video-game we would use for our research topic. The following list is our targeted contribution to the game, which we then will detail going forward, keeping in mind, other implementation such as the audio will have a huge impact on the games narrative expression, but is not something we will directly impact the game with.
In general, our contribution will consist of implementing and designing core functionalities to the game, taking part in the whole process within photogrammetry being the main artistic expression, as well as a complete new design approach to how the actual core gameplay will be. Specifically,

- Plan, record and implement photogrammetric 3D assets.
- Design and implement the the core challenges of the game
- Design and implement an intuitive tutorial for the game to make the game easily understandable
- Design and implement an in-game feedback tool to iteratively evaluate the game

These are the main areas we have been responsible for with other minor changes in collaboration with others. Since we are dealing with many areas that can seemingly be separated and all being relevant to both our main and sub research question we will divide the following into specific areas. We will at best try to keep the order of development chronological, but we will prioritize discussing one aspect at a time and cross reference when some areas overlap in decisions. First, we will detail the contribution of the actual gameplay and narrative to get a complete understanding to what kind of game our participants will be playing.

4.2 Narrative and Gameplay

We established the design prior to our contribution at Initial design 4.1. This section will highlight the most notable changes to the actual gameplay design of the game to understand the player's journey for our evaluation.

4.2.1 Summary of the game

To understand the context of what we exactly are creating both in terms of the narrative and gameplay the following is derived from the development teams design document giving a summary of the most important aspects of the narrative and gameplay:

The player is one of the Mind Divers, a group of researchers, who have developed technology to scan people's brains and dive into their minds to restore the memories. A young woman, Lina, approaches the Mind Divers to get help with finding her missing boyfriend, Sebastian. The player must now dive into her most intimate memories to search for clues that can reveal where Sebastian has disappeared to.

The Mind Diver is a first-person character in a 3D world inside a character's (Lina's) mind, represented as an ocean. Frozen scenes, depicting memories from the past, are scattered throughout the ocean, stored inside large bubbles of air. The 'Mind Diver' is deliberately non-personalized

to keep the focus on the characters of the memories.

Inside every memory is a 'memory hole', that the player must fix. They can pick up any object or person in any memory and try to place it where the hole is. The challenge of the game lies in figuring out the correct object, by carefully studying the scenes, and listening to the dialogue, in order to deduce what happened in the scene. When they place the correct object, more of the memory is restored, which gives new information that can be used to solve more memories.

Figure 16: Concept art for Mind Diver *made by our 2D artist on the production team

4.2.2 Gameplay

In the game, the player will generally be doing two different things, note that these design choices has been made in collaboration with previous designers and the director of the game:

- Traversal (walking when inside the memory bubbles and swimming in the ocean)
- Correcting memories

There are two states for traversal:
While inside the memories (memory bubbles), you walk, and when outside a memory, you swim.

A memory can be corrected by picking up any item or person in a scene and placing it in a 'memory hole.' If the memory correction is wrong, the player receives corresponding feedback, and nothing else changes.

In the case where the correction was true, an elaborate transition effect will be activated, changing the memory from the broken version to the newly restored version. Resulting in a new audio clip regarding the narrative.

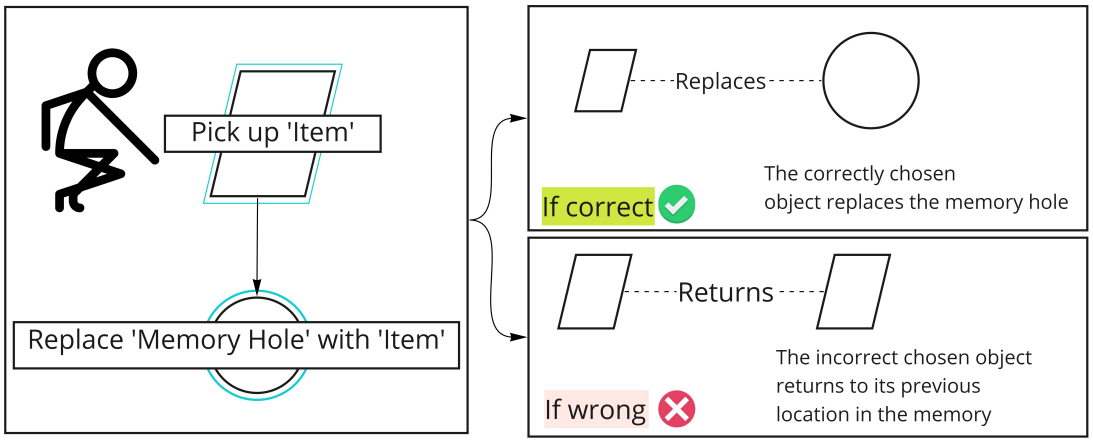


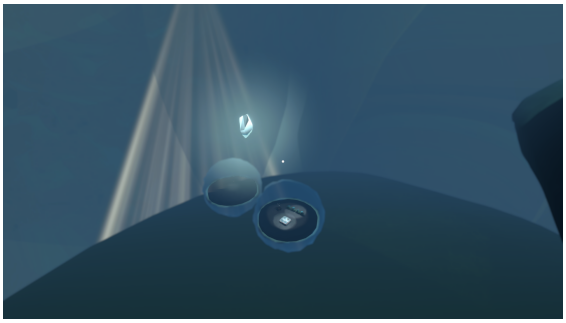
Figure 17: An illustration of correcting memories

For harder challenges, which we refer to as 'puzzles', we ensure that there are many objects that physically, in size and type of object, could be the correct object. This prevents the player to some degree of overly guessing their way to the correct answer and instead forces them to use deductive reasoning through understanding the narrative, similar to the video-game, 'Return of The Obra Dinn', see Return of the Obra Dinn 2.2.2.

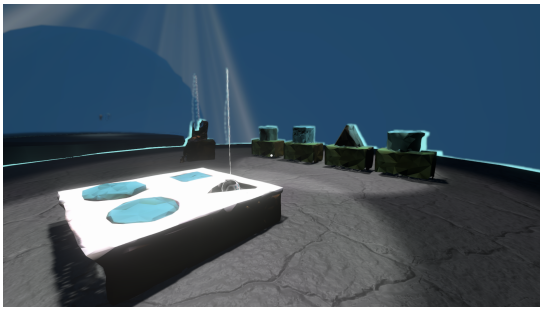
Our design is largely geared towards the principle of puzzle revelation equaling dramatic story revelation, which means that every puzzle must be solved through a realization. The aim is to have puzzle realizations depend on dramatic story developments with a strong emotional connection. Our in-game feedback tool for evaluating narrative intelligibility is based on this principle, as we can evaluate the reasoning behind a particular decision if we are able to understand what drove that player to take that particular decision in the first place. Our goal is to reach the point where a player's realization of the story makes them able to solve the puzzle and not by simply guessing the correct answer using purely logic alone.

4.2.3 Game Journey

We will now detail the player journey from start to finish to understand what the player will experience, specifically the shortened version for our evaluation.
The game starts with a short text sequence, explaining the player's task as a Mind Diver.
In the 'Mind Ocean', the player begins with an enclosed tutorial area - the 'Ice Cube' - where light exploration is taught to the player, along with how to correct memories and move around the world. Here, they solve their first puzzles, see Figure 18.



(a) Swimming to tutorial 'Memory Bubbles'



(b) First 'Memory Bubble' to learn how to play the game

(c) Another 'Memory Bubble' to learn how to play the game

Figure 18: Ice Cube: Where the player learn the basics of the game.

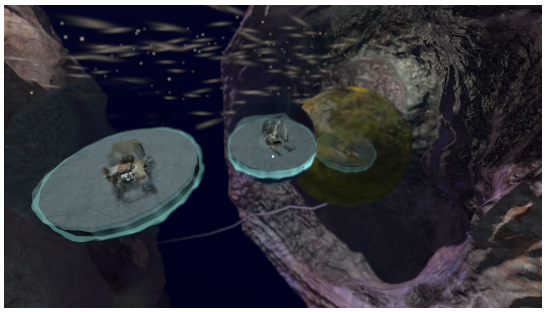
Upon completion of the tutorial (Ice Cube), the player is provided with a case file that represents the case of the memories of the main character, Lina, followed by swimming to the 'Central Memory'.
The 'Central Memory' is the main memory the player gets told they have to solve. There are multiple moments from the night of the party that the main character has asked the Mind Diver for help remembering. In our test, we only tested the first moment during the party. The player can then activate the memory, which produces a sound recording that captures that moment. However, some of the audio is muffled, see Figure 19.

(a) Swimming to the Central Memory

(b) Central Memory: First moment

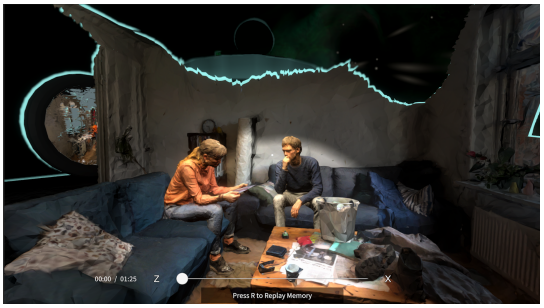
Figure 19: Central Memory: Where the player will get introduced to the narrative.

During this step, the player has to locate and place the correct object in the memory hole to reveal the entire memory, resulting in the entire audio recording.
The player is guided to a new area in which there are three separate moments from another memory, referred to as the 'Worry Memory'. Using information from these three moments, the player will have to fix each memory by finding inconsistencies in the arrangement and replacing items that have been arranged incorrectly, see Figure 20.

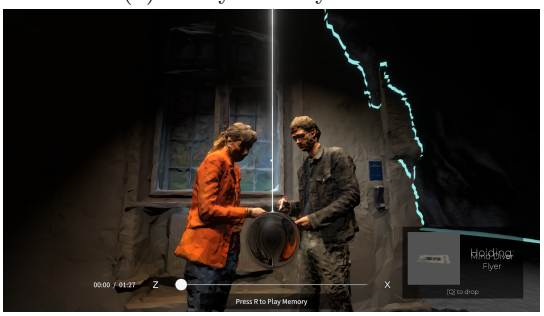


(a) Swimming to Worry Memory Bubbles

(b) Worry Memory Bubble 1



(c) Worry Memory Bubble 2



(d) Worry Memory Bubble 3

Figure 20: Worry Memory: Where the player will have to fix memories, also where we evaluate continuous narrative intelligibility

As part of our testing, we will end the game after fully completing the 'Worry Memory'. Throughout the game, an in-game feedback tool will be prompted at specific points. Before we describe that however, we will first detail the design and implementation of photogrammetry.

4.3 Photogrammetry

This section will detail our process from start to finish working with photogrammetry. We will start with the initial plan before going to locations to get the needed photos to translate it into 3D. Next, we will detail some examples with how it went, finally going over our process in importing the raw 3D files to a 3D software to clean them up to the implementation in Unity.

4.3.1 A Plan for Photogrammetry

In our case, how you play the game and how you experience the narrative are all dependent on the photogrammetric 3D assets we produce. In the context of gameplay design, we would need to know exactly how the game will function before it is implemented and, based on that information, decide what can be changed and what cannot be changed when the photogrammetric 3D assets are available. Due to limited resources and time we had, we would only have one opportunity to book locations and actors without the possibility of re-doing certain parts later if we suddenly decided to change how the scene would appear. With five locations throughout Copenhagen, Denmark, three main actors and a dozen extras, it was essential that everything was planned out correctly and we knew how to prioritize each case appropriately.

Keep in mind our goal was to setup complete locations authentically rather than create individual objects and place them together in a virtual scene, which means most of the time we would need to do a lot of preparation in order to set up the scene.

Prior to visiting any location, the following steps must be taken:

- An illustration of the most important aspects of how the scene and gameplay relates to each other.
- A list of all objects and actors needed for the scene.
- An overview of what we would need to prepare prior to going on set and what we would naturally get on location.
- An overview of what we would need to do in post-edit.
- iPad Pro fully charged to use photogrammetry software on set.

4.3.2 Hardware and Software for Photogrammetry

Once we receive the real locations and actors, there will not be a lot of room for changes to the game. It would therefore be necessary, to test out the game's design using photogrammetry so we could iterate based on that feedback without having to iterate the game design once the correct 3D assets were generated by photogrammetry.

For this, our method was to use an iPad Pro with 3D depth scanning with the software application, PolyCam as detailed in Photogrammetry Software 2.1.4. Here we used LiDAR 3D scanning, see Defining LiDAR 2.1.2 since it allowed us for a quick and easy to use method to get a real physical object translated into 3D, see Figure 21.

(a) Overview of scene

(b) Close-up of an actor

Figure 21: Examples of our quick versions for testing

In particular, as can be seen at Figure 21b, the character's face does not appear very appealing, but considering how long it took to develop this asset, the result is certainly worthwhile. During the scanning process, there is a waiting period of approximately five minutes, followed by approximately fifteen to thirty minutes to cut the edges and export the asset to Unity.

As soon as these temporary assets were acquired, we could begin testing the new iteration of the design with assets that were closer to the final look than the greybox version at Figure 15.

When getting to the actual recording of final scans for asset creation we prepared both the iPad Pro 12 with Polycam now using their 'photo mode', based on a photogrammetric method. We also took photos with a Digital Camera that would be used in conjunction with Agisoft's MetaShape software application.

We decided we wanted to try out both methods for our first location so we could determine whether or not the extra time and effort of using a higher quality software and camera would be worth it for our use.

We came to the conclusion that using Polycam's 'photo mode' would give us the most effective results based on the time and effort it takes compared to using a digital camera and the MetaShape software. Since we are dealing with a low fidelity of photogrammetry, Polycam's lower quality aligned better with what we wanted and was much more effective to produce.
We therefore stuck to Polycam for the entirety of the creation. Figure 22 shows a behind-the-scenes look at what that looked like.

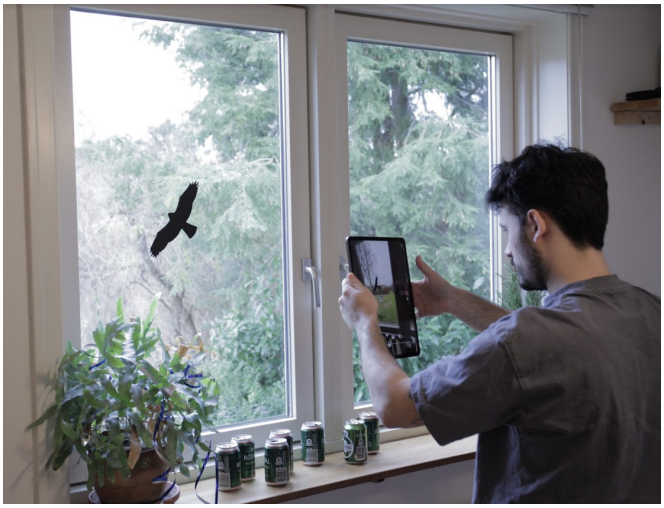


Figure 22: Behind-the-scenes using Ipad Pro 12 with Polycam

4.3.3 From Photogrammetry Scan to Unity

The 3D assets were mostly processed by a 3D artist and another designer while we implemented them into the Unity Engine to make them playable. A simplified description of the process would be like the following:

1. Scan Room for the scene (using Polycam)
2. Process the scan to fit the art style (using Blender)
3. Cutout each object from the scan (Using Blender)
4. Make all objects interactable (Using Unity)

For our contribution to this process we have mainly been a part of the first, second and the last part of the process.

1. Scan Room for the scene
The Polycam application was used to take photos of an entire room. The polycam application had a limit of 250 photos per scan, so to get a reasonable level of detail, we would split up a scene into 2-3 parts, containing enough detailed images for each area. A screenshot of Figure 23 shows a typical Polycam process:
We would choose the 'photo' and 'record' option. 'Photo' being the photogrammetry mode and 'record' would make the application automatically take a new photo when moved to a slightly new angle of the desired frame. The process of taking photos would then be to keep having a mental reference point each time you move to a new angle. When we come to a point in the

scan where we would not be able to have the same reference point to get it right, we would need to find a new reference point, and from transitioning from one reference point to another the previous photo should contain both reference points so that the complete scan would accurately simulate the 3D space. As object masking is intended for scanning a specific object, we would export it as a raw file and deselect object masking for processing the scan. As you can see in the last screenshot on Figure 23, you can review the scans that you have created shortly after they are created and therefore retake the scans right on location immediately. This helped us immensely, due to the fact that we were mostly learning by doing and would not generally have the opportunity to redo scenes later. Therefore, the process was very iterative and flexible, and we had the opportunity to make sure we were getting the exact processed scan we wanted before moving forward.

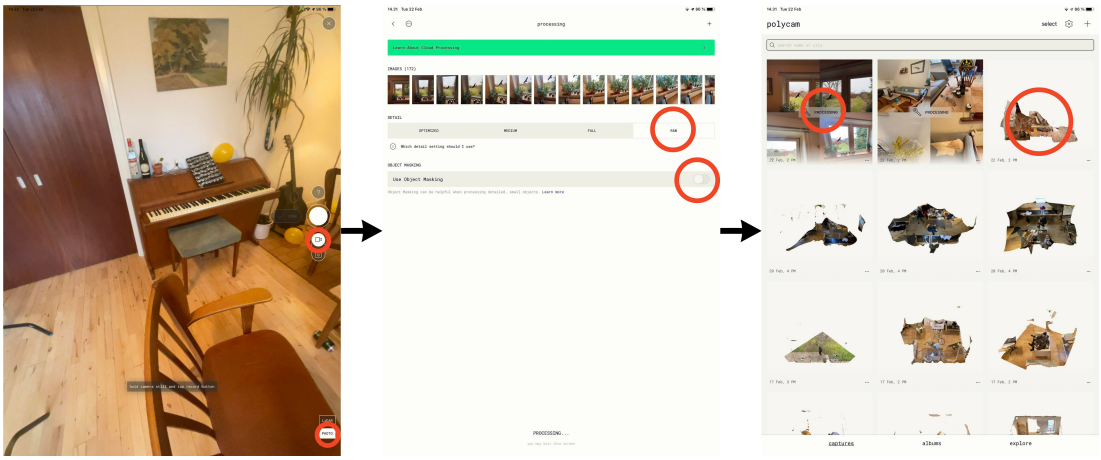


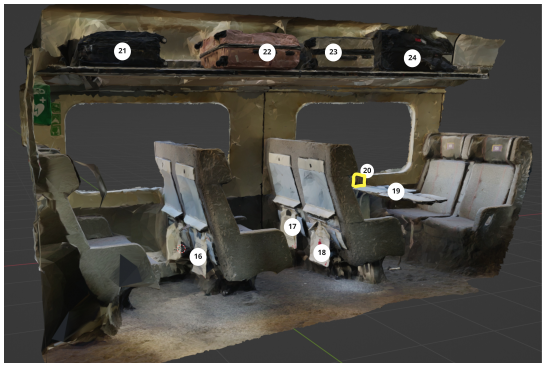
Figure 23: Process for scanning a scene using PolyCam

2. Process the scan to fit the art style

Our next step was to process the scan to match our style. In this instance, we will utilize the free open-source 3D computer graphics program, blender[Blender.org 2002] to process the scene in order to achieve the desired low fidelity style. In addition to scaling and setting up the scene to ensure it is consistent with the other scenes, we would most importantly use Blender's 'Decimate Modifier' in order to decimate the 3D objects. By doing this, we will be able to reduce the vertex/face count of the mesh with minimal changes to the mesh itself [Blender.org 2022].

3. Cutout each object from the scan
For this, we had 3D artist performing the task. In essence, we noted down everything in a scene that required a separate cutout in order to adhere to the game design of the player being able to pick up any person or object. Two examples of how we would label each object that needed to be cutout individually to a 3D artist are demonstrated in Figure 24.

(a) Each bike needed to be cut out



(b) Different objects around the train needed to be cut out

Figure 24: Examples for cutting out 3D Objects of the entire scan

4. Make all objects interactable
An additional programmer on the team would write a script, called Memory Interactable.cs, for the Unity Engine[Unity 2005], the game engine we used to develop the game. As a result of this script, a 3D object could be manually made interactive, allowing you to replace an object with another object that is also interactive. A lot of tedious manual work was put in to this process as one scene alone would contain 50+ objects that needed to be manually sorted. Figure 25 is a screenshot of how the final result would look like from a gameplay perspective.

Figure 25: An example on the visuals in game right before you replace an object with another

4.4 In-Game Questionnaire

The purpose of this section is to cover the process of designing questions for the In-Game Questionnaire as well as the actual player journey of when and where the player should be prompted to answer the questions.

Before implementing an automated in-game evaluation system, we would act as facilitators, asking the questions we would later require the in-game system to ask and store.

Testing early versions of the game, one test facilitator was present to note down the following:

- What did the participant know about the game prior to testing:
- What games have the participant played recently that they enjoyed.
- How long did it take to complete the test

While the above is the general data we were interested in gathering, whenever the participant took an object and replaced it, we would ask them the following questions before telling them if the replacement was correct or not.

- Why did you choose this object?
- From 1-5, How confident are you that this is correct? (1 = completely guessing and 5 = Sure of this being correct)

An example of how we would note that down can be seen below at Figure 26, while Early testing of Evaluating In-Game Feedback G contains all the raw documents for the early testing.

Pink Door → M2.2 (4) (CORRECT)
"She touches something newly painted, she has pink paint on her and there is a pink door here."

Figure 26: *"statement"* = Why did participant choose this object
Pink Door = the object they chose to replace
M2.2 refers to what memory hole they replaced it with
(4) = how confident was the participant that it would be correct from 1-5

It was critical for us to keep this approach of asking the participants before revealing the truth, since it allowed us to capture their immediate responses regarding why they chose a given object before being influenced by whether it was right or wrong. The statement, *"She touches something newly painted, she has pink paint on her and there is a pink door"* directly tells us that the participant understood the puzzle and actually solved it with the exact mental model we intended. In less successful examples, we would get an answer that would be based purely on logical guesswork since the object stood out in the scene, which is not what we woud the sole reason to be for harder designed puzzles.
Our participants tested several different versions of the game, both on paper prototypes and computer prototypes with temporary LiDAR versions that emulated the photogrammetric style we were attempting. All tests were conducted according to the same methodology, with minor changes to the way the questions were asked in order to iterate on the best formulated questions that provides the most profound responses.

Whenever we received answers regarding 'why a person would choose an object', we generally, if it was not completely guesswork, received a general explanation, which may be in accordance with the size, shape, or direct clues in the environment, or it may be based upon narrative reasoning that some object must be found at this location because it makes sense for the narrative progression. It turned out that both of these reasons were valuable for providing insight into how to improve the game. General reasoning can provide us with an indication of the ease with which the environment itself would assist a player in solving puzzles. Narrative reasoning will provide us with a direct indication whether or not the solution is accompanied by a narrative explanation, and as a result, if it is aligned with the intended narrative. The Likert item concerning a participant's perceived confidence could be used to determine whether the general and narrative reasoning was a calculated guess or something they strongly believed to be accurate. However, we observed that in many cases when a participant was very confident about something, but it did not turn out to be correct, we observed a tendency for those particular participants to be less confident in their next attempts, even though they would generally and narratively be accurate in their reasoning. This indicates that we should regard that question carefully and not place too much weight on the perceived confidence itself, but rather look into why the participant felt confident or insecure about their choices.

All in all, the process for the evaluation method for narrative intelligibility in our in-game feedback tool ended up as followed:

1. Participant would pick up an item
2. Replace that item with a memory hole
3. Game will freeze.
4. An in-game feedback canvas would prompt, see Continuous in-game questionnaire for evaluating Narrative Intelligbility 27
5. After answering questions
6. Game will unfreeze revealing whether or not it was correct.
7. This loop will continue until the game has been completed.

-*How confident are you that this item is correct?* (From a scale of 1 (not at all confident) to 5 (Very confident))

-*Why did you choose this item in general?*

-*Why did you choose this item in the context of the story?*

Figure 27: Continuous in-game questionnaire for evaluating Narrative Intelligbility

This approach would only provide as an understanding to the exact moments of the story we also wanted to ensure that the participants could summarize the whole story collectively. To do this, we would at two points of the game ask them a question. After completing the introduction to the story, getting a sense of how much they understood so far and after fully completing the Worry memory. The question that would prompt in-game at these two points in time were:

What do you think the story is about so far?

In addition to that we see it fitting to also include continuation desire adapting two questions that are designed to be asked continuously while playing:

I want to continue playing now! from a scale of 1-7 (strongly disagree to strongly agree)

Why/Why not do you want to continue playing

These additional questions will potentially provide us data to triangulate with the 'Narrative Engagement Questionnaire', see Narrative Engagement Questionnaire E.2 as well as help us correlate the desire to continue playing with how profoundly they could or would elaborate on their narrative reasoning for solving puzzles in the game.

5 Implementation

In this section, we will examine two parts of the in-game feedback tool, namely the core system and the Mind Diver integration, along with instructions on how to set up the tool's settings. In addition, we will discuss and describe various systems we developed in conjunction with Mind Diver as part of our collaborative effort. While these systems have an effect on the game experience, they were not specifically developed for the purpose of the test, as the Mind Diver integration aspect of the in-game feedback tool was. The code of the core system is available on GitHub at:

`https://github.com/Nicrodk/UnityGoogleFormResponse`

5.1 System Overview

The system we designed has been implemented in two parts. We have the core system and the Mind Diver integration. There are two key components to the core system, which are the storage of values and the sending of those values to the Google Forms server, as detailed in Core System 5.3. The diagram as one image is in the appendix in Implementation Class Diagram C.

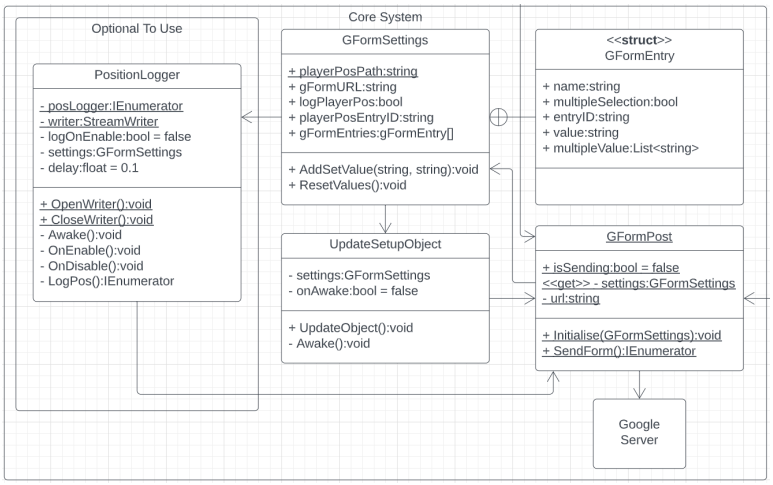


Figure 28: Figure showing the class diagram of the core system

The second component, the integration of the Mind Diver game, is concerned with how the participant is prompted to answer the questions during the test, and providing this information to the core system, as detailed in Integration With Mind Diver 5.4.

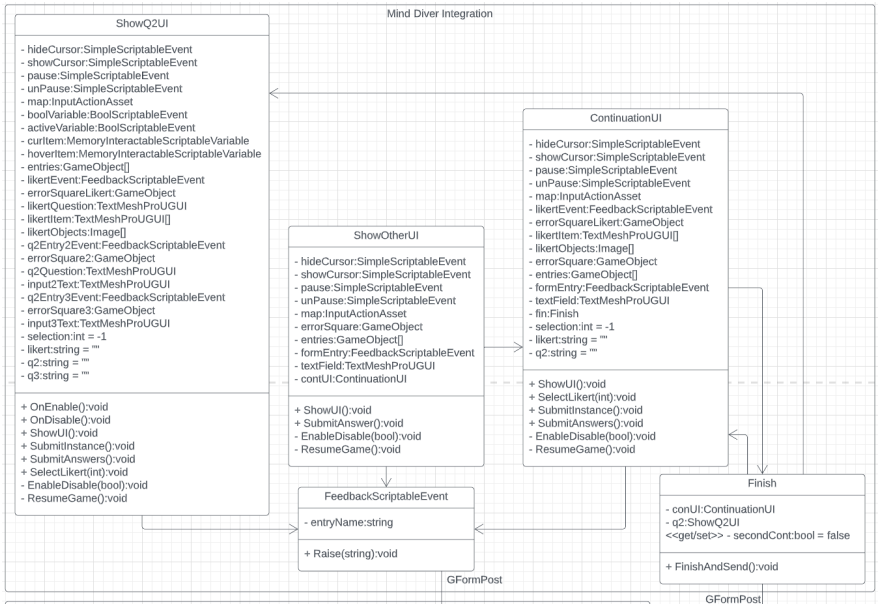


Figure 29: Figure showing the class diagram of the Mind Diver integration

5.2 Google Form Request

Our feedback system mimics the POST request that the Google Forms website sends when the send button is pressed on a Google Form. A few pieces of information from the Google Form website are required in order to properly replicate the request. First and foremost, the URL of the Google Form, so that it is known where to send the request. In addition, it is necessary to know the designation of each of the questions on the form. These come in the form of "entry.XXXXXXXXX", where the "X"s are the numbers constituting the entry's ID, as shown on Figure 30.

```
entry.594644774: test2
entry.333899234: 5
entry.782532863: Single 3
entry.842598442: Multiple 1
entry.842598442: Multiple 2
entry.842598442: Multiple 4
```

Figure 30: Figure showing the payload of the google form POST request

There are two ways to find this ID; the first is to examine the page source of the website, which contains the form. Among the mess of characters, it is possible to utilize the search function of the browser to locate the name of the question, and then to find the ID of the question, as shown on Figure 31.

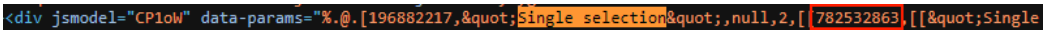


Figure 31: Figure showing the page source of google form with the question name highlighted and the entry id in the red square.

When one has become familiar with finding the ID number, this is likely the fastest way. For the second method, one must inspect the POST request sent by the browser upon pressing the submit button. This can be accomplished in the Google Chrome browser by opening the developer tools. To do this, hit the F12 key or go to the browser options under "More Tools" and then "Developer tools". When the tools are open, select the Network tab at the top, then just under the line with the tabs, select "Preserve log". When this feature is enabled, as long as the developer tools are opened and on the network tab, the requests and responses will be logged, meaning the POST request sent by the form will be displayed, and the user can also filter the results by typing in "formResponse", thereby removing the majority of the clutter and making it easier to locate the desired POST request. Lastly, with the selected request, it is possible to view the payload of the request. Figure 32 shows how the developer tools are displayed when the request is selected, and opened to display its payload.

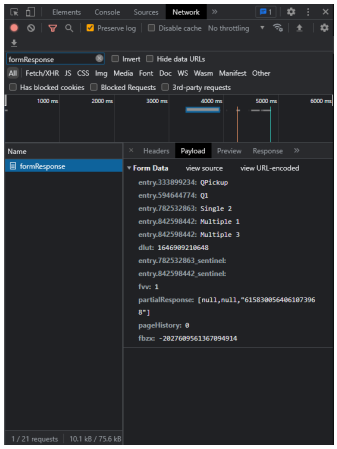


Figure 32: Figure showing the network tool in the google chrome browser filtered to formResponse and showing the payload of the request.

Figure 32 shows that there is nothing to distinguish which question has which ID, since all of the questions are listed as "entry." followed by their ID number. Therefore, since a dummy response is always necessary, if there is an option on each question, which is the name of the question, it is easy to figure out which question has which ID. From Figure 32, it can also be observed that the questions allowing multiple selections simply pack multiple fields into the payload with the same entry ID and the name of each selected option. Multiple choice questions also have the same structure, except that only one response may be selected, resulting in only one field. It was theorized that since Google Forms is able to show old responses to questions, even when the selected option has been replaced, it might be possible to send any string in response to a multiple choice question. Testing has revealed that this is not possible, since if the value of the field does not correspond to one of the current options of the question, it results in an 'HTTP 400 error', which gives the error message Bad Request, meaning that the request that was send did not meet the servers expectations.

5.3 Core System

The functionality of the core of the system we developed is primarily derived from two scripts, one being GFormSettings and the other being GFormPOST.

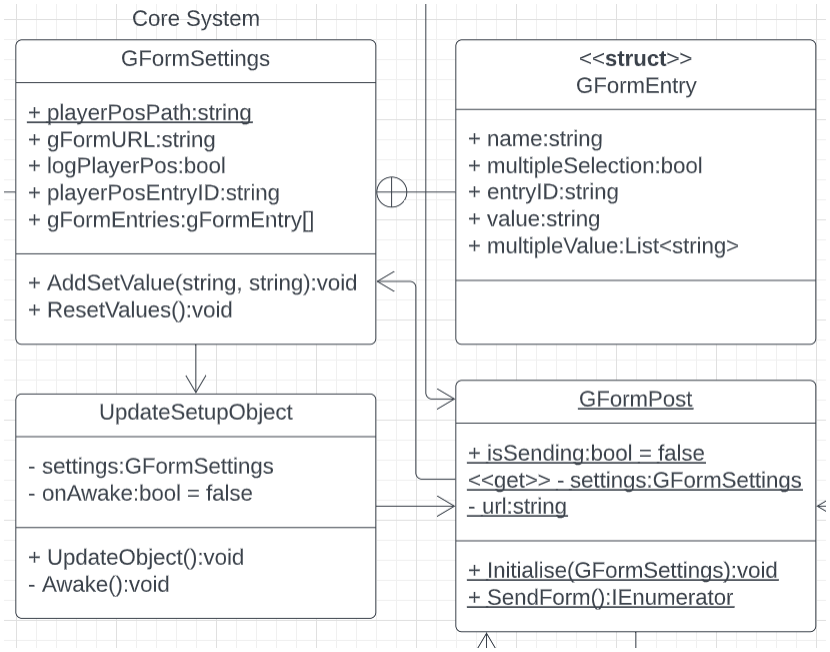


Figure 33: Figure showing the core system without positionlogger

GFormSettings represents the data container of the core system. It contains the URL for the Google Form, as well as an array of a struct which carries the ID for the question and the data to be sent. A GFormSettings instance is created as its own file, since it inherits from the Unity class ScriptableObject. An advantage, but possibly also a disadvantage of ScriptableObjects is that since they are their own files, they hold their data between plays in the editor. Therefore, when the game is tested in the editor, the ScriptableObject will preserve the data, and unless that data is removed prior to building, it will be available within the game. Although this is a problem within the editor, when the build has been made, the ScriptableObject no longer saves data to a file, but to memory only. Meaning that upon boot-up of the game, the ScriptableObject will contain the same data as when the game was built. The class GFormPost, which is a static class, serves as a container for the GFormSettings script, allowing other scripts to interact with the settings object through this script statically, and also includes the functionality to send the Google Form response. With this setup, it is possible to utilize multiple Google Forms within the same game. The only thing required is a GFormSettings object per Google Form and some integration in the game to allow switching between the setting objects during runtime. This can be accomplished by simply providing the $Initialise(GFormSettings)$ function of GFormPost, with a different GFormSettings object. It is important to note that, since GFormSettings objects are ScriptableObjects, a method of data preservation will be required if information is collected over time for the response, rather than a simple response that is sent immediately, so that answers can be saved in the event of a crash or a user closing the game prematurely.

5.3.1 Setting up GFormSettings

When creating the GFormSettings object, the primary component will be to obtain the Google Form URL, as well as the various entry IDs for each of the questions on the Google Form. The procedure for doing so is outlined in Google Form Request 5.2. After that, it is simply a matter of inserting the information within the GFormSettings object, with the URL at the top and an entry in the array for each question. To fill out an array entry it is given the question's entry ID as "entry.XXXXXXXXX", and the name of the array entry. Within the application, the name of the entry serves as an identifier to be able to provide data to this entry. This identification of an entry could have just as easily been accomplished by the entry ID, however, the name member was added so that a custom name could be specified and as such connect to the entry through the name in other scripts. Figure 34 illustrates how a filled out settings object would look.

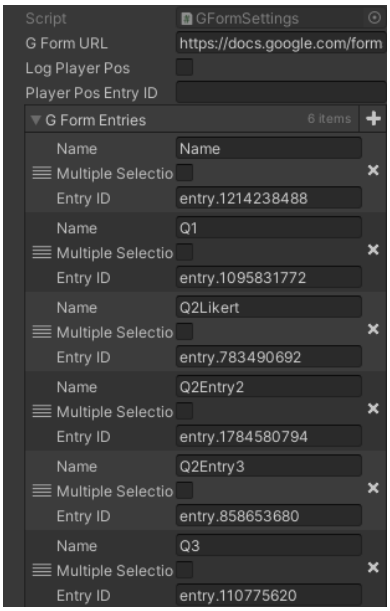


Figure 34: Figure showing the inspector of a GFormSettings ScriptableObject

On the figure it can also be seen that there is the option for logging a player's location and providing an entry ID, in which the contents of the file containing the player positions will be output.

5.3.2 GFormPost

While GFormSettings is the data body of the structure, the GFormPost script is the container and sender of that data. In designing the core system, the purpose was to allow others to use it by integrating it with their own applications. Keeping this in mind, we have integrated GFormSettings with GFormPost so that the function for adding or updating the values in the GFormSettings object is on the GFormSettings object itself. The result of this is that it is possible to send data to a GFormSettings object, which is not the currently active object in GFormPost. The intended way of providing data to GFormSettings, is by the static reference in GFormPost. This will allow the function to be called from anywhere as

$GFormPost.settings.AddSetValue(string, string)$, and so to ensure that this intention is followed, it would also be possible to implement it more strictly. By moving the $AddSetValue(string, string)$ function into GFormPost, the static function call becomes
$GFormPost.AddSetValue(string, string)$. In this case, the data will be restricted to only be provided through GFormPost, and only to the currently active GFormSettings object. Our decision was to go with an open implementation, since as noted, the intention was for others to utilize the system and they may have a use case where they wish to provide data to multiple GFormSettings objects at the same time. The only concern with the open implementation is that, if mismanaged, it may be possible to send data to the incorrect GFormSettings object, as well as creating uncertainty about which one is the currently active one, when sending data to a Google Form. In order to create and format the data for the request, the Unity WWWForm class is used. While UnityWebRequest is the class used to send the request and to examine the response. As the data is set up by calling $WWWForm.AddField(string, string)$, the setup of a WWWForm is fairly straightforward.

5.3.3 UpdateSetupObject and PositionLogger

For our core system, we have developed two optional scripts, the UpdateSetupObject script and the PositionLogger script. The UpdateSetupObject script's purpose is to provide a GFormSettings object to the GFormPost script, either through the Unity life-cycle function $Awake()$ or via the $UpdateObject()$ function. This script is optional due to the fact that it is completely possible, as indicated at the end of Core System 5.3, to integrate the provision of a GFormSettings object with any script. Therefore, if this integration is not done, then UpdateSetupObject is a necessary part of the system, since this will be the script that provides the GFormSettings object to GFormPost.
PositionLogger allows the location of the GameObject, which it is attached to as a component in Unity's inspector, to be logged into a local text file. Then GFormPost is able to read the file and output the contents as an answer to a paragraph question on a Google Form. For the PositionLogger to enable logging of position, it must first be attached to a GameObject and then, on the GFormSettings object, the entry ID for the paragraph question must be specified and a boolean value must be set to true.

5.4 Integration With Mind Diver

To integrate the core system with the Mind Diver game, some UI components to get user inputs was needed, as well as a handful of scripts to manage these UI components, send the information in the components to the core system, and also in the end, start the process for the core system, to send the answers to the google form.

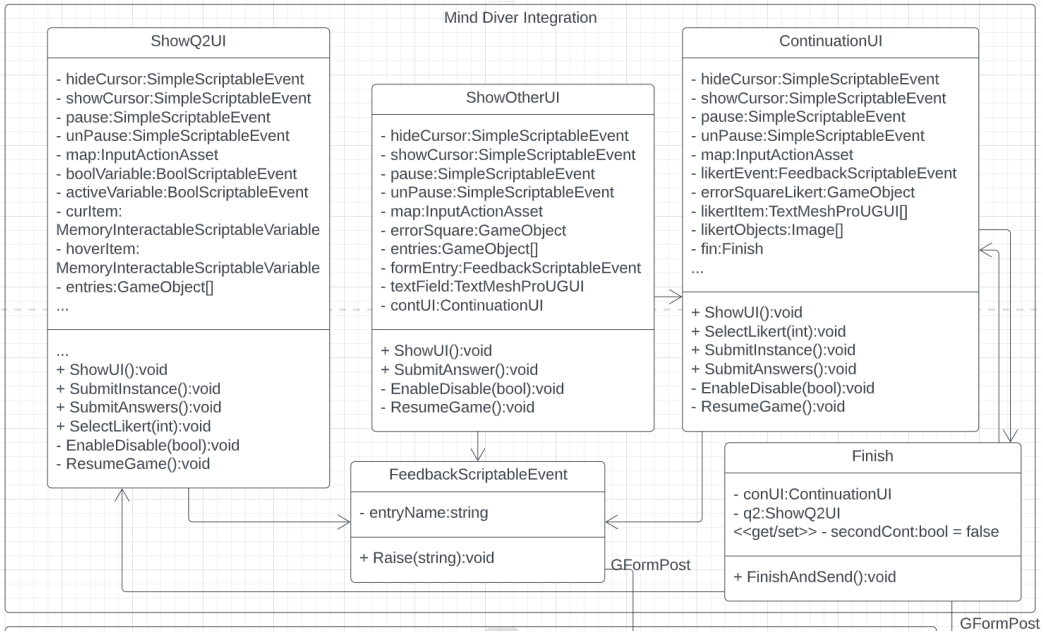


Figure 35: Figure showing the Mind Diver integration shortened

5.4.1 Scriptable Events

The Mind Diver project, with which we are collaborating, utilizes a package called Scriptable Events[Danevičius 2022]. This package was developed by another programmer of the Mind Diver team. With this package, it is possible to set up events as ScriptableObjects, which are paired with listener scripts. When an event a listener script is listening for is raised, the corresponding Unity event to the listener will be activated. Unity events can be defined in the inspector according to what they are supposed to do when they are activated. A Unity event in this situation may set a GameObject active or inactive or invoke a script function. The boon of these events being ScriptableObjects, is the possibility of cross scene referencing, where an event is raised in one scene, and a listener in a different or the same scene, activates some functionality.

5.4.2 FeedbackScriptableEvent

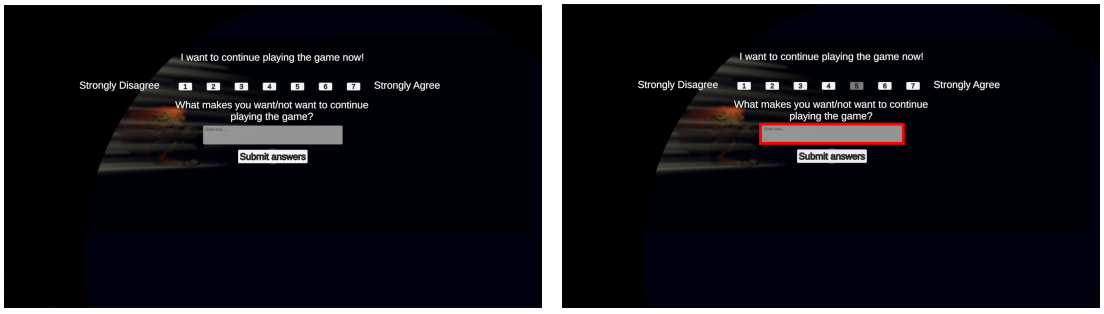
In order to provide information to the GFormSettings object, we needed a simple script that would hold the name of an entry and would include a function that, given another string, would provide information. In order to accomplish this, FeedbackScriptableEvent inherited from StringScriptableEvent, which comes from the Scriptable Events package. However, because of the way the functionality was implemented, the inherited members were hidden anyway. FeedbackScriptableEvent could have inherited from Unity's class ScriptableObject, since

it was directly provided to the UI scripts, as described in ShowOtherUI ShowQ2UI and ContinuationUI 5.4.3. The UI scripts then call the function on the feedback object, passing it information gathered from the UI elements. The feedback object then provides this information to the GFormSettings object, by having the entry name defined for the feedback object in the inspector, and then providing the value through the function call, where the feedback object then calls the $AddSetValue(string, string)$ function on the GFormSettings object, through the static reference in GFormPost.

5.4.3 ShowOtherUI ShowQ2UI and ContinuationUI

Three scripts were designed in order to manage the various UI elements, which the user can interact with to provide us with feedback. There is one script for one-off questions, called ShowOtherUI, one script for repeated questions, as the user attempts to place an item into a 'memory hole', called ShowQ2UI, and one script for the continuation desire questions, called ContinuationUI. An example of the UI elements of the continuation desire question can be seen on Figure 36a.

All three scripts contain references to four SimpleScriptableEvents from the Scriptable Events package and one InputActionAsset from Unity. The five references are used to disable player inputs, properly pause and resume the game, as well as show and hide the cursor. The one-off questions, such as asking for a person's name in order to identify the response with the pre/post questionnaire response, and asking what they think of the story so far, are handled by the ShowOtherUI script. Upon calling the $ShowUI()$ function, the script will pause the game, display the user interface elements, and then wait for something to call the $SubmitAnswer()$ function. In our case, this is accomplished by a Unity UI button that is set to call the function on pressing the button, and as the function runs, it checks whether the text written in the Unity InputField is not null and if it is longer than two characters. In the case that these requirements are not met, instead of sending the answer to the feedback object, it will display a red square around the inputfield to alert the user to the problem. Figure 36b provides an example of this red square.



(a) Continuation Desire Questions UI

(b) Red error message if not fulfilled

Figure 36: Continuation Desire - In-Game

In the event that the requirements are met, it will send the answer through the FeedbackScriptableEvent, provided to the script in the inspector, and then resume the game. The ShowQ2UI script is responsible for handling the repeatable questions that appear when the user attempts to place an item in a 'memory hole'. As with ShowOtherUI, this script has a $ShowUI()$ function that pauses the game, displays the cursor and enables the UI elements. However, unlike

ShowOtherUI, ShowQ2UI waits for a *SubmitInstance*() call. The script also checks to ensure the likert item has been selected, as well as that the input fields contain more than two characters, with red squares indicating errors. If the requirements are met, the function adds the current answers to private strings of the class, and resumes the game. ShowQ2UI will not send the responses directly to the GFormSettings object until the *SubmitAnswers*() function is called, which is handled by the Finish script described below in Finish Script 5.4.4. Lastly, for continuation desire questions, these are handled by the ContinuationUI script, which functions similarly to the ShowQ2UI by means of the *SubmitInstance*() and *SubmitAnswers*() functions. Difference between the two is that for continuation desire questions there is only one input field.

5.4.4 Finish Script

The last script of the feedback system integration with Mind Diver, is the Finish script. This script was developed to provide a final check to ensure that the answers from both the ShowQ2UI script and the ContinuationUI script have been passed to the GFormSettings object before the call to *SendForm*() on the GFormPost script. There is a private boolean called secondCont in the Finish Script. It was introduced because at the end of the *ResumeGame*() function, which is called by *SubmitInstance*() in the ContinuationUI script, it will make a *FinishAndSend*() call for the Finish script. Therefore, the boolean will be set to true by the submit button of the last question, before the second set of continuation desire questions are answered, so that the *FinishAndSend*() function will be executed when these continuation desire questions are answered, which are the final questions of the test.

5.5 Other Systems for the game to function

In our collaboration with Mind Diver, we also developed other features and systems that are part of the test. However, these features and systems have not been designed specifically for the test or feedback system. As part of these other systems, we designed a system for changing the lighting color of the environment, a tutorial manager system that could show images and text as required with integration of FMOD event emitters for audio. In addition, functionality for the ability to drop an object currently held by the user, as well as a system to trigger a dynamically modified audio event, when the player collides with the terrain. Furthermore, we developed a batch script for some of the earlier iterations of the photogrammetry pipeline.

5.5.1 Colour Change

The first system that we developed for the Mind Diver game, as part of the collaborative effort, was a system that changed the light elements and unity fog. The changes were implemented by using a ScriptableObject that contained a set of settings for the colors of each light and fog. This was implemented using a simple script, which would receive the settings via Scriptable Events.

5.5.2 Tutorial Manager

Another slightly larger system we created was a tutorial manager, which had the functionality to display subtitles, an image, and play an FMOD audio clip as part of a game tutorial, helping the user understand how the game works and how the controls work. Initially, the system contained an array of a struct containing a reference to a Unity localized string, from the Unity localization package, for keyboard, and one for controller, as well as an image for each type of control. The struct also stores a reference to an audio event emitter from FMOD and a GameObject that holds a listener for a Scriptable Event to interact with the tutorial. As the tutorial progressed to the entry, the tutorial manager would enable this GameObject on its own, and then disable it as the tutorial progressed further. The design was intended to be a slideshow of sorts, where a function named $ProgressTutorial()$ would move to the next entry in the array, in the same way a slideshow would move to the next slide. As time passed, it was discovered that there was also a need to display an image and text in response to user actions if the user did something incorrectly. As a result of this, in conjunction with the, at this point, rather inefficient array setup, a new iteration of the system was developed. As a result of the overall design idea derived from the 'Colour Change' system, this updated design utilized a ScriptableObject for each entry instead of an array of entries. This resulted in the tutorial manager simply showing the image and text associated with the ScriptableObject, as provided to the script. Additionally, this would simplify the process for the game designers who would be able to attach a TutorialEntry ScriptableObject to a trigger rather than have to set a new item in the array on the TutorialManager script.

5.5.3 Drop Item

Another feature we added was the ability to drop an item that had previously been picked up. At the time, the controls would only permit the "dropping" of a held item, by trying to replace it with something else, correct or incorrect. Hence, after reviewing the various scripts related to memory interactables as well as ScriptableVariables, which are included in the Scriptable Events package, it was discovered that there was a script, which had a function that needed to be called and also ScriptableEvents that had to be raised for the proper resetting of the held item, so that another item could be picked up afterward and the old item would be reactivated.

5.5.4 Collision Audio

In order to increase player experience, the audio part of the team requested a system that when there is a collision with terrain, a sound would be played. Nevertheless, it was not simply a matter of playing an audio clip from a FMOD event emitter, but also interacting with FMOD, since there was a variable, as part of the audio event, that could be used to modify the audio clip. With this modification, it was possible to play audio clips related to the terrain that the player was colliding with, as well as adjust the volume of the clip based on the speed of the player. Therefore, it was designed with a lower and upper bound for the velocity of the player character, which was then normalized, since FMOD volume values range from zero to one.

5.6 Software use and batch processing

The first objective we had when coming up with a batching process was to learn how to even create a batch script. Batch works in a rather simple manner, where the main part of the script uses the echo command to put a line into a command prompt, therefore it effectively automates what you would normally do in a command prompt line by line. One of the first

things the batch script would need to do is to be able to extract frames from a video and output each one as a separate image file. For this purpose, we looked at both VLC media player and also a library called ffmpeg. VLC was found to be lacking, as there seemed to be no way to manage it contained to the command prompt. VLC must be running and the video must also be playing in order to extract images. For this reason, ffmpeg was used, as it would allow the process to be contained within the command prompt environment. A file selection was required for the purpose of image extraction, so that the script knew where it could find the video file, which was to be passed to ffmpeg. Using batch to select files is a very restricted version of the Windows file selection window; a file can only be found by traversing through the directories, with no search bar or option to input a directory, as is common in Windows Explorer-like file selection windows. Because of this, a part of the batch script was set up to generate a temporary PowerShell script, and then to execute this script, so we could utilize the proper Windows Explorer-like window to select files. The setup for this did require a bit of work, as PowerShell by default does not allow script execution; as a result, a security setting had to be alleviated a bit to allow locally created scripts to run. After these images have been extracted, we would want to supply them to a photogrammetry application. At this point we encountered a stumbling block, as the software that the Mind Diver team was using at the time did not permit script integration unless a much higher price was paid. However, there was a batching process within the program itself, which utilized XML files. The final step of the batch script is to generate a text file into a specified folder which will contain a filled out XML file that can be passed along to the photogrammetry program. In the end, when the photogrammetry software output was compared with the output of Polycam, it was decided to go with Polycam, which runs entirely independently. Polycam is responsible for both capturing imagery and generating 3D models, so the batch script was of some assistance up until this point. However, since Polycam does everything on its own, it was no longer necessary.

6 Evaluation

The purpose of this section is to provide more information on the testing procedure by describing first what part of the game is used for our evaluation, as well as a step-by-step test procedure. This is followed by an explanation of how we intend to analyze the data we are collecting as part of the test process.

6.1 Test Procedure

The section of the video game 'Mind Diver' we used for the evaluation was the the first quarter of the entire game. The test was sent out as a google forms containing both the pre- and post questionnaire aswell as a link to the downloadable executable hosted on the gaming platform itch.io. The test was distributed online on a Facebook forum group for students at Aalborg University, aswell as direct messaging people we knew that would be considered the target audience. A few participants were conducted purely online with a facilitator following along online while most participants were conducted by finding people at Aalborg University, Copenhagen Campus. To get as many participants as possible within a short amount of time. While also wanting to ensure that participants could do the test without needing help from a facilitator. We would ask people in groups (between 2-3 people) to test getting multiple participants at once making them sit far enough from each other to not influence one another, but still making it possible for us as facilitators to observe if anything went wrong.
The participants were informed to answer the first part of our questionnaire, then play through the prototype, before finally answer the post-questionnaire.
While this is not a fully controllable environment we thought it would represent the vision we had for the system best while still maintaining some level of control to quickly notice any potential confounding variables.
With that said, prior to the real evaluation we conducted five pilot tests that helped us locate game breaking errors as-well as changes to some questions to ensure a more valid evaluation. Which in turn, gave us the confidence to that test could be done in a less controlled matter.
The following is a detailed step by step process of how the test took place:

Part one: Prior to the experience Participants would read and agree to the consent form, see Appendix D Demographics(name, gender, age, video-game experience, photogrammetry experience, see Demographics / Background Information E.1 and The Visual Experience E.4)
Part two: Start the game Participants would start the game and the first prompt would be to state their name similar to the pre-questionnaire in order for us to compare the right data sets.
Part three: Play through the tutorial Participants would play the tutorial of the game getting acquainted with the controls and the logic behind solving puzzles.
Part four: Play the first part of the 'Central Memory' Participants would get the introduction to what the narrative is about.
Part five: Narrative Intelligibility and Continuation Desire As the participant would go to the next section of the game a in-game prompt would freeze the game asking the following questions: - *Describe what you think the story is about so far?* (Related to narrative intelligibility) - *I want to continue playing the game now!* (From 'strongly disagreeing' to 'strongly agreeing', related to Continuation Desire) - *What makes you want/not want to continue playing?* (Qualitative answer to understand the previous answer, related to Continuation Desire)
Part six: Play the 'Worry Memory' Participants would solve the puzzles between three moments related to the same memory. While doing so, each time the participant would try to correct a memory the game would freeze before revealing the truth prompting the following questions all related to narrative intelligibility: - *How confident are you that putting "Name of object" here is correct?* (From 'not at all confident' to 'very confident') - *Describe why you in general chose "Name of object".* - *Describe how you think "Name of object" belongs here in the context of the story.*
Part seven: Completing the game After completing the Worry Memory, the game would prompt one last questionnaire asking the same questions as in part five.
Part eight: Post-Questionnaire The game would stop and ask the player to return to the google forms questionnaire which would continue them to answer the post questionnaire. - Narrative Engagement Questionnaire, see Narrative Engagement Questionnaire E.2 - Narrative Tension, see Narrative Tension Questionnaire E.3 - Photogrammetry related questions, see The Visual Experience E.4 - Questions regarding the annoyance and intrusiveness to the in-game questionnaire, see In-Game Questionnaire Experience E.5 - Overall experience, questions related to the game in general, see The Virtual Environment E.6

Table 1: Procedure - Step by step

6.2 Data Analysis

When analyzing the data for our in-game questionnaire, there are a number of perspectives that can be explored. With that, we will focus on the relationship between the narrative intelligibility and puzzle design. There will be four different puzzles, each requiring you to find the correct object and place it in the correct 'memory hole' so that the memory can be restored and the story can continue.
We will refer to these four puzzles as the following:

1. Trashcan, see Figure 37
2. Photo: Sebastian, see Figure 38
3. Police Officer, see Figure 39
4. Mind Diver Flyer, see Figure 40

The four names refer to the object the player will need to use to restore the memory, all four intended to be related to the story the player will experience. The following four figures, Figure 37, Figure 38, Figure 39 and Figure 40 are shown to provide a mental image of how these four puzzles are presented in the game, showing both the *unsolved* version with a memory hole and the *solved* version with the respective item.

(a) Unsolved (b) Solved

Figure 37: Puzzle to solve: Trashcan (1)

(a) Unsolved (b) Solved

Figure 38: Puzzle to solve: Photo: Sebastian (2)

(a) Unsolved (b) Solved

Figure 39: Puzzle to solve: Police Officer (3)

(a) Unsolved (b) Solved

Figure 40: Puzzle to solve: Mind Diver Flyer (4)

Considering that we do not have a control group to compare data points with in the form of Likert scales and items within the post-questionnaire, we will instead rely on the neutral viewpoint of a Likert scale of 1 - 7, that being 4, to determine whether the engagement, intrusiveness or likeness of our case study is positive, neutral or negative. To make this process more readable, all items were converted for representation from 3 to -3 (0 representing neutrality). We additionally analyze individual items from Likert scales due to a low sample size as well as in order to facilitate a deeper interpretation of the results.

We conducted interpretive analysis for the qualitative findings, looking at each participant's individual responses. This enabled us to quantify answers into different categories to gain a better understanding of the subjective opinions of the participants. Wordclouds and a sorted Excel file of all participants' answers are provided to limit subjective analysis bias.

7 Results

In this section, we outline the demographic characteristics of our participants and then present the results of our data using boxplots, histograms, and word clouds along with text to explain the meaning of the figures, but without further elaboration as this will be addressed in Findings 7.9 and Discussion 8.

7.1 Participants

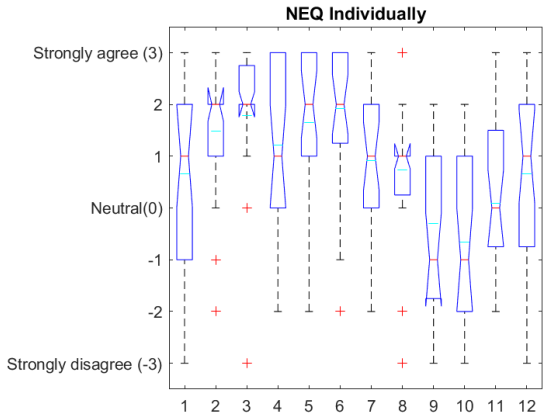
Gender	
Male	20
Female	2
Other	1
Age	
20-24	8
26-29	10
31+	5
Would you consider yourself to be a ...	
Casual Gamer (e.i. you dabble in games but in short sessions or infrequently)	5
Core / Mid-Core Gamer (i.e. you regularly play video games but are not super serious or competitive)	11
Hardcore Gamer (i.e. you are a dedicated gamer and play seriously or competitively)	6
In a typical week, about how many days do you spend at least 30 minutes playing a video game?	
0-1 days	5
2-3 days	9
4-5 days	3
6-7 days	6
Do you work in the video game industry	
Yes	3
Study games in academia	11
No	9
Do you know what photogrammetry is? (before playing this game)	
No, I don't know what it is.	3
I have seen something like this before, but did not know it was called photogrammetry.	5
Yes, I know what photogrammetry is.	8
Yes, I know what photogrammetry is and have worked with it.	7
	n(23)

Table 2: Demographics, frequency of participants per answer n(23)

Our demographics show that the sample size consists primarily of males (20/23) with most participants being fairly well rounded video game players playing games at least twice a week (18/23).(14/23) participants either work in the video game industry (3/23) or study games in academia (11/23). (15/23) participants know (8/23) or have worked with photogrammetry

before (7/23) indicating that this is a sample group with a strong interest in video games and some pre-established opinions regarding photogrammetry.

7.2 Narrative Engagement



(a) Narrative Engagement Questions shown per item.

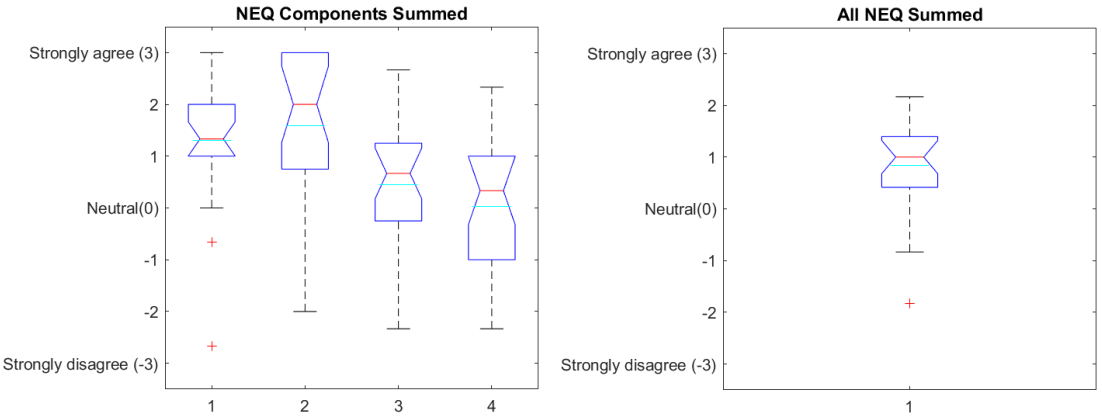
Question	Median	Mean
1	1	0,65
2	2	1,48
3	2	1,78
4	1	1,22
5	2	1,65
6	2	1,91
7	1	0,91
8	1	0,74
9	-1	-0,3
10	-1	-0,65
11	0	0,09
12	1	0,65

(b) Table: Narrative Engagement Questionnaire

Figure 41: The 12 individual Narrative Engagement Questions

1. At points, I had a hard time making sense of what was going on in the game. (-)
2. My understanding of the characters are unclear. (-)
3. I had a hard time recognizing the thread of the story. (-)
4. I found my mind wandering while playing the game. (-)
5. While the game was on I found myself thinking about other things. (-)
6. I had a hard time keeping my mind on the game. (-)
7. During the game, my body was in the room, but my mind was inside the world created by the story
8. The game created a new world, and then that world suddenly disappeared when the game ended
9. At times during the game, the story world was closer to me than the real world
10. The story affected me emotionally
11. During the game, when a main character succeeded, I felt happy, and when they suffered in some way, I felt sad
12. I felt sorry for some of the characters in the game

Figure 41a shows the answers given by the participants per question of the twelve questions of the Narrative Engagement Scale, (-) = reverse coded. Question one was reverse coded and has a median of 1 and a mean of 0,65. Question two was reverse coded and has a median of 2 and a mean of 1,48. Question three was reverse coded and has a median of 2 and a mean of 1,78. Question four was reverse coded and has a median of 1 and a mean of 1,22. Question five was reverse coded and has a median of 2 and a mean of 1,65. Question six was reverse coded and has a median of 2 and a mean of 1,91. Question seven has a median of 1 and a mean of 0,91. Question eight has a median of 1 and a mean of 0,74. Question nine has a median of -1 and a mean of -0,3. Question ten has a median of -1 and a mean of -0,65. Question eleven has a median of 0 and a mean of 0,09. Question twelve has a median of 1 and a mean of 0,65.



(a) All 4 Narrative Engagement components summed and scaled down.

(b) All 12 Narrative Engagement Questions summed and scaled down.

Figure 42: Narrative Engagement Questionnaire: Subscales and Total

Figure 42a shows a plot of the answers divided into their four subscales of the Narrative Engagement Questionnaire, while Figure 42b shows all twelve questions, hence all four subscales summed up and then scaled down to provide an average of all answers.
The 1st subscale represents **Narrative Understanding** with a median of 1,33 and mean of 1,3. The 2nd subscale represents **Attentional Focus** with a median of 2 and mean of 1,59. The 3rd subscale represents **Narrative Presence** with a median of 0,67 and a mean of 0,45. The 4th subscale represents **Emotional Engagement** with a median of 0,33 and mean of 0,03. Combing all subcales gives a median of 1 with a mean of 0,84.

Subscale	Median	Mean
(1) Narrative Understanding	1,33	1,3
(2) Attentional Focus	2	1,59
(3) Narrative Presence	0,67	0,45
(4) Emotional Engagement	0,33	0,03

Table 3: This table shows the median and the mean of the four subscales of the Narrative Engagement Questionnaire on a 7-point scale

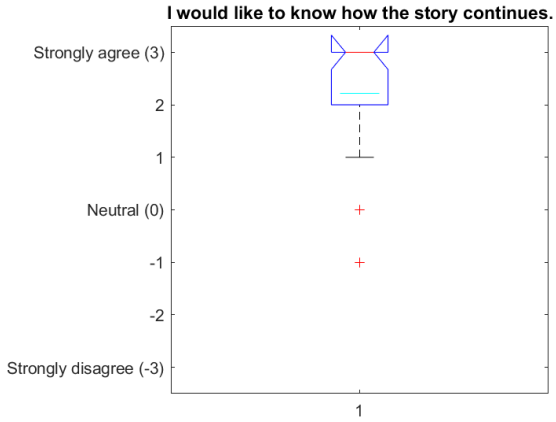
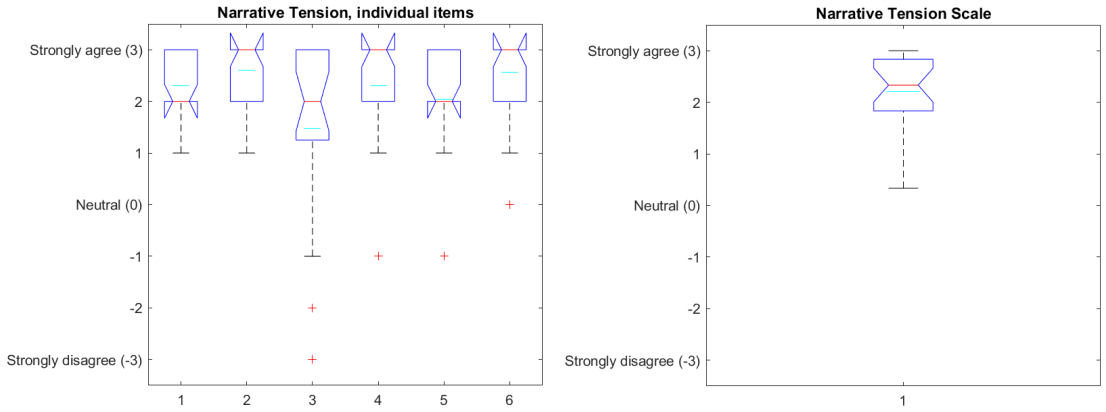


Figure 43: An extra question pertaining to narrative engagement

Figure 43 show a plot of the answers given to an additional thirteenth question pertaining to narrative engagement, which has a median of 3 and a mean of 2,22.

7.3 Narrative Tension



(a) Items individually

(b) Summed and scaled down

Figure 44: Narrative Tension Scale: Individual Items and Summed

Figure 44a shows the answers to six questions relating to the narrative, question 1, 3, 4 and 6 are reverse coded. 1 has a median of 2 and a mean of 2,3. 2 has a median of 3 and a mean of 2,61. 3 has a median of 2 and a mean of 1,48. 4 has a median of 3 and a mean of 2,3. 5 has a median of 2 and a mean of 2,04. 6 has a median of 3 and a mean of 2,57.

Figure 44b shows the answers to the six narrative questions summed and then scaled down, this has a median of 2,33 and a mean of 2,22.

7.4 Narrative Intelligibility

7.4.1 What do you think the story is about so far

(a) During Gameplay

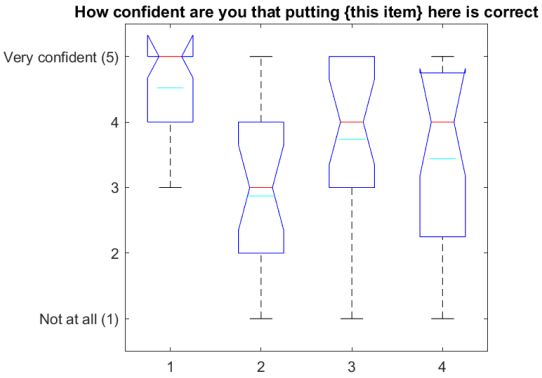
(b) After Gameplay

Figure 45: What do you think the story is about so far?

Figure 45 illustrates two wordclouds showing the 50 most frequent words from the in-game questions participants would get during the gameplay experience, see Figure 45a and after the gameplay, see Figure 45b. The words are only filtered to being stemmed e.g. memory and memories being categorized as the same word. Additionally, 6/23 particpants has been removed from 'During Gameplay' and 8/23 participants has been removed from 'After Gameplay', due to them not answering the question and instead stating their opinion on about the game. This resulted in 10 participants in total whereas 4/10 participants did not answer the question as intended in both 'During' and 'After' Gameplay.

7.4.2 Participants Perceived Confidence

Quantitative: Participants Perceived Confidence



(a) Confidence levels of participants of the correct item placements.

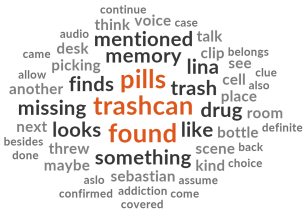
Item	Median	Mean
(1) Trashcan	5	4,52
(2) Photo: Sebastian	3	2,87
(3) Police Officer	4	3,73
(4) Mind Diver Flyer	4	3,43

(b) Table showing the medians and means

Figure 46: How confident participants were in their corrections

Figure 46a shows the answers participants gave pertaining to how confident they were that the item they were trying to put into a memory hole was correct with the figure showing the answers that were given to the actually correct placements. An elaboration of the figure entries is shown on Figure 46b.

Qualitative: Participants Perceived Confidence



(a) 1. when the correct object to replace was: 'trashcan'

(b) 2. when the correct object to replace was: 'Photo: Sebastian

Figure 47: Wordcloud for the 1st and 2nd puzzle challenge

Figure 47 illustrates a wordcloud containing the 50 most common words when describing the general and story explanation to why they believed that the object was correct. Again, only

illustrating for the actual correct objects; (1st = trashcan) and (2nd = Photo: Sebastian). Same goes for Figure 48 which instead shows; (3rd = Police officer) and (4th = Mind Diver Flyer).

(a) 3. when the correct object to replace was: 'Police Woman'

(b) 4. when the correct object to replace was: 'Mind Diver Flyer'

Figure 48: Wordcloud for the 3rd and 4th puzzle challenge

Figure 49 categorized the frequency of answers from participants guessing or not having a coherent statement to their answers in the general explanation and their story explanation.

Correct Object	General, No Coherent Statement	Story, No Coherent Statement
Trashcan	2/23	1/23
Photo: Sebastian	8/23	9/23
Police Officer	0/23	1/23
Mind Diver Flyer	1/23	4/23

Figure 49: Frequency of guessing or no coherent statement between general explanation and story explanation (n=23)

7.5 Continuation Desire

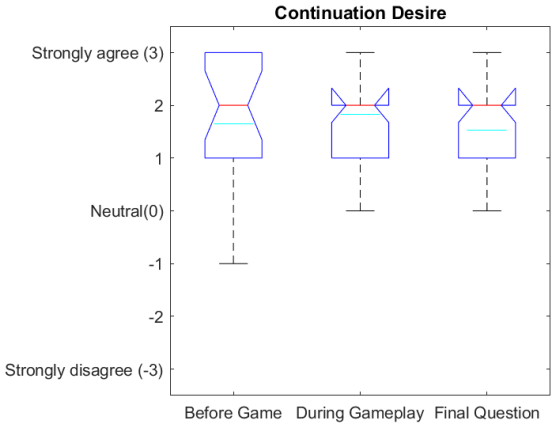


Figure 50: Continuation desire answers from pre questionnaire, during the test and at the end of the test.

Figure 50 shows a plot of the participants answers to the question "I want to begin playing the game" in the pre-questionnaire and then the question "I want to continue playing the game now!" during gameplay and the final question of the test is also in the setting of the participant playing the game so "Final Question" is also while they are playing the game. Before Game has a median of 2 and a mean of 1,65. During Gameplay has a median of 2 and a mean of 1,82. Final Question has a median of 2 and a mean of 1,52.

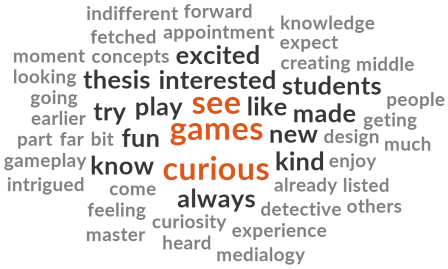


Figure 51: Wordcloud: Continuation Desire, before starting the game

Figure 51 shows a wordcloud containing the 50 most used words before playing the game. What is shown is that the word curious is a major factor in their description, while also words as "thesis" and "medialogy" indicating that people are excited due to it being "excited" to play a game made by other students, which makes sense due to a significant amount participants that are students of a video game education.

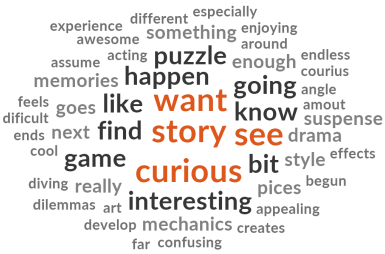


Figure 52: Wordcloud: Continuation Desire, during the game

Figure 52 shows a wordcloud containing the 50 most used words in the middle of the game. What is shown is that the word curious and story are major factor in their description, while also words like "interesting" and "puzzle" are common words used.

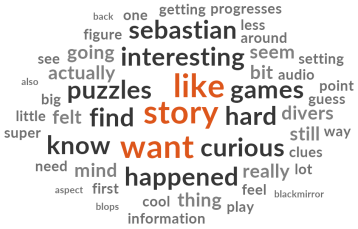


Figure 53: Wordcloud: Continuation Desire, after the game

Figure 53 shows a wordcloud containing the 50 most used words after the game ended, while also words like "Sebastian" and "Interesting" are common words use. This is due to participants wanting to know what has happened to Sebastian, Sebastian being the missing person in the narrative.

7.6 Photogrammetry

A part of the post questionnaire was about the art style of the game, this being photogrammetry regarding our sub-research question.

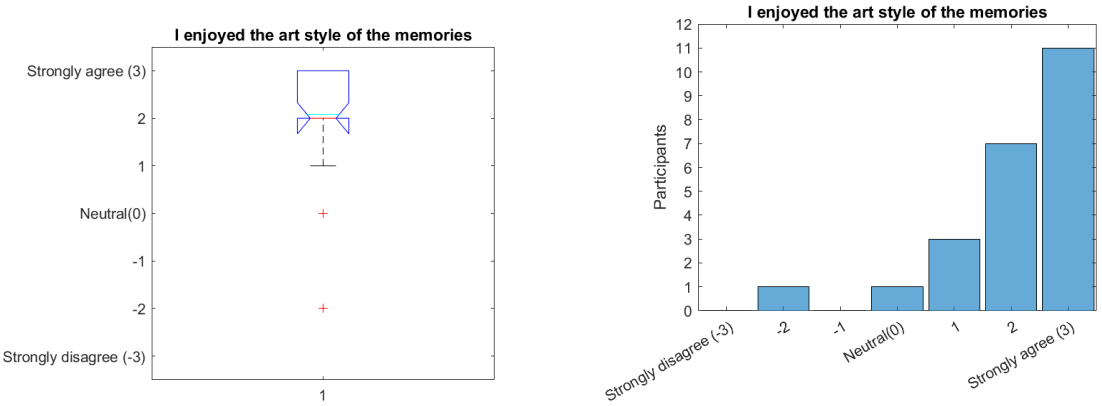


Figure 54: A boxplot and histogram of answers to if the participant enjoyed the art style

Figure 54 shows a boxplot and histogram of the answers to the question about if the participant enjoyed the art style of the memories. These answers has a median of 2 and a mean of 2,09.

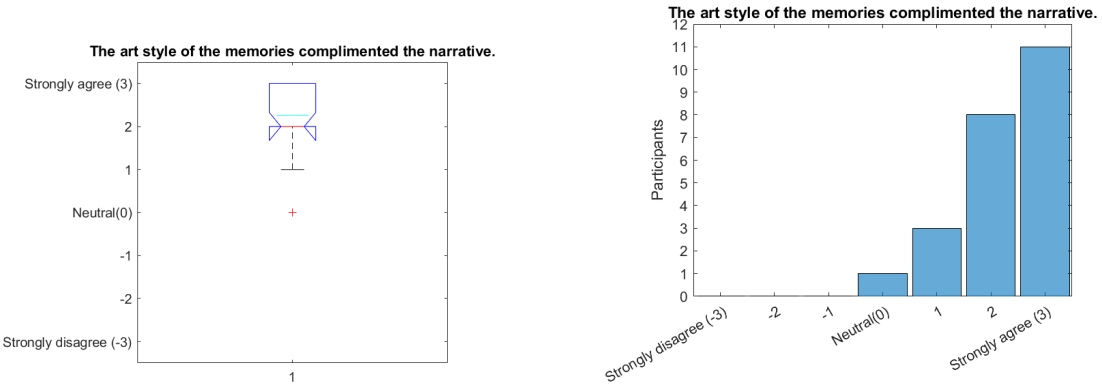


Figure 55: A boxplot and histogram of the answers to if the participant thought the art complimented the narrative.

Figure 55 shows a boxplot and histogram of the answers to the question about if the participant thought the art style complimented the narrative. These answers has a median of 2 and a mean of 2,26.

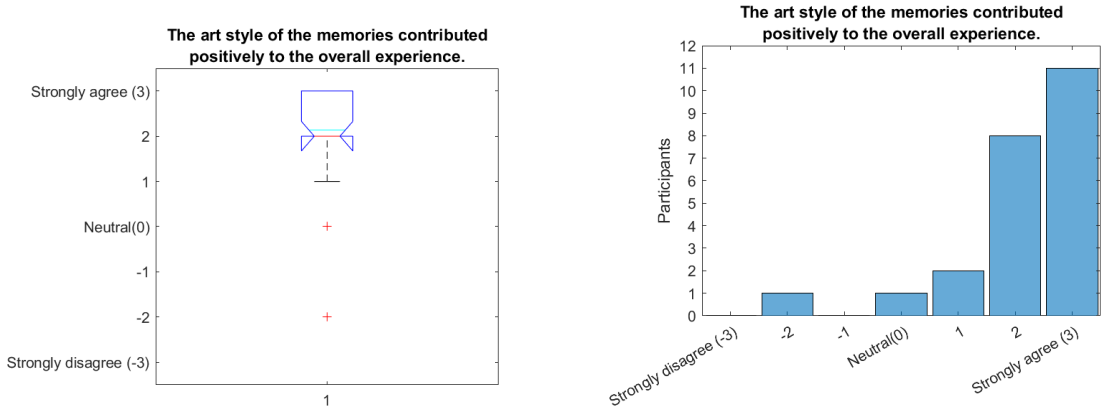


Figure 56: A boxplot and histogram of the answers to if the participant thought the art style contributed to the experience.

Figure 56 shows a boxplot and histogram of the answers to the question about if the participant thought the art style was a positive contribution to the overall experience. These answers has a median of 2 and a mean of 2,13.

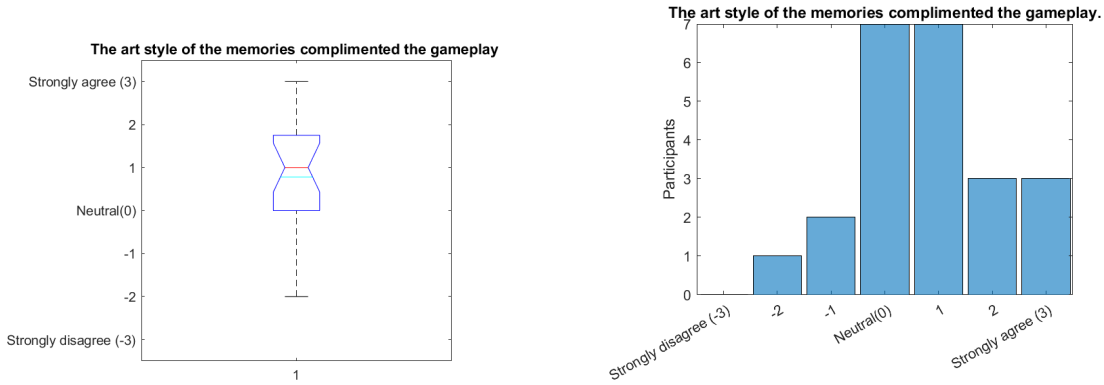
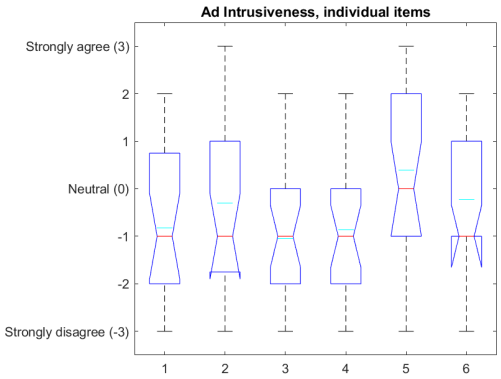


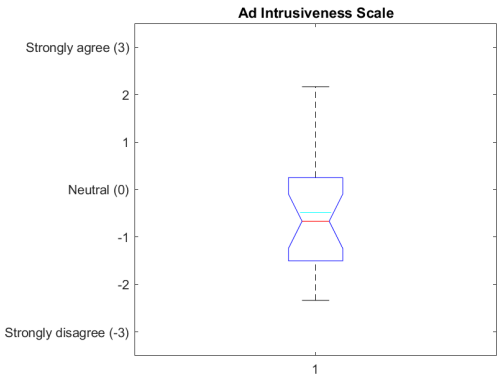
Figure 57: A boxplot and histogram of the answers to if the participant thought the art style complimented the gameplay.

Figure 57 shows a boxplot and histogram of the answers to the question about if the participant thought the art style was complimenting the gameplay. These answers has a median of 1 and a mean of 0,78.

7.7 Intrusiveness towards the In-Game Questionnaire



(a) In-game feedback system intrusiveness answers shown as individual items



(b) In-game feedback system intrusiveness answers summed and scaled down

Figure 58: Intrusiveness of the In-Game Questionnaire

Figure 58a show a plot of the answers given to the six questions pertaining to the intrusiveness of the in game questions being asked to the participant, these questions can be seen on the following list and the median and mean for each of these are shown on Table 4.

1. When the in-game questionnaire was shown, I thought it was [Distracting (to divert the attention)]
2. When the in-game questionnaire was shown, I thought it was [Disturbing (to confuse a quiet, constant state or a calm, continuous flow)]
3. When the in-game questionnaire was shown, I thought it was [Forced (to do something involuntarily)]
4. When the in-game questionnaire was shown, I thought it was [Instrusive (to cause disruption or annoyance through being unwelcome or uninvited)]
5. When the in-game questionnaire was shown, I thought it was [Invasive (to invade a person's thoughts or privacy)]
6. When the in-game questionnaire was shown, I thought it was [Obtrusive (to be noticeable or prominent in an unwelcome and unpleasant way)]

	1	**2**	**3**	**4**	**5**	**6**
Median	-1	-1	-1	-1	0	-1
Mean	-0.83	-0.3	-1.04	-0.87	0.39	-0.22

Table 4: Intrusiveness of in game questions

Figure 58b shows a plot of the answers to all six questions summed up and then scaled down to provide an average of all answers with a median of -0,67 and a mean of -0,48.

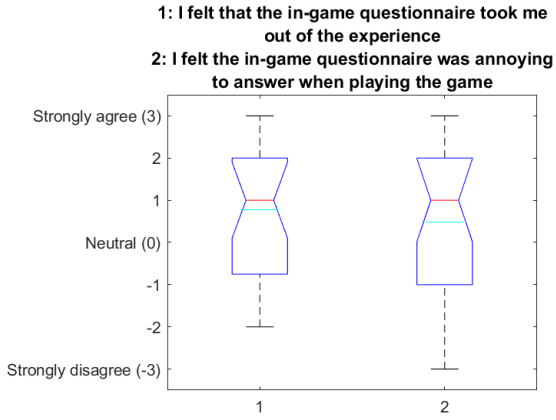


Figure 59: An extra two questions about the in-game feedback system intrusiveness

Figure 59 shows two boxplots for two additional questions we posed outside of the scale which is shown on Figure 58. The answers for 1 has a median of 1 and a mean of 0,78. The answers for 2 has a median of 1 and a mean of 0,48.

7.8 Virtual Environment

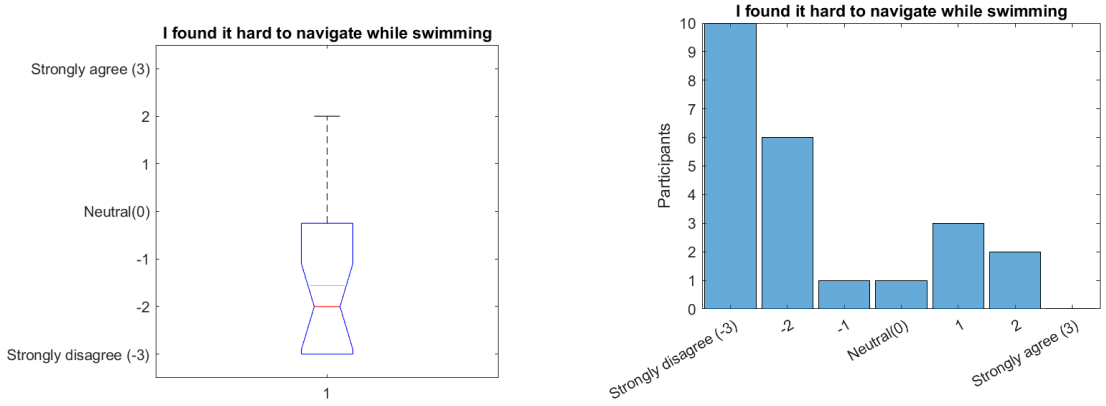


Figure 60: Boxplot and histogram of answers to how hard it was to swim

Figure 60 shows a boxplot and histogram of the answers to the question about if the participant thought that it was hard to navigate in the game. These answers has a median of -2 and a mean of -1,57.

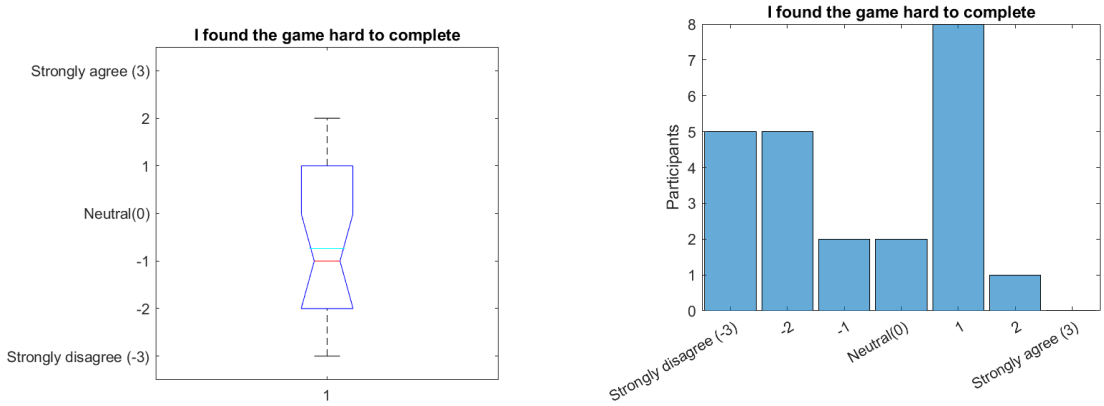


Figure 61: Boxplot and histogram of answers to how hard it was to complete the game

Figure 61 shows a boxplot and histogram of the answers to the question about if the participant thought that the game was hard to complete. These answers has a median of -1 and a mean of -0,74.

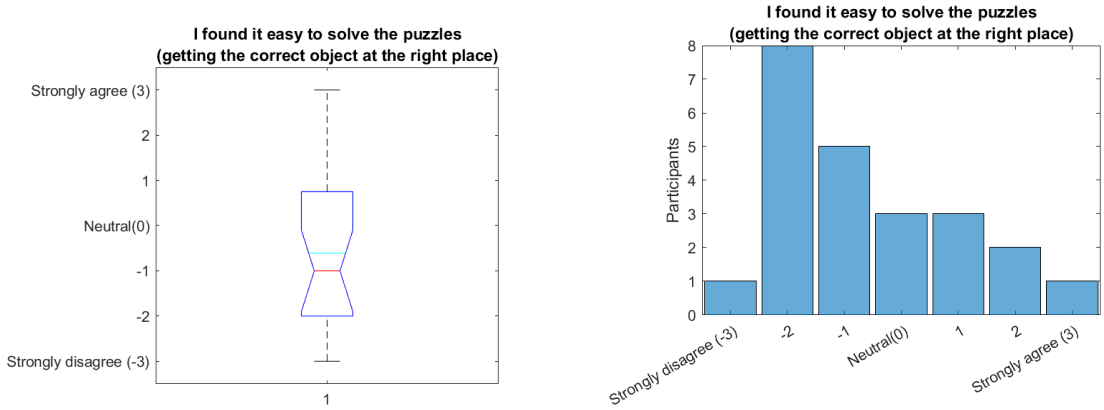


Figure 62: Boxplot and histogram of answers to how easy it was to solve puzzles

Figure 62 shows a boxplot and histogram of the answers to the question about if the participant thought that the puzzles of the game was easy to solve. These answers has a median of -1 and a mean of -0,61.

7.9 Findings

We can speculate based on the qualitative and quantitative data from the results. We will begin by examining the answers to our main research question, namely, in-game narrative intelligibility. When looking at the perceived confidence participants had with correcting a memory (1-5), trashcan(median=5), Photo: Sebastian(median=3), Police Officer(median=3), and Mind Diver Flyer(median=3). On the basis of these numbers, we see that the 'Trashcan' was the most reliable answer, indicating that the trashcan puzzle was the easiest to solve. In addition, from the comments provided by the participants, it appears that the reasons for their choices were largely in accordance with the intended general reasons we wanted:
"I can see a trashcan lid on the floor."

"Logical choice at first, then made near definite by the audio in the previous room saying she found the pills in the trashcan."

While the greater story explanation is not as clear, but still contains some predictions from the participant:

"Maybe Sebastian has taken some kind of drug and threw the drug case in to the trashcan thinking that no one would find it there. But Lina is smarter than that!"

"Sebastian is doing drugs."

"Sebastian threw his pills in the trash."

"Not sure yet."

When looking at the most confusing puzzle to solve being 'Photo: Sebastian', which also appears as answers being generally:

"No Idea" or *"Guessing."*

Observing it in retrospect, it becomes apparent that the clues are not explicit enough. In this puzzle, there is no verbal explanation of why the object should be that particular one. In the physical space the only hint is that 'Lina' is holding something, without any indication of what that might be. This story explanation of why this object exists shows that some participants could guess why this might be the case. However, they still expressed little confidence in the perceived confidence item:

She is trying to find this missing person named "Sebastian" but she can't. Now she is looking at his image.

"She could be looking at a picture of him because she is missing him."

Those quotes are what the author intended. Some participants could have assumed it, however from the overall responses, it has not been adequately portrayed in either the environment or the narrative that this was the case.

To determine whether the in-game questionnaire did influence the narrative engagement we can see that the summed median score is 1 from (-3 to - 3), where 0 would be the comparable

neutral. We would want narrative engagement to be above 0 to indicate that there is some form of narrative engagement. When looking at 'Narrative Tension' we see a median of 2, which is also higher than the comparable neutral standpoint. With regard to the desire to continue playing the game, there is a significantly positive level of engagement. This is reflected in the median score of 2 before, during and after gameplay, with a neutral score of 0. Indicating that there is a general desire to start and that general desire persists all the way to the end of the game, while not increasing or decreasing. At the same time the continued desire is generally shown to be because of curiosity towards the game, while at the end of the game we see that the continued desire to play leans over to being about wanting more of the story:

"I want to know what happened to Sebastian".

All this indicates that the narrative experience was engaging and that it was the narrative itself that generally led to the continued desire to play.

The in-game questionnaire potentially negatively influenced the narrative engagement, meaning the narrative is even more engaging than what we can depict from the data. All items on the intrusiveness scale scored below the comparative neutral with a median score of -1. Additionally, the two separate items that we formulated to understand the reasoning for the in-game questionnaire being annoying or not had a median score of 1 in both items, which indicates a negative result due to the negatively loaded questions formulated. Participants were asked to explain their reasoning for these two items where the general reasons were negatively loaded with:

6/23 participants disagreed to the questionnaire taking them out of the experience, for example:

"I mean, as part of a testing phase, it is to be expected, and it wasn't annoying, but rather helped get my thoughts and feelings out as they occurred. In the finished game of course it would be pointless."

3/23 participants neither agreed or disagreed to the questionnaire taking them out of the experience, for example:

"It was actually nice to reflect on what is happening, but wiring it down made it tedious."
14/23 participants agreed to the questionnaire taking them out of the experience, for example:

"It kind of put me in a different mindset, where I had to make justifications for what may have been loose guesses or playful exploration. It slightly took some of the flow out of the experience I had."

8/23 participants disagreed to the in-game questionnaire being annoying to answer when playing the game, for example:

"It went quite fast so I did not mind."

Another common statement that could make it look better due to the test participants being mainly students that are used to test other projects:

"after 6 years at university I'm used to being a lab-rat for other students."

2/23 participants neither agreed or disagreed to the in-game questionnaire being annoying to answer when playing the game. Neither of these two participants elaborated on their answer as to why.

13/23 participants agreed to the in-game questionnaire being annoying to answer when playing the game, for example:

"The main issue was that I felt I needed to answer every potential solution the same way, which could take up a lot of mental energy and motivation. For example if I have tried 2-3 different things before getting the right answer, I think I would answer the questions more and more superficial, because I didn't want to spent time on describing an answer if it was incorrect."

7.10 Photogrammetry: sub-research question

Reviewing the findings for our photogrammetry sub-research question shows to be a strong indication that the art-style of the game, and in particular the narrative, was positively influenced by the photogrammetric approach. From (3 to -3, with 0 representing neutral), a positive influence on liking the art style (median=2), complimenting the narrative (median=2), complimenting the gameplay (median=1), and complimenting the overall experience (median=2) are shown:

Photogrammetry	**negative**	**neutral**	**positive**
Liking the art-style	1/23	1/23	21/23
Complimenting the narrative	0/23	1/23	22/23
Complimenting the gameplay	3/23	7/23	13/23
Overall complimenting the experience	1/23	1/23	21/23

Figure 63: Frequency of opinions towards the photogrammetric art-style, n(23)

Overall, this indicates that the art-style positively influenced the gameplay experience, with the only concerns being about some lack of details in the 3D assets, which affected the gameplay experience as the game relies on visually identifying objects, for example:

It's too clutter for my eyes. It confused me more when I was trying to find stuff.

While this is a concern it shows to be minor for the sake of this experience as more than half of the participants (14/23) explicitly stated our intended design philosophy of using the imperfections of the art style to enhance the theme of exploring broken memories, for example:

the art style works well with the fragmented memories theme

I like that the art style is like this, because when I think of the mind, I think of it not clear and clean but more imperfect. The memory feels like it isn't always crystal clear.

I believe that photogrammetry and its imperfections are an ideal way of representing memories because of the inherent inaccuracies that they have.

7.11 Virtual Environment

Aside from the general technical difficulties, we also looked at some other gameplay elements to determine if we had any other concerns about the experience that may have negatively affected the experience.

As far as swimming in the game is concerned, it showed no difficulty, with 5/23 participants finding it difficult while 1/23 found it neither difficult nor easy.

As for the difficulty of the game or the difficulty of the puzzles, there have been mixed results which make it difficult to provide a specific explanation. (9/23) participants reported difficulty in completing the game, while (14/23) participants experienced difficulty with the puzzles.

Overall, it indicates that in general the game can be completed, which we know because all participants completed it. The game does seem to be challenging for many participants, but whether or not this negatively affected the game is hard to tell based on the data we have.

8 Discussion

The results shown in the previous section, Results 7, will be discussed in this section, beginning with the research question for narrative intelligibility. Following that, photogrammetry will be discussed, followed by the in-game questionnaire, and the final section will discuss the testing process as a whole, with focus on the reliability and validity of the evaluation.

8.1 Narrative Intelligibility: Research question

Will using our implemented in-game evaluation tool provide usable player insights within narrative intelligibility, without being intrusive and negatively impacting narrative engagement? - main research question, see Research question 2.7 for further details.

Based on these results, we can conclude that the in-game questionnaire could provide us with some interesting indications as to how our video game, 'Mind Diver,' can be improved. In my analysis of Findings 7.9, it appears that we have generally succeeded in getting answers that can be used to determine the areas of our design that have been successful and those that require improvement. Considering the four main objects a player would need for correcting the memory, it is possible to identify what was the most confusing item by simply examining the perceived confidence each participant had in his or her ability to correct the memory. We can confirm this by looking at the qualitative reasoning from each participant as it is clear that the object *'Photo: Sebastian'* was the one that scored the lowest in their perceived confidence and also appeared to be the one where participants mainly stated that they either were guessing or did not have a clear answer as to why they chose this object over a set of other objects. In light of these findings, it is evident that the puzzle has some issues. However, identifying how to solve that problem is not as clear from the data alone. However, because we did have puzzles that were successful, we could use this to guide our decision-making. Using this information, we can determine how these other puzzles differ from those that are unsuccessful and design solutions that are more similar to those that are a success. This works in our case, but if the scenario were that none of them were successful, we would not be able to indicate the underlying reasons for their failure. In general, it is easier to localize a problem than to find a solution to it. Therefore, localizing the issues is a helpful first step. If the case were that significant issues were shown in all puzzles, it would then be a clear indication that the entire design philosophy needed to be rethought.

Since we do not have a control group to compare our data points against, it is hard to determine whether our methodology negatively impacts narrative engagement. In the ideal scenario, we would test the in-game questionnaire on one group of participants while having a control group that merely played the game and responded to the NEQ, and compare data between an experimental group required to undergo an in-game questionnaire evaluation and a control group not required to undergo an in-game questionnaire evaluation. However, due to the lack of participants we could obtain, we decided that splitting up our test size into two n(23) was not a useful method. Therefore, 11 participants would be in one group and 12 in another, which is far too small for an experimental design to provide any meaningful findings.

In our exploratory approach, instead, we compared each Likert item to its middle point. Based on this, we can see that the narrative scored higher than our neutral threshold in all metrics except for a few items on the Narrative Engagement Questionnaire examined individually. A total of 2/12 items scored below our neutral threshold, and 1/12 items scored just at our neu-

tral threshold, mainly questions regarding emotional engagement, which we were expecting to be lower because the narrative that the player would experience would only serve as an introduction to the story and therefore not allow the player to establish an emotional connection with the characters. In light of these findings, there is no immediate evidence that the in-game questionnaire is influencing narrative engagement to the extent that it negatively impacts it, keeping in mind that reliability and validity issues remain, which will be addressed in the discussion later on.

It is worth noting that the in-game questionnaire does show intrusiveness to some players and does not represent an authentic experience of how the game will run correctly, as we previously stated that continuous surveys could have an adverse effect on the player experience. In our case, the in-game questionnaire had the effect of forcing participants to conduct an in-depth reflection on their choices in the game, which either made it tiresome or made them better at the game as we forced them to consider their actions before acting. Regardless of the case, this does not represent how the game will authentically be played and therefore one should be aware that using such systems does not give an accurate representation of the gameplay experience.

Following this, we will continue to discuss the in-game questionnaire and how we would like to improve it, both for general future use and for our specific case study.

8.2 Photogrammetry: Sub-research question

What impact does photogrammetry have on the narrative experience both from a production and an artistic perspective? - sub-research question, see Research question 2.7 for further details.

In order to discuss the sub-research question, it is necessary to address two different areas. Firstly, the impact of photogrammetry's artistic impact on the experience. In this case, we have participants' feedback on which to base our claims. While for the production aspect of things, we would have to speculate based on our analysis and personal experience working with the technology, as well as what we think we could have done better in hindsight to prevent issues that occurred.

As we noted in Photogrammetry: sub-research question 7.10, the only real issue we were able to determine from participants regarding the photogrammetric art style was that some areas were visually difficult to divine what the object was. At an artistic level, we chose a fractured and somewhat blurry visual style to convey the sense of damaged memories, however, it was not intended for us to have unintelligible objects that could only be interpreted by reading the actual text that illustrated the game. Ultimately, this results in some difficulties from a production perspective. For convenience and efficiency reasons, we decided to utilize the Ipad Pro 12 we had with the photogrammetric software, Polycam. This strategy, however, came at a significant cost in the form of the poor quality of the iPad's camera. When it comes to photogrammetry, the quality of the camera and lighting is vital to the quality of the output. Despite knowing this would be a risk, we proceeded ahead because of the pros it had for our particular situation. One game that we could have further explored for valuable lessons on how to use photogrammetry is "The Signifier"[Playmestudio 2020a] which was released back in October 2020. During this game, the player explores other people's memories in an attempt to unravel the truth. The game follows similar themes to ours, employing the inconsistencies of photogrammetry similarly to our design. See Figure 64 for visuals on that particular game. While there is no immediate information regarding their photogrammetry production processes,

if time permits, we could have interviewed the developers on the team to gain insight into the methods they employed when creating their assets. However, this game has more evidence demonstrating that the themes we are dealing with for the game are compatible with the artistic approach we have chosen for the game utilizing photogrammetry, and the recent release of the game may suggest that this approach to using the technology is a new one and one that will perhaps be used in a more common manner in the future.

Figure 64: Examples of the photogrammetric style of the game "The Signifier" [Playmestudio 2020b]

We contend that our questions provided information that would assist us in answering the artistic aspect of our sub-research question. As such, we acknowledge that we are only scratching the surface of what could have been a much more deepening understanding of the photogrammetric approach we chose. In this thesis, we were mainly focused on the in-game questionnaire, but further research could definitely investigate the effects that photogrammetry has on the video game industry and provide more insight into where this technology has the potential to be useful and where it will not. In our case as a small team of individuals with almost no budget, photogrammetry worked almost perfectly for the purposes we intended. We achieved this primarily by acknowledging the inherent inconsistencies that photogrammetry can have, making it appear as an intentional part of the game, which enabled us to create 3D assets at a much faster rate than we could have done from scratch or even by employing photogrammetry with the intend of photo-realism. In the case of photogrammetry being used for photorealism, our approach would prove to be ineffective. In light of the limited number of images that PolyCam is able to process per scan as well as the lower quality of mobile cameras, the texture quality will not be comparable to what can be achieved if a professional camera is used with a professional photogrammetry program such as Agisoft Metashape.

8.3 In-game Questionnaire

The way we implemented the in game questionnaire, and set up the google forms worked quite well. We were able to obtain the data and process it. That said, one possible improvement to how we did it, could be to keep the focus on only the correct replacements, so the google form could output the data in a more concise and easily processed format, to avoid a lot of manual work that goes into sorting qualitative data. To accomplish this, one might create four sets of questions on the google form, specifically for the correct placements, so that these have already been extracted from the overall set of data for attempted corrections, and then have one set of questions for the collection of incorrect placements. The reason one could choose to do it this way is that this would allow google forms to place each of the correct placements into their own cells in excel, while still allowing for potentially unlimited responses to incorrect placements. Another reason for doing it this way would be, that if the questions were simply set

to not show on incorrect placements, then the participants would most likely pick up on this, and become aware that if the questions pop up, then they have made a correct placement. In order to solve this problem, the participant could be presented with the questionnaire prompt at random intervals when they perform incorrect placements as well. This would prevent the participants from identifying a pattern. Because of this, they would still be able to answer the questions as if they did not know if they were correct or not. A different improvement could be, to think of a way to limit the time participants spend answering the in game questions. From observing the test and its results, it is evident that participants would start being much more selective about what they try to place in 'memory holes'. Probably because they had to answer the questions every time discouraged them from trying different items, they would diverge from the actual game experience, where they would receive immediate feedback from the game when they were right or wrong and then be able to move on. This could be accomplished by including a button within the question user-interface that participants could press instead of answering the questions when they are guessing completely. Participants would be free to make guesses at their own discretion since they would only need to press this button. This potential improvement could also improve another aspect, namely the forced answering of questions. Eventually, as the participant answers the same questions again and again, as they place something in a memory hole, and so it is not unlikely that they will start putting less thought into their answers, and just putting something into their answers so that they could continue. In regards to forced answers, we see this as a benefit of the system, as opposed to the usual method of opening a google form behind the game and then requesting participants to switch between them. If the test is conducted solely by the participant without supervision, it is possible that something could go wrong. It will be possible with our system to make sure that the questions that need to be asked as part of the game experience will be asked and answered when they are supposed to, and as such provide a potentially better structure to tests that require these types of questions.

8.4 Validity and Reliability

During this particular case study and evaluation, there are many biases and confounding variables to consider. For one, we did not make this test in a controlled environment; the participants tested on their own computers using different hardware specifications and different external environments, such as at home or in an open area surrounded by other people and distractions. Observing that some of the participants got distracted by the environment around them, they might not have invested in the narrative as deeply as some other participants who were completely isolated. Luckily, we were able to provide headphones to all participants to minimize this bias.

When looking at our in-game formulated questions we primarily based it on [Larsen, Bruni, et al. 2019] and [Bruni and Baceviciute 2013]. Due to it being qualitative answers we can extract whether we did get answers that were linked to what the question intended. In general, there is a clear indication for improvements here. The most notable one, which we can directly analyze had issues was the question: *"What do you think the story is about so far?"*.

Some participants (6/23 during gameplay, and 8/23 after gameplay) answered their own personal opinions rather than what we intended to ask, which was what they could retell about the narrative as it appeared in their mind. Unfortunately, this issue was discovered during the final evaluation of the project and therefore the question could not be modified. For future purposes, we think the question will be more accurate if 'think' is changed to 'believe', result-

ing in the following question: *What do you **believe** the story is about so far.* Due to a lack of data supporting this, we assume that the word 'think' has led them to expect the question to focus on their personal opinions of the narrative, while 'believe' could potentially divert them from that understanding. Alternatively, we could more directly state that we wish them to retell the story so far. As it stands now, the validity of that question is arguably not optimal, but since it is qualitative we could easily disregard these results in the interpretive analysis.

When we look at the 'NEQ' only, we can observe that there is a large variation between some items, which should not be the case since they should all refer to the same construct. In this case, it may illustrate that the NEQ scale is not as accurate as it is being represented, but the more likely explanation is due to the relatively small sample size of 23 participants. Another possibility is that we may have incorrectly believed we could translate "NEQ," which is intended for film/television (non-interactive narratives), into a video game (interactive narratives). Although we are largely certain that this is not directly the case due to the strong parallels between the narrative structures of the video game we are reviewing and traditional films. The fundamental difference between the two is the interactive nature of a video game, and it can be argued that some items should be rewritten to reflect the interactive nature of a game. Consequently, most likely the fault lies with the sample size, and whether it is our specific evaluation or the measures that are at fault, it can be correctly assumed that the validity of our approach to using the NEQ is not substantial. In general, it is difficult to evaluate the opinions of a narrative without taking into consideration the extensive predispositions of each individual. This is another reason why we have presented the Likert scale as a separate item and coupled with the substantial amount of qualitative data we have obtained through the in-game questionnaire, we are in a position to claim that there was narrative engagement with the experience. As far as reliability, however, we cannot assure the same results under the same conditions due to the somewhat uncontrolled environment, nature of our sampling procedure as well as a low sample size. With the resources available, we decided to do a convenience sampling and as a result, found many participants to have some knowledge of the theory and practice involved in creating a videogame or biases due to a fair amount of them having a personal relationship with us. Consequently, we can see that the results could have appeared quite differently with larger sample size and a more general video game experience among participants.

9 Conclusion

In conclusion, this thesis provides a detailed look at the development of a video game using photogrammetry and explores the use of continuous evaluation methods to provide usable player insights into the game's narrative intelligibility. This thesis case study is centered around the video game, Mind Diver, which we co-created with other students from various universities and various backgrounds. Mind Diver is a video-game set up as an ocean in the mind, with memories frozen inside large bubbles of air. The player must explore memories and fix the 'memory hole', which is done by picking up objects and placing them in the correct spot. The goal of the game is to to piece together a story of the person whose mind they are exploring. This led to our main research question:

Will using our implemented in-game evaluation tool provide usable player insights within narrative intelligibility, without being intrusive and negatively impacting narrative engagement?

Narrative intelligibility is the ability of a story to make sense to its audience from the authors intended narrative. This is important to evaluate because it ensures that the story is understandable and that the audience is able to follow the plot as intended by the author. We therefore set to design an in-game feedback tool with questions that would give insight to the narrative intelligibility of a video-game. The way we implemented the in-game feedback tool, and set up the google forms worked quite well. Our in-game questions helped us identify areas where the player would have difficulty understanding how the story was connected to the puzzles the player had to solve which provided feedback to help improve the game. That said, one possible improvement to how we did it, could be to keep the focus on only the correct replacements, so the google form could output the data in a more concise and easily processed format, to avoid a lot of manual work that goes into sorting qualitative data. From that, the bigger contribution of the in-game feedback tool is the variety of use cases it can be used for to any researchers that want streamlined continuous qualitative and quantitative data during a gameplay experience.

With regards to the intrusiveness of the in-game feedback tool, our data does not seem to indicate that it has a severely negative impact on the narrative engagement. However, the constant requirement to the participant to answer questions while playing the game seem to be indicating the players still thought that the constant in-game questions either felt intrusive and annoying or for others it helped them pause and reflect on their choices in game. For one it shows that even though it was intrusive it did not hinder the participant to follow along in the narrative, however it still shows that the continuous approach does not represent an authentic gameplay experience as it is still very clear for the participant that they are testing a certain aspect of a game, which in our case-study, forces them to reflect on the narrative of the game.

Additionally, we discussed our experience using photogrammetry (a method for extracting 3D assets from 2D images in the real world), as it played a crucial role in the development of our case study. This is why we focused on the following subtopic:

What impact does photogrammetry have on the narrative experience both from a production and an artistic perspective?

From an artistic point of view the photogrammetric art style had a clear, positive influence on the game and especially its narrative. While some minor details were lacking in the 3D

assets, this did not majorly affect the gameplay experience for the majority of participants. By embracing the imperfections of photogrammetry we managed to create a unique and visually pleasing art style that enhanced the game's theme of exploring broken memories. From a production point of view, we believe that using the iPad Pro 12 was a good decision for our purposes. Despite the lower quality of the camera, the photogrammetric results were still usable and fitted the theme of the game. Additionally, the convenience of cloud processing with Polycam was a major benefit to a production with limited time and resources. However, if the aim of using photogrammetry is for photo-realism, then our approach would not be ideal and instead one should use a professional camera and software to create 3D assets, as the details and texture quality would be much better than if they were to use a mobile device and Polycam.

With all this being said, due to the small sample size, relying solely on convenience sampling and a somewhat uncontrolled environment, the study's validity and reliability is limited.

10 Future work

It is our aim, for future work, to share our in-game feedback tool as a unity package that can be used by other researchers and developers to conduct their own studies within interactive digital media.

As a result of designing the system with a core system and integrating on top of it, it was intended that the core system would be able to be distributed. It is possible to host the core system on a distribution platform such as GitHub without any documentation. However, it will be difficult to integrate the core system without this documentation. In the future, it would be ideal if proper documentation was created for the core system and if it were even packaged into a Unity package that could be included in the Unity Asset Store.

References

Csikszentmihalyi, Mihaly (Apr. 2014). *Flow and the foundations of positive psychology: The collected works of Mihaly Csikszentmihalyi*, pp. 1–298. ISBN: 978-94-017-9087-1. DOI: 10.1007/978-94-017-9088-8.

Schønau-Fog, Henrik and Thomas Bjørner (2012). "Sure, I Would Like to Continue: A Method for Mapping the Experience of Engagement in Video Games". In: *Bulletin of Science, Technology & Society* 32.5. url: https://doi.org/10.1177/0270467612469068, pp. 405–412. DOI: 10.1177/0270467612469068.

Przybylski, Andrew K., C. Scott Rigby, and Richard M. Ryan (2010). "A Motivational Model of Video Game Engagement". In: *Review of General Psychology* 14.2. url: https://doi.org/10.1037/a0019440, pp. 154–166. DOI: 10.1037/a0019440. eprint: https://doi.org/10.1037/a0019440.

Abbasi, Amir, Ding Hooi Ting, and Helmut Hlavacs (Jan. 2017). "Engagement in Games: Developing an Instrument to Measure Consumer Videogame Engagement and Its Validation". In: *International Journal of Computer Games Technology* 2017, pp. 1–10. DOI: 10.1155/2017/7363925.

Schoenau-Fog, Henrik (2011). "Hooked! – Evaluating Engagement as Continuation Desire in Interactive Narratives". In: *Interactive Storytelling*. Ed. by Mei Si et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 219–230. ISBN: 978-3-642-25289-1.

Bruni, Luis Emilio and Sarune Baceviciute (2013). "Narrative Intelligibility and Closure in Interactive Systems". In: *Interactive Storytelling*. Ed. by Hartmut Koenitz et al. Cham: Springer International Publishing, pp. 13–24. ISBN: 978-3-319-02756-2.

Statham, Nataska (2020). "Use of Photogrammetry in Video Games: A Historical Overview". In: *Games and Culture* 15.3, pp. 289–307. DOI: 10.1177/1555412018786415.

Poznanski, Andrzej (2014). "Visual Revolution of The Vanishing of Ethan Carter". In: URL: https://www.theastronauts.com/2014/03/visual-revolution-vanishing-ethan-carter/. Last accessed 07/02/2022.

ISPRS-Foundation (2016). *ISPRS Brochure*. URL: https://www.isprs.org/publications/brochure/ISPRS_brochure.pdf.

Luhmann, Thomas et al. (2019). *Close-Range Photogrammetry and 3D Imaging*. ISBN: 9783110607253. DOI: 10.1515/9783110607253.

NOAA (2012). *"Lidar 101: An Introduction to Lidar Technology, Data, and Applications*. URL: https://coast.noaa.gov/data/digitalcoast/pdf/lidar-101.pdf.

Foster, Shaun and David Halbstein (Jan. 2014). *Integrating 3D Modeling, Photogrammetry and Design*. ISBN: 978-1-4471-6329-9. DOI: 10.1007/978-1-4471-6329-9.

Unity (2022). *Unity Documentation Mesh.indexFormat*. URL: https://docs.unity3d.com/ScriptReference/Mesh-indexFormat.html. Last accessed 31/03/2022.

Übel, Max von (2021). "Best Photogrammetry Software of 2021 (Some are Free)". In: URL: https://all3dp.com/1/best-photogrammetry-software/. Last accessed 04/02/2022.

Polycam (2020). *Polycam*. URL: https://poly.cam/. Last accessed 05/03/2022.

Fuhrmann, Simon, Fabian Langguth, and Michael Goesele (n.d.). *Multi-View Environment*. URL: https://www.gcc.tu-darmstadt.de/home/proj/mve/. Last accessed 04/02/2022.

Agisoft (n.d.). *Metashape*. URL: https://www.agisoft.com/. Last accessed 04/02/2022.

Reality, Capturing (n.d.). *Reality Capture*. URL: https://www.capturingreality.com/. Last accessed 04/02/2022.

Fuhrmann, Simon, Fabian Langguth, Michael Goesele, et al. (2022). *Multi-View Environment GitHub*. URL: https://github.com/simonfuhrmann/mve. Last accessed 25/05/22.

Francis, Tom (2019). *Design Talk: Information Games*. URL: https://www.youtube.com/watch?v=LFMEmS4PN00&t. Last accessed 07/03/2022.

Barlow, Sam (2015). *Her Story Game*. URL: http://www.herstorygame.com/. Last accessed 07/03/2022.

Pope, Lucas (2018). *Return of the Obra Dinn*. URL: https://obradinn.com/. Last accessed 07/03/2022.

Annapurna-Interactive (2019). *Outer Wilds*. URL: https://annapurnainteractive.com/games/outer-wilds. Last accessed 07/03/2022.

Course-Hero (2018). *Motivation*. URL: https://www.coursehero.com/study-guides/wsu-sandbox/motivation/.

Blacow, Glenn (1980). *Aspects of Adventure Gaming*. URL: https://www.darkshire.net/jhkim/rpg/theory/models/blacow.html. Last accessed 12/03/2022.

Bartle, Richard (1996). *Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs*. URL: https://mud.co.uk/richard/hcds.htm. Last accessed 12/03/2022.

Laws, Robin D. (2002). *Robin's Laws Of Good Game Mastering*. URL: https://i.4pcdn.org/tg/1368190742811.pdf. Last accessed 12/03/2022.

Yee, Nick (2005). *A Model of Player Motivations*. URL: http://www.nickyee.com/daedalus/archives/print/001298.php. Last accessed 12/03/2022.

Yee, Nick and Nicolas Ducheneaut (2019a). *Gamer Motivation Model*. URL: https://quanticfoundry.com/wp-content/uploads/2019/04/Gamer-Motivation-Model-Reference.pdf. Last accessed 12/03/2022.

– (2019b). *A Deep Dive Into The 12 Motivations*. URL: https://quanticfoundry.com/wp-content/uploads/2019/03/Quantic-Foundry-GDC-2019-Talk.pdf. Last accessed 12/03/2022.

– (2020). *Player Segments Based On Gamer Motivations*. URL: https://quanticfoundry.com/wp-content/uploads/2020/08/9-Player-Segments-GDC-2020-Excerpt.pdf. Last accessed 12/03/2022.

Ensslin, Astrid (May 2015). "Video Games as Unnatural Narratives". In.

Toolan, M. (Dec. 2006). "Narrative: Linguistic and Structural Theories". In: pp. 459–473. ISBN: 9780080448541. DOI: 10.1016/B0-08-044854-2/00528-9.

Herman, David (May 2013). "Narrative Theory and the Sciences of Mind". In: *Literature Compass* 10. DOI: 10.1111/lic3.12062.

Larsen, Bjarke Alexander and Henrik Schoenau-Fog (2016). "The Narrative Quality of Game Mechanics". In: *Interactive Storytelling*. Ed. by Frank Nack and Andrew S. Gordon. Cham: Springer International Publishing, pp. 61–72. ISBN: 978-3-319-48279-8.

Ryan, M.L., Johns Hopkins University Press, and ProQuest (Firm) (2001). *Narrative as Virtual Reality: Immersion and Interactivity in Literature and Electronic Media*. Parallax (Baltimore, Md.) Johns Hopkins University Press. ISBN: 9780801864872. URL: https://books.google.dk/books?id=cjAWAQAAIAAJ.

Busselle, Rick and Helena Bilandzic (2009). "Measuring Narrative Engagement". In: *Media Psychology* 12.4, pp. 321–347. DOI: 10.1080/15213260903287259. eprint: https://doi.org/10.1080/15213260903287259. URL: https://doi.org/10.1080/15213260903287259.

Nielsen, Thomas Lund and Eoin Ivan Rafferty (2020). *Embedded Narratives in Procedurally Generated Environments*. eng.

Christy, Katheryn R. and Jesse Fox (2016). "Transportability and Presence as Predictors of Avatar Identification Within Narrative Video Games". In: *Cyberpsychology, Behavior, and Social Networking* 19.4. PMID: 26919032, pp. 283–287. DOI: 10.1089/cyber.2015.0474. eprint: https://doi.org/10.1089/cyber.2015.0474. URL: https://doi.org/10.1089/cyber.2015.0474.

Richardson, Daniel C. et al. (2018). "Measuring narrative engagement: The heart tells the story". In: *bioRxiv*. DOI: 10.1101/351148. eprint: https://www.biorxiv.org/content/early/2018/09/13/351148.full.pdf. URL: https://www.biorxiv.org/content/early/2018/09/13/351148.

Brockmyer, Jeanne H. et al. (2009). "The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing". In: *Journal of Experimental Social Psychology* 45.4, pp. 624–634. ISSN: 0022-1031. DOI: https://doi.org/10.1016/j.jesp.2009.02.016. URL: https://www.sciencedirect.com/science/article/pii/S0022103109000444.

Larsen, Bjarke Alexander, Luis Emilio Bruni, and Henrik Schoenau-Fog (2019). "The Story We Cannot See: On How a Retelling Relates to Its Afterstory". eng. In: *Interactive Storytelling*. Vol. 11869. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 190–203. ISBN: 9783030338930.

Li, Hairong, Steven M. Edwards, and Joo-Hyun Lee (2002). "Measuring the Intrusiveness of Advertisements: Scale Development and Validation". In: *Journal of Advertising* 31.2, pp. 37–47. DOI: 10.1080/00913367.2002.10673665. eprint: https://doi.org/10.1080/00913367.2002.10673665. URL: https://doi.org/10.1080/00913367.2002.10673665.

Wehmeyer, Kai (2007). "Mobile ad intrusiveness-The effects of message type and situation". In: *BLED 2007 Proceedings*, p. 6.

Ying, Lou, Tor Korneliussen, and Kjell Grønhaug (2009). "The effect of ad value, ad placement and ad execution on the perceived intrusiveness of web advertisements". In: *International Journal of Advertising* 28.4, pp. 623–638. DOI: 10.2501/S0265048709200795. eprint: https://doi.org/10.2501/S0265048709200795. URL: https://doi.org/10.2501/S0265048709200795.

Bauer, Raymond A and Stephen A Greyser (1968). *Advertising in American: The Consumer View*. Boston, MA: Harvard University.

UnknownWorlds (2014). *Subnautica*. URL: https://unknownworlds.com/subnautica/. Last accessed 17/03/2022.

STOMT (n.d.). *STOMT: Player Feedback Platform for Game Developers & Publishers*. URL: https://www.stomt.com/. Last accessed 18/03/2022.

Jeremy, Hugh (2015). *Subnautica Feedback System*. URL: https://ubm-twvideo01.s3.amazonaws.com/o1/vault/gdc2015/presentations/Jeremy_Hugh_Tech_Toolbox.pdf. Last accessed 17/03/2022.

UnkownWorlds (2014). *Subnautica Feedback Website*. URL: https://subnautica.unknownworlds.com/. Last accessed 17/03/2022.

Jeremy, Hugh and GDC (2015). *Subnautica Feedback System Talk*. URL: https://www.youtube.com/watch?v=Urx7WQE6NY0. Last accessed 17/03/2022.

Zentner, Philipp (2018). *In-Game Feedback: Top 6 Advantages Of Doing It Right*. URL: https://www.stomt.com/blog/in-game-feedback-benefits/. Last accessed 18/03/2022.

Schukies, Daniel (2015). *Integrate STOMT with Unity*. URL: https://www.stomt.com/integrations/unity. Last accessed 18/03/2022.

Bjørner, Thomas (2015). *Qualitative methods for Consumer Research*. English. Denmark: Hans Reitzels Forlag. ISBN: 9788741258539.

Blender.org (2002). *Blender*. URL: https://www.blender.org/about/. Last accessed 03/05/2022.

– (2022). *Blender - Decimate Modifier*. URL: https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/decimate.html. Last accessed 03/05/2022.

Unity (2005). *Unity Real-time Development Platform*. URL: https://unity.com/. Last accessed 11/05/2022.

Danevičius, Edvinas (2022). *Scriptable Events*. URL: https://github.com/chark/scriptable-events. Last accessed 29/04/2022.

Playmestudio (2020a). *The Signifier*. URL: https://www.thesignifier.com/. Last accessed 17/05/2022.

– (2020b). *The Signifier Director's Cut*. URL: https://store.steampowered.com/app/1082930/The_Signifier_Directors_Cut/. Last accessed 17/05/2022.

11 Appendix

A Data from in-game questionnaire

For the data from the in-game questionnaire please refer to the excel sheet which can be found in the extras named "In_game_questionnaire_data.xlsx".

B Data from pre- and post-questionnaire

For the data from the pre- and post-questionnaire please refer to the excel sheet which can be found in the extras named "Pre__and_post_questionnaire_data.xlsx".

C Implementation Class Diagram

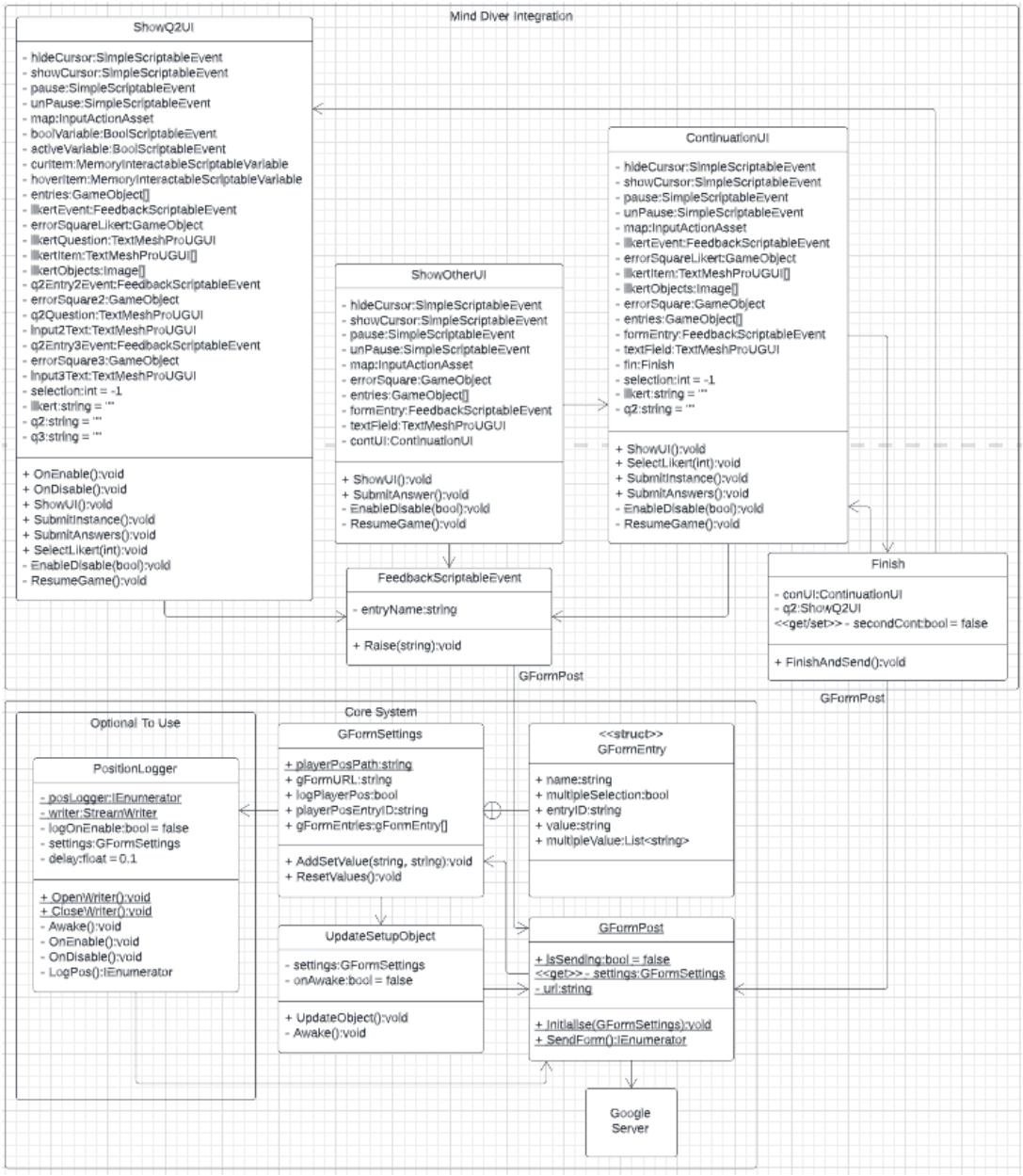
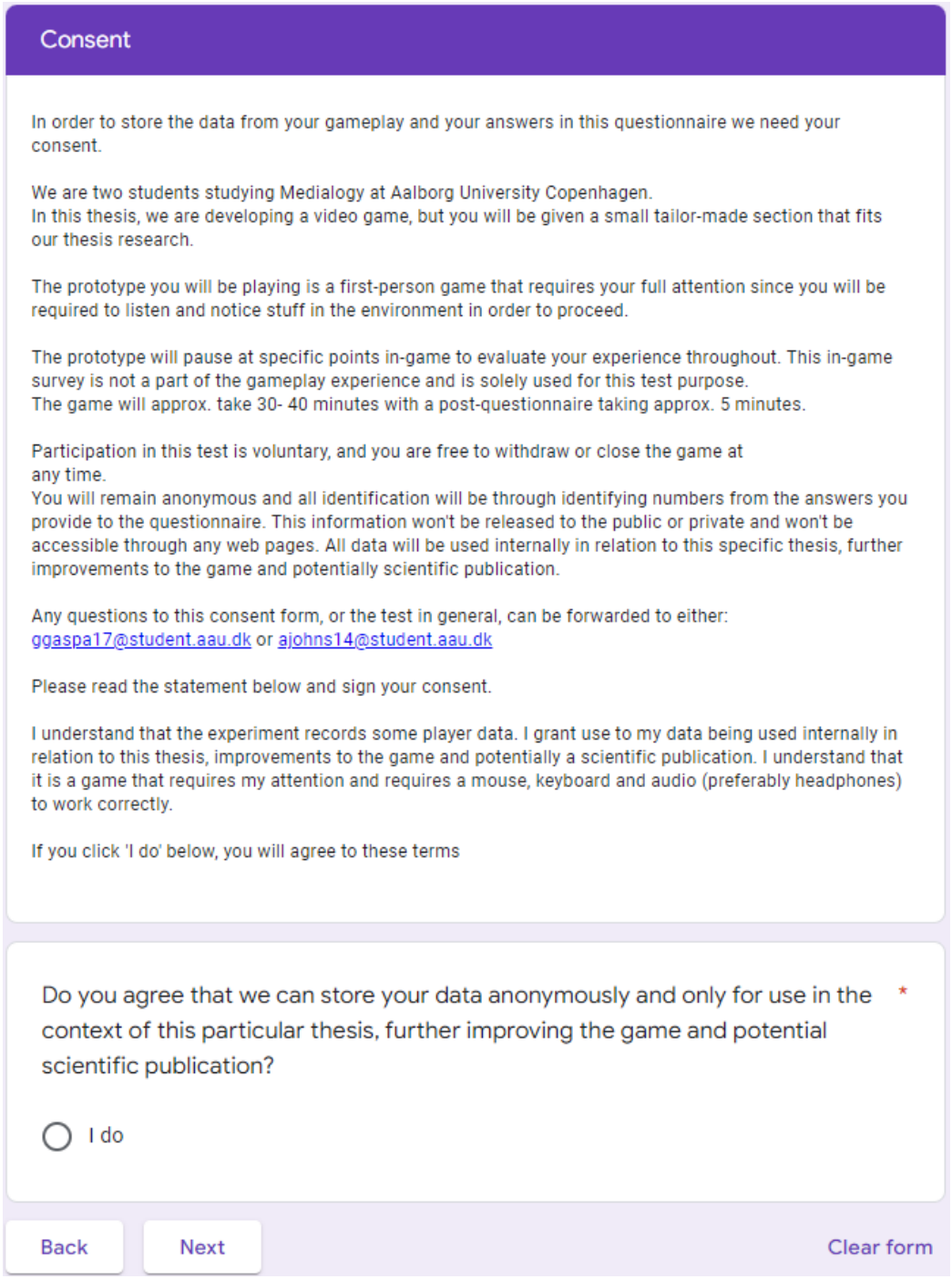


Figure 65: Figure showing the full class diagram

D Consent Form

The following is the consent form provided to each participant for the final evaluation.



Consent

In order to store the data from your gameplay and your answers in this questionnaire we need your consent.

We are two students studying Medialogy at Aalborg University Copenhagen.
In this thesis, we are developing a video game, but you will be given a small tailor-made section that fits our thesis research.

The prototype you will be playing is a first-person game that requires your full attention since you will be required to listen and notice stuff in the environment in order to proceed.

The prototype will pause at specific points in-game to evaluate your experience throughout. This in-game survey is not a part of the gameplay experience and is solely used for this test purpose.
The game will approx. take 30- 40 minutes with a post-questionnaire taking approx. 5 minutes.

Participation in this test is voluntary, and you are free to withdraw or close the game at any time.
You will remain anonymous and all identification will be through identifying numbers from the answers you provide to the questionnaire. This information won't be released to the public or private and won't be accessible through any web pages. All data will be used internally in relation to this specific thesis, further improvements to the game and potentially scientific publication.

Any questions to this consent form, or the test in general, can be forwarded to either:
ggaspa17@student.aau.dk or ajohns14@student.aau.dk

Please read the statement below and sign your consent.

I understand that the experiment records some player data. I grant use to my data being used internally in relation to this thesis, improvements to the game and potentially a scientific publication. I understand that it is a game that requires my attention and requires a mouse, keyboard and audio (preferably headphones) to work correctly.

If you click 'I do' below, you will agree to these terms

Do you agree that we can store your data anonymously and only for use in the context of this particular thesis, further improving the game and potential scientific publication? *

I do

Back Next Clear form

E Final Evaluation: Pre- and Post-Questionnaire

E.1 Demographics / Background Information

This section contains general background information we would want for each participant. Participants would answer these questions prior to playing the game.

Background Information

Before playing the game, we would like you to answer a couple of general questions.

Name: (used for identifying your answers with your in-game data. If you prefer using an ID other than your name you can do so aswell. We just need you to write the same here and in-game *

Your answer

What is your email address? (Only if you would like us to contact you again for a follow-up interview)

Your answer

How old are you? *

Your answer

Gender *

- ◯ Female
- ◯ Male
- ◯ Non-Binary
- ◯ Prefer not to say

Which country do you currently live in? *

Your answer

In a typical week, about how many days do you spend at least 30 minutes playing a video game? *

- 0-1 days
- 2-3 days
- 4-5 days
- 6-7 days

Would you consider yourself to be a ... *

- Casual Gamer (e.i. you dabble in games but in short sessions or infrequently)
- Core / Mid-Core Gamer (i.e. you regularly play video games but are not super serious or competitive)
- Hardcore Gamer (i.e. you are a dedicated gamer and play seriously or competitively)
- I do not play videogames

Which of these platforms do you usually play games on? *

- [ ] Console (Xbox, Playstation, Nintendo switch (Docked w/ TV))
- [ ] PC/Mac (Desktop, Laptop)
- [ ] Smartphone/Tablet (IOS, Android, etc.)
- [ ] VR (Oculus Quest/Rift, Sony Playstation VR, etc.)
- [ ] None

What games have you played recently that you enjoyed (if any)? *

Your answer

How much have you played the following games? *

	I have never heard about the game.	I have heard about the game, but I have never played it.	I played a portion of the game, but I did not complete it	I have completed the game
Return of the Obra Dinn	○	○	○	○
Outer Wilds	○	○	○	○
Her Story	○	○	○	○
Telling Lies	○	○	○	○
Detective: The Modern Board Game (Board Game)	○	○	○	○
To the Moon	○	○	○	○
The Signifier	○	○	○	○
Twelve Minutes	○	○	○	○
Any Sherlock Holmes game by Frogwares	○	○	○	○
Painscreek Killings	○	○	○	○

Do you work in the video game industry? *

- ◯ Yes
- ◯ I study video games in academia
- ◯ No

Please indicate below the extent to which you agree or disagree with this sentence: “I want to begin playing the game” *

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

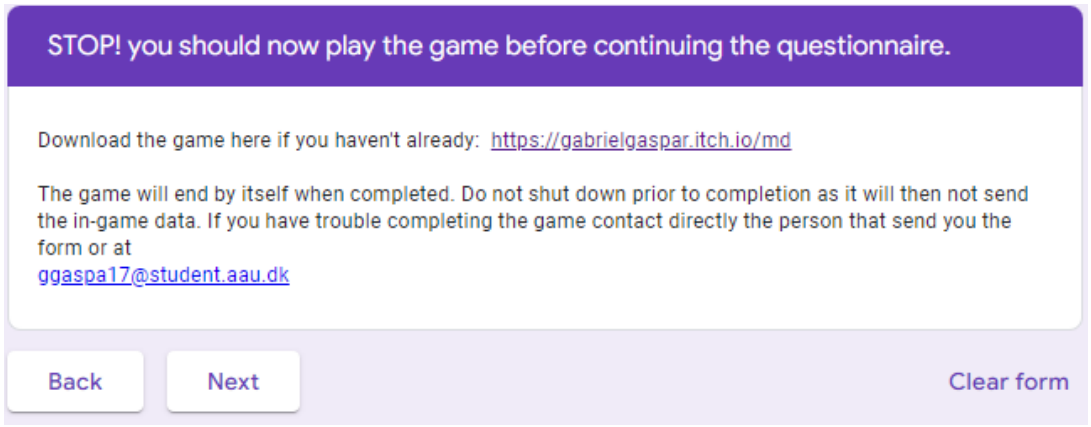
What makes you want/not want to begin playing the game? *

Your answer

You will be playing a prototype for a game in development called 'Mind Diver' - If you have any previous knowledge of 'Mind Diver', please share here (gameplay, story, etc.).

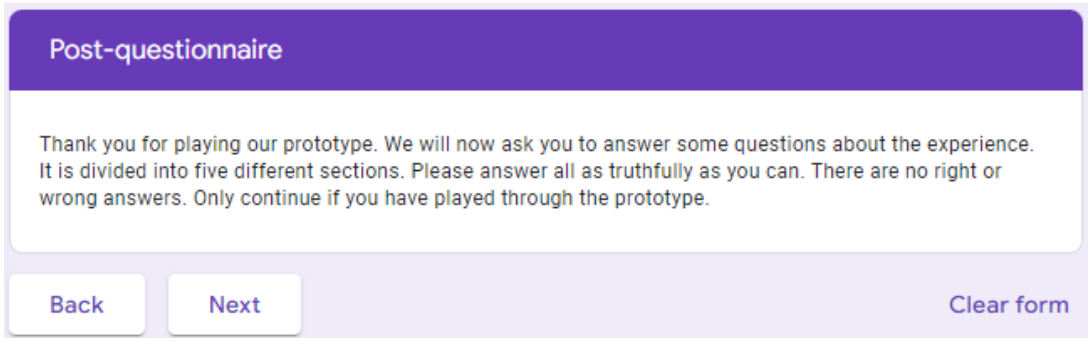
Your answer

Back Next Clear form



E.2 Narrative Engagement Questionnaire

The following contains how we presented the 'Narrative Engagement Questionnaire', which would be the first series of questions the participants would answer after playing through the video game.



Game Experience

In this section we ask you to rate the game experience based on how much you agree with 9 different statements, all on a scale from 1 to 7 where 1 is strongly disagree and 7 is strongly agree. There are no right or wrong answers.

At points, I had a hard time making sense of what was going on in the game. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

My understanding of the characters are unclear *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I had a hard time recognizing the thread of the story *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I found my mind wandering while playing the game *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

While the game was on I found myself thinking about other things *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I had a hard time keeping my mind on the game *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

During the game, my body was in the room, but my mind was inside the world created by the story *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The game created a new world, and then that world suddenly disappeared when the game ended *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

At times during the game, the story world was closer to me than the real world *

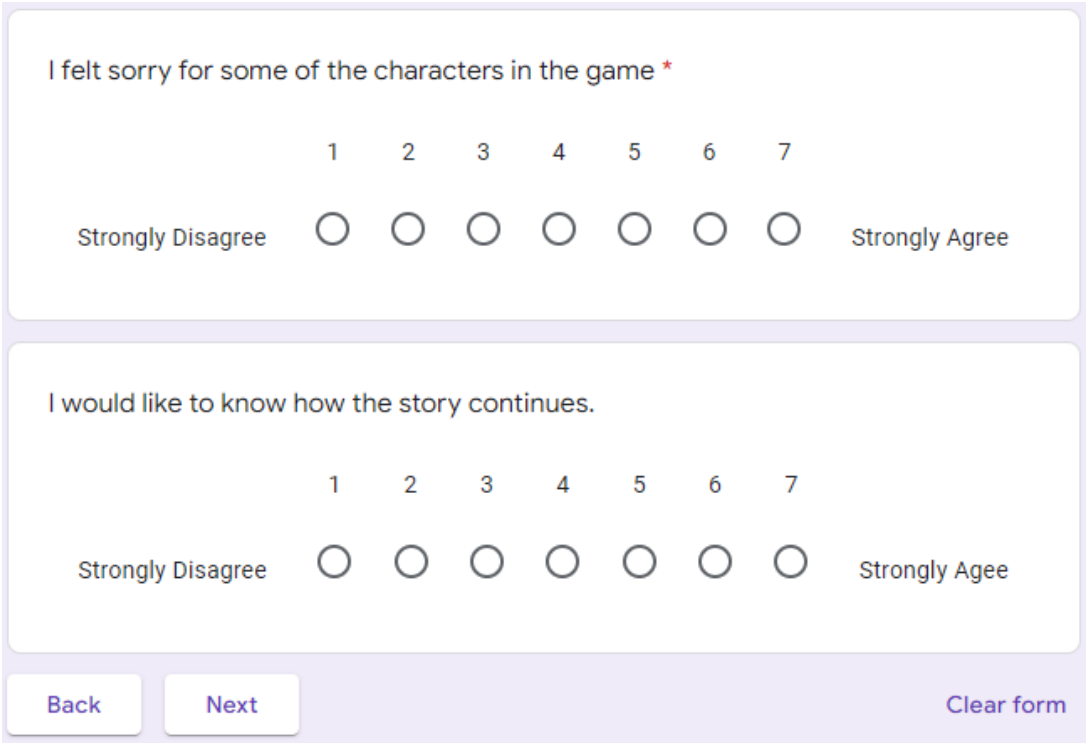
	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

The story affected me emotionally *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

During the game, when a main character succeeded, I felt happy, and when they suffered in some way, I felt sad *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree



E.3 Narrative Tension Questionnaire

The following contains how we presented the 'Narrative Tension Questionnaire', which would be the second series of questions the participants would answer after playing through the video game.

Narrative Experience

In this part we, similar to the first part, ask you indicate how you feel about a few different statements. This time about your experience of the narrative in the game. Please indicate to what degree you agree in the following statements where 1 is strongly disagree and 7 is strongly agree

I did not experience any narrative progression *

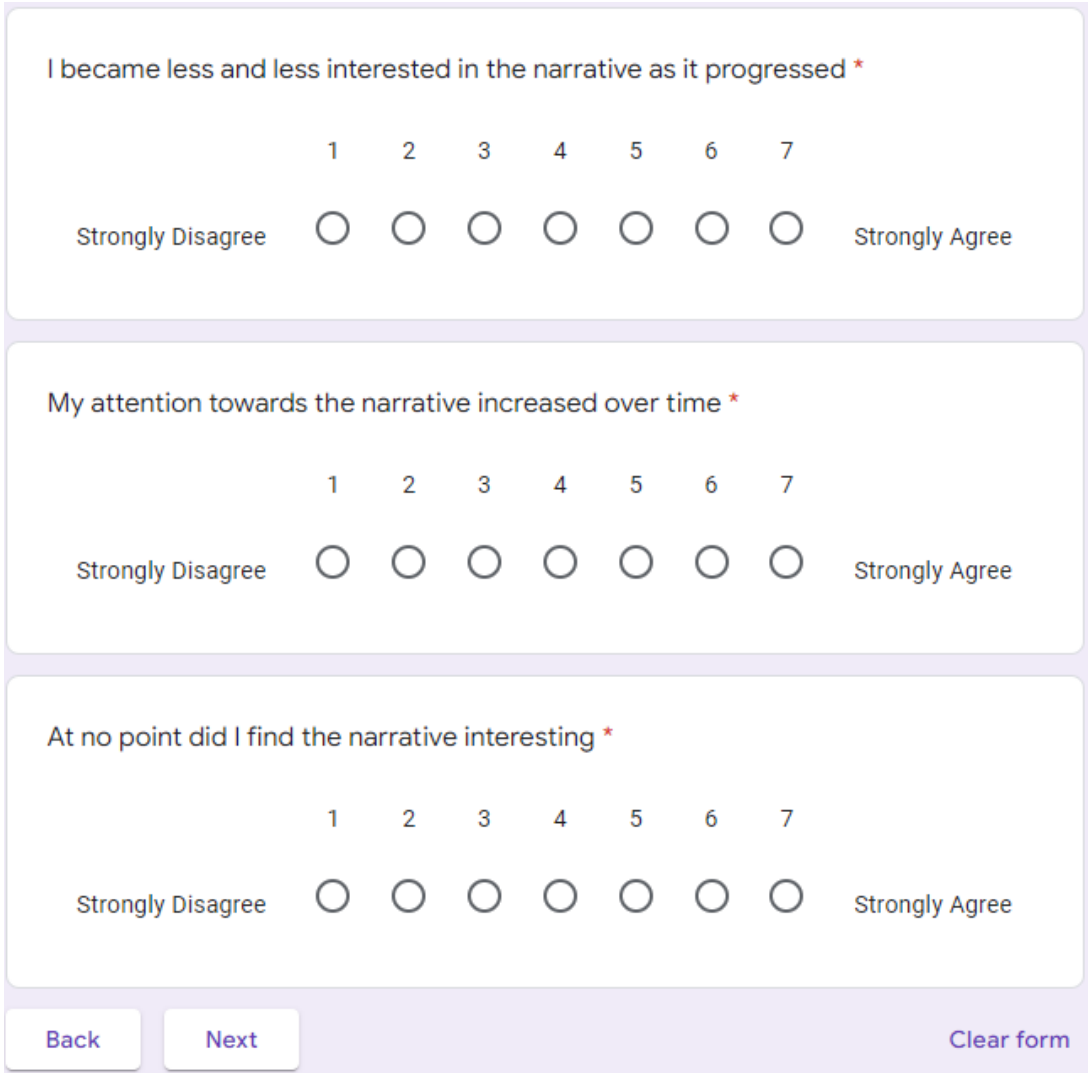
	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

At the end, I knew more about the narrative than when I began *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I was more interested in the narrative at the beginning than the end *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree



E.4 The Visual Experience

The following contains how we presented the third part of the post-questionnaire, which is a series of questions that would help us answer our sub-research question regarding photogrammetry.

The Visual Experience

In this part we would like to get your opinion on the visual expression of the scenes, specifically the art style of the memories using a method called 'photogrammetry'.

Do you know what photogrammetry is? (before playing this game) *

- ◯ No, I don't know what it is.
- ◯ I have seen something like this before, but did not know it was called photogrammetry.
- ◯ Yes, I know what photogrammetry is.
- ◯ Yes, I know what photogrammetry is and have worked with it.
- ◯ Other: _____

Art style example of a memory

I enjoyed the art style of the memories *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

If you can, please elaborate on your answer to the previous statement

Your answer

The art style of the memories complimented the narrative. *

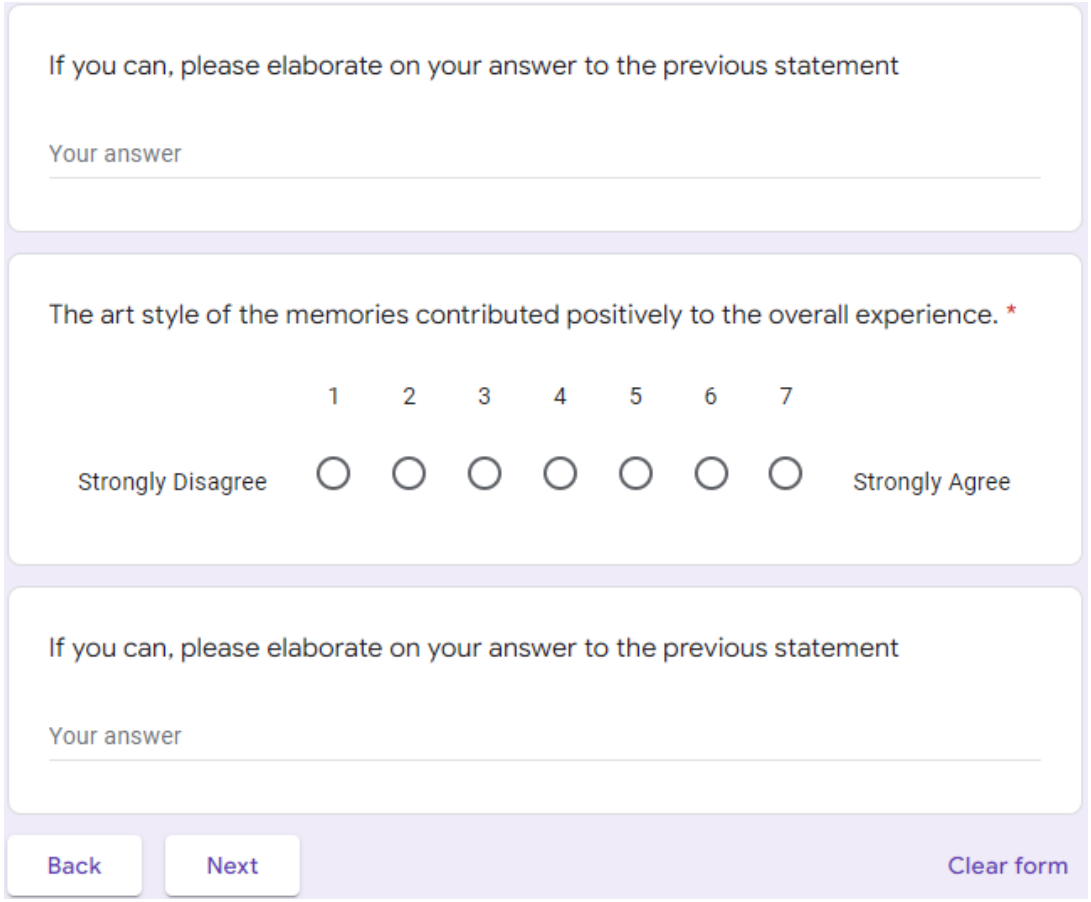
	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

If you can, please elaborate on your answer to the previous statement

Your answer

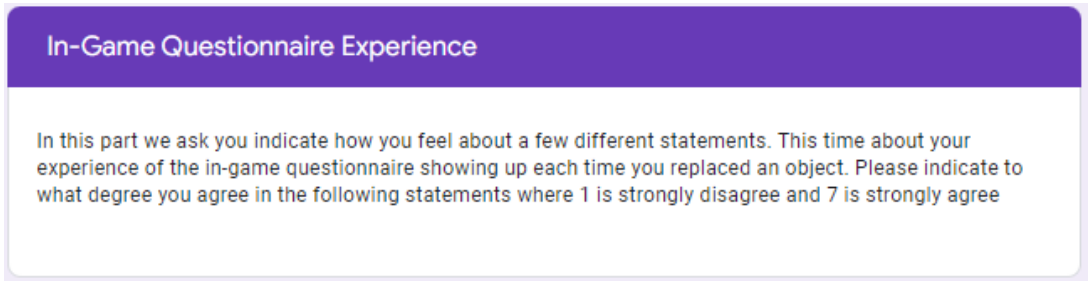
The art style of the memories complimented the gameplay. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree



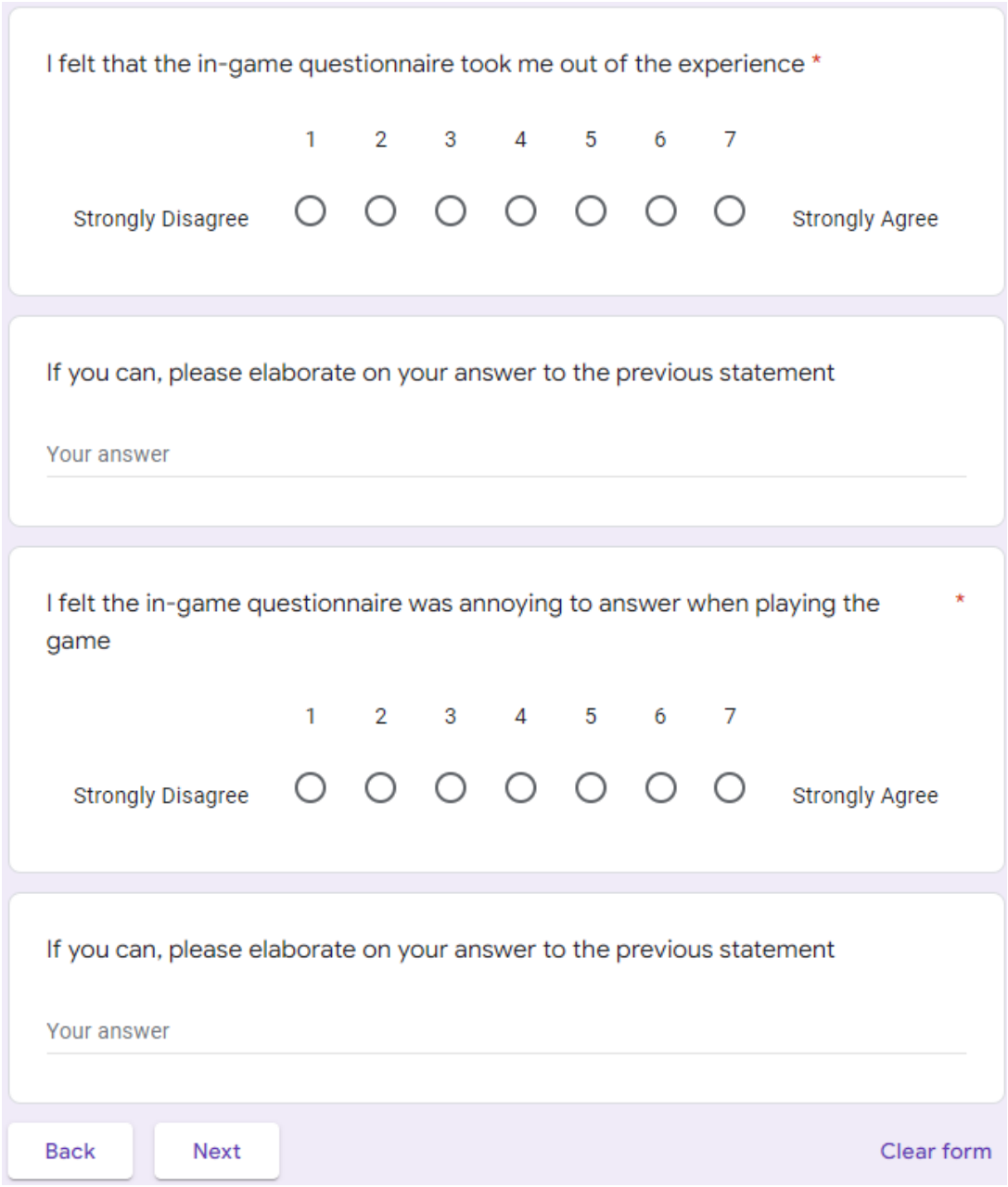
E.5 In-Game Questionnaire Experience

The following contains how we presented the fourth part of the post-questionnaire. This part contains the 'Instrusivenes Scale' aswell as two additional questions to help us evaluate the intrusiveness of the in-game questionnaire.



When the in-game questionnaire was shown, I thought it was *

	I strongly disagree	I moderately disagree	I disagree a little	I neither agree or disagree	I agree a little	I moderately agree	I strongly agree
Distracting (to divert the attention)	◯	◯	◯	◯	◯	◯	◯
Disturbing (to confuse a quiet, constant state or a calm, continuous flow)	◯	◯	◯	◯	◯	◯	◯
Forced (to do something involuntarily)	◯	◯	◯	◯	◯	◯	◯
Instrusive (to cause disruption or annoyance through being unwelcome or uninvited)	◯	◯	◯	◯	◯	◯	◯
Invasive (to invade a person's thoughts or privacy)	◯	◯	◯	◯	◯	◯	◯
Obtrusive (to be noticeable or prominent in an unwelcome and unpleasant way)	◯	◯	◯	◯	◯	◯	◯



E.6 The Virtual Environment

The following contains how we presented the last part of the post-questionnaire. This part contains questions regarding the overall gameplay experience including technical issues that might have occurred.

Virtual Environment

This is the last part where we ask you to answer some general questions about the experience.

Any technical errors concerning the experience? *

Your answer

What was the worst thing about the game (excluding the in-game survey)? *

Your answer

What was the best thing about the game (excluding the in-game survey)? *

Your answer

Please give us your overall thoughts about the game (excluding the in-game survey) *

Your answer

I found it hard to navigate while swimming *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I found the game hard to complete *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

I found it easy to solve the puzzles (getting the correct object at the right place) *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Would you play a full version of this game? (Why/Why Not) *

Your answer

Back Next Clear form

Thank you!

Your help is valuable for our Master Thesis and is very much appreciated.

You only played an unfinished short section of this demo. The full demo will be released free on Steam with more content than what you just played and potential updates on what you played. If you are interested in playing the full demo, we would appreciate if you could wishlist 'Mind Diver' on Steam:
- https://store.steampowered.com/app/1945810/Mind_Diver/

You can also check out our socials for updates to when we will release the game:
- https://twitter.com/MindDiverGame
- https://www.facebook.com/minddivergame
- https://discord.gg/y4hgyb2S

Please remember to submit by clicking the button below!

If you have any questions to this test feel free to reach out to us at ggaspa@student.aau.dk or ajohns14@student.aau.dk

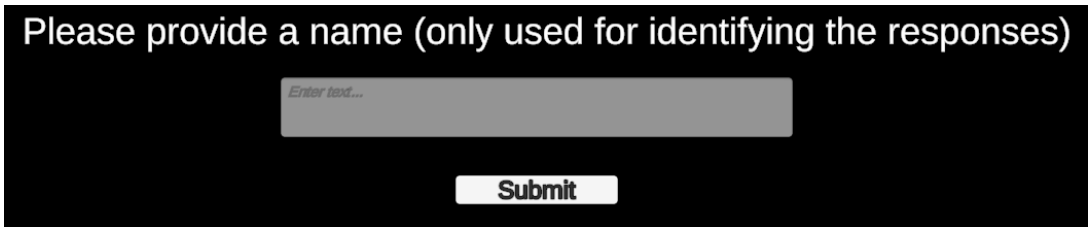
Andreas Johnsen & Gabriel Gaspar

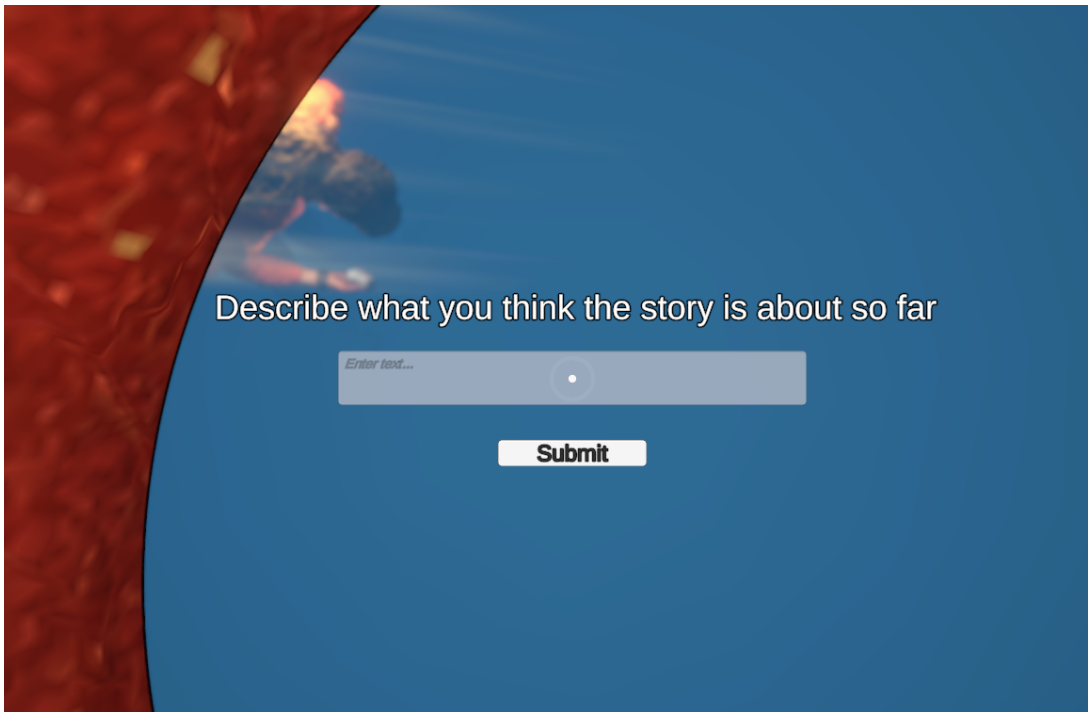
Medialogy Master Students,
Aalborg University, Copenhagen

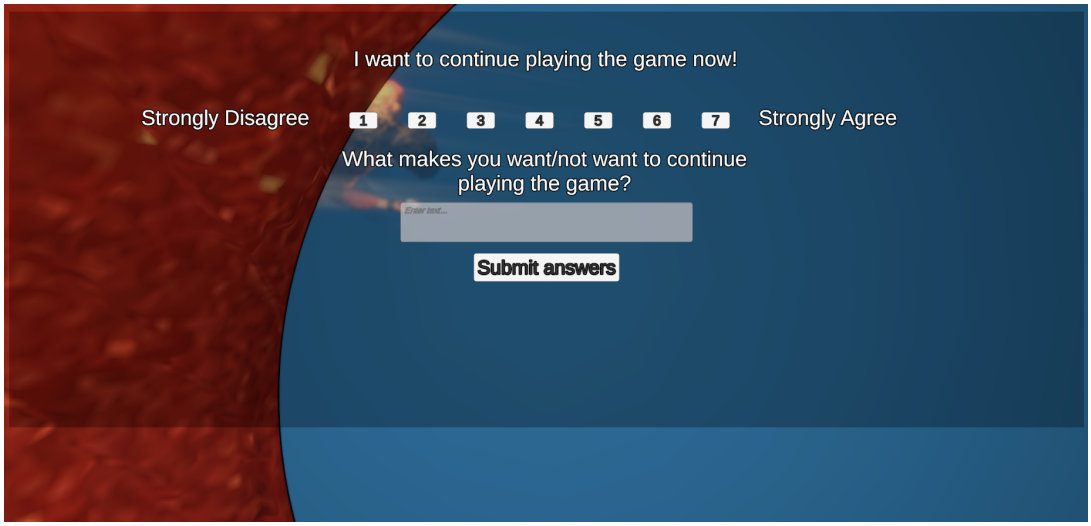
Back Submit Clear form

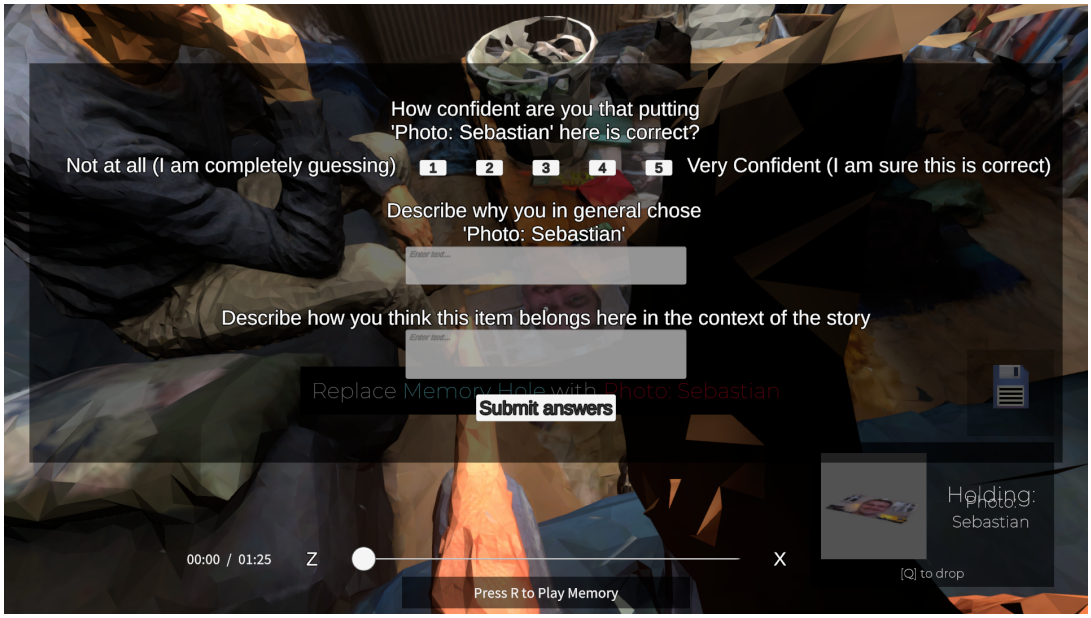
F In-Game Questionnaire

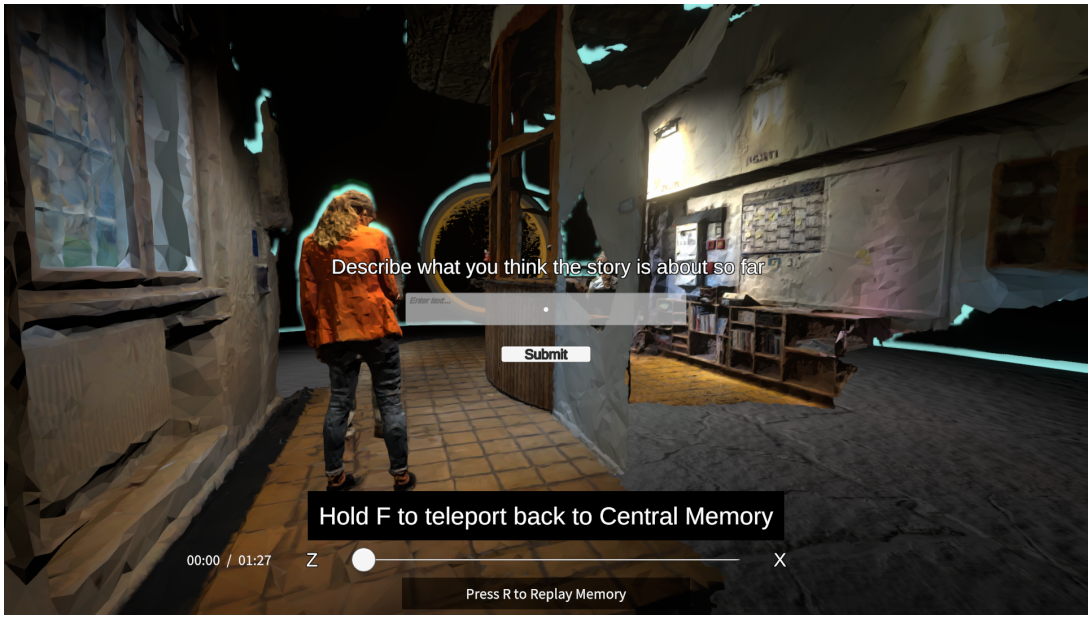
The following contains how we presented the in-game questions shown in a chronological order from the point of view of the player.

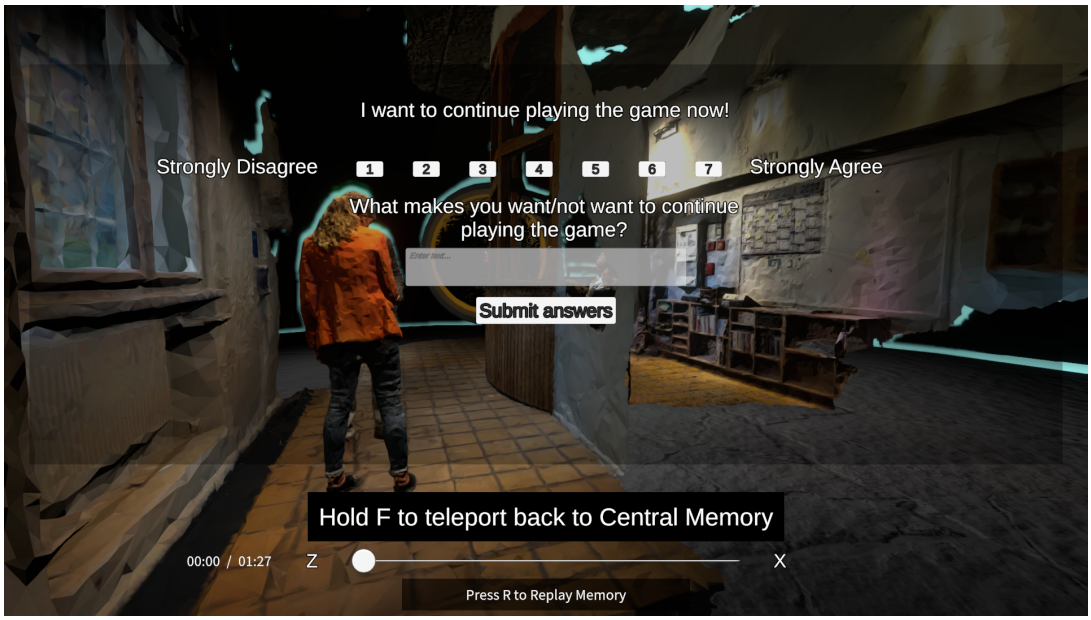





I want to continue playing the game now!
Strongly Disagree
1
2
3
4
5
6
7
Strongly Agree
What makes you want/not want to continue playing the game?
Enter text...
Submit answers


How confident are you that putting 'Photo: Sebastian' here is correct?
Not at all (I am completely guessing)
1
2
3
4
5
Very Confident (I am sure this is correct)
Describe why you in general chose 'Photo: Sebastian'
Enter text...
Describe how you think this item belongs here in the context of the story
Enter text...
Replace Memory Hole with Photo: Sebastian
Submit answers
Holding:
Photo: Sebastian
[Q] to drop
00:00 / 01:25
Z
X
Press R to Play Memory


Describe what you think the story is about so far
Enter text...
Submit
Hold F to teleport back to Central Memory
00:00 / 01:27
Z
X
Press R to Replay Memory


I want to continue playing the game now!
Strongly Disagree
1
2
3
4
5
6
7
Strongly Agree
What makes you want/not want to continue playing the game?
Enter text...
Submit answers
Hold F to teleport back to Central Memory
00:00 / 01:27
Z
X
Press R to Replay Memory

G Early testing of Evaluating In-Game Feedback

What participant knew before test:

Remember from last time:
Met sebastian for first time and got really close.
went to the rooftop and got to know each other.

Right before participant replaces objects participant is asked:
- Why did you choose this item?
- How confident are you that your answer is correct? (1-5) (1 = completely guessing, 5 = Completely sure)

_____
Listens to first m1.1
Looks at the memory, reads the memory again.

Listens to second memory

going a bit back and forward between m1 and m2

listens to m3

Sebastian(m1) → ??(tourist shop) -(5) (10min)
Sebastian has black shoes on all the other memories.

Wanted to take the clock to change out with what lina is struggling with, but then got told that it's the central station clock.

Feels stuck.

thinks notes are relevant, but dont know which one to choose.
getting stuck by thinking the notes are something that should be placed somewhere else, instead of using it as informative clues.

remember platform 5 → lina struggling (1)
Stands out from the rest of the notes

Wanted to take coffee that was spilled on sebastian, faciliator said it is not an object but just something on sebastian.

suitcase(danish flag) → struggling Lina (2)
stands out, she has a handbag and you can easily have more than one bag/suitcase.

Coffee which Lina is holding → m2 holding (2)
coffee spill on sebastian so maybe from that

Seems lost.

Suitcase slovak stickers → Lina m2 (3)
slovak sticker on handbag in m1

RE875 →Under Lina m2 (3)

platform 5 → under lina (4)
Note from m1 about platform 5.

note platform 5 → lina holding m2 (3)
They are on platform 5.

Pausing the playthrough

Dont know what to do.

Suitcase slovak sticker → Lina (3)
Has this suitcase in other memories.

Brown wallet → lina m2 (2)
Something about this wallet, maybe she has gotten it.

Coffee “Black” → lina m2 (4)
Spilled coffee on him.

Lina → Old Woman 3.1(4)
Lina is on the train sebastian is sitting alone.

“So lina is missing a chair.”

Seat01 → m3 (3)
It is missing a suitcase, and lina is missing something to sit on.
Lina has maybe giving her spot to the old lady.

Seat37 → m3 (2)
Number two in the row here, just like her

seat45 → m3 (4)
That time she was there, 8.15, and that is the seatnumber

Brown wallet → sebastian (4)
He had that receipt in his wallet

Regret summary

Going out to some job, Lina has to interview a soldier. Which means she needs to travel some days.
Something is important for sebastian and there was something with his mom.
—-

Satisfying:
Seat number, going back from m1 and m2 and solve it in m3.

Take a whole room and change that.
Seems interesting.

Interested in changing the environment rather than the object.

Least satisfying:
Brown wallet, was to ez.
the wallet has significance, and the receipt is too obvious of a hint.

Participant is mostly a FPS gamer (Counterstrike and Garrys Mod)
Have completed Portal 2 and played a bit of LA NOIRE but stopped due to not understanding it at the time.

Playthrough video here: https://youtu.be/Vt3HNGZBQN0

What participant knew before test:

- *"some clusters you can gain access to. You gain access to them to solve mysteries"*
- *"I know you are swimming"*
- *"Two characters, a man and woman (Lina and Sebastian)"*
- *"A HUB in the game."*
- *"These memories can be faked, there are flaws in it, and that is what you should do to solve the mystery."*
- *"Changing props with other props. "*

  _____

Old tutorial level (swimming between two clusters and using the portal)
Relatively easy to swim
Dont know why I fell.
There is a portal. "Now they are linked" - Understand that they are connected to eachother.
Thinks he should press 'E'

Worry Cluster

Right before participant replaces objects participant is asked:
- Why did you choose this item?
- How confident are you that your answer is correct? (1-5) (1 = completely guessing, 5 = Completely sure)

Participant listens to false memory1.

****immediately thinks to find something Sebastian has forgotten****

Facilitator explains mechanics and hints that the misplaced object is in another memory.

Participant first explores a bit around the first memory.
Says he would want to trade out with random stuff, but does not do it since he knows it's in another memory.

Finds new memory easily
—---

Participant listens to false memory2. (00:09:09)

Dog is gone and missing,
Police is here
Looking at photo of dog
A guy named Michael

Participant looks at the different objects in memory2

Participant knows that he should take something in memory2 and replace it in memory1, but first wants to see if he can replace something in memory2 with memory2 flaw.

Politiken Newspaper → photo of dog. (2) (00:11:19)
Other things on the table didn't look to be that. It felt more natural that the 'politiken article' should be here.

_____
Pills → Photo of dog (2) (00:12:38)
It looks weird that there are pills on the table. *"it's out of place"*

_____

Looks at the police officer, but does not choose it as he cannot stand on the table.

Participant now thinks he should take something from 2nd memory to 1st memory.
Treats sound weird. Something he has a lot that he forgot.

Pills → Keys (4) (CORRECT) (00:13:34)
Do not think it would be any of the food since it does not make sense to be something that is not good to forget.
It seems to be something he uses regularly that he has forgotten.
Pills is something Sebastian might need.

_____
Participant listens to true memory1. (00:15:27)

Didn't sound worried, now she seems worried.
Pills are very important for Sebastian.
Seems like Sebastian needs them.

Faciliator explains that this memory is completed, but there can still be objects here that should be replaced other places, and that memory3 is now unlocked.

_____
Woman Magazine → photo of dog (1) (00:17:04)
Just random since I dont lose anything to try.

_----------

Participant listens to false memory3 (00:18:24)
Two red things
Michael is here. (knows that since it says michael when you hover over the person)
Immediately gets the idea of getting the police officer at the chair instead
**goes briefly away from it to first see if he can get an idea about the water bottle

replacement. He could not and therefore goes back to the police officer/book theory**

Police officer → Books (5) (00:20:00)
Police officer make sense both gameplay and story wise.

Chocolate Bar → Water Bottle (4) (00:21:34)
It make sense to take some water since you are nervous, since its not water it could be chocolate. Eat a snickers commercial.

laughs at it being wrong **"It was a good bait"**

**checks out memory1 and memory2, concludes that its not something from the same memory to replace with a water bottle.

Wants to replace the dog with water bottle to make her relax, but knows he cannot do that. He thinks that memory3 is about the dog.

Has not noticed Mind Diver flyer.

Now sees the photos on the wall at memory3.

Photo of sebastian → Water bottle (4) (00:24:35)
Maybe its not the dog that is gone, maybe it is sebastian that is gone and why she is so nervous. A picture of him on the wall seems odd
Participant did not think it would be connected to the photo of dog

Oat Cookies → Water bottle (1)
A full on guess since he does not know more of the story.

Brownies → Water bottle (1) (00:26:37)
Completely guessing, "feel like im missing something. Something Im not seeing, it probably is not food."

Goes to memory1 and sees mind diver flyer but says: (00:28:14)
"Mind Diver Flyer, it has nothing to do with that"

Still thinks a dog is also gone.

_____
Notebook → Water bottle (2)
To stress off, not a completely random guess.

_____

completely lost now and just want to try to guess.
Paper → Water bottle (1) (00:30:05)
just guessing

House → Water bottle (2) (00:30:23)
Still a guess, but maybe a lead to where he is.

Bread → Water Bottle (2) (00:31:00)
Still guessing, but basing on the theory of something to eat instead of drinking something.
Still seems weird that it should be that.

Closed Binder Book → Water bottle (2) (00:31:37)
Guess, Maybe some pictures in it to help.

Wrist clock → Water bottle (1) (00:32:31)
Just a guess.

_____
Participant seems to have forgotten about the photo of the dog: Been to focus on the third memory probably because it opened up after memory1.
Facilitator asks the participant how many flaws he is missing. Participant first says he is only missing the water bottle, but then realizes he is also missing the photo of the dog.

_____

Now focuses on the photo of the dog to then make that memory true which maybe would give hint to memory3.

Photo of Sebastian → Photo of dog. (2) (00:34:55)
Maybe looking at Sebastian and not the dog.
Still thinks the dog is relevant. Thinks by doing this he will switch the dog picture over to memory3.

Participant listens to true memory2. (00:37:40)
Immediately thinks its a flyer since it is stated and remember that there is a flyer in memory1

Mind Diver Flyer → Water Bottle (5) (00:37:50)
Very confident since it is directly stated in memory2.
Its a flyer about the Mind Diver, Michael gives flyer so they can find Sebastian.

Participant listens to true memory3.
—----------

Final Notes:
In your own words. What do you think the story is about? (00:41:21)

Been to some party. Sebastian (boyfriend) has talked about some place he has been. Maybe something he wants to pursue. Something has happened, he forgot his pills, which is abnormal since he is used to take his medicine / pills.
He might be in danger (Sebastian).
She wants to try this Mind Diver.
Dont know who Michael is, is he a friend? or what is he trying to do.
Someone you should keep an eye out for. What is the relation between Lina and Michael?
Police is extremely busy which may push her towards the mind diver tech.

Emotional thoughts: (00:44:00)

A bit confused to start out with.
Did not know beforehand what was opened (in terms of bubbles).
Heard the truth at the memory3, is hooked and want to know more

Some confusion with the dog. Forgot the dog. Information overload.

Getting something correct was nice.
No favourites.
Most change with memory3. That is where I got the feeling of going forward.

Not so nice to just guess. Want to guess it correctly based on something.
Did not feel nice when being in the complete guessing phase.
Motivation was still high to keep going.

Teleports make perfect sense
Swimming made sense.
“E” is use, Mouse1 is shoot for me. But when knowing Mouse1 was correct it felt natural.
the photogrammetry part is a cool feature.
Easy to navigate around (tps, swimming and scans).

Playthrough video here: https://www.youtube.com/watch?v=DJ9O9XbI7R4

Games played? (any puzzles or detective games?)

FPS Mainly, some Platformers (Mario).
No puzzle or detective games.

What participant knew before test:
Nothing

Old tutorial level (swimming between two clusters and using the portal)
Quickly understood how to use the swim stroke.

some acceleration with wasd movement while walking.

Worry Cluster

Right before participant replaces objects participant is asked:
- Why did you choose this item?
- How confident are you that your answer is correct? (1-5) (1 = completely guessing, 5 = Completely sure)

Participant listens to false memory1.
** **

not sure what to do. Dont think he has enough info. I know to replace Linas Keys

The facilitator explains mechanics and hints that the misplaced object is in another memory.

Participant initially understands it as taking the key and replacing it with another object. (00:07:24)

Participant listens to false memory2

Seems confused and wanted to go to memory3 for more info, but was not allowed.

Receipt Pills → Keys (2) (00:10:42)
random guess, the most evident thing to put in hands.
Must have forgotten or not understood that the correct object is in the other memory

Now he asks if he could pick something from another memory and put it back.

initially takes the photo of the dog, but gets reminded from the facilitator that he cannot do that.

He noticed the red around the photo initially, but it is not very clear that the red color is meant for the dog.

Politiken Article → Keys (1) (00:13:01)
totally random

Ibyen article → Keys (1) (00:13:26)
totally random

Tries most things in memory1 (00:13:51)
(Alarm Clock, magazine1, magazine2, magazine3, plant) → keys (1)
random guessing. Just want to try everything.

Bottle of Pills → Keys (1) (00:14:31)
Completely guess, No clue on why it should be pills.

Participant listens to true memory1
****participant asks if she mentions pills in the first false memory** facilitator says no.**

Goes to memory2 and reminds himself that he has to replace the dog and thereafter goes to memory3.

Participant listens to false memory3 (00:17:32)
need to replace a bottle.
don't know if he should solve the second or third one first.

Snickers → Water bottle (2) (00:18:00)
Random, but it fits in the hand.

Michael → Police officer (4) (00:19:02)
She was talking to Michael and not a police officer.

Police Officer → Michael (1) (0019:32)
Just to see if it fits the other way around.

All food → Water bottle. (1) (00:20:50)
Totally random

Photo of sebastian → water bottle (1)
Total guess.

Faciliator is now giving hints (trying to make participant think of certain things.)

Facilitor asks:
What do you think is right and wrong in this memory (memory2) (00:24:48)
“Don’t know” “everything seems fine to me”

Facilitor asks:
What do you think the story is about so far? (00:25:45)
“Lina is missing her dog.”
“Michael took some pills.”
“It is worrying that michael took some pills”

Hint from facilitator
repeating memory1 script, trying to indicate that it was Sebastian that took pills. (00:27:09)

Receipt pills → photo of dog (1) (00:27:15)
Total guess.

Hint from facilitator:
Telling him directly that the person in memory1 is a person named Sebastian and memory3 is a person named Michael. (00:28:00)

Participant still does not know what to replace.

Hint from facilitator:
telling him directly that the police officer is not suppose to be in memory2 (00:29:22)

Police officer → Books (4) (00:29:33)
because of hints and sitting on a chair.

Hint from facilitator:
All the misplaced objects are from another memory than the one that should be replaced.

Photo of sebastian → Photo of dog (1) (00:30:52)
Total guessing.

Participant listens to true memory2
immediately goes for the flyer that he remembered was in memory1.

Mind Diver Flyer → Water bottle (3) (00:31:51)
Michael just talked about a flyer.

Hard to solve the mind diver flyer without anything else.

Final Notes:
In your own words. What do you think the story is about? (00:35:03)

About sebastian missing, michael friend tries to help lina, Police understaffed because of covid, left to find Sebastian. Sebastian is mentally ill, he has pills and is a danger to himself. Need to find him quickly.

Emotional thoughts: (00:36:00)
Like the idea at first, replacing stuff and finding the right answer.
If things disappear it could maybe be fun.
It was really too hard to understand what to replace with what.
It took too much time. Frustrating this felt like the end of the game. Wanting something easier to begin with. Maybe it would be easy if having something way more easy to begin with.

Liked that there were a limited amount of items. Make it easier to have less things to interact with. Liked that he could not take walls and all tables.

Reduce items in first levels and expand from there.

Memories and scans work well together.

Playthrough here: https://youtu.be/EIVDzxpoTSc

Games:
Playing a lot of different genres.
Played a lot of puzzle games.
Detective games
Beneath a steel sky
beyond a steel sky
Blade Runner
X-Files

What participant knew before test:

- Has read the Mind Diver story script
- It's a Story about Lina and she gets help to remember something. Looking after a guy and where he is.
- You go through the memories and replace them with correct objects.

Old tutorial level (swimming between two clusters and using the portal)
Already have some understanding beforehand with the mechanic, so it's easy to swim.

Worry Cluster

Right before participant replaces objects participant is asked:
- Why did you choose this item?
- How confident are you that your answer is correct? (1-5) (1 = completely guessing, 5 = Completely sure)

Facilitator explains mechanics and hints that the misplaced object is in another memory.

Participant listens to false memory1.

Participant listens to false memory2.

Bottle of Pills → Keys (2)
Mostly a guess, but is highlighted in a different way.

Facilitator tells him that the different highlights should not be a factor and is a technical error.

Participant listens to true memory1.

Participant listens to false memory3.

water bottle is wrong. Goes directly to try to find out what should be placed instead of water bottle.

Photo of House Location → Water bottle (4)
Something she is searching for, maybe searching for a house.

Photo of House → water bottle (3)
Same as before, maybe searching for a house.

Goes back to memory2.

Politiken paper article → Water bottle (1)
Could be but a guess.

Takes books since it is highlighted, facilitator reminds that he cannot do that and that is also something that has to be changed.

Photo of sebastian → water bottle. (2)
Searching after someone, cant remember the name.

Mind Diver Flyer → Water bottle (4)
The game's name and maybe getting introduced to what mind divers are, and that could be the solution to her problem.

Police officer → Michael (4)
Why is michael giving it in a police station. It would be a police man giving it to her.

Police officer → Books (2)
Seems like the police officer should not be in memory2 and books are wrong in memory3

Participant listens to true memory3.

wants to go through through to the next area
Finds out he is still missing something and goes back.

goes to memory2:
Looking at the photo of the dog.
Trying to find something that she would look at.

Photo of Casper → Photo of Dog (3)
Maybe she is talking about this casper guy or talking with him

Going back to the photos.

Photo of Sebastian → Photo of dog (2)
Not sure

In your own words. What do you think the story is about?
Lina is looking for Sebastian. She is close to Sebastian and he is missing and we don't know where he is.

Don't understand where the books went?
Thought things switched places.

Emotional Thoughts:
Nice intro with the Mind Diver start.
Have some intro about the Mind Diver would be nice (good since we are planning that with ice cube).
Faciliator took him out from going with intuition.
As soon as the mind diver flyer it was easier to get through it.
Moving around doing other things while people are talking. He likes that.
Became fun to solve the puzzles, got in the flow.

Test of Hope cluster on 18/2-2022

Summary:

Found it nice and accessible with some satisfying moments.
Frustrated at being nauseous (we can't do much about this)
Biggest puzzle problem was being able to brute-force the door.
Some apprehension at re-listening to scenes.
Camera puzzle could be harder - was solved first.

Play context:

Participant was playing with three other people sitting next to participant, watching and coming with ideas once in a while.

Participant is the game director's brother.
Participant is an experienced game player who especially loves strategic games (X-Com, Civilization) and story games (Disco Elysium) but also many other kinds of games.
Participant is a professional script-writer.
Participant always gets quite serious nausea from playing first-person games.

The people watching are inexperienced and usually don't play games themselves.

What participant knew before test:

Had read the script in an old version and given feedback, so he was familiar with much of the story. However, at that point, the story was from both Sebastian's and Lina's perspectives.
He specifically could recognize the story of the Hope cluster - meeting at a protest and going to a rooftop.
He also knew about the core mechanic of correcting memories, but also in an older version.

Playtest transcription:

Figured out the swimming with no trouble.
Facilitator told him specific details about how to complete the tutorial, but he figured out to place the box and Lina on his own.

He tried to swim to the giant figures outside of the algae in the beginning.

On arriving to the CM, he first looks around at all the visible scenes.

Bottle → C1 Lina's Head (INCORRECT)
He doesn't see the difference between the hole and the head, so he tries to place the bottle where the head is.

Facilitator explains he has to place things where the red spheres are and that it's not supposed to be functionally possible place things other places than the holes.

Gin → C1 Hole (INCORRECT)

Dress → C1.1 Hole (CORRECT)

Hat → C1 Michael head (CORRECT)

Shirt → C1 Sebastian chest hole (CORRECT)
Here the participant gets stuck in the scan.

Facilitator explains that he can jump on spacebar.

Participant jumps out and is no longer stuck.

Wallet → C1 Lina hand hole (INCORRECT)

Wallet → C1 Sebastian hand hole (CORRECT)

Participant plays R to play the memory, and the subtitles play. Facilitator provides live voice acting ;)

Gin → C-Final hole (CORRECT)

Facilitator explains that we'll skip the Worry cluster and instead activates the green lights that guide the way to Hope, and tells participant that he should follow them.
Participant finds it without difficulty.

Hope cluster

Participant starts with looking at M2, then M1, then M3.

Couch → M3.1 (CORRECT)
Because they were sitting in the air, and the couch was right next to them.

Listens to M2.1 (facilitator reads aloud from script)

Sebastian's key → M2.1 (CORRECT)

Because Sebastian said that she could have the key.
He barely noticed the key, while randomly walking around while the script was being read aloud.

Listens to M2.2 (facilitator reads aloud from script)
He understands it must be a newly painted door from "Eww, my hand".

Listens to M3.2
The sound of the camera shutter several times must mean that she is sitting with a camera.
“But which camera?”

Listens to M1.1

Sebastian → M1.1 (CORRECT)
Because he is a main character. He recognized the name and the person from the other memories.
It was partly a wild shot.

Listens to M1.2
Participant and audience realizes that she must be on a bike. “ohh, she is not just jumping away from a light pole”

Regular camera from M1 → M3.1 (INCORRECT)
Because he knew she had a camera, but she seemed like the type that would have a professional camera.

Polaroid camera from M1 → M3.1 (3) (CORRECT)
Because he had tried the other one, and there are only two in the temp scan…
(The polaroid camera is not in M2 in the temp scan either)

Listens to M3.3
Loves the scene. Loudly exclaims that it’s beautiful, the characters are tender and fragile, and the motivations make sense.
Everyone in the audience reacts with “awwww”.

Relistens to M1.2
He realizes the horn must be important, and goes to look at the bikes in M2.
He doesn’t spot it, but one of the on-lookers spots the horn on a bike.

Lina’s broken bike → M2.2 (CORRECT)
Because of the horn sound in M1.
He says that if he hadn’t seen the horn, he would have tried to look for a broken chain.
(He never noticed there were two bikes with the horn)

Relistens to M2.2
He now starts trying to solve M2.2.
He is inclined to brute-force and just try all four doors, but facilitator asks him for the sake of the test to not brute-force.

He starts by looking at the doors in M3 to see if they open the right way.
Realizes they all open the same way.
He thinks it must have something to do with the color.
He positions himself to get an overview of the four doors, and then notices the paint on Lina’s hand in M3.

Blue door → M2.2 (CORRECT)
Because

- The door part was obvious from them talking about opening something
- “Eww my hand” -> newly painted door
- The paint on her hand in M3 matches the color of the blue door

---

In your own words. What do you think the story is about?
(This was answered by one of the on-lookers, as the participant was already familiar with the story)
Lina is supposed to cover a demonstration.
Sebastian steps in front of her, so she stops.
He repairs her bike.
She gets fired.
Sebastian has access to the rooftop.
He explains that his mom is unwell.
They take a selfie at the end.

Gameplay thoughts:
The one with the couch was easiest.
The door was the hardest and the most satisfactory. He just barely got to the thought “could she have something on her hand” and then saw the paint in the same moment.
The key could have been super hard. He didn’t expect there to be anything useful on the wall, and didn’t expect useful things to be small.
He only noticed the horn, because the facilitator read it aloud in a different way second time.
He loved the rooftop scene. He liked that “thank you for helping with the bike” signals that time has passed.
He liked that the hole for the wallet in Central Memory was a bit hidden. Felt like a nice small puzzle to start with.

Frustrations:
Nausea. Not too bad, but it affected the experience
He notices a tension between listening to long scenes and the player’s urge to move around and do that stuff. He says it was fine when the scenes were really good (rooftop), but less so, when he listened several times to an all-right scene.
He suggests having an angle that the player is placed in when they play the memory, to encourage attention and making sure you don’t miss it.
He really liked the idea of having a journal. And he **strongly** encouraged us to skip the audio you’ve already heard, when unlocking new audio for a memory (or just play the last line of the audio you’ve already heard).
Could have been stumped by key.

Overall:
It feels easy and accessible.

He thought the introduction to the mechanic, from Central Memory to Hope was very smooth - he didn't even think about the tutorilization before we talked about it afterward.
There were a few more notes, but I forgot them at home….

On-lookers commented that it took a while before they understood what the red spheres were.
Comment by one of the on-lookers (who is a painter): "Red is a very present color. Maybe consider white, as that represents a lack of presence and is a distance color."

Test of Hope cluster on 18/2-2022

Summary:

Found it nice and accessible with some satisfying moments.
Frustrated at being nauseous (we can't do much about this)
Biggest puzzle problem was being able to brute-force the door.
Some apprehension at re-listening to scenes.
Camera puzzle could be harder - was solved first.

Play context:

Participant was playing with three other people sitting next to participant, watching and coming with ideas once in a while.

Participant is the game director's brother.
Participant is an experienced game player who especially loves strategic games (X-Com, Civilization) and story games (Disco Elysium) but also many other kinds of games.
Participant is a professional script-writer.
Participant always gets quite serious nausea from playing first-person games.

The people watching are inexperienced and usually don't play games themselves.

What participant knew before test:

Had read the script in an old version and given feedback, so he was familiar with much of the story. However, at that point, the story was from both Sebastian's and Lina's perspectives.
He specifically could recognize the story of the Hope cluster - meeting at a protest and going to a rooftop.
He also knew about the core mechanic of correcting memories, but also in an older version.

Playtest transcription:

Figured out the swimming with no trouble.
Facilitator told him specific details about how to complete the tutorial, but he figured out to place the box and Lina on his own.

He tried to swim to the giant figures outside of the algae in the beginning.

On arriving to the CM, he first looks around at all the visible scenes.

Bottle → C1 Lina's Head (INCORRECT)
He doesn't see the difference between the hole and the head, so he tries to place the bottle where the head is.

Facilitator explains he has to place things where the red spheres are and that it's not supposed to be functionally possible place things other places than the holes.

Gin → C1 Hole (INCORRECT)

Dress → C1.1 Hole (CORRECT)

Hat → C1 Michael head (CORRECT)

Shirt → C1 Sebastian chest hole (CORRECT)
Here the participant gets stuck in the scan.

Facilitator explains that he can jump on spacebar.

Participant jumps out and is no longer stuck.

Wallet → C1 Lina hand hole (INCORRECT)

Wallet → C1 Sebastian hand hole (CORRECT)

Participant plays R to play the memory, and the subtitles play. Facilitator provides live voice acting ;)

Gin → C-Final hole (CORRECT)

Facilitator explains that we'll skip the Worry cluster and instead activates the green lights that guide the way to Hope, and tells participant that he should follow them.
Participant finds it without difficulty.

Hope cluster

Participant starts with looking at M2, then M1, then M3.

Couch → M3.1 (CORRECT)
Because they were sitting in the air, and the couch was right next to them.

Listens to M2.1 (facilitator reads aloud from script)

Sebastian's key → M2.1 (CORRECT)

Because Sebastian said that she could have the key.
He barely noticed the key, while randomly walking around while the script was being read aloud.

Listens to M2.2 (facilitator reads aloud from script)
He understands it must be a newly painted door from "Eww, my hand".

Listens to M3.2
The sound of the camera shutter several times must mean that she is sitting with a camera.
“But which camera?”

Listens to M1.1

Sebastian → M1.1 (CORRECT)
Because he is a main character. He recognized the name and the person from the other memories.
It was partly a wild shot.

Listens to M1.2
Participant and audience realizes that she must be on a bike. “ohh, she is not just jumping away from a light pole”

Regular camera from M1 → M3.1 (INCORRECT)
Because he knew she had a camera, but she seemed like the type that would have a professional camera.

Polaroid camera from M1 → M3.1 (3) (CORRECT)
Because he had tried the other one, and there are only two in the temp scan…
(The polaroid camera is not in M2 in the temp scan either)

Listens to M3.3
Loves the scene. Loudly exclaims that it’s beautiful, the characters are tender and fragile, and the motivations make sense.
Everyone in the audience reacts with “awwww”.

Relistens to M1.2
He realizes the horn must be important, and goes to look at the bikes in M2.
He doesn’t spot it, but one of the on-lookers spots the horn on a bike.

Lina’s broken bike → M2.2 (CORRECT)
Because of the horn sound in M1.
He says that if he hadn’t seen the horn, he would have tried to look for a broken chain.
(He never noticed there were two bikes with the horn)

Relistens to M2.2
He now starts trying to solve M2.2.
He is inclined to brute-force and just try all four doors, but facilitator asks him for the sake of the test to not brute-force.

He starts by looking at the doors in M3 to see if they open the right way.
Realizes they all open the same way.
He thinks it must have something to do with the color.
He positions himself to get an overview of the four doors, and then notices the paint on Lina’s hand in M3.

Blue door → M2.2 (CORRECT)
Because

- The door part was obvious from them talking about opening something
- “Eww my hand” -> newly painted door
- The paint on her hand in M3 matches the color of the blue door

---

In your own words. What do you think the story is about?
(This was answered by one of the on-lookers, as the participant was already familiar with the story)
Lina is supposed to cover a demonstration.
Sebastian steps in front of her, so she stops.
He repairs her bike.
She gets fired.
Sebastian has access to the rooftop.
He explains that his mom is unwell.
They take a selfie at the end.

Gameplay thoughts:
The one with the couch was easiest.
The door was the hardest and the most satisfactory. He just barely got to the thought “could she have something on her hand” and then saw the paint in the same moment.
The key could have been super hard. He didn’t expect there to be anything useful on the wall, and didn’t expect useful things to be small.
He only noticed the horn, because the facilitator read it aloud in a different way second time.
He loved the rooftop scene. He liked that “thank you for helping with the bike” signals that time has passed.
He liked that the hole for the wallet in Central Memory was a bit hidden. Felt like a nice small puzzle to start with.

Frustrations:
Nausea. Not too bad, but it affected the experience
He notices a tension between listening to long scenes and the player’s urge to move around and do that stuff. He says it was fine when the scenes were really good (rooftop), but less so, when he listened several times to an all-right scene.
He suggests having an angle that the player is placed in when they play the memory, to encourage attention and making sure you don’t miss it.
He really liked the idea of having a journal. And he **strongly** encouraged us to skip the audio you’ve already heard, when unlocking new audio for a memory (or just play the last line of the audio you’ve already heard).
Could have been stumped by key.

Overall:
It feels easy and accessible.

He thought the introduction to the mechanic, from Central Memory to Hope was very smooth - he didn't even think about the tutorilization before we talked about it afterward.
There were a few more notes, but I forgot them at home….

On-lookers commented that it took a while before they understood what the red spheres were.
Comment by one of the on-lookers (who is a painter): "Red is a very present color. Maybe consider white, as that represents a lack of presence and is a distance color."

Same person as test participant#3 for worry 2nd pass test.

Summary:
One easy in each memory felt good
It seems like the game develops in a way that it starts out hard and then becomes easier with more and more info. Or that the player then gets used to it and to the mindset of how to solve the puzzles.

What participant knew before test:

Knew the story of worry cluster:
Lina had been to the police
Got the Mind Diver people to help find Sebastian

Participant listened to the memory bubble 1.1

“She is chrashing”

Looks around memory, sees car

Car → M1.1 (4) (INCORRECT)
She crashed, there were sounds of wheels which could be a bike, but perhaps also a car.

“Could be one of the protesters”
Note that Sebastian is marked as “Protester Sebastian” and other protesters are marked “Protester (name)”

Protester Alexander → M1.1 (3) (INCORRECT)
Could be this guy she is talking to since he is one of the protesters who are physically closer to Lina

Polaroid camera with photo → M1.1 (3) (INCORRECT)
she mentions something with a photo, this seems to stick out.

The participant goes to memory2 to see if there is something he is missing.

Participant listened to the memory bubble 2.1

Lina’s **“**bell horn” → M1.2 (4) (INCORRECT)
This bell horn is referenced in the script.

Looks at script again, “aah maybe it is one of the keys, does any of them have anything on them..”

Key with Sebastian tag → M2.1 (3) (CORRECT)
Sebastian and Lina are in a relationship, so she could have the keys.

Participant listened to the memory bubble 2.2

“If I take someone here then they are not in this memory?” - Facilitator confirms this.
“Uuh, I want to try something out, it is probably not correct”

Sebastian from M2 → M1.1 (2) (INCORRECT)
Not sure, but maybe they didn’t meet here at all and Sebastian is the one here.

“Aah well then it is Sebastian Protester”

Sebastian Protester → M1.1 (4) (CORRECT)
It is Sebastian helping her, I don't know why I didn't see that in the beginning.

Mentions he likes the red thread as he forgot Sebastian in the beginning, but the second memory then reminded him of him.

Participant listened to the memory bubble 1.2

Was about to take polaroid with photo to Lina, but then noticed that Lina already had a polaroid camera so it could not be that.

Demonstration Banner#1 → M1.2 (2) (INCORRECT)
just a guess, I understand that there is something under her so maybe a banner.

Goes to memory3

Participant listened to the memory bubble 3.1

Notices oil on Sebastian and pink paint on Lina. Asks if he can take those things. - Faciliator explains that it is not something you can take individually, but is just something on the them.

Pink Door → M2.2 (4) (CORRECT)
“She touches something newly painted, she has pink paint on her and there is a pink door here.”

Participant listened to the memory bubble 2.3
“Aaah a bike”

Bikehorn with chain broken → M1.2 (4) (CORRECT)
Don't know why I didn't think about it before, it makes perfect sense that she is on her bike.

Participant listened to the memory bubble 1.3
Sees he is done with memory1 and memory2 - goes therefore to memory3.

Listens to memory script again.

Bench → M3.1 (4) (CORRECT)

“they are sitting on something”

Participant listened to the memory bubble 3.2
“Aah there was the polaroid camera back in memory1”

About to take the polaroid camera from Lina in m1, but then reminds himself that then that camera will not be here, which does not make sense. Then remembers that there is another polaroid camera with a photo on it.

Polaroid Camera with photo → M3.2 (4) (CORRECT)
They are taking a picture hence the photo with a polaroid camera.

Participant listened to the memory bubble 3.3

In your own words. What do you think the story is about?
Lina goes to work, meets sebastian who is protesting. Her chain brakes.
Gets fired and she was late. Sebastian suggests to go to a rooftop. It goes well and they take a photo together.

Gameplay thoughts:
A bit frustrated in the beginning, probably a bit “out of training” and forgot Sebastian.
Story was exciting.
Fun puzzles to solve, working across multiple bubbles.
Really liked that the first puzzle for each was only within the same bubble. “Not super hard mode to begin with” which is also good for people that just might have taken a break from the game.

The facilitator asks if he can tell the most satisfying puzzle and most boring puzzle:
Really liked the paint on Lina. Gave the feeling of interacting with the surroundings and also that the people change accordingly from what happened in the previous memory. Other puzzles did this as well, but this was direct with a person which was nice.

The bench was least satisfying since it was too obvious, there was nothing to solve. The other easy ones at least had something to solve. “something to think about”

Facilitator asks about this way of playing it compared to the previous way of “false memories”:
Likes this version a lot. It feels like you are “tying a bow around it”. It is all scrambled, all pieces influence the story. Also liked the old version. Does not see this version missing out on something the old version had.

Same person as test participant#3 for worry 2nd pass test.

Summary:
One easy in each memory felt good
It seems like the game develops in a way that it starts out hard and then becomes easier with more and more info. Or that the player then gets used to it and to the mindset of how to solve the puzzles.

What participant knew before test:

Knew the story of worry cluster:
Lina had been to the police
Got the Mind Diver people to help find Sebastian

Participant listened to the memory bubble 1.1

“She is chrashing”

Looks around memory, sees car

Car → M1.1 (4) (INCORRECT)
She crashed, there were sounds of wheels which could be a bike, but perhaps also a car.

“Could be one of the protesters”
Note that Sebastian is marked as “Protester Sebastian” and other protesters are marked “Protester (name)”

Protester Alexander → M1.1 (3) (INCORRECT)
Could be this guy she is talking to since he is one of the protesters who are physically closer to Lina

Polaroid camera with photo → M1.1 (3) (INCORRECT)
she mentions something with a photo, this seems to stick out.

The participant goes to memory2 to see if there is something he is missing.

Participant listened to the memory bubble 2.1

Lina’s **“**bell horn” → M1.2 (4) (INCORRECT)
This bell horn is referenced in the script.

Looks at script again, “aah maybe it is one of the keys, does any of them have anything on them..”

Key with Sebastian tag → M2.1 (3) (CORRECT)
Sebastian and Lina are in a relationship, so she could have the keys.

Participant listened to the memory bubble 2.2

“If I take someone here then they are not in this memory?” - Facilitator confirms this.
“Uuh, I want to try something out, it is probably not correct”

Sebastian from M2 → M1.1 (2) (INCORRECT)
Not sure, but maybe they didn’t meet here at all and Sebastian is the one here.

“Aah well then it is Sebastian Protester”

Sebastian Protester → M1.1 (4) (CORRECT)
It is Sebastian helping her, I don't know why I didn't see that in the beginning.

Mentions he likes the red thread as he forgot Sebastian in the beginning, but the second memory then reminded him of him.

Participant listened to the memory bubble 1.2

Was about to take polaroid with photo to Lina, but then noticed that Lina already had a polaroid camera so it could not be that.

Demonstration Banner#1 → M1.2 (2) (INCORRECT)
just a guess, I understand that there is something under her so maybe a banner.

Goes to memory3

Participant listened to the memory bubble 3.1

Notices oil on Sebastian and pink paint on Lina. Asks if he can take those things. - Faciliator explains that it is not something you can take individually, but is just something on the them.

Pink Door → M2.2 (4) (CORRECT)
“She touches something newly painted, she has pink paint on her and there is a pink door here.”

Participant listened to the memory bubble 2.3
“Aaah a bike”

Bikehorn with chain broken → M1.2 (4) (CORRECT)
Don't know why I didn't think about it before, it makes perfect sense that she is on her bike.

Participant listened to the memory bubble 1.3
Sees he is done with memory1 and memory2 - goes therefore to memory3.

Listens to memory script again.

Bench → M3.1 (4) (CORRECT)

“they are sitting on something”

Participant listened to the memory bubble 3.2
“Aah there was the polaroid camera back in memory1”

About to take the polaroid camera from Lina in m1, but then reminds himself that then that camera will not be here, which does not make sense. Then remembers that there is another polaroid camera with a photo on it.

Polaroid Camera with photo → M3.2 (4) (CORRECT)
They are taking a picture hence the photo with a polaroid camera.

Participant listened to the memory bubble 3.3

In your own words. What do you think the story is about?
Lina goes to work, meets sebastian who is protesting. Her chain brakes.
Gets fired and she was late. Sebastian suggests to go to a rooftop. It goes well and they take a photo together.

Gameplay thoughts:
A bit frustrated in the beginning, probably a bit “out of training” and forgot Sebastian.
Story was exciting.
Fun puzzles to solve, working across multiple bubbles.
Really liked that the first puzzle for each was only within the same bubble. “Not super hard mode to begin with” which is also good for people that just might have taken a break from the game.

The facilitator asks if he can tell the most satisfying puzzle and most boring puzzle:
Really liked the paint on Lina. Gave the feeling of interacting with the surroundings and also that the people change accordingly from what happened in the previous memory. Other puzzles did this as well, but this was direct with a person which was nice.

The bench was least satisfying since it was too obvious, there was nothing to solve. The other easy ones at least had something to solve. “something to think about”

Facilitator asks about this way of playing it compared to the previous way of “false memories”:
Likes this version a lot. It feels like you are “tying a bow around it”. It is all scrambled, all pieces influence the story. Also liked the old version. Does not see this version missing out on something the old version had.

H Code Files

AgisoftPrep.bat

```
@echo off & SETLOCAL EnableDelayedExpansion

:MakeTemp
set "tempFile=%temp%\file-%random%"
if exist "%tempFile%" goto :MakeTemp

:MakeffmpegTemp
set "tempFile2=%temp%\file-2%random%"
if exist "%tempFile2%" goto :MakeffmpegTemp

call :MakeTempScript
call :MakeffmpegScript

echo Please select the ffmpeg exe

for /f %%a in ('powershell "%tempFile2%.ps1"') do (
        set "ffmpeg=%%~fa"
)

set i = 0

echo Please select all the videos to process

for /f %%a in ('powershell "%tempFile%.ps1"') do (
        set /a "i += 1"
        set "vidLoc=%%~dpa"
        set "vidPath=%%~fa"
        set "vidName=%%~na"

        if not exist "!vidLoc!!vidName!\" mkdir !vidLoc!!vidName!
        if not exist "!vidLoc!!vidName!\pictures\" mkdir !vidLoc!!vidName!\pictures

        echo Starting ffmpeg
        start /wait %ffmpeg% -i !vidPath! -vf "select=not(mod(n\,7))" -vsync vfr -q:v 1 -
    bitexact "!vidLoc!!vidName!\pictures\photo%%05d.jpg"
        echo Finished ffmpeg

        echo Creating XML file
        set "xml=!vidLoc!!vidName!\BatchProcess"
        call :CreateXML
        echo Finished Creating XML File
        echo Finished run for file !i! !vidName!
)

pause
del "%tempFile%.ps1"
del "%tempFile2%.ps1"
goto :EOF

:MakeTempScript
(
echo Add-Type -AssemblyName System.Windows.Forms
echo $fileBrowser = New-Object System.Windows.Forms.OpenFileDialog
echo $fileBrowser.initialDirectory = [Environment]::GetFolderPath^("Desktop"^)
echo $fileBrowser.Multiselect = $True
echo $fileBrowser.filter = "Video(*.mp4;*.mov)|*.mp4;*.mov|All(*.*)|*.*"
echo $fileBrowser.ShowDialog^(^) ^| Out-Null
echo $fileBrowser.FileNames
) > "%tempFile%.ps1"
exit /B 0

:MakeffmpegScript
(
echo Add-Type -AssemblyName System.Windows.Forms
echo $fileBrowser = New-Object System.Windows.Forms.OpenFileDialog
echo $fileBrowser.initialDirectory = [Environment]::GetFolderPath^("Desktop"^)
```

```
echo␣$fileBrowser.filter␣=␣"Executable(*.exe)|*.exe|All(*.*)|*.*"
echo␣$fileBrowser.ShowDialog^(^)␣^|␣Out-Null
echo␣$fileBrowser.filename
)␣>␣"%tempFile2%.ps1"
exit␣/B␣0

:CreateXML
(
echo␣^<?xml␣version="1.0"␣encoding="UTF-8"?^>
echo␣^<batchjobs␣version="1.8.1"^>
echo␣␣^<job␣name="AlignPhotos"␣target="all"^>
echo␣␣␣␣^<mask_tiepoints^>false^</mask_tiepoints^>
echo␣␣␣␣^<reference_preselection_mode/^>
echo␣␣^</job^>
echo␣␣^<job␣name="BuildDenseCloud"␣target="all"^>
echo␣␣␣␣^<downscale^>2^</downscale^>
echo␣␣␣␣^<reuse_depth^>true^</reuse_depth^>
echo␣␣^</job^>
echo␣␣^<job␣name="BuildModel"␣target="all"^>
echo␣␣␣␣^<downscale^>2^</downscale^>
echo␣␣␣␣^<reuse_depth^>true^</reuse_depth^>
echo␣␣␣␣^<source_data^>6^</source_data^>
echo␣␣^</job^>
echo␣␣^<job␣name="BuildTexture"␣target="all"/^>
echo␣␣^<job␣name="ExportModel"␣target="all"^>
echo␣␣␣␣^<embed_texture^>true^</embed_texture^>
echo␣␣␣␣^<format^>10^</format^>
echo␣␣␣␣^<path^>!vidLoc!!vidName!\Model.fbx^</path^>
echo␣␣␣␣^<texture_format^>3^</texture_format^>
echo␣␣^</job^>
echo␣^</batchjobs^>
)␣>␣"!xml!.xml"
exit␣/B␣0
```

ColourChangeApplication

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

namespace MemoryMysteryGame {
    public class ColourChangeApplication : MonoBehaviour {

        #region Editor
        [SerializeField] private bool isPlayerVisibilitySphere = false;
        [SerializeField] private bool isEnvironment = false;
        [SerializeField] private bool isDirectionalLight = false;
        #endregion

        #region Private Fields
        [SerializeField]
        private List<ColourChangeSettings> colourChangeSettings;

        private ColourChangeSettings currentlyActiveSettings;

        private Color matColour;
        private Color shadowColour;
        private Color ambientColour;
        private Color fogColour;
        private float fogDensity;
        private Color directionalColour;
        private float directionalIntensity;

        private IEnumerator fadeMat;
        private IEnumerator fadeEnv;
        private IEnumerator fadeLight;

        private int currentPriority = 0;
        #endregion

        private void Awake() {
            colourChangeSettings = new List<ColourChangeSettings>();
        }
```

```
38
39          #region Public Methods
40          public void OnEnterEvent(ColourChangeSettings settings) {
41
42              colourChangeSettings.Add(settings);
43              if (colourChangeSettings.Count == 0) return;
44              if (colourChangeSettings[colourChangeSettings.Count - 1].priority > settings.
                    priority) return;
45              PrioritizedActivateColourChangeSetting();
46          }
47
48          public void OnLeaveEvent(ColourChangeSettings settings)
49          {
50              var removeIndex = colourChangeSettings.LastIndexOf(settings);
51              if (removeIndex < 0)
52              {
53                  return;
54              }
55
56              colourChangeSettings.RemoveAt(removeIndex);
57              if (colourChangeSettings.Count == 0)
58              {
59                  return;
60              }
61
62              if (colourChangeSettings.Count == 0) return;
63              if (currentlyActiveSettings != colourChangeSettings[colourChangeSettings.Count -
                    1]) {
64                  PrioritizedActivateColourChangeSetting();
65              }
66          }
67
68          #endregion
69
70          private void PrioritizedActivateColourChangeSetting() {
71              ColourChangeSettings highestPrioritySetting = colourChangeSettings[0];
72              foreach (ColourChangeSettings settings in colourChangeSettings) {
73                  if (settings.priority >= highestPrioritySetting.priority) {
74                      highestPrioritySetting = settings;
75                  }
76              }
77              ActivateColourChangeSetting(highestPrioritySetting);
78          }
79
80          private void ActivateColourChangeSetting(ColourChangeSettings settings) {
81              currentlyActiveSettings = settings;
82              currentPriority = settings.priority;
83              StopAllCoroutines();
84              if (isPlayerVisibilitySphere) {
85                  Material mat = this.GetComponent<Renderer>().material;
86                  matColour = mat.color;
87                  fadeMat = CrossFadeMaterial(mat, settings);
88                  StartCoroutine(fadeMat);
89              }
90              else if (isEnvironment) {
91                  shadowColour = RenderSettings.subtractiveShadowColor;
92                  ambientColour = RenderSettings.ambientLight;
93                  fogColour = RenderSettings.fogColor;
94                  fogDensity = RenderSettings.fogDensity;
95                  fadeEnv = CrossFadeEnvironment(settings);
96                  StartCoroutine(fadeEnv);
97              }
98              else if (isDirectionalLight) {
99                  Light light = this.GetComponent<Light>();
100                 directionalColour = light.color;
101                 directionalIntensity = light.intensity;
102                 fadeLight = CrossFadeLight(light, settings);
103                 StartCoroutine(fadeLight);
104             }
105         }
106
107
108         #region Coroutines
```

```
109         private IEnumerator CrossFadeMaterial(Material mat, ColourChangeSettings settings) {
110             float timer = 0;
111             while (timer < 1) {
112                 timer += Time.deltaTime / settings.crossFadeDuration;
113                 mat.color = Vector4.Lerp(matColour, settings.visibilitySphereColour, timer);
114                 yield return null;
115                 if (timer >= 1) {
116                     mat.color = settings.visibilitySphereColour;
117                     currentPriority = 0;
118                 }
119             }
120         }
121 
122         private IEnumerator CrossFadeEnvironment(ColourChangeSettings settings) {
123             float timer = 0;
124             while (timer < 1) {
125                 timer += Time.deltaTime / settings.crossFadeDuration;
126                 RenderSettings.subtractiveShadowColor = Vector4.Lerp(shadowColour, settings.
                        realtimeShadowColour, timer);
127                 RenderSettings.ambientLight = Vector4.Lerp(ambientColour, settings.
                        ambientColour, timer);
128                 RenderSettings.fogColor = Vector4.Lerp(fogColour, settings.fogColour, timer)
                        ;
129                 RenderSettings.fogDensity = Mathf.Lerp(fogDensity, settings.fogDensity,
                        timer);
130                 yield return null;
131                 if (timer >= 1) {
132                     RenderSettings.subtractiveShadowColor = settings.realtimeShadowColour;
133                     RenderSettings.ambientLight = settings.ambientColour;
134                     RenderSettings.fogColor = settings.fogColour;
135                     RenderSettings.fogDensity = settings.fogDensity;
136                     currentPriority = 0;
137                 }
138             }
139         }
140 
141         private IEnumerator CrossFadeLight(Light light, ColourChangeSettings settings) {
142             float timer = 0;
143             while (timer < 1) {
144                 timer += Time.deltaTime / settings.crossFadeDuration;
145                 light.color = Vector4.Lerp(directionalColour, settings.
                        directionalLightColour, timer);
146                 light.intensity = Mathf.Lerp(directionalIntensity, settings.
                        directionalLightIntensity, timer);
147                 yield return null;
148                 if (timer >= 1) {
149                     light.color = settings.directionalLightColour;
150                     light.intensity = settings.directionalLightIntensity;
151                     currentPriority = 0;
152                 }
153             }
154         }
155         #endregion
156     }
157 }
```

ColourChangeSettings.cs

```
1  using UnityEngine;
2  
3  namespace MemoryMysteryGame {
4      [CreateAssetMenu(fileName = "ColourChangeSettings", menuName = "Game/Scriptable␣Objects/
           Colour␣Change␣Settings")]
5      public class ColourChangeSettings : ScriptableObject {
6  
7          [Tooltip("In␣cases␣of␣conflict␣between␣several␣ColourChangeSettings,␣one␣is␣chosen␣
               based␣on␣this␣number.␣Higher␣values␣=␣higher␣priority.")]
8          public int priority;
9  
10         [Tooltip("Duration␣for␣the␣crossfade␣defined␣in␣seconds")]
11         public float crossFadeDuration = 3;
12 
13         [Header("Player␣Sphere")]
```

```
        [ColorUsage(true, true)] public Color visibilitySphereColour = new Vector4(0, 0, 0, 1);

        [Header("Environment")]
        [ColorUsage(true, true)] public Color realtimeShadowColour = new Vector4(0, 0, 0, 1);
        [ColorUsage(true, true)] public Color ambientColour = new Vector4(0, 0, 0, 1);
        [ColorUsage(true, true)] public Color fogColour = new Vector4(0, 0, 0, 1);
        public float fogDensity = 0.008f;

        [Header("Directional Light")]
        [ColorUsage(true, true)] public Color directionalLightColour = new Vector4(0, 0, 0, 1);
        public float directionalLightIntensity = 1;
    }
}
```

ContinuationUI.cs

```
using System.Globalization;
using System.IO;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.InputSystem;
using TMPro;
using ScriptableEvents.Events;

namespace MemoryMysteryGame.Feedback {
    public class ContinuationUI : MonoBehaviour {
        [SerializeField] private SimpleScriptableEvent hideCursor;
        [SerializeField] private SimpleScriptableEvent showCursor;
        [SerializeField] private SimpleScriptableEvent pause;
        [SerializeField] private SimpleScriptableEvent unPause;
        [SerializeField] private InputActionAsset map;
        [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent likertEvent;
        [SerializeField] private GameObject errorSquareLikert;
        [SerializeField] private TextMeshProUGUI[] likertItem;
        [SerializeField] private Image[] likertObjects;
        [SerializeField] private GameObject errorSquare;
        [SerializeField] private GameObject[] entries;
        [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent formEntry;
        [SerializeField] private TextMeshProUGUI textField;
        [SerializeField] private Finish fin;

        private int selection = -1;
        private string likert = "";
        private string q2 = "";

        public void ShowUI() {
            Time.timeScale = 0;
            map.Disable();
            showCursor.Raise();
            pause.Raise();
            EnableDisable(true);
        }

        public void SelectLikert(int entry) {
            if (selection != -1)
                likertObjects[selection].color = Color.white;
            selection = entry;
            likertObjects[entry].color = Color.grey;
        }

        public void SubmitInstance() {
            if (selection == -1) {
                errorSquareLikert.SetActive(true);
                return;
            } else {
                errorSquareLikert.SetActive(false);
            }

            if (textField.text == null || textField.text.Length <= 2) {
```

```
                errorSquare.SetActive(true);
                return;
            } else {
                errorSquare.SetActive(false);
            }

            likert += "\n" + likertItem[selection].text;
            q2 += "\n" + textField.text;
            CultureInfo culture = new CultureInfo("en-GB");
            StreamWriter writer = new StreamWriter(Application.persistentDataPath + "/
                GFormValues.txt", true);
            writer.WriteLine(likertItem[selection].text);
            writer.WriteLine(textField.text + "␣[" + System.DateTime.UtcNow.ToString(culture
                ) + "]");
            writer.Close();
            writer.Dispose();
            ResumeGame();
        }

        public void SubmitAnswers() {
            likert = likert.Substring(1);
            q2 = q2.Substring(1);
            likertEvent.Raise(likert);
            formEntry.Raise(q2);
        }

        private void EnableDisable(bool value) {
            foreach (GameObject obj in entries) {
                obj.SetActive(value);
            }
        }

        private void ResumeGame() {
            likertObjects[selection].color = Color.white;
            selection = -1;
            textField.GetComponentInParent<TMP_InputField>().text = null;
            hideCursor.Raise();
            unPause.Raise();
            EnableDisable(false);
            map.Enable();
            Time.timeScale = 1;
            fin.FinishAndSend();
        }
    }
}
```

FeedbackScriptableEvent.cs

```
using UnityEngine;
using ScriptableEvents.Events;

namespace MemoryMysteryGame.ScriptableEvents.Events {
    [CreateAssetMenu(fileName = "FeedbackScriptableEvent", menuName = "Game/Scriptable␣
        Events/Feedback␣Scriptable␣Event")]
    public class FeedbackScriptableEvent : StringScriptableEvent {
        [SerializeField] private string entryName;

        new public void Raise(string value) {
            PlayerFeedback.GFormPost.Settings.AddSetValue(entryName, value);
        }
    }
}
```

FeedbackSelection.cs

```
using UnityEngine;
using UnityEngine.UI;
using TMPro;

namespace MemoryMysteryGame.Feedback {
    public class FeedbackSelection : MonoBehaviour {
        [SerializeField] private Image[] buttons;
        [SerializeField] private TextMeshProUGUI[] buttonTexts;
```

```
9
10          [HideInInspector] public string selectionText;
11
12          private int selection = -1;
13
14          public void SelectOption(int index) {
15              if (selection != -1)
16                  buttons[selection].color = Color.white;
17              selection = index;
18              buttons[selection].color = Color.grey;
19              selectionText = buttonTexts[selection].text;
20          }
21
22          public void ResetValues() {
23              if (selection != -1)
24                  buttons[selection].color = Color.white;
25              selection = -1;
26          }
27      }
28  }
```

FeedbackUI.cs

```
1   using UnityEngine;
2   using UnityEngine.UI;
3   using UnityEngine.InputSystem;
4   using TMPro;
5   using ScriptableEvents.Events;
6   using Sirenix.OdinInspector;
7
8   namespace MemoryMysteryGame.Feedback {
9       public class FeedbackUI : MonoBehaviour {
10          #region Editor
11          [FoldoutGroup("Simple␣Events␣and␣Keybinds")]
12          [SerializeField] private InputActionReference keybind;
13          [FoldoutGroup("Simple␣Events␣and␣Keybinds")]
14          [SerializeField] private InputActionAsset inputMap;
15          [FoldoutGroup("Simple␣Events␣and␣Keybinds")]
16          [SerializeField] private SimpleScriptableEvent hideCursor;
17          [FoldoutGroup("Simple␣Events␣and␣Keybinds")]
18          [SerializeField] private SimpleScriptableEvent showCursor;
19          [FoldoutGroup("Simple␣Events␣and␣Keybinds")]
20          [SerializeField] private SimpleScriptableEvent pause;
21          [FoldoutGroup("Simple␣Events␣and␣Keybinds")]
22          [SerializeField] private SimpleScriptableEvent resume;
23
24          [FoldoutGroup("Feedback␣Events")]
25          [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent textEvent;
26          [FoldoutGroup("Feedback␣Events")]
27          [SerializeField] private TextMeshProUGUI textField;
28          [FoldoutGroup("Feedback␣Events")]
29          [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent buttonEvent
                ;
30          [FoldoutGroup("Feedback␣Events")]
31          [SerializeField] private FeedbackSelection btnSelection;
32          [FoldoutGroup("Feedback␣Events")]
33          [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent posRotEvent
                ;
34          [FoldoutGroup("Feedback␣Events")]
35          [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent
                curMemoryEvent;
36          [FoldoutGroup("Feedback␣Events")]
37          [SerializeField] private ScriptableVariables.Variables.MemoryZoneScriptableVariable
                curMemory;
38          [FoldoutGroup("Feedback␣Events")]
39          [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent
                saveEntitiesEvent;
40          [FoldoutGroup("Feedback␣Events")]
41          [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent timeEvent;
42          [FoldoutGroup("Feedback␣Events")]
43          [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent
                curItemEvent;
44          [FoldoutGroup("Feedback␣Events")]
```

```
45          [SerializeField] private ScriptableVariables.Variables.
                MemoryInteractableScriptableVariable curItem;
46
47          [FoldoutGroup("UI␣Components")]
48          [SerializeField] private GameObject textSquare;
49          [FoldoutGroup("UI␣Components")]
50          [SerializeField] private GameObject typeSquare;
51          [FoldoutGroup("UI␣Components")]
52          [SerializeField] private GameObject[] uiObjects;
53          #endregion
54
55          #region Private Fields
56          private Cameras.CameraTransformProvider camProvider;
57          #endregion
58
59          #region Unity Lifecycle
60          private void Awake() {
61              camProvider = this.GetComponent<Cameras.CameraTransformProvider>();
62          }
63
64          private void OnEnable() {
65              keybind.action.performed -= ctx => ShowUI();
66              keybind.action.performed += ctx => ShowUI();
67          }
68
69          private void OnDisable() {
70              keybind.action.performed -= ctx => ShowUI();
71          }
72          #endregion
73
74          #region Public Methods
75          public void SubmitAnswers() {
76              textSquare.SetActive(false);
77              typeSquare.SetActive(false);
78
79              if (textField.text.Length > 2)
80                  textEvent.Raise(textField.text);
81              else {
82                  textSquare.SetActive(true);
83                  return;
84              }
85
86              if (btnSelection.selectionText.Length > 2)
87                  buttonEvent.Raise(btnSelection.selectionText);
88              else {
89                  typeSquare.SetActive(true);
90                  return;
91              }
92
93              posRotEvent.Raise(camProvider.GetPosRot());
94
95              if (curMemory.Value != null)
96                  curMemoryEvent.Raise(curMemory.Value.name);
97              else
98                  curMemoryEvent.Raise("Not␣currently␣in␣a␣memory");
99
100             saveEntitiesEvent.Raise("Not␣implemented␣yet");
101             timeEvent.Raise(Time.time.ToString());
102
103             if (curItem.Value != null)
104                 curItemEvent.Raise(curItem.Value.PrettyName);
105             else
106                 curItemEvent.Raise("No␣item␣held");
107
108             StartCoroutine(PlayerFeedback.GFormPost.SendForm());
109             ResumeGame();
110         }
111
112         public void ResumeGame() {
113             resume.Raise();
114             inputMap.Enable();
115             btnSelection.ResetValues();
116             hideCursor.Raise();
```

```
            DisableEnableObjects(false);
            Time.timeScale = 1;
        }
        #endregion

        #region Private Methods
        private void ShowUI() {
            pause.Raise();
            inputMap.Disable();
            showCursor.Raise();
            DisableEnableObjects(true);
            Time.timeScale = 0;
        }

        private void DisableEnableObjects(bool value) {
            foreach (GameObject obj in uiObjects) {
                obj.SetActive(value);
            }
        }
        #endregion
    }
}
```

Finish.cs

```
using UnityEngine;

namespace MemoryMysteryGame.Feedback {
    public class Finish : MonoBehaviour {
        [SerializeField] private ContinuationUI conUI;
        [SerializeField] private ShowQ2UI q2;

        private bool secondCont = false;

        public bool SecondCont {
            get { return secondCont; }
            set { secondCont = value; }
        }

        public void FinishAndSend() {
            if (secondCont) {
                conUI.SubmitAnswers();
                q2.SubmitAnswers();
                StartCoroutine(PlayerFeedback.GFormPost.SendForm());
            }
        }
    }
}
```

GFormPost.cs

```
using System.Collections;
using System.IO;
using UnityEngine;
using UnityEngine.Networking;

namespace PlayerFeedback {
        public static class GFormPost {

                public static bool isSending = false;

                private static GFormSettings settings;
                private static string url;

                public static GFormSettings Settings { get => settings; }

                public static void Initialise(GFormSettings settings) {
                        GFormPost.settings = settings;
                        if (settings.gFormURL.EndsWith("viewform")) {
                                url = settings.gFormURL.Substring(0, settings.gFormURL.
                                    LastIndexOf("viewform")) + "formResponse";
                        } else if (!settings.gFormURL.EndsWith("/")) {
                                url = settings.gFormURL + "/formResponse";
```

```
                        } else if (!settings.gFormURL.EndsWith("formResponse")) {
                                url = settings.gFormURL + "formResponse";
                        }
                }

                public static IEnumerator SendForm() {
                        isSending = true;
            if (settings.logPlayerPos) {
                PositionLogger.CloseWriter();
            }
                        WWWForm form = new WWWForm();

                        //Fill in entries
                        foreach (GFormSettings.gFormEntry entry in settings.gFormEntries) {
                                if (entry.multipleSelection) {
                                        foreach (string str in entry.multipleValue) {
                                                form.AddField(entry.entryID, str);
                                        }
                                } else {
                                        form.AddField(entry.entryID, entry.value);
                                }
                        }

                        //Read playerPos
            if (settings.logPlayerPos) {
                StreamReader playerPosReader = new StreamReader(GFormSettings.playerPosPath,
                      true);
                form.AddField(settings.playerPosEntryID, playerPosReader.ReadToEnd());
                playerPosReader.Close();
                PositionLogger.OpenWriter();
            }

                        //Send the form
                        UnityWebRequest www = UnityWebRequest.Post(url, form);
            Debug.Log("Sending form");
                        yield return www.SendWebRequest();
                        if (www.result == UnityWebRequest.Result.ConnectionError) {
                                Debug.LogError("Connection error " + www.result + " " + www.
                                     error);
                        } else if (www.result == UnityWebRequest.Result.DataProcessingError)
                             {
                                Debug.LogError("Data error " + www.result + " " + www.error)
                                     ;
                        }
                        Debug.Log("Form Sent " + www.result);
            www.Dispose();
                        isSending = false;
            UnityEngine.SceneManagement.SceneManager.LoadScene(4);
                }
        }
}
```

GFormSettings.cs

```
using System.Collections.Generic;
using UnityEngine;

namespace PlayerFeedback {
        [CreateAssetMenu(fileName = "GFormSetup", menuName = "ScriptableObjects/GFormSetup")
            ]
        public class GFormSettings : ScriptableObject {
                public static string playerPosPath;
                public string gFormURL;

        public bool logPlayerPos;
                public string playerPosEntryID;
                public gFormEntry[] gFormEntries;

                [System.Serializable]
                public struct gFormEntry {
            [Header("Entry")]
                        public string name;
                        public bool multipleSelection;
```

```
                        public string entryID;
                        [HideInInspector] public string value;
                        [HideInInspector] public List<string> multipleValue;
                }

        public void AddSetValue(string name, string value) {
            int index = System.Array.FindIndex(gFormEntries, entry => entry.name == name);
            if (index == -1) {
                Debug.LogError("Could not find " + name + " within gForm entries");
                return;
            }

            if (gFormEntries[index].multipleSelection && !gFormEntries[index].multipleValue.
                Contains(value))
                gFormEntries[index].multipleValue.Add(value);
            else if (!gFormEntries[index].multipleSelection)
                gFormEntries[index].value = value;
        }

        public void ResetValues() {
            for (int i = 0; i < gFormEntries.Length; i++) {
                gFormEntries[i].value = null;
                gFormEntries[i].multipleValue.Clear();
            }
        }
        }
}
```

ImmobilizeCamera.cs

```
using UnityEngine;

namespace MemoryMysteryGame.Cameras {
    public class ImmobilizeCamera : MonoBehaviour {
        [SerializeField] private Cinemachine.CinemachineInputProvider input;

        public void Enable() {
            input.enabled = true;
        }

        public void Disable() {
            input.enabled = false;
        }
    }
}
```

PositionLogger.cs

```
using System.Collections;
using System.Globalization;
using System.IO;
using System.Threading.Tasks;
using UnityEngine;

namespace PlayerFeedback {
        public class PositionLogger : MonoBehaviour {

        [SerializeField] private bool logOnEnable = false;
                [SerializeField] private GFormSettings settings;
                [SerializeField] private float delay = 0.1f;

                private static IEnumerator posLogger;
                private static StreamWriter writer;

                private void Awake() {
                        if (GFormSettings.playerPosPath == null || GFormSettings.
                            playerPosPath == "")
                                GFormSettings.playerPosPath = Application.persistentDataPath
                                        + "/playerPos.txt";

                        if (GFormPost.Settings == null && settings != null)
                                GFormPost.Initialise(settings);
                }
```

```
24
25                  private void OnEnable() {
26              if (logOnEnable) {
27                  OpenWriter();
28                  posLogger = LogPos();
29                  StartCoroutine(posLogger);
30              }
31                  }
32
33                  private void OnDisable() {
34                          StopCoroutine(posLogger);
35                          CloseWriter();
36                  }
37
38          public void StartLogging() {
39              OpenWriter();
40              posLogger = LogPos();
41              StartCoroutine(posLogger);
42          }
43
44          public void StopLogging() {
45              StopCoroutine(posLogger);
46              CloseWriter();
47          }
48
49                  private IEnumerator LogPos() {
50                          CultureInfo culture = new CultureInfo("en-GB");
51                          while (true) {
52                                  yield return new WaitForSeconds(delay);
53                                  if (writer != null) {
54                                          string line = "\n[" + System.DateTime.UtcNow.
                                                ToString(culture) + "]␣:␣" + this.transform.
                                                position;
55                                          try {
56                                                  writer.Write(line);
57                                          } catch (System.Exception e) {
58                                                  Debug.Log(e);
59                                          }
60                                  }
61                          }
62                  }
63
64                  public static async void OpenWriter() {
65                          await Task.Run(() => {
66                  while (GFormPost.isSending == true) {
67                      System.Threading.Thread.Sleep(100);
68                  }
69              });
70                          writer = new StreamWriter(GFormSettings.playerPosPath, true);
71                  }
72
73                  public static void CloseWriter() {
74                          writer.Close();
75                          writer.Dispose();
76                  }
77          }
78  }
```

QuitGame.cs

```
1   using UnityEngine;
2
3   namespace MemoryMysteryGame {
4       public class QuitGame : MonoBehaviour {
5
6           public void Quit() {
7   #if UNITY_EDITOR
8               UnityEditor.EditorApplication.isPlaying = false;
9   #else
10              Application.Quit();
11  #endif
12          }
13
```

```
        private float timer = 0;
        private float delay = 5;

        private void Update() {
            timer += Time.deltaTime;
            if (timer > delay) {
                timer -= delay;
                Quit();
            }
        }
    }
}
```

SendForm.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

namespace MemoryMysteryGame.Feedback {
    public class SendForm : MonoBehaviour {
        public void SendGoogleForm() {
            PlayerFeedback.GFormPost.SendForm();
        }
    }
}
```

ShowOtherUI.cs

```
using System.Globalization;
using System.IO;
using UnityEngine;
using UnityEngine.InputSystem;
using TMPro;
using ScriptableEvents.Events;

namespace MemoryMysteryGame.Feedback {
    public class ShowOtherUI : MonoBehaviour {
        [SerializeField] private SimpleScriptableEvent hideCursor;
        [SerializeField] private SimpleScriptableEvent showCursor;
        [SerializeField] private SimpleScriptableEvent pause;
        [SerializeField] private SimpleScriptableEvent unPause;
        [SerializeField] private InputActionAsset map;
        [SerializeField] private GameObject errorSquare;
        [SerializeField] private GameObject[] entries;
        [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent formEntry;
        [SerializeField] private TextMeshProUGUI textField;
        [SerializeField] private ContinuationUI contUI;

        public void ShowUI() {
            Time.timeScale = 0;
            map.Disable();
            showCursor.Raise();
            pause.Raise();
            EnableDisable(true);
        }

        public void SubmitAnswer() {
            if (textField.text != null && textField.text.Length > 2) {
                CultureInfo culture = new CultureInfo("en-GB");
                errorSquare.SetActive(false);
                formEntry.Raise(textField.text + "␣[" + System.DateTime.UtcNow.ToString(
                    culture) + "]");
                StreamWriter writer = new StreamWriter(Application.persistentDataPath + "/
                    GFormValues.txt", true);
                writer.WriteLine(textField.text + "␣[" + System.DateTime.UtcNow.ToString(
                    culture) + "]");
                writer.Close();
                writer.Dispose();
                ResumeGame();
            } else {
                errorSquare.SetActive(true);
            }
```

```
        }

        private void EnableDisable(bool value) {
            foreach (GameObject obj in entries) {
                obj.SetActive(value);
            }
        }

        private void ResumeGame() {
            if (contUI != null) {
                EnableDisable(false);
                contUI.ShowUI();
            } else {
                hideCursor.Raise();
                unPause.Raise();
                EnableDisable(false);
                map.Enable();
                Time.timeScale = 1;
            }
        }
    }
}
```

ShowQ2UI.cs

```
using System.IO;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.InputSystem;
using TMPro;
using ScriptableEvents.Events;

namespace MemoryMysteryGame.Feedback {
    public class ShowQ2UI : MonoBehaviour {
        [SerializeField] private SimpleScriptableEvent hideCursor;
        [SerializeField] private SimpleScriptableEvent showCursor;
        [SerializeField] private SimpleScriptableEvent pause;
        [SerializeField] private SimpleScriptableEvent unPause;
        [SerializeField] private InputActionAsset map;
        [SerializeField] private ScriptableVariables.Variables.BoolScriptableVariable
            boolVariable;
        [SerializeField] private ScriptableVariables.Variables.BoolScriptableVariable
            activeVariable;
        [SerializeField] private ScriptableVariables.Variables.
            MemoryInteractableScriptableVariable curItem;
        [SerializeField] private ScriptableVariables.Variables.
            MemoryInteractableScriptableVariable hoverItem;
        [SerializeField] private GameObject[] entries;
        [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent likertEvent
            ;
        [SerializeField] private GameObject errorSquareLikert;
        [SerializeField] private TextMeshProUGUI likertQuestion;
        [SerializeField] private TextMeshProUGUI[] likertItem;
        [SerializeField] private Image[] likertObjects;
        [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent
            q2Entry2Event;
        [SerializeField] private GameObject errorSquare2;
        [SerializeField] private TextMeshProUGUI q2Question;
        [SerializeField] private TextMeshProUGUI input2Text;
        [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent
            q2Entry3Event;
        [SerializeField] private GameObject errorSquare3;
        [SerializeField] private TextMeshProUGUI input3Text;

        private int selection = -1;
        private string likert = "";
        private string q2 = "";
        private string q3 = "";

        private void OnEnable() {
            boolVariable.OnChanged -= (oldV, newV) => { if (newV) ShowUI(); };
            boolVariable.OnChanged += (oldV, newV) => { if (newV) ShowUI(); };
        }
```

```
42
43          private void OnDisable() {
44              boolVariable.OnChanged -= (oldV, newV) => { if (newV) ShowUI(); };
45          }
46
47          public void ShowUI() {
48              if (!activeVariable.Value) {
49                  boolVariable.Value = false;
50                  return;
51              }
52              Time.timeScale = 0;
53              map.Disable();
54              likertQuestion.text = "How␣confident␣are␣you␣that␣putting\n" + "'" + curItem.
                    Value.PrettyName + "'" + "␣here␣is␣correct?";
55              q2Question.text = "Describe␣why␣you␣in␣general␣chose\n" + "'" + curItem.Value.
                    PrettyName + "'";
56              pause.Raise();
57              showCursor.Raise();
58              EnableDisable(true);
59          }
60
61          public void SubmitInstance() {
62              if (selection != -1) {
63                  errorSquareLikert.SetActive(false);
64              } else {
65                  errorSquareLikert.SetActive(true);
66                  return;
67              }
68
69              if (input2Text.text != null && input2Text.text.Length > 2) {
70                  errorSquare2.SetActive(false);
71              } else {
72                  errorSquare2.SetActive(true);
73                  return;
74              }
75
76              if (input3Text.text != null && input3Text.text.Length > 2) {
77                  errorSquare3.SetActive(false);
78              } else {
79                  errorSquare3.SetActive(true);
80                  return;
81              }
82
83              likert += "\n" + likertItem[selection].text;
84              q2 += "\n" + input2Text.text + "\nHeld␣item␣was:␣" + curItem.Value.PrettyName +
                    "\nTrying␣to␣replace:␣" + hoverItem.Value.MemoryInteractableId + "\nEntry␣
                    done.";
85              q3 += "\n" + input3Text.text;
86              StreamWriter writer = new StreamWriter(Application.persistentDataPath + "/
                    GFormValues.txt", true);
87              writer.WriteLine(likertItem[selection].text);
88              writer.WriteLine(input2Text.text + "␣Held␣item␣was:␣" + curItem.Value.PrettyName
                     + "␣Trying␣to␣replace:␣" + hoverItem.Value.MemoryInteractableId);
89              writer.WriteLine(input3Text.text);
90              writer.Close();
91              writer.Dispose();
92              ResumeGame();
93          }
94
95          public void SubmitAnswers() {
96              likert = likert.Substring(1);
97              q2 = q2.Substring(1);
98              q3 = q3.Substring(1);
99              likertEvent.Raise(likert);
100             q2Entry2Event.Raise(q2);
101             q2Entry3Event.Raise(q3);
102         }
103
104         public void SelectLikert(int entry) {
105             if (selection != -1)
106                 likertObjects[selection].color = Color.white;
107             selection = entry;
108             likertObjects[entry].color = Color.grey;
```

```
        }

        private void EnableDisable(bool value) {
            foreach (GameObject obj in entries) {
                obj.SetActive(value);
            }
        }

        private void ResumeGame() {
            likertObjects[selection].color = Color.white;
            selection = -1;
            input2Text.GetComponentInParent<TMP_InputField>().text = null;
            input3Text.GetComponentInParent<TMP_InputField>().text = null;
            input2Text.GetComponentInParent<TMP_InputField>().ForceLabelUpdate();
            input3Text.GetComponentInParent<TMP_InputField>().ForceLabelUpdate();
            hideCursor.Raise();
            unPause.Raise();
            EnableDisable(false);
            map.Enable();
            Time.timeScale = 1;
            boolVariable.Value = false;
        }
    }
}
```

SoundCollision.cs

```
using UnityEngine;

namespace MemoryMysteryGame.Audio {
    public class SoundCollision : MonoBehaviour {

        #region Editor
        [SerializeField] private Rigidbody rb;
        [SerializeField] private string materialName = "Collision Material";
        [SerializeField] private string[] tagNames;
        [SerializeField] private string speedName = "Player Speed";
        [SerializeField] private float minSpeed;
        [SerializeField] private float maxSpeed;
        [SerializeField] private string audioPath = "event:/MOVEMENT events/Collision";
        #endregion

        #region Fields
        private float normalizedSpeed;
        private FMOD.Studio.EventInstance track;
        #endregion

        #region Unity Lifecycle
        private void Start() {
            if (rb == null) {
                rb = GetComponentInParent<Transform>().GetComponentInParent<Rigidbody>();
            }
            track = FMODUnity.RuntimeManager.CreateInstance(audioPath);
        }

        private void OnTriggerEnter(Collider col) {
            for (int i = 0; i < tagNames.Length; i++) {
                if (col.CompareTag(tagNames[i])) {
                    if (minSpeed > maxSpeed) {
                        float tempSpeed = minSpeed;
                        minSpeed = maxSpeed;
                        maxSpeed = tempSpeed;
                    }
                    float vel = rb.velocity.magnitude;
                    if (vel < minSpeed) {
                        normalizedSpeed = 0;
                    } else if (vel > maxSpeed) {
                        normalizedSpeed = 1;
                    } else {
                        vel -= minSpeed;
                        normalizedSpeed = vel / (maxSpeed - minSpeed);
                    }

```

```
                    float value = i;
                    track.setParameterByName(materialName, value);
                    track.setParameterByName(speedName, normalizedSpeed);

                    track.start();
                    track.release();

                    ReinitializeTrack();

                    //FMODUnity.RuntimeManager.PlayOneShot(audioPath, collision.gameObject.
                        transform.position);
                }
            }
        }
        #endregion

        #region Private Methods
        private void ReinitializeTrack() {
            track = FMODUnity.RuntimeManager.CreateInstance(audioPath);
        }
        #endregion
    }
}
```

TextToForm.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

namespace MemoryMysteryGame.Feedback {
    public class TextToForm : MonoBehaviour {
        [SerializeField] private ScriptableEvents.Events.FeedbackScriptableEvent
            feedbackEvent;
        [SerializeField] private TextMeshProUGUI textObject;

        public void SendText() {
            feedbackEvent.Raise(textObject.text);
        }
    }
}
```

TutorialEntry.cs

```
using UnityEngine;
using UnityEngine.Localization;
using ScriptableEvents.Events;
using Sirenix.OdinInspector;

namespace MemoryMysteryGame.Tutorial
{
    [CreateAssetMenu(
        fileName = "TutorialEntry",
        menuName = "Game/Tutorial/Tutorial Entry"
    )]
    public class TutorialEntry : ScriptableObject
    {
        #region Editor

        [Tooltip("Event to raise when this entry is reached")]
        [SerializeField]
        public SimpleScriptableEvent raiseEvent;

        [FoldoutGroup("Keyboard Content")]
        [Tooltip("Text to show when using keyboard and mouse")]
        [SerializeField]
        public LocalizedString keyboardString;

        [FoldoutGroup("Keyboard Content")]
        [Tooltip("Image to show when using keyboard and mouse")]
        [SerializeField]
        public Texture2D keyboardImage;
```

```
29
30          [FoldoutGroup("Keyboard Content")]
31          [Tooltip("Height and width of image")]
32          [SerializeField]
33          public Vector2 keyboardImageSize;
34
35          [FoldoutGroup("Controller Content")]
36          [Tooltip("Text to show when using controller")]
37          [SerializeField]
38          public LocalizedString controllerString;
39
40          [FoldoutGroup("Controller Content")]
41          [Tooltip("Image to show when using controller")]
42          [SerializeField]
43          public Texture2D controllerImage;
44
45          [FoldoutGroup("Controller Content")]
46          [Tooltip("Height and width of image")]
47          [SerializeField]
48          public Vector2 controllerImageSize;
49
50          [FoldoutGroup("Hiding")]
51          [Tooltip("Whether the entry should hide automatically after a set amount of seconds")]
52          [SerializeField]
53          public bool hideAfterSeconds;
54
55          [FoldoutGroup("Hiding")]
56          [Tooltip("How long to show the entry for, if hideAfterSeconds is enabled")]
57          [SerializeField]
58          public float secondsToHide;
59
60          [FoldoutGroup("Hiding")]
61          [Tooltip("Whether the TutorialManager should progress to the next entry in the list instead of hiding the entry, when the timer runs out")]
62          [SerializeField]
63          public bool progressInstead;
64
65
66          #endregion
67      }
68  }
```

Old TutorialManager.cs

```
1   using System.Collections;
2   using UnityEngine;
3   using UnityEngine.UI;
4   using UnityEngine.InputSystem;
5   using UnityEngine.Localization;
6   using ScriptableEvents.Events;
7   using TMPro;
8
9   namespace MemoryMysteryGame {
10      public class TutorialManager : MonoBehaviour {
11
12          [SerializeField] private TextMeshProUGUI text;
13          [SerializeField] private RawImage image;
14          [SerializeField] private TutorialEntry[] tutorialEntries;
15          [SerializeField] private TutorialEntry[] oneShotEntries;
16
17          private Texture2D transparent;
18          private TutorialEntry latestEntry;
19          private int controlIndex = 0;
20          private int index = 0;
21          private float timer = 0;
22          private IEnumerator hideAfterSeconds;
23
24          [System.Serializable]
25          public struct TutorialEntry {
26              public bool hideAfterSeconds;
27              public bool progressInstead;
28              public float secondsToHide;
```

```
29              public LocalizationStruct instruction;
30              public SimpleScriptableEvent audio;
31              public GameObject listenerObject;
32          }
33
34          [System.Serializable]
35          public struct LocalizationStruct {
36              public LocalizedString kbm;
37              public Texture2D kbmImage;
38              public LocalizedString controller;
39              public Texture2D controllerImage;
40          }
41
42          private void OnEnable() {
43              ShowCurrentTutorial();
44          }
45
46          private void OnDisable() {
47              RemoveListeners();
48          }
49
50          private void Awake() {
51              transparent = new Texture2D(1, 1);
52              for (int i = 0; i < transparent.height; i++) {
53                  for (int j = 0; j < transparent.width; j++) {
54                      transparent.SetPixel(i, j, Color.clear);
55                  }
56              }
57              transparent.Apply();
58              hideAfterSeconds = HideAfterSeconds(0, false);
59              if (tutorialEntries.Length != 0)
60                  latestEntry = tutorialEntries[0];
61          }
62
63          public void CheckControlType(PlayerInput input) {
64              switch (input.currentControlScheme) {
65                  case "Keyboard&Mouse":
66                      controlIndex = 0;
67                      break;
68                  case "Gamepad":
69                      controlIndex = 1;
70                      break;
71              }
72              RemoveListeners();
73              ShowByControlType(latestEntry);
74          }
75
76          //Main function to give a slideshow type of tutorial
77          public void ProgressTutorial() {
78              if (index >= tutorialEntries.Length) {
79                  text.gameObject.SetActive(false);
80                  image.gameObject.SetActive(false);
81                  if (!latestEntry.hideAfterSeconds && latestEntry.listenerObject != null)
82                      latestEntry.listenerObject.SetActive(false);
83                  return;
84              }
85
86              NextSetup();
87
88              TutorialEntry entry = tutorialEntries[index];
89
90              ShowByControlType(entry);
91
92              if (entry.audio != null) {
93                  entry.audio.Raise();
94              }
95
96              if (!entry.hideAfterSeconds && entry.listenerObject != null) {
97                  entry.listenerObject.SetActive(true);
98              } else if (entry.hideAfterSeconds) {
99                  hideAfterSeconds = HideAfterSeconds(entry.secondsToHide, entry.
                        progressInstead);
100                 StartCoroutine(hideAfterSeconds);
```

```
            }

            latestEntry = entry;
            index++;
        }

        //Function to show a specific "slide" of the tutorial
        public void ShowByIndex(int idx) {
            if (idx >= tutorialEntries.Length) {
                text.gameObject.SetActive(false);
                image.gameObject.SetActive(false);
                if (!latestEntry.hideAfterSeconds && latestEntry.listenerObject != null)
                    latestEntry.listenerObject.SetActive(false);
                return;
            }

            NextSetup();

            TutorialEntry entry = tutorialEntries[idx];

            ShowByControlType(entry);

            if (entry.audio != null) {
                entry.audio.Raise();
            }

            if (entry.hideAfterSeconds) {
                hideAfterSeconds = HideAfterSeconds(entry.secondsToHide, entry.
                    progressInstead);
                StartCoroutine(hideAfterSeconds);
            }

            latestEntry = entry;
        }

        //Function to set tutorial image and text back to the current "slide"
        public void ShowCurrentTutorial() {
            if (index >= tutorialEntries.Length) {
                text.gameObject.SetActive(false);
                image.gameObject.SetActive(false);
                if (!latestEntry.hideAfterSeconds && latestEntry.listenerObject != null)
                    latestEntry.listenerObject.SetActive(false);
                return;
            }

            NextSetup();

            TutorialEntry entry = tutorialEntries[index];

            ShowByControlType(entry);

            if (!entry.hideAfterSeconds && entry.listenerObject != null) {
                entry.listenerObject.SetActive(true);
            } else if (entry.hideAfterSeconds) {
                hideAfterSeconds = HideAfterSeconds(entry.secondsToHide, entry.
                    progressInstead);
                StartCoroutine(hideAfterSeconds);
            }

            latestEntry = entry;
        }

        //Function to hide the tutorial
        public void HideTutorial() {
            text.gameObject.SetActive(false);
            image.gameObject.SetActive(false);
            RemoveListeners();
        }

        public void OneShotByIndex(int idx) {
            if (idx >= oneShotEntries.Length) {
                text.gameObject.SetActive(false);
                image.gameObject.SetActive(false);
```

```
172                 if (!latestEntry.hideAfterSeconds && latestEntry.listenerObject != null)
173                     latestEntry.listenerObject.SetActive(false);
174                 return;
175             }
176 
177             NextSetup();
178 
179             TutorialEntry entry = oneShotEntries[idx];
180 
181             ShowByControlType(entry);
182 
183             if (entry.audio != null) {
184                 entry.audio.Raise();
185             }
186 
187             if (!entry.hideAfterSeconds && entry.listenerObject != null) {
188                 entry.listenerObject.SetActive(true);
189             } else if (entry.hideAfterSeconds) {
190                 hideAfterSeconds = HideAfterSeconds(entry.secondsToHide, entry.
                        progressInstead);
191                 StartCoroutine(hideAfterSeconds);
192             }
193 
194             latestEntry = entry;
195         }
196 
197         private void ShowByControlType(TutorialEntry entry) {
198             if (controlIndex == 0) {
199                 if (!entry.instruction.kbm.TableEntryReference.ToString().Contains("Empty"))
                        {
200                     entry.instruction.kbm.StringChanged += UpdateText;
201                 } else {
202                     text.text = "";
203                 }
204                 if (entry.instruction.kbmImage) {
205                     image.texture = entry.instruction.kbmImage;
206                 } else {
207                     image.texture = transparent;
208                 }
209             } else if (controlIndex == 1) {
210                 if (!entry.instruction.controller.TableEntryReference.ToString().Contains("
                        Empty")) {
211                     entry.instruction.controller.StringChanged += UpdateText;
212                 } else {
213                     text.text = "";
214                 }
215                 if (entry.instruction.controllerImage) {
216                     image.texture = entry.instruction.controllerImage;
217                 } else {
218                     image.texture = transparent;
219                 }
220             }
221         }
222 
223         private void NextSetup() {
224             if (latestEntry.listenerObject != null)
225                 latestEntry.listenerObject.SetActive(false);
226 
227             StopCoroutine(hideAfterSeconds);
228             timer = 0;
229 
230             text.gameObject.SetActive(true);
231             image.gameObject.SetActive(true);
232 
233             RemoveListeners();
234         }
235 
236         private IEnumerator HideAfterSeconds(float seconds, bool progress) {
237             while (true) {
238                 yield return null;
239                 timer += Time.deltaTime;
240                 if (timer > seconds) {
241                     timer = 0;
```

```
                    StopCoroutine(hideAfterSeconds);
                    if (progress)
                        ProgressTutorial();
                    else
                        HideTutorial();
                }
            }
        }

        private void UpdateText(string s) {
            text.text = s;
        }

        private void RemoveListeners() {
            latestEntry.instruction.kbm.StringChanged -= UpdateText;
            latestEntry.instruction.controller.StringChanged -= UpdateText;
        }
    }
}
```

TutorialManager.cs

```
using System.Collections;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.InputSystem;
using UnityEngine.Localization;
using ScriptableEvents.Events;
using TMPro;
using MemoryMysteryGame.Saving;

namespace MemoryMysteryGame.Tutorial {
    public class TutorialManager : MonoBehaviour, ISaveable {

        #region Editor


        [SerializeField] private RectTransform textPanel;
        [SerializeField] private TextMeshProUGUI textDisplay;

        [SerializeField] private RawImage imageDisplay;
        [SerializeField] private TutorialEntry[] tutorialEntries;

        #endregion

        #region Fields

        private Texture2D transparent;
        private TutorialEntry latestEntry;
        private int controlIndex = 0;
        private int index = 0;
        private float timer = 0;
        private IEnumerator hideAfterSeconds;

        #endregion

        #region Unity Lifecycle

        private void OnEnable() {
            ShowCurrentTutorial();
        }

        private void OnDisable() {
            RemoveListeners();
        }

        private void Awake() {
            transparent = new Texture2D(1, 1);
            for (int i = 0; i < transparent.height; i++) {
                for (int j = 0; j < transparent.width; j++) {
                    transparent.SetPixel(i, j, Color.clear);
                }
            }
```

```
52              transparent.Apply();
53              hideAfterSeconds = HideAfterSeconds(0, false);
54              if (tutorialEntries.Length != 0)
55                  latestEntry = tutorialEntries[0];
56          }
57
58          #endregion
59
60          #region Public Methods
61
62          public void CheckControlType(PlayerInput input) {
63              switch (input.currentControlScheme) {
64                  case "Keyboard&Mouse":
65                      controlIndex = 0;
66                      break;
67                  case "Gamepad":
68                      controlIndex = 1;
69                      break;
70              }
71              RemoveListeners();
72              ShowByControlType(latestEntry);
73          }
74
75          //Main function to give a slideshow type of tutorial
76          public void ProgressTutorial() {
77              if (index >= tutorialEntries.Length) {
78                  textPanel.gameObject.SetActive(false);
79                  imageDisplay.gameObject.SetActive(false);
80              }
81
82              TutorialEntry entry = tutorialEntries[index];
83              DisplayEntry(entry);
84
85              index++;
86          }
87
88          //Function to show a specific "slide" of the tutorial
89          public void ShowByIndex(int idx) {
90              if (idx >= tutorialEntries.Length) {
91                  textPanel.gameObject.SetActive(false);
92                  imageDisplay.gameObject.SetActive(false);
93                  return;
94              }
95
96              TutorialEntry entry = tutorialEntries[idx];
97              DisplayEntry(entry);
98          }
99
100         //Function to set tutorial image and text back to the current "slide"
101         public void ShowCurrentTutorial() {
102             if (index >= tutorialEntries.Length) {
103                 textPanel.gameObject.SetActive(false);
104                 imageDisplay.gameObject.SetActive(false);
105                 return;
106             }
107
108             TutorialEntry entry = tutorialEntries[index];
109             DisplayEntry(entry, false);
110         }
111
112         //Function to hide the tutorial
113         public void HideTutorial() {
114             textPanel.gameObject.SetActive(false);
115             imageDisplay.gameObject.SetActive(false);
116             RemoveListeners();
117         }
118
119         public void OneShot(TutorialEntry entry) {
120             DisplayEntry(entry);
121         }
122
123         public object CaptureState()
124         {
```

```
125             return index;
126         }
127 
128         public void RestoreState(object state)
129         {
130             int savedIndex = (int)state;
131 
132             index = savedIndex;
133 
134             ShowCurrentTutorial();
135         }
136 
137         #endregion
138 
139         #region Private Methods
140 
141         private void DisplayEntry(TutorialEntry entry, bool playEvent = true) {
142             NextSetup();
143 
144             ShowByControlType(entry);
145 
146             if (entry.raiseEvent != null && playEvent) {
147                 entry.raiseEvent.Raise();
148             }
149 
150             if (entry.hideAfterSeconds) {
151                 hideAfterSeconds = HideAfterSeconds(entry.secondsToHide, entry.
                        progressInstead);
152                 StartCoroutine(hideAfterSeconds);
153             }
154 
155             latestEntry = entry;
156         }
157 
158         private void ShowByControlType(TutorialEntry entry) {
159             if (controlIndex == 0) {
160                 if (!entry.keyboardString.TableEntryReference.ToString().Contains("Empty"))
                        {
161                     entry.keyboardString.StringChanged += UpdateText;
162                 } else {
163                     textPanel.gameObject.SetActive(false);
164                     textDisplay.text = "";
165                 }
166                 if (entry.keyboardImage) {
167                     imageDisplay.texture = entry.keyboardImage;
168                     imageDisplay.rectTransform.sizeDelta = entry.keyboardImageSize;
169                 } else {
170                     imageDisplay.texture = transparent;
171                 }
172             } else if (controlIndex == 1) {
173                 if (!entry.controllerString.TableEntryReference.ToString().Contains("Empty")
                        ) {
174                     entry.controllerString.StringChanged += UpdateText;
175                 } else {
176                     textPanel.gameObject.SetActive(false);
177                     textDisplay.text = "";
178                 }
179                 if (entry.controllerImage) {
180                     imageDisplay.texture = entry.controllerImage;
181                     imageDisplay.rectTransform.sizeDelta = entry.controllerImageSize;
182                 } else {
183                     imageDisplay.texture = transparent;
184                 }
185             }
186         }
187 
188         private void NextSetup() {
189             StopCoroutine(hideAfterSeconds);
190             timer = 0;
191 
192             imageDisplay.gameObject.SetActive(true);
193 
194             RemoveListeners();
```

```
        }

        private IEnumerator HideAfterSeconds(float seconds, bool progress) {
            while (true) {
                yield return null;
                timer += Time.deltaTime;
                if (timer > seconds) {
                    timer = 0;
                    StopCoroutine(hideAfterSeconds);
                    if (progress)
                        ProgressTutorial();
                    else
                        HideTutorial();
                }
            }
        }

        private void UpdateText(string s) {
            textPanel.gameObject.SetActive(true);
            textDisplay.text = s;
        }

        private void RemoveListeners() {
            latestEntry.keyboardString.StringChanged -= UpdateText;
            latestEntry.controllerString.StringChanged -= UpdateText;
        }

        #endregion
    }
}
```

UpdateSetupObject.cs

```
using UnityEngine;

namespace PlayerFeedback {
    public class UpdateSetupObject : MonoBehaviour {
        [SerializeField] private GFormSettings settings;
        [SerializeField] private bool onAwake = false;

        private void Awake() {
            if (onAwake)
                GFormPost.Initialise(settings);
        }

        public void UpdateObject() {
            GFormPost.Initialise(settings);
        }
    }
}
```