PlayGuide: A Visual Gaze-Based Player Guidance System for Three-Dimensional Computer Games

Andreas Haslinger

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im Juli 2018

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, July 1, 2018

Andreas Haslinger

Contents

Abstract

Gaze direction is a technique that relies on visual cues that involuntarily pull the eyes of the viewer towards a specific region on an analog or digital screen. Game designers and developers use overt visual stimuli, such as contrasting colors, movements, shapes or anything else that will attract the attention of the player, to direct the gaze of the player to important objects in the game, often disregarding the impact of overt visual cues on factors like competence, immersion and flow. Up until now, however, no alternative solution was known to ensure unobtrusive gaze direction in games. *PlayGuide*, the visual gaze-based player guidance system for three-dimensional computer games, developed in the course of this master's thesis uses a novel technique called subtle gaze direction that is capable of unnoticeably guiding the viewer's eyes anywhere on the screen. Twenty participants tested *PlayGuide* and evaluated the potential capabilities of the player guidance system. Not only did *PlayGuide* turn out to be successful in guiding the player's gaze but also, when the overall aim was to explore the environment of the game, players preferred *PlayGuide* over traditional visual cues used in computer games. Furthermore, players spent almost twice as much time exploring the environment of the game, while benefiting subjective impressions of the players, such as immersion and flow.

Kurzfassung

Blickführung stellt eine Technik dar, die visuelle Hinweise nutzt, um die Augen eines Betrachters in Richtung einer bestimmten Region innerhalb eines digitalen oder analogen Bildschirmes zu ziehen. Spieledesigner und -entwickler nutzen auffällige visuelle Reize, wie zum Beispiel kontrastierende Farben, Bewegungen oder Formen, um den Blick des Spielers auf wichtige Objekte im Spiel aufmerksam zu machen. Dabei wird oft vernachlässigt, dass sich diese visuellen Reize auf Faktoren wie Kompetenz, Immersion und Flow nachteilig auswirken können. Allerdings waren bis heute keine alternativen Lösungen bekannt, um eine unaufdringliche Blickführung zu gewährleisten. *PlayGuide*, das visuelle, blickbasierte Spielerführungssystem für dreidimensionale Computerspiele, welches im Zuge dieser Masterarbeit entwickelt wurde, verwendet eine neuartige subtile Blickrichtungstechnik, die es ermöglicht, die Augen des Spielers unbemerkt zu steuern. In der vorliegenden Masterarbeit wurde *PlayGuide* von 20 Teilnehmern getestet, die die potenzielle Leistungsfähigkeit des Spielführungssystems evaluierten. Dabei stellte sich heraus, dass *PlayGuide* nicht nur erfolgreich darin war, den Blick des Spielers zu leiten, sondern auch, wenn es Ziel des Spiels war die Umgebung zu erkunden, dass Spieler *PlayGuide* den traditionellen visuellen Hinweisen in Computerspielen vorzogen. Darüber hinaus verbrachten die Spieler doppelt so lange damit, die Umgebung im Spiel zu erkunden, während die subjektiven Eindrücke der Spieler, wie Immersion und Flow profitierten.

Chapter 1

Introduction

Our gaze is guided by attention. Wherever our attention is drawn to, our eyes will follow in order to be able to better explore what caught our attention in the first place. A phenomenon that is exploited by researchers to guide our gaze, which can be useful to focus our attention on things that matter, such as a single piece of important information within a flood of visual sensory impressions. In the online business there is even a "market for consumer attention," which, according to Teixeira from the *Harvard Business School*, "has become so competitive that attention can be regarded as a currency" [74, p. 1], driving marketeers to invent and research increasingly aggressive methods to compete for the attention of the customer. These methods are usually overt visual triggers that pull the eyes towards a specific location on the screen with contrasting colors, movement, shapes or anything else that will draw the attention of the viewer. However, Bailey et al. [1] propose a novel and different solution that abandons these simple overt image modifications and offer a different approach. By exploiting the basic characteristics of the human visual system and the use of an eye-tracking device they are capable of directing the viewer's gaze by merely applying subtle image modulations to an image that are terminated before the viewer can scrutinize them. In other words, the viewer will not be able to accurately determine what guided his or her gaze in the first place. In their experiments, the technique, coined *subtle gaze direction*, proved to be highly successful in static images and Bailey et al. concluded that this subtle manner of guiding the viewer's gaze might be applicable to other areas as well.

PlayGuide, proposed in this master's thesis, is a visual gaze-based player guidance system for three-dimensional computer games that applies the idea of *subtle gaze direction* to a highly dynamic context and builds upon the basic principles first proposed by Bailey et al. While playing computer games, just like any other activity that involves a screen, the viewer is flooded with visual sensory impressions that need to be processed and evaluated to act accordingly. To single out important information and to offer unobtrusive assistance during game play, *PlayGuide* applies subtle visual cues within the scene to guide the player's gaze towards important game objects.

1.1 Background

In order for *subtle gaze direction* to perform accurately, the position of the viewer's gaze on the screen must be known. Although eye-tracking devices were found mostly in research labs only a couple of years ago, smaller and cheaper sensors made consumer version eye trackers possible. Companies like *Tobii* [72] create hardware that is priced more inexpensively and is smaller in size than eye trackers used for research, yet precise enough to provide gaze data that can be used for analysis. Additionally, advanced consumer eye trackers do not only provide gaze points on the screen but also more detailed information such as the rotation of the head and the distance of the player to the screen [57]. Although the information coming from eye-tracking devices is very precise, it still needs to be filtered and processed before it can be used for further analysis or player input. This is not only because of a certain inaccuracy of the hardware but also due to the nature of how our eyes work [43, p. 544].

Subtle gaze direction and other gaze-dependent gaze direction techniques, such as *overt gaze direction*, also consider the nature of how our eyes work by making use of how the human visual system processes image information. While we are capable of sensing great image details in the fovea, this integral part of the eye only makes up a small central part of our vision. The rest (roughly 98%) is made up of the peripheral vision, which performs badly at sensing image details but is highly sensible towards detecting motion [44]. In their work, Bailey et al. [1] use this circumstance to create motion by subtly modifying a region of an image that is outside the foveal vision of the viewer but hence highly discernible by the motion-sensitive peripheral vision. However, since the peripheral vision only receives a blurred view of the image, the eyes must relocate the fovea to get a clear vision of the region that is being modulated. As soon as the eyes perform a saccade (rapid positional change) towards the modified part of the image, the modulation is terminated and the viewer does not know what attracted his or her attention in the first place. This is not only beneficial to the user, who is guided throughout an image without any discernible stimuli but also for researchers, educators, marketeers and developers who can use this technology to reap the benefits of *subtle gaze direction* by applying it to unobtrusively, but effectively, guide the gaze of the viewer (more on which can be read in Section 4.2.2).

Although Bailey et al. only assessed the performance of *subtle gaze direction* in a static context, they mentioned that it might perform equally as well in a dynamic context [1]. While there are experiments involving video footage [39] and simple virtual reality scenes [59], there is no evaluation of the performance of *subtle gaze direction* in a truly complex dynamic setting such as computer games, which strongly depend on the adequate guidance of their players. Especially in level design it is crucial to support players in orienting themselves quickly and effectively within the environment, so that they do not get stuck and frustrated while playing the game [66]. This master's thesis explores the possibility of *subtle gaze direction* as an unobtrusive solution to assist players during game play.

1.2 Objectives

The technique of directing the viewers' gaze without them knowing has already been evaluated in static settings, such as a set of different images [1], narrative art [27] or mammography training [41] and was found to be highly effective in each of the aforementioned examples. Nevertheless, little can be found about the application of *subtle gaze direction* in dynamic settings and conducted experiments, which include 360° videos [39] and a simple virtual reality scene [59]. *PlayGuide*, the visual gaze-based player guidance system developed for this master's thesis, sets out to apply the idea of *subtle gaze direction* to the complex and dynamic context of computer games, guiding a player's gaze within a three-dimensional game scene. Furthermore, two games have been developed to test the influence of *PlayGuide* on games according to efficiency and effectivity. With the help of the *Game Experience Questionnaire* developed by IJsselsteijn et al. [17], *PlayGuide* is also evaluated in terms of sensory and imaginative immersion, flow, tension and challenge, as well as negative affects and positive affects.

1.3 Structure

This master's thesis is comprised of nine chapters and structured as follows. The preceding sections in Chapter 1 gave a brief introduction to the gaze-based player guidance system. The second chapter will provide a comprehensive insight into the basic knowledge required and the current technologies that facilitated the development of the gaze-based player guidance system. A special focus will be given to the human visual system, an integral part of the body that enables us to see and forms our visual perception of the world. As the guidance system also makes use of eye-tracking technology, Chapter 2 will further include a section about the history and methodology of eye-tracking research as well as the operating principles of eye-tracking devices. Moreover, to create an understanding of what can be achieved with this technology, Chapter 2 lists different possible eye interactions for games. Chapter 3 then shares the knowledge and best practices of measuring gaze behaviour in games, including how to classify different eye movements, map the gaze of viewers to objects in a three-dimensional scene and structure the data that needs to be captured in order to properly analyze gaze data. Chapter 4 introduces the concept of visual attention and how it can be used to direct a viewer's gaze about an image. This method, also called gaze direction, represents the fundamental idea, on which this thesis and the guidance system are based. Also, the validity of the proposed system was tested through a series of user tests. Therefore, the guidance system and two different test games had to be implemented, the process of which is depicted in Chapter 5. The subsequent chapter covers the evaluation of the conducted experiments and the proposed thesis. This includes an explanation of the testing procedure, a summary of the study's participants, an insight on the setup of the study and most importantly the results and evaluation of *PlayGuide*, the gaze-based player guidance system. Thereupon, Chapter 7 explores potential applications of the system as well as problems and limitations that were encountered during the experiments. Finally, Chapter 8 gives a thorough conclusion of what has been learned.

Chapter 2

Theoretical Background

This chapter shares insights into the most important topics needed to convey a better understanding of the subject matter of the presented work. First, there is a brief overview of the human visual system, as the thesis and the therefore conducted experiments make use of the basic workings of the human eye by analyzing and evaluating the player's gaze. Subsequently, there will be an introduction to the mechanics of eye-tracking devices followed by a summary of eye interactions in computer games to exemplify how gaze can be used as an alternative or additional input method. Together the shared insights from this chapter form the basis to understand the modus operandi of *PlayGuide*, the gaze-based player guidance system.

2.1 The Human Visual System

The human visual system is a complex organism comprised of different interlinked parts that are required to observe the environment that is surrounding us. It plays an essential role in our lives, offers an invaluable opportunity for game developers to study the behaviour of people who play games and its characteristics are also made use of in *PlayGuide*. The two main parts are the receptive eyes and the processing brain. The latter acts as an interpreter of the information coming from the eyes, creating a comprehensible image [16].

2.1.1 The Eye

The eyes themselves are comprised of a multitude of different building blocks such as the sclera (recognizable by its distinct white appearance) which surrounds and protects the eye ball – apart from the cornea. The cornea itself is a see-through membrane that covers the colourful iris and the pupil, both of which sit in front of the eye ball. This can be seen in Figure 2.1, which illustrates a cross section of the inner workings of an eye [28, pp. 6f].

The iris continually controls the amount of light that permeates the eye by either relaxing or contracting its muscles to change the pupil's size. Behind the iris, the lens, "a biconvex multilayered structure" as Morimoto and Mimica [28, p. 7] call it, can adapt its shape to change the focus in the retina, which is located at the back of the eye ball

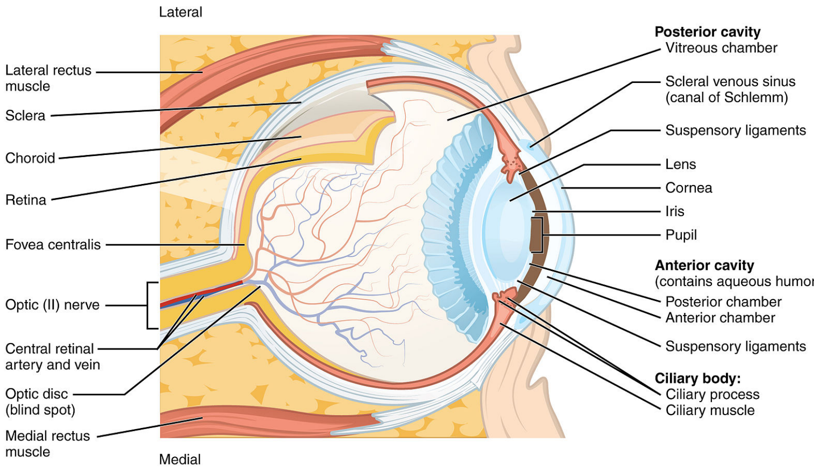


Figure 2.1: Illustration of the structure of the eye. Image source: [54].

and coated with photosensitive cells. Just like the lens of a photo camera the cornea together with the lens can bring individual objects into focus [28, pp. 6f].

At long last, the light waves that travelled through the eye and reach the retina are transformed into nerve signals and processed by the brain. Depending on the wavelength, there are two types of photo receptors located on the retina that are responsible of interpreting the different signals. Rods are in charge of handling reduced light, meaning that they are already activated by a few photons and responsible for vision in low light. However, they are not capable of sensing colors and only offer weak visual acuity. Blue, green and red cones, most of them situated on the fovea, require more light but are capable of differentiating between colors and conveying high visual acuity [60].

Although the central part of the retina, called fovea, is exclusively comprised of cones and thus responsible for sensing sharp image details, it only makes up roughly 2% of our vision, the size of your thumbnail held at arm's length [30]. The majority of the retina, called peripheral vision, is covered in rods and cones, whereas the number of the latter decreases with an increasing distance to the retina's center. The peripheral vision thus receives a blurred image and makes up about 98% of our field of view. Nevertheless, its worse visual acuity is compensated by a high sensitivity and faster response time towards detecting stimuli [7, 44].

2.1.2 Eye Movements

When our gaze is fixated on particular features of an object it seems as though the eyes are completely still. However, the human visual system is not capable of holding the eyes in one place for a long period of time and different movement patterns occur. These movement patterns are enabled by six muscles that are attached to the sclera [40]:

- **superior and inferior oblique muscles** capable of torsional eye movements,
- **superior and inferior rectus muscles** capable of vertical eye movements and
- **lateral and medial rectus muscles** capable of horizontal eye movements.

As soon as the eyes are supposed to move into a certain direction, the above mentioned pairs of muscles have to coordinate the intensity of their contractions and relaxations. While one muscle contracts the other one relaxes, both with an equal level of intensity [40]. In total, this coordinated interplay of muscles yields six different types of eye movements:

- **Saccades** are fast, voluntary and reflexive eye movements during which no visual information can be processed [7, pp. 42–45]. They are used to relocate the fovea, which is responsible for sensing image details, yet only makes up a relatively small area of the human visual field [7, p. 15].
- **Fixations** are eye movements that stabilize the retina when the gaze remains on an object or a feature of an object for a longer period of time (150–600 milliseconds). This can be controlled voluntarily and allows people to properly examine objects they are interested in. However, fixations can also occur involuntarily, whenever an object holds striking features that attract the attention of the viewer's gaze. 90% of the time the eye remains in a state of fixation [7, pp. 46f].
- **Smooth persuits** occur when people want to fixate on an object that is in motion. In order to fix their gaze on approximately the same spot, the eyes have to carry out the same motion as the moving object, which results in a smooth movement of the eyes [12].
- **Vergence** movements appear when different objects are located at varying distances, forcing the eyes to move in opposite directions. This means that whenever the gaze is shifted from an object that is further away to one that is closer, the eyes have to converge, whereas in the opposite case the eyes will diverge [14].
- **Compensatory eye movements**, also called vestibular movements or vestibular-ocular reflexes, are eye movements that mirror and therefore counteract the motion of the head. With the same velocity the eyes will rotate in the opposite direction of the head's movement [4].
- **Optokinetic nystagmus** are eye movements enabled by an intricate interplay between smooth persuits and saccades in order to be able to fixate on moving objects. For example, optokinetic nystagmus occurs when looking through a train window trying to fixate objects outside [7, p. 47].

Even though humans exhibit all of the six mentioned eye movements, only three of them (saccades, fixations and smooth persuits) usually play an important role in the analysis of gaze behaviour.

2.2 Eye-Tracking

For *PlayGuide* to apply its subtle image space modulations, an eye-tracking device is required to track and evaluate the eye movements of the player. Already in 1905 Charles H. Judd was one of the first people to design and use a kinetoscopic eye-tracking device that was capable of recording eye movements in two axes. While this was eye-tracking at its infant stage, he would later be known as a pioneer who used eye-tracking devices (also called eye trackers for short) as a scientific method to understand human eye movements during the process of reading [25, pp. 71–86].

Since then, many different means of tracking eye movements, such as electrodes placed on the skin around the eyes or wearing contact lenses (the latter of which still poses one of the most precise methods), have occurred and were applied to advance research. Nevertheless, it is worth noting that nowadays the most commonly used solutions for tracking eye movements are video-based, due to their ease of use and the low price of the hardware [7, p. 51]. The following subsections will provide an insight into the current state of eye-tracking systems.

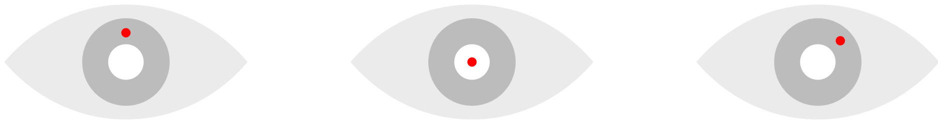
2.2.1 Eye-Tracking Methodology

Eye-tracking technology enables us to record and observe eye movements. Amongst other applications, this information can then be used to analyze what someone is interested in, how a viewed scene was perceived or what caught someone's attention [7, p. 3]. Eye-tracking devices work by measuring the rotation of the eyes to calculate gaze points in the form of two-dimensional vector objects that include horizontal and vertical coordinates [43, p. 557]. For the digital realm the gaze itself can be defined as the vector from the eye to the gaze point on the screen [15, p. 87].

For an eye tracker to be of use in the field of gaming or research it needs to continuously deliver accurate information about the observer's gaze and be able to efficiently gather data at high speeds, as different eye movements occur in quick succession. The *Tobii EyeX*, for example, is a commercially used eye tracker that has a frequency of 70 Hertz, gathering seventy samples per second [58]. Gaze samples are usually comprised of x- and y-coordinates, the distance of the observer to the screen and a timestamp [58]. Even though commercially used eye trackers deliver very precise results, inaccuracies within the samples require processing and filtering of the raw gaze data before it can be used. After the removal of noise and other disruptive factors, the processed data can give researchers a deep insight into the observer's behaviour. Besides an analytic use, eye-tracking technology can also be used as an input device for digital devices to replace the traditional mouse and keyboard setup. This is especially helpful for disabled people that might be in desperate need of a different form of interaction. Additionally, eye-tracking technology has not only shown to be able to replace but also to coexist with traditional input devices as a means to enhance the user experience [33]. This thesis exclusively focuses on the use of eye-tracking systems within a digital context, where there are possible applications for many different scientific fields including computer graphics, virtual reality and games. Nevertheless, gaze behaviour outside the digital realm has been studied extensively as well, including ordinary tasks like hand washing and sandwich-making [43].

2.2.2 Eye-Tracking Systems

As mentioned in the beginning of this chapter, the most commonly used eye-tracking systems are video-based due to their ease of use and low hardware prices. Nevertheless, there is a multitude of different eye-tracking systems that are capable of measuring eye movements and providing gaze data. Duchowski categorizes these systems as follows [7, pp. 51–59]:



(a) (b) (c)

Figure 2.2: This figure shows an illustration of an eye and a red dot representing the corneal reflection caused by an infrared light. Depending on the point of regard the position of the corneal reflection changes. Illustration (a) shows the corneal reflection when the head is positioned below the camera, (b) when the head is directed at the camera and (c) when the head is below and to the right of the camera. Image based on [7].

Figure 2.3: An example of a head-mounted eye-tracking device. Image source: [51].

Electro-Oculography

Established nearly 40 years ago, electro-oculography is still used today, albeit being a vanishing technology. In its early days, however, it was the “most widely applied eye movement recording method” [7, p. 52] and delivered results by measuring the difference of the electric potential of the skin, created by electrodes that were placed around the eye [50]. Even though it poses a viable option of measuring eye movements, this method is not applicable for gathering gaze points on a screen, as the head’s movement is not considered throughout the measuring process and would necessitate a separate tracking of the head [7, p. 52].

Figure 2.4: An example of a built-in eye-tracking device. Image source: [52].

Scleral Contact Lens

Eye-tracking systems that use scleral contact lenses pose one of the most precise methods to measure eye movements. Amongst many different lens measuring methods, the most common one is the "search coil technique", in which a wire coil is attached to a lens that moves through a electromagnetic field and thereby determines the position of the gaze [50]. The downsides of these systems are that a large contact lens must be worn by the user, which might be uncomfortable for some individuals and that, as in electro-oculography, the measurement method is decoupled from the head's movement [7, pp. 52f].

Photo-/Video-Oculography

Photo-/video-oculography represents a wide variety of systems that are capable of measuring eye movements. Usually they work by capturing distinct characeteristics of the eye and automatically (or manually) calculating the difference in rotation and translation thereof. Most devices that use this method require the head to be stationary [50] and just as the last two preceding methods, photo-/video-oculography measurements usually do not provide information about the head's movement [7, pp. 53f].

Video-Based Combined Pupil/Corneal Reflection

The most used eye-tracking devices utilize reflections in the cornea caused by an infrared light. They consist of cheap hardware, namely infrared cameras and infrared light emitting diodes. To compute the gaze coordinates, the camera not only records the position of the eyes but also considers the head's movements by calculating the difference between the pupil's center and reflections in the cornea (also called *Purkinje reflections*) caused by the infrared light emitted by the diodes. The method of defining the position of the head in world-space is named point of regard and allows a slight movement of the user [7, pp. 54–58]. An illustration of this positioning process can be seen in Figure 2.2. To ensure a proper functionality and accurate results these eye-tracking devices have to be calibrated after each new use [9]. However, they are usually very versatile and can be used in two different setups: either as a head-mounted system that can be worn by the user (as seen in Figure 2.3) or as a desk-mounted system, being directly built into the screen (as seen in Figure 2.4) or placed on the desk in front of the user.

Such a video-based eye-tracking system was also used for the successful use of *PlayGuide*, to evaluate where a player is currently looking and what eye movement he or she is currently exhibiting. For example, fixations, which indicate an area of importance to the viewer, or saccades, which are rapid eye movements to reposition the fovea, the only part of the eye that can sense image details. Thus, all of the components described in this chapter play an important role for the effective functioning of *PlayGuide*. The following chapter will describe how this collected information can be used to measure and evaluate the gaze behaviour of players in computer games.

Chapter 3

Measuring Gaze Behavior in Games

To gain a deeper insight into the gaze behaviour of the player, it is necessary to design and structure games in a specific way, already in the early stages of the game development process [43, pp. 543–583]. This chapter shares insights on how to prepare eye-tracking data to measure gaze behaviour in games.

3.1 Classifying Eye Movements

In the previous chapters a comprehensive overview of the eye-tracking technology and its use cases was given. Eye trackers deliver raw gaze data, which (as mentioned in Section 2.2.1) usually consists of x- and y-coordinates, the distance of the observer to the screen and a timestamp. This information can then be used to gather information about the saliency (interesting areas) of an image or to categorize the raw gaze data into the different eye movement patterns that occur during game play. On the one hand, a fixation on certain parts of an image reveals what an observer was paying attention to. Saccades, on the other hand, unveil the behaviour pattern of the eye, i.e. how it moves about in a scene [43, pp. 554–559].

Although eye-tracking devices deliver data about a viewer's gaze, the information obtained from the devices only represent the coordinates on the screen. In order to properly evaluate the gaze data there are a wide variety of approaches to derive the different types of eye movements from the gathered screen coordinates. Common metrics to analyze include gaze duration, saccadic velocities or saccadic amplitudes [37]. Not only are these metrics important for research but also for games that might need a real-time evaluation of the current eye movements exhibited by the player. In games, such an evaluation can then be used as an input mechanism (e.g., use fixations for object selection), as a tool to trigger in-game events (e.g., gaze direction) or to analyze player behaviour (e.g., measuring the immersion of the player) [46].

The two most important and thus useful eye movements are saccades and fixations, a pattern of which can be seen in Figure 3.1. Fixations occur when the eyes are dwelling on a certain object or feature of an object to gather image details and hence can be interpreted as the human visual system paying attention to and expressing interest in what is currently examined. Saccades are used to reposition the eye and are identified as rapid and sudden eye movements, during which no image information can be processed,

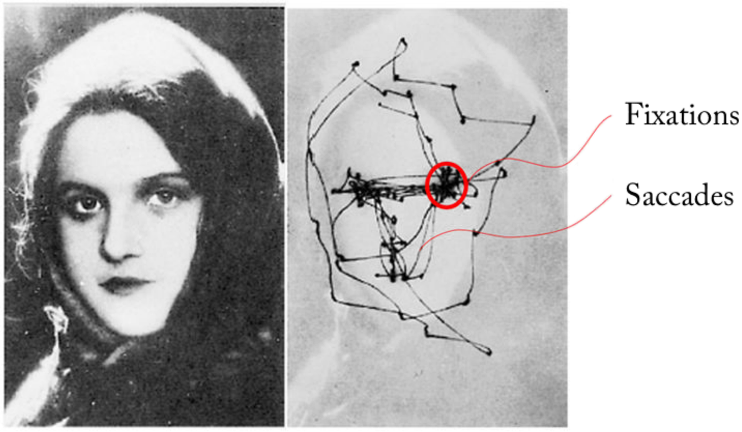


Figure 3.1: To visualize the most important eye movements, the above figure shows saccadic movements (spread apart) and fixations (clustered closely together). Image source (modified): [55].

leaving the viewers oblivious to the events happening in their surroundings. Studies, however, suggest that 90% of the time is spent in a fixation state. It also needs to be considered, that due to camera movements and dynamic objects in computer games, the eyes can also demonstrate a movement pattern called smooth persuits. Smooth persuits compensate the movement of an object to keep the eyes on the same point of interest during a fixation [7, pp. 45f].

Classifying eye movements usually means to identify fixations, for the simple reason that saccades merely represent the movements that happen in between fixations. It also serves as a practical approach to reduce the great amount of data collected by eye trackers while keeping the most essential information and is commonly referred to as *fixation identification* or simply *identification*. During the identification procedure the raw gaze coordinates that are marked as fixations are usually grouped together and represented as a single weighted point (called *tuple*). This has two reasons: Firstly, as means to represent a multitude of individual gaze points caused by drifts, flicks and tremors that naturally occur during any eye movements as a single entity. Secondly, fixations are usually the only relevant data needed for research because during saccadic eye movements the human visual system cannot process any perceived information. The process of *fixation identification* is needed in almost any situation where an eye-tracking device is involved and constitutes the basis to any analysis that evaluates eye movement patterns. Yet, only few details on the different evaluation methods and algorithms can be found. A fact that exacerbates the search for a reliable *identification* algorithm is that it is still subject of discussion, when exactly a fixation begins and when it ends. Thus, the different ways to classify fixations is still a subjective endeavor. Especially poorly designed algorithms can yield wildly different results and drastically affect the findings of research. Well-grounded identification algorithms, however, produce viable results that allow for an unobjectionable use in most research cases. An often used solution to prove the validity of the outcomes of fixation identification algorithms has been to compare the computed fixation patterns with the, admittedly subjective, statements of the test subjects through interviews. In their work, Salvucci et al. present a detailed definition as well as classification of common *fixation identification* algorithms to offer a comprehensive overview, which allows researchers to make an informed decision about a suitable algorithm to choose for their application. To ensure a comparability throughout the different approaches they also introduce a new taxonomy, which makes it easier to compare algorithms by distinct criteria, namely spatial and temporal features. Spatial features can be categorized into three different types: Velocity-based algorithms make use of the fact that during fixations eye movements have a low velocity compared to the rapid movement of saccades. Dispersion-based algorithms consider the distance between the sampled gaze points, whereas area-of-interest information-based algorithms also determine and consider the saliency of image areas. Temporal features can be categorized into the two following types: duration-sensitive algorithms take into account that the eyes' usually exhibit a dwell time of 200–400 milliseconds during a fixation and locally adaptive algorithms consider the types of temporally vicinal gaze points [37].

Unlike Salvucci et al., whose work dealt with the processing gaze data after it has been recorded, Kumar et al. proposed an approach to better detect saccades in real-time, thus making it an ideal candidate for games [22]. As a means to classify the gaze information, each of the gaze points coming from the eye tracker is labelled as either a saccade or fixation. To achieve this classification they determine whether a given gaze point is the beginning of a saccade, the continuation of the current fixation or just a corrupted sample, that poses an outlier compared to the temporally adjacent gaze samples. As a basis they use the dispersion-based algorithm proposed by Salvucci et al. [37], which calculates the distance between two subsequent gaze points and compares them to a predefined threshold. If the measured distance is bigger than the threshold, the gaze point will be categorized as saccade. However, in order to deal with outliers Kumar et al. introduce two modifications to this algorithm. Instead of calculating the distance of the current gaze point to the preceding one, they calculate the distance to their estimation of where the current gaze point should be. Furthermore, they automatically rule out a saccadic classification of the current gaze point, if the following gaze point is a part of the current fixation. These methods successfully prevented outliers but it must be noted that the latter modification also introduces latency to the classification process [22]. For the sake of completeness, it should be mentioned that eye movements can also be classified as *smooth persuits*, as proposed in [21]. However, the categorization of smooth persuits was not necessary for the successful implementation of the gaze-based player guidance system and thus is not explained in greater detail.

3.2 Gaze-to-Object Mapping

Even though a pixel-based frame-wise analysis of eye movements can be used in static scenes, the same frame-by-frame analysis would hardly make any sense within a dynamic context. In computer games, where the viewport is controlled by the player and the camera exhibits constant and mostly rapid movements, each image to be analyzed within a sequence of images drastically differs from the one before. Additionally, considering the sample rate of modern eye trackers, 70 Hertz and above [58], the amount of data collected requires a lot of processing power and tedious work to analyze the data. A more practical approach for games is proposed by Sundstedt et al., which shows that

(a)

(b)

Figure 3.2: These images visualize the process of creating an item buffer. Each object in the game scene is assigned an individual color and then rendered to a separate image. The original image (a) is the final rendered scene, whereas the image (b) shows the computed item buffer. Source of the 3D models for the renderings: [53].

an analysis based on mapping fixations to objects yields accurate results and serves as a great tool to analyze fixation behaviour in games [42]. This allows for data to be collected on an object basis, disregarding unnecessary information on objects that are not relevant for the analysis.

The process of assigning the viewer's fixation to an object in the scene is called *gaze-to-object mapping* (GTOM). GTOM algorithms evaluate the player's gaze and derive from additionally collected image information, what game object the player is actually looking at. This mapping process is vital for a proper evaluation of the gaze behaviour, since it identifies objects in a scene the viewer has attributed his or her attention to – the most valuable resource that reveals profound insights into the viewer's gaze behaviour.

However, it is not always correct to just directly map the processed gaze data to an object in the scene, which is especially important when complex scene structures come into play. In games where the mapping process plays a particularly important role, common complex scene structures include a high density of smaller objects, objects that move at high speeds or objects that become occluded by other moving parts in the scene. In such cases a 1:1 mapping of fixations to objects can lead to a false interpretation of image material and bias the results of the viewer's gaze behaviour [3].

Sundstedt et al. [43] first devised a novel methodology for GTOM algorithms. To map a viewer's gaze to objects in a three-dimensional scene, they make use of an *item buffer* (sometimes also called *id buffer*), which represents each individual object in the game scene as a single color. This color is unique to each object and thus each pixel in the image serves as a fully distinguishing feature for any object in the game scene. For computer games this item buffer can be easily computed on the graphics processing unit, where the viewport image then has to be rendered twice. First, the scene is rendered normally, as it should be presented to the player. Then the same scene can be rendered with unlit materials and the unique color identifiers applied to each game object. A comparison of the two rendering processes can be seen in Figure 3.2. Furthermore, the item buffer allows for the grouping of objects in a game according different criteria. The object space, for example, represents the game objects as they are named in the game (e.g., “entity_1”), the property space comprises the name as well as the different sates and behaviours of the game objects (e.g., name: “entity_1”, status: “alive”, behaviour: “waiting”) and the semantic space, which allows for a high-level grouping of objects (e.g., “background”, “foreground” or “danger”). Such *item buffers* are necessitated in different effective GTOM methods, explained in the next paragraphs.

Point-based methods represent the easiest GTOM approach and make use of the gaze samples, by casting a ray through the center of a fixation into the game scene and detecting which object was intersected by the ray or simply by sampling the affected pixel from the *item buffer*. Although the mapping of a single fixation point to an object might work in simple scenes with a few big spatially separated objects, it does not always pose the most accurate solution. Therefore, area-based methods consider the region that is encompassed by the foveal vision and the sampled gaze points through which the fixation point is obtained. Foveal sensor density models further take into account that the perceived image details in foveal vision decrease with an increasing distance to the center of the fovea. Additionally, these methods consider the possibility that more than one object can be fixated at once. Gaze-point distribution models consider that gaze data coming from an eye-tracking device is usually tainted by a certain inaccuracy. Although Sundstedt et al. note that all these methods pose a viable option for the GTOM process, they would fail at mapping the viewer’s gaze to an object that is moving at high speeds or whose silhouette is being thoroughly examined [43, p. 566]. Bernhard et al. [3] try to address this problem and propose an algorithm that improves the accuracy of the GTOM process during smooth persuits.

Although GTOM plays an important part in the analysis of the viewer’s gaze behaviour (especially in three-dimensional and dynamic game settings), up until now it has received little attention in research and existing methods are still in great need of an evaluation in terms of their accuracy [3]. As Schön mentions in [38], the leading eye-tracking company *Tobii Technology*[1] recommends a technique called *Ray Casting Shotgun* to be used in games, due to the technique’s simple implementation process and low performance costs while maintaining relatively exact results. To map an object to a gaze, a number of rays (usually 3–6) are cast into the game scene within a certain radius. The game object that is intersected by the most rays is then considered to be the fixated object.

Besides the simple purpose of mapping a player’s gaze to game objects, GTOM has manifold areas of application. To name a very unconventional example, in their work, Vidal et al. [47], present an application called *Pursuits*, which provides a way to utilize eye-tracking devices, without the user having to go through the hassle of a dedicated calibration phase. This liberation from the time-consuming and complex calibration phase is enabled by the idea that the eyes perform the same path as the object they are fixating on during a smooth pursuit. Hence, Vidal et al. use the trajectory of the

[1] https://www.tobii.com/

eyes instead of individual gaze points and dwell times to map the user's gaze to moving objects on the screen.

As already mentioned in the beginning of this chapter, it is necessary to design and structure games in a specific way, already in the early stages of the game development process. The aforementioned methods to categorize eye movements into saccades and fixations as well as the process of mapping gaze coordinates to game objects in a scene are made use of in any game that implements the gaze of the player. The same holds true for *PlayGuide*, which is dependent on the player's eye movements. The GTOM methods, on the other hand, are needed to evaluate the performance of *PlayGuide* and measure how well it performs in subtly guiding the player's gaze towards game objects. The methods and details of guiding a player's gaze are explained in the following chapter.

Chapter 4

Guiding Attention

The gaze guidance system proposed in this thesis, *PlayGuide*, makes use of two tightly correlated subjects: visual attention and *gaze direction*. The classification of eye movements and the understanding of what a player is actually looking at, as described in the last chapter, can be used to understand what the mind is currently focused on and how to guide the attention, thus, gaze towards any desired location on the screen. However, in order to understand why visual attention is so effective in guiding a viewer's gaze, first there needs to be an understanding of how visual attention works.

4.1 Visual Attention

Naturally, the processing power of our brain is limited. The sensory impressions around us, however, are seemingly infinite. Attention defines the process of how our mind decides what is worth our attention and what is not. This means that while the brain is confronted with a multitude of sensory stimuli like smell, sound, touch, vision or taste it needs to focus on a particular noteworthy sensation, partially disregarding other impressions. Especially noteworthy is that the eyes closely follow this principle too, due to the nature of the limited range of the foveal vision. As mentioned in Section 2.1.2, the fovea is responsible for sensing image details but only makes up roughly 2% of our vision, "the size of your thumbnail held at arm's length" [31]. The majority of the retina, called the peripheral vision, receives a blurred image and makes up most of our field of view. To get a clear vision of whatever caught our attention, it is essential for the human visual system to continuously relocate the fovea towards the things we deem as interesting. Instead of perceiving a scenery as a whole, an image is formed by capturing a large number of small pieces of visual information, depending on what attracts our visual attention [7].

For our vision to properly process information it is composed of two different but integral parts: bottom-up and top-down processes, also known as perception and cognition. Bottom-up processes, also called perception, are preattentive mechanisms that subconsciously shift the attention "rapidly and involuntarily" [5] and react to the saliency of an image – low-level features such as color, movement, form and contrast [5, 45]. Examples that illustrate bottom-up process well are the contrast of a black dot on an otherwise white surface or a fast approaching object that is about to hit someone. Top-down

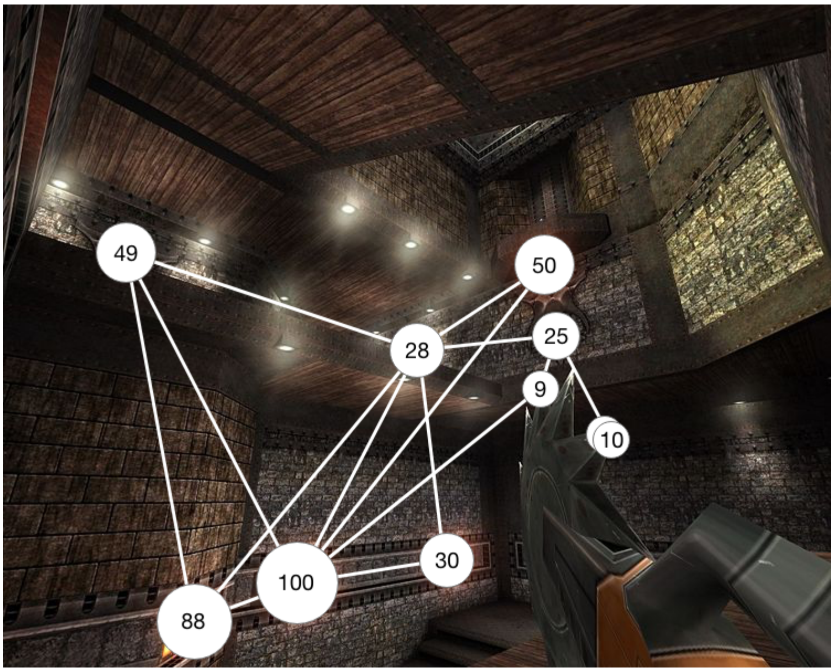
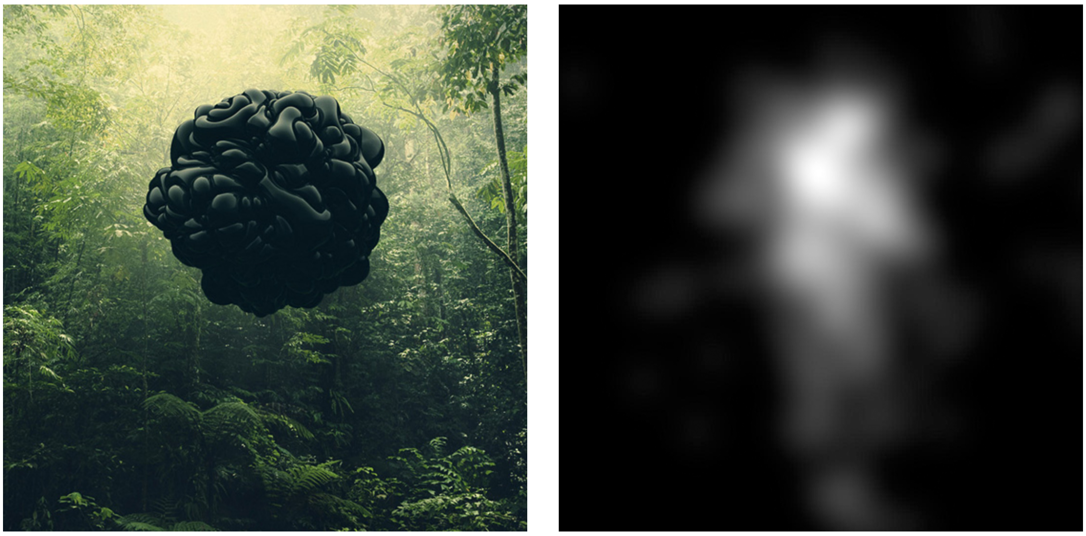


Figure 4.1: This figure shows an example of a *scan path* for a computer game inspired by [43]. The lines represent saccades, the circles depict fixations with a reference to how often a player fixated on a particular feature of the image. Image source (modified): [56].

processes are cognitive mechanisms (thus, also called cognition) that shift the attention "when we need to look for something specific" [5], e.g., a distinct visual feature of an object that represents pain to us, such as a turned on stove top glowing red [5]. Cognition evaluates image features through thought and experience [31]. *Gaze direction*, however, is highly affected by the bottom-up processes; a jarring color, a strongly different shape or a moving object amid static ones can draw an observer's attention and thus his or her gaze [49].

This means that our gaze is always directly fixated on objects we pay attention to. Thus, in seeking to establish what a person was or is interested in, we can record their eye movements and derive a sequence of fixations, which gives insight on what caught someone's attention [7]. Duchowski calls this sequence *path of attention* [7], Jacob and Karn *scan path* [18]. According to Sundstedt et al. [43] *scan paths* are commonly visualized by using a snapshot with an overlay that shows the fixations as circles, which are connected through lines that show the saccadic movements of the eye. An example of such an overlay in a game scenario can be seen in Figure 4.1.

While visual attention can be measured with an eye-tracker, different efforts have been made to compute and predict visual attention. Early models tried to predict gaze behaviour based on the saliency of objects, which is described by Land et al. as the "intrinsic ability to stimulate early parts of the visual pathway" [23]. The resulting model, based on the measurement of saliency per pixel within an image is called a saliency map (see Figure 4.2) [43]. However, Land et al. [23] showed that people who were given a task did not abide to the rules of saliency. As soon as participants were



(a) (b)

Figure 4.2: Figure (a) shows the original image, while figure (b) shows the calculated saliency map, which indicates potential areas of interest to the viewer. Image source: [61].

given tasks, "objects were fixated because of their relevance, not because they were big, bright, or visually exciting in other ways." [23].

4.2 Gaze Direction

As explained in the previous section, visual attention is guided by a number of different features. These natural responses to visual stimuli can be exploited to trigger bottom-up processes that guide a viewer's visual attention (and therefore gaze) within an image. In research, this method is often called *visual guidance*, *gaze manipulation*, *gaze guidance* or *gaze direction* and has been applied to a wide variety of different exciting contexts, such as art [27], medicine [41] and virtual reality [34].

Gaze direction relies on visual cues that involuntarily pull the eyes towards a specific region of an image, hence they are also called *pull cues* [19]. Attempts to direct the viewer's gaze can be divided into two categories, namely *subtle* and *overt gaze direction* techniques [11]. As the name suggests, *overt gaze direction* is clearly visible to the viewer's eyes, yet highly effective in guiding the gaze. For example, an overt visual cue can be the use of depth-of-field in an image, if applied strongly. In their work Ennset al. [8] mention that the viewer's eyes were predominantly drawn to regions of an image containing sharp details rather than to those regions that were blurred. However, attention can also be guided with the viewer barely or not noticing it at all. *Subtle gaze direction* techniques rely on visual cues that are only subconsciously perceived by the viewer. For example, McNamara et al. [27] propose a solution for guiding the gaze in narrative art, based on slight changes in the luminance channel of an image. This way the participants of the study did not realize their gaze was guided and their viewing experience was not disrupted during the contemplation of the art.

In general, *gaze direction* has been proven to foster the ability to solve problems [10] and improve the performance in search tasks [26]. Additionally, *gaze direction* was successfully applied as an instructional tool. Thereby the fixation behaviour of experts is used to provide support and guide the gaze of inexperienced people confronted with new tasks. Sadasivan et al. [36] deployed this strategy for aircraft inspections. With the use of an eye-tracking device they recorded eye movements and developed scanpaths to demonstrate the flight inspectors' search strategy to novices in training. Sridharan et al. [41] used the same principle to guide novices through the process of finding abnormalities in mammograms.

4.2.1 Overt Gaze Direction

Overt gaze direction techniques rely on obvious pull cues that try to attract the attention of the viewer. As soon as the viewer's attention has been caught, the human visual system will try to reposition the fovea (the central part of the eye where image details can be sensed) and perform a saccadic eye movement towards the perceived position of the *pull cue*, which enables the viewer to examine the region of interest with greater detail. These cues are global image transformations, which can be applied throughout the image [11, p. 4:2]. The following paragraphs highlight four different examined methods to guide a viewer's gaze with the help of *overt gaze direction* techniques.

In their work Hata et al. [13] propose an overt method based on blur. To guide the gaze of the observer they introduced a system that gradually blurs an image but skips the parts that the observer's gaze should be directed to. As the human visual system is meant to process information regarding our surroundings, it attributes less importance to regions of an image containing fewer information. The greater presence of detail – and hence information – naturally lead the viewer's eyes towards unblurred areas of an image. Although the system proved to be successful, Hata et al. mentioned that this technique did not work properly as soon as high saliency objects, such as faces, were introduced to the image.

In 2005 Khan et al. [20] proposed an *overt gaze direction* technique called *Spotlight*. To rapidly attract the attention of an audience during a presentation on a wall-sized display (or video projector), they simply darken the whole image except for a region that is displayed normally. This creates the look of a spotlight shedding light on a specific part of the image where the viewers are supposed to look at. During their experiments Khan et al. showed that their idea was able to increase the performance of target acquisition significantly.

Another method to overtly guide a viewer's gaze is to adjust the saliency of an image, which specifies how probable it is that a viewer will fixate on a certain part of an image due to its low-level features such as color or contrast [23]. Vig et al. [48] showed that they could adjust the saliency of video regions to attract the viewer's gaze. With the help of a comprehensive sample of eye movements and a machine learning algorithm they experimented with local saliency adjustments in videos to direct the viewer's gaze.

A far simpler *overt gaze direction* method was proposed by Lin et al. [24] in 2017. As a means to assist the viewer of 360° video material in focusing and refocusing targets they introduced a simple overlay that indicated the direction of movement of a target. This overlay, visualized as a green arrow on top of the 360° video footage, appeared whenever the viewer lost track of the target and disappeared as soon as the viewer focused on target again. Lin et al. then concluded that "participants felt more engaged, enjoying, and receptive to video content" [24] when they were visually aided by the green arrows. This and all of the aforementioned *overt gaze direction* techniques have proven to be successful in guiding the viewer's gaze. Nevertheless, such apparent methods may disrupt the viewing experience, which might not be acceptable for some applications. A solution to this problem poses *subtle gaze direction*, discussed in the subsequent section.

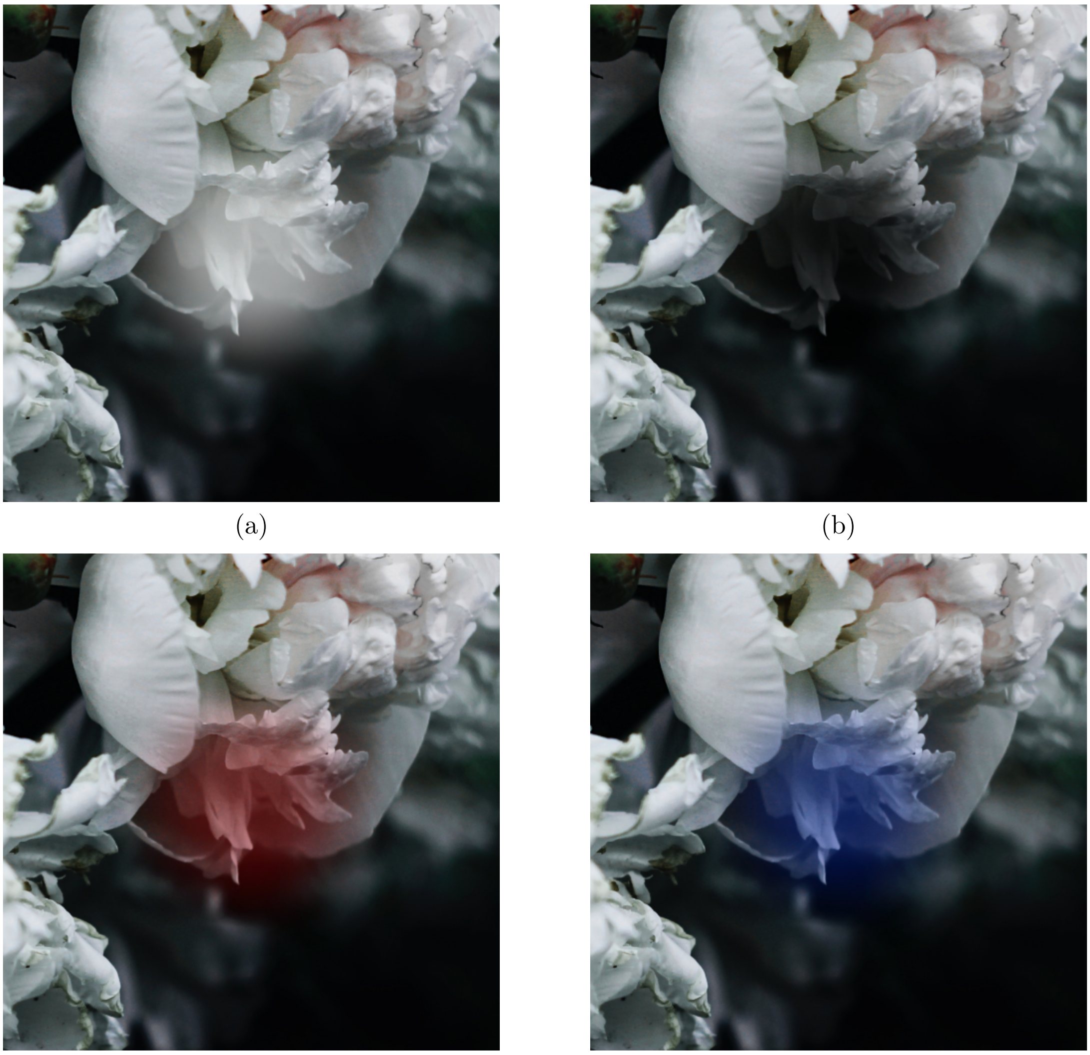
4.2.2 Subtle Gaze Direction

Subtle gaze direction defines the process of guiding the attention – and therefore gaze – throughout an image without the viewer noticing. This process had been subject of many papers, whereas the proposed methods usually related to the fact that the human foveal vision is more likely to be drawn to image areas containing high detail. Bailey et al. [1], however, first coined the term *subtle gaze direction* and introduced a novel *gaze direction* technique based on an eye-tracking device and the use of subtle image-space transformations.

To direct the gaze, Bailey et al. presented short visual stimuli in the peripheral vision of the viewer, that draw the gaze towards the modulated region of the image. However, in contrast to *overt gaze direction* techniques Bailey et al. used image modulations that were indiscernible to the viewer's foveal vision and terminated the modulation as soon as the foveal vision came close to the modulated area, thus leaving the viewer unsure of what attracted his or her attention in the first place [1].

To achieve a *subtle gaze direction*, Bailey et al. successfully exploit two distinct characteristics of the human visual system. First, they briefly modulate a specific point of interest in an image, which is in the peripheral vision of observer's field of view. As a result of the high sensitivity towards motion and the blurry image received by the peripheral vision, the viewer notices the image modulations but cannot determine the exact changes to the image. Secondly, to further investigate the motion caused by the image-space modulations, the eyes will perform a saccade to reposition the fovea towards the point of interest. A phenomenon called saccadic masking, however, prevents the human visual system from processing perceived impressions during saccadic movements of the eye. Bailey et al. use this to their advantage and automatically terminate the image-space modulations as soon as the eye tries to reposition the fovea to the modulated region, hidden by saccadic masking [1].

In the experiments conducted by Bailey et al. ten participants were separated into a control group with no modulations as well as a *subtle gaze direction* group that was presented with the image-space modulations at various pre-selected points of interest. These modulations altered the image regions containing the points of interest in either the luminance or the color channel (as depicted in Figure 4.3). Ultimately, the experiments led to the conclusion that in static images *subtle gaze direction* is highly effective in guiding the viewer's gaze, although the test subjects reported that the modulated images subjectively appeared to be of less quality than the images that were not modulated. Furthermore, Bailey et al. saw a potential application of their subtle gaze guidance technique in manifold areas such as perceptually adaptive rendering, flight or driving simulations, online training as well as pervasive advertising [1]. Besides Bailey's



(a) (b) (c) (d)

Figure 4.3: Bailey et al. use subtle image-space modulations in the peripheral vision to direct the viewer's gaze about an image. Figure (a) and (b) show the illumination modulations, while (c) and (d) show the warm and cold color modulations used in [1]. To be clearly visible, the modulations in these example images have been exaggerated. Images based on [1].

advances in this domain of gaze guidance there are other works that can be categorized as *subtle gaze direction*. For example, Barth et al. [2] attracted the attention of viewers by briefly showing red dots in the periphery of their vision, increasing the saliency of certain areas in video footage. During their experiments Barth et al. noticed that they were able to guide their participants' gaze but even a simple solution such as the red dot used in their study needs a comprehensive investigation to perform efficiently [2]. However, Bailey's work poses one of the most interesting contributions in this field.

4.2.3 Gaze Direction in Games

Although video gaming has become hugely popular over the last few decades, with an estimated global value of the video games market of almost 80 billion dollars in 2017 [70], the market for eye tracking devices has only recently gained traction due to the fact that cheaper sensors and smaller electronics made eye trackers more accessible to the end consumer [46]. This increasing prevalence of the eye-tracking technology introduces new and exciting possibilities for *gaze direction* as a tool for developers or to aid players throughout their gaming experience. As a potential application Bailey et al. [1] mention a process to progressively render scenes in games. By analyzing the players' behaviour and actively guiding their gaze away from complex geometry, the game could take more time during a time-consuming rendering process while the player is distracted looking at parts of the image that take less time to render. Another practical use of *gaze direction* in games is mentioned in a paper by Ben-Joseph et al. [59]. In a virtual reality setting, where the feeling of immersion is essential, they tried to guide the player's gaze with the help of image-space modulations. By introducing subtle flicker effects to the peripheral vision of the viewer, they observed that the test subjects would turn their head towards the flicker effect. With the help of this technique participants of the study were able to find an object, placed directly behind them, much faster. Although the previous two examples only represent methods based on eye-tracking techniques, *gaze direction* has always played an essential role in level design. Throughout a level, there needs to be a clear visual communication on how a path is structured in order for the player to successfully navigate through the environment. Amongst other things, this can be achieved through a coordinated placement of clues, the use of coherent color palettes and an understandable iconography so that the player's gaze is automatically guided towards visual cues placed by the level designers [66].

Nevertheless, with the help of eye-tracking devices, *subtle* and *overt gaze direction* techniques constitute new and promising solutions to guide the player's gaze, although each of the two techniques has its own pitfalls. On the one hand, *subtle gaze direction*, which deploys unobtrusive visual cues, might leave players unsure of what to do next when it is not capable of guiding their gaze into the right direction. On the other hand, *overt gaze direction*, which makes use of highly visible visual cues, is more likely to succeed in guiding the player's gaze but the apparent alteration of the image material might break the feeling of immersion for some players or spoil the sense of accomplishing demanding tasks on their own. As a relatively young research field there are still a lot of unanswered questions and countless possibilities to further incorporate and make use of *gaze direction* based on eye-tracking in games. *PlayGuide* tries to answer some of these questions and uses *subtle gaze direction* to guide the player's gaze. More on its implementation and the development of two test games to evaluate *PlayGuide* can be read in the following chapter.

Chapter 5

Implementation

The following sections will provide insight into the different steps necessary to implement and test the visual gaze-based player guidance system for three-dimensional computer games, short *PlayGuide*. As the system makes use of eye-tracking technology, the first subsequent section describes the process of how the gaze data is filtered and analyzed to ensure that the gaze samples coming from the eye tracker are exact enough to be used for any other processes that rely on the gaze samples. Then, details on how *PlayGuide* works will be shared. Furthermore, to compare the results of *PlayGuide* gathered in the conducted experiments, two additional *gaze direction* mechanisms were implemented: An *overt gaze direction* (OGD) technique that, like *PlayGuide*, makes use of the player's gaze as well as a traditional way of guiding the player's gaze that resorts to common visual cues used in game development and is independent of the player's gaze. Finally, in order to test *PlayGuide* within different game genres, two test games were developed. Therefore, the last two sections will describe the implementation of the test games.

As a basis for the implementation *Unity*[1] was chosen as a game development platform. This choice was based on the fact that the version of the *Unity Editor* used for developing the system (*Unity* 2017.4.2)[2] can be run on almost any hardware [71] and that *Unity* provides comprehensive tools to develop three-dimensional games that can be deployed on various different operating systems, such as Windows, MacOS, Android and iOS. Moreover, application programming interfaces (APIs) were readily provided by the manufacturer of the eye-tracking hardware, *Tobii Technology*[3], that was used to analyze the players' gaze behaviour. Since *Unity* makes use of C# as a programming language, the implementation was entirely written in C# as well.

5.1 Providing Gaze Input

The most important task in order to realize *PlayGuide* was to properly filter and classify the gaze data coming from the eye-tracking device as the guidance system primarily depends on the player's gaze. Even though the eye tracking device used, the *Tobii EyeX*[4],

[1] https://unity3d.com
[2] https://unity3d.com/de/get-unity/download/archive
[3] https://tobii.com
[4] https://help.tobii.com/hc/en-us/articles/212818309-Specifications-for-EyeX

provides accurate gaze data most of the time, there are occasional outliers in the gaze samples that have to be taken care of. Unfiltered, these outliers would influence the accuracy of *PlayGuide* and result in the player scrutinizing the otherwise unobtrusive *gaze direction* cues. Additionally, the individual gaze samples are classified into the different eye movement patterns (saccades and fixations) according to a method proposed by Salvucci and Goldberg [37]. To ensure that the current gaze coordinates (filtered or unfiltered) can be accessed anywhere in the implementation, a *singleton* manager-class called `GazeManager` collects individual gaze samples in form of a `GazeSampleObject` that contains the screen-coordinates of the gaze sample and a timestamp, determining when the gaze sample was recorded by the eye tracker. Additionally, the `GazeManager` provides other important data in form of

- the current raw and unfiltered gaze sample,
- the current filtered gaze sample,
- which eye movement the current gaze sample can be classified as,
- whether the player's gaze is available or not and
- the distance from the player to the screen.

The categorization of a gaze sample into different eye movements is important as *PlayGuide* relies on the accuracy of this information. The categorization of a gaze sample into either a fixation or saccade is determined by the `GazeType` class that uses a technique based on a paper by Salvucci and Goldberg [37], who established different methods of identifying fixations and saccades in eye-tracking protocols. Salvucci and Goldberg group those methods into velocity-based, dispersion-based and area-based algorithms. While each of the algorithms presented in their paper offers specific advantages and disadvantages, the basis for the algorithm used in this implementation, called *Velocity-Threshold Identification*, was chosen due to the fact that it "runs very efficiently, and can easily run in real time." [37]. This constitutes a major advantage considering that performance is a crucial metric in games. The idea of this approach is that the velocity between each of the individual gaze samples is measured and a simple threshold then defines whether a gaze sample must be a saccade or a fixations. The velocity is defined as degrees per second, determined by how fast the eyes move from one point to another on the screen. Thereby, low velocities (below 100 degrees per second) indicate a fixation and fast velocities (above 300 degrees per second) indicate that the eye exhibits a rapid spatial relocation and thus can be categorized as saccade. Knowing the distance from the player to the screen, the velocities can be easily calculated for each gaze sample using the law of cosine. Unfortunately, the *Tobii EyeX* eye tracker used for this thesis does not scan the distance of the observer to the screen and the value of the distance had to be set manually set before the start of each game. This information was then used in the implementation to categorize each gaze sample as saccade or fixation.

The categorization of the gaze samples was then further needed by the `GazeInput` class to filter the data accordingly. During a saccade, the gaze sample coming from the eye tracker could directly be used, since the data needed to be accurate and immediately available during these volatile eye movements. During fixations, however, the eye tries to focus on a specific part of the image and exhibits only small movements called tremors, that occur because the eye can never be completely still (as mentioned in Section 2.1.2). These individual gaze samples were grouped together and

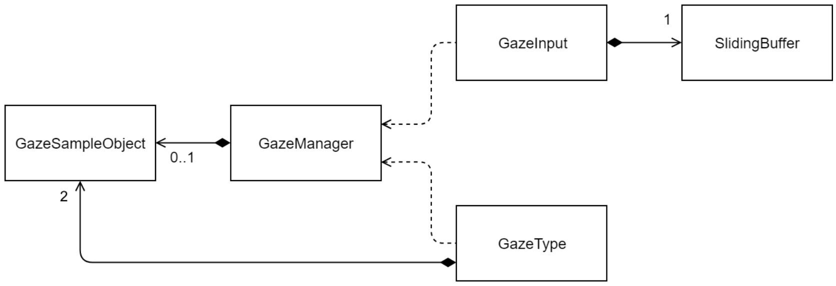


Figure 5.1: This diagram represents a simplified overview of the classes that were responsible for filtering and classifying the gaze data coming from the eye-tracking device.

averaged to obtain a smooth result, called the centeroid of the fixation. This was achieved with the class `SlidingBuffer`, which contained a *queue* that stored every consecutive `GazeSampleObject` that was categorized as fixation. As soon as a `GazeSampleObject` was marked as saccade, the *queue* was cleared and only one data object (the current gaze sample) represented the saccade. For clarification, Figure 5.1 gives an outline of how the system and its classes are structured. After the gaze samples are categorized and freed from input noise (such as outliers and tremors), they could be used for the core of this implementation, *PlayGuide*.

5.2 Realizing PlayGuide

Usually, computer games use visual cues to guide players throughout the gaming experience. Amongst many, these cues can be color codes, the careful placement of in-game properties or the way a level is structured. *PlayGuide*, however, sets out to achieve the same kind of guidance in a more subtle manner by using cues that guide the players' attention unconsciously towards a region of interest. This means that it attempts to draw the players' attention to specific regions or objects within the game that might help the players to accomplish their tasks. As a means to achieve the subtlety of the cues, the *subtle gaze direction* (SGD) technique by Bailey et al. [1] was used and adapted for the context of gaming. In their paper, the authors successfully attempt to alter the gaze behaviour of the viewer. To draw the attention to previously uninteresting areas in an image they make use of eye-tracking and subtle image-space modulations. When inspecting an image, the peripheral vision (which is highly sensitive towards detecting stimuli) first identifies regions that might be of interest to the viewer and signals the foveal vision to focus on that specific region to get a clear and sharp view on whatever is deemed as interesting [29]. By applying the subtle modulations to regions in the peripheral vision of the viewer in an image, Bailey et al. signal the viewer's eye to move the foveal vision towards the modulation and investigate the newly detected stimulus, resulting in a rapid movement (saccade) from the current fixated area of interest to the new one. In their studies, Bailey et al. used black and white (luminance) and warm-cold modulations to assess the validity of their hypothesis and concluded that although both were successful in guiding the viewer's gaze, luminance modulations were more effective than warm-cold modulations.

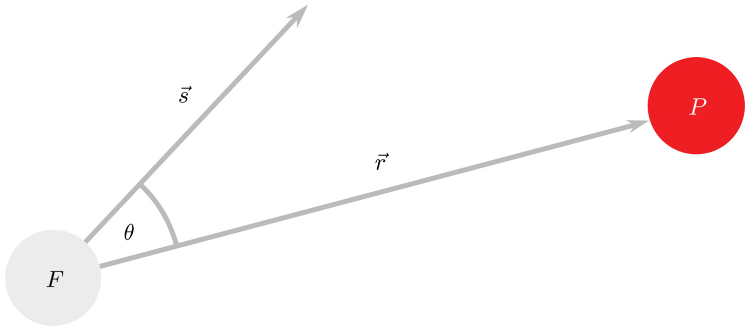


Figure 5.2: This illustration depicts the angle θ between the vector $\vec{r}$ (from the last recorded fixation F to the current area of interest P) and the vector $\vec{s}$ (from F to the current position of the saccade). Illustration based on [1].

As a means of being subtle in applying the gaze direction stimuli, Bailey et al. used modulations that were just suffice enough to alert the peripheral vision and calculated a threshold by using samples collected in separate studies. Furthermore, the viewer was never able to fixate on and investigate the modulations, as they were terminated, as soon as the viewers foveal vision performs a saccade towards the modulated region. Although saccadic movements are extremely fast, saccadic masking, a phenomenon described in Section 4.2.2, leaves enough time to stop the ongoing modulations. Figure 5.2 visualizes the idea of the aforementioned process. Assuming that P is the area of interest located in the peripheral vision of the viewer and F represents the last recorded fixation, the angle θ between the vector of $\overrightarrow{FP}$, called $\vec{r}$, and the current saccade, defined as the direction vector $\vec{s}$, can be calculated. The modulation in P is terminated, as soon as θ is smaller than a certain threshold, indicating that the viewer's eye is performing a saccade towards P. Therefore, the angle θ is calculated as

$$\theta = \arccos\left(\frac{\vec{r} \cdot \vec{s}}{|\vec{r}||\vec{s}|}\right). \tag{5.1}$$

Furthermore, it is important to terminate the modulation during a saccade, to make proper use of saccadic masking. To create movement, the modulations alternate between colors at a speed of 10 Hertz. In the case of the luminance modulations, the colors are alternated between black and white and overlayed on top of the original image by using the formula

$$col'(p) = ((l \cdot i) + col(p) \cdot (1 - i)) \cdot f(p) + col(p) \cdot (1 - f(p)). \tag{5.2}$$

The modulated color $col'(p)$, where $\boldsymbol{p} = (x, y)$ represents a color value of a pixel in the region of interest P and is calculated by using l, the color of the luminance modulation (either black or white), the intensity of the modulation i, the original color of the pixel $col(p)$ and a Gaussian falloff function called $f(p)$.

This successful approach of the SGD technique was also used for *PlayGuide* and describes the core functionality of the guidance system, which was implemented as the

(a) (b)

Figure 5.3: *PlayGuide* applies subtle image-space modulations in the luminance channel during a game. The system alternates between white (a) and black (b) modulations to create movement in the peripheral vision of the players to attract their gaze and trigger a saccade towards specific regions of interest. The modulations were outlined red in both figures.

`SubtleGazeDirection` class. Additionally, the original idea proposed by Bailey et al. was adapted and improved by making the size and the intensity of the modulation dependent of the distance from the gaze sample to the modulated area and the distance of the player to the screen. The closer the gaze is to the modulated area the smaller the intensity and size of the modulation. Thus, the `SubtleGazeDirection` class, includes two values that store the minimum and maximum intensity of the modulation that can individually adapted to each player and set during the calibration phase of the guidance system. The base value of the modulation radius, defined by the function $f(p)$ in Equation 5.2, was adjusted according to the distance from the player to the screen and the distance from the gaze sample to the modulated area. An example of the modulation applied by *PlayGuide* to a game can be seen in Figure 5.3.

To provide the `SubtleGazeDirection` class with the position of the area that needs to be modulated on the screen, individual game objects in a scene can be marked as `PointOfInterest`. The `PointsOfInterest` class then supplies *PlayGuide* with a random point of interest that is currently in the viewport of the rendered scene. Also, classes that implement the interface `IPointOfInterestSelectionProcess` can be used, if the points of interest need to be displayed according to specific criteria, like a prespecified order of appearance.

5.3 Additionally Implemented Gaze Direction Techniques

To examine the potential advantages and disadvantages of *PlayGuide*, two additional *gaze direction* techniques have been implemented: an OGD technique that also relies on the the player's gaze and a traditional *gaze direction* technique that is independent of the player's gaze, already popularized in computer games. The latter represents traditional visual cues, like lights, colors or image overlays that already are a standard in computer games and used, just like *PlayGuide*, to guide the players' attention towards important objects in a scene. For more information on the individual *gaze direction* techniques, see Section 4.2.

Furthermore, the three gaze direction techniques – *PlayGuide*, the OGD technique and the traditional visual cues – were applied to the two different prototype game scenarios representing contrasting game genres. This deliberate effort is supposed to provide an extensive insight into the capabilities and shortcomings of *PlayGuide* in the context of different game genres as well as other existing emphgaze direction techniques. This means that within each of the two game scenarios, all three *gaze direction* techniques were compared to each other. The following subsections will describe the two additionally implemented gaze direction techniques.

5.3.1 Overt Gaze Direction

As mentioned in Section 4.2.1, OGD techniques rely on obvious pull cues that try to attract the attention of the viewer and represent a contrary approach to SGD. To use OGD in the conducted experiments and apply it to a game scenario, the core mechanics of *PlayGuide* were used and adapted. The `OvertGazeDirection` class is responsible to present and apply the same image-space modulations as *PlayGuide* and its `SubtleGazeDirection` class (see Figure 5.3). However, the `OvertGazeDirection` class functions without using the techniques proposed by Bailey et al. [1]. While *PlayGuide* terminates the modulations as soon as the player's gaze saccades towards the region of interest, the `OvertGazeDirection` class continuously shows the modulations and does not stop modulating, even when the player's gaze is directly fixating on the region of interest. This implies that the modulations are occurring regardless of whether the modulated area is in the peripheral or foveal vision of the player. Furthermore, the modulation intensity cannot be individually adapted for subtlety but is always clearly visible, meaning that during maximum intensity the black and white colors almost completely cover the underlying pixel color values at the center of the Gaussian falloff function, mentioned in Equation 5.2.

To get a better understanding of the matter, Program 5.1 exemplifies how the method `CalculateModulationIntensity` is used to determine the intensity of the modulation in both *PlayGuide* and the `OvertGazeDirection` class. First, the position (i.e., the pivot) of the game object that is marked as point of interest is mapped to the *Unity* viewport coordinates to calculate the distance of the gaze to the point of interest. Afterwards, the intensity is calculated by linearly interpolating between the minimum and maximum modulation intensity values. Finally, the method returns the calculated intensity value, which is then used by the image effect shader that applies the modulation to the rendered image. A very similar process is used to calculate the modulation radius, which can be seen in Program 5.2.

5.3.2 Gaze Independent Direction Techniques

While SGD (used by *PlayGuide*) and OGD techniques are dependent of the player's gaze and require the input of an eye-tracking device, traditional techniques to guide

Program 5.1: This method describes the process of calculating the modulation intensity applied to the image by *PlayGuide* and the overt gaze direction technique.

```
private float CalculateIntesityModulation(Vector3 pointOfInterest) {
  Vector3 viewportPoint = Camera.main.WorldToViewportPoint(pointOfInterest);

  float distanceGazeToPointOfInterestNormalized = Mathf.Min(Vector2.Distance(new
    Vector2(viewportPoint.x, viewportPoint.y), GazeManager.Instance.
    SmoothGazeVectorNormalized), 1f);

  return Mathf.Lerp(MinModulationIntensity, MaxModulationIntensity,
    distanceGazeToPointOfInterestNormalized);
}
```

Program 5.2: This method describes the process of calculating the modulation size applied to the image by *PlayGuide* and the overt gaze direction technique.

```
private float CalculateSizeModulation(Vector3 pointOfInterest) {
    Vector3 viewportPoint = Camera.main.WorldToViewportPoint(pointOfInterest);

    float distanceGazeToPointOfInterestNormalized = Mathf.Min(Vector2.Distance(new
    Vector2(viewportPoint.x, viewportPoint.y), GazeManager.Instance.
    SmoothGazeVectorNormalized), 1f);

    float modulationRadiusNormalized = ConvertDegreesToViewportDimension(Mathf.Lerp(
    MinModulationSize, MaxModulationSize, distanceGazeToPointOfInterestNormalized));

    return modulationRadiusNormalized * 2;
}
```

the player's gaze already used in computer games do not require the use of an external hardware. Such techniques are implemented in the form of visual cues that try to direct the attention of the player to areas of interest during game play. Amongst many, common approaches are the deliberate placement of lights, the structuring of game levels, image overlays, color codes, the use of clarity and blur or objects that visually stand out in a scene to signify importance or interaction possibilities [66, 35]. To compare *PlayGuide* to these traditionally used visual cues, the prototype game scenarios were also tested using these visual cues that do not consider the gaze of the player. Naturally, not every visual cue is appropriate and fitting for every game setting and they have to be carefully integrated, as not to distract the player from playing the game but rather to support the player and serve as assistance. This is why the players were guided by individual visual cues, adjusted to fit the setting and mood of each of the two tested game prototypes.

The first game scenario, *Game A*, was set in space. With keyboard and mouse, players directed a spaceship viewed from the outside throughout the infinite vastness of space. To increase their score, players had to shoot small blue planets that appeared in the player's field of view. As a means to indicate the position of a planet that was about to appear, the game's head-up display (HUD) displayed white markers centered on the planet's position, indicating that the HUD had already detected the target (as

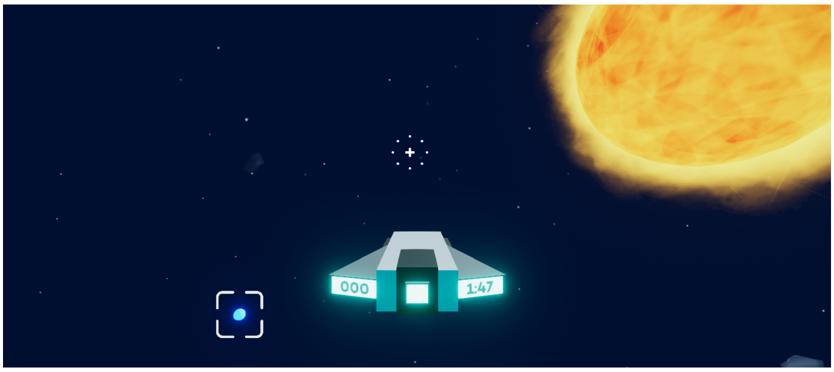


Figure 5.4: To indicate targets that were about to appear in the field of view of the player, the head-up display showed white markers at the target's position.

Figure 5.5: The green glowing arrow pointed at game objects in the scene that were collectible and thus important to the player.

can be seen in Figure 5.4). This is a common approach in the genre of space shooters. An example of such an implementation can be seen in the action-focused single-player space shooter *Everspace* [65], where the game developers used image overlays to tag enemies, show important information and indicate objects of significance to the player.

The second game scenario, *Game B*, was set in medieval times and used thematically fitting three-dimensional models from the *Unity Asset Store*[5]. The player could navigate a first-person player through an alchemist's house and had to retrieve as many hidden coins as possible. While an image overlay would have been possible in this game Game As well, it would not have fitted the mood and the style of the setting. A technique often used as a game mechanic in first-person games such as *Planetside 2* [68] or *Dishonored 2* [67] is to tag enemies, allies or targets by displaying a colorful arrow above the target. Hence, to indicate the positions of the coins that needed to be retrieved in the test game, a green glowing three-dimensional arrow appeared and hovered above the coin to arouse interest as soon as the player got close. The visual cue clearly stood out from the rest of the scene, as its bright green color and slow movements stood in stark contrast to the plain brown static interior of the game. The implementation of visual

[5] https://assetstore.unity.com/packages/3d/environments/fantasy/alchemist-s-house-interior-47318

cue used in *Game B* can be seen in Figure 5.5. An even more detailed insight into the implementation of both test games is given in the following section.

5.4 Designing the Test Games

Besides the additional *gaze direction* techniques that have been realized, two game scenarios were developed in order to test the performance of *PlayGuide* in the context of two contrasting game genres, named *Game A* and *Game B*. To explore the limits of *PlayGuide* each of the games was deliberately designed to be vastly different from the other. This section shares details about the implementation and content of the two games developed.

5.4.1 Game A

Game A was set in space. From a third-person view the player could steer a spaceship by moving his or her mouse into the according direction on the screen (see Figure 5.6). The further the player moved the mouse towards the edge of the screen, the faster the ship would turn. Thus, in order to stop the spaceship from turning at all, the player had to center the mouse cursor in the middle of the screen. While this seems intuitive, the position of the screen center can only be roughly estimated by the player and succumbs to a subjective impression of where the center actually is located. To counteract frustrations with the controls and that the mouse cursor always had to be placed directly at the center of the screen to stop turning, the method `CenterAndSquareMousePosition` used the equation

$$f(m') = \begin{cases} m^2 & \text{if } s > 0 \text{ and } m \geq 0, \\ -m^2 & \text{if } s > 0 \text{ and } m < 0, \\ m^4 & \text{if } s = 0 \text{ and } m \geq 0, \\ -m^4 & \text{if } s = 0 \text{ and } m < 0. \end{cases} \tag{5.3}$$

where m is the mouse input in the range of $[-1, 1]$ on either the x- or y-axis and s the speed of the spaceship. In other words, the method used both a square function while the spaceship was moving and a quartic function while the spaceship was standing still. This ensured that towards the center, the mouse had less influence on the rotation of the spaceship and gained sensibility towards the edge of the screen. Additionally, while the spaceship was standing still, the mouse input had even less influence towards the center of the screen so that the player could aim better, without influencing the rotation of the spaceship right away.

Considering that the player was controlling the spaceship from outside (as can be seen in Figure 5.6) and to convey an engaging game experience, the camera was not simply attached to the spaceship but rather followed the spaceship's movements smoothly by linearly interpolating the position and rotation of the camera. This was supposed to enhance the severity of the spaceship's movements, make the game environment more dynamic and create an engaging experience for the player. Program 5.3 shows how this effect was implemented.

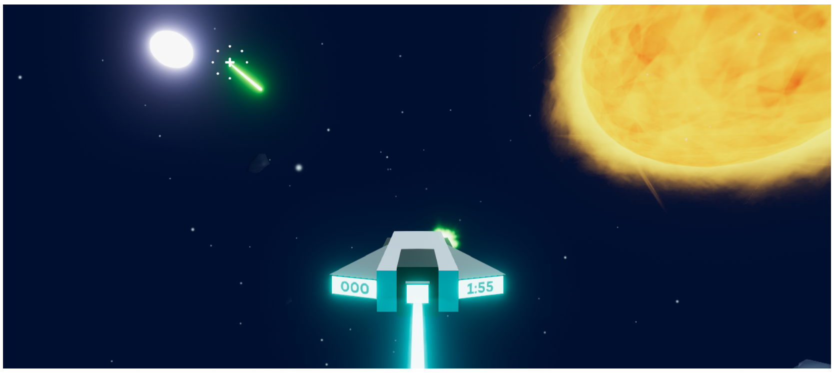


Figure 5.6: The first scenario of the test games was set in space. The player had to control a spaceship by using mouse and keyboard input. To increase the score, blue targets that appeared throughout the game had to be shot within a given time frame.

Program 5.3: The camera continuously follows the spaceship in a smooth manner to create the feeling of a dynamic game environment. To achieve this effect the position and rotation of the camera is linearly interpolated towards the position and rotation of the spaceship.

```
void Update ()  {
    Vector3 desiredPosition = Target.position;
    Quaternion desiredRotation = Target.rotation;

    Quaternion smoothedRotation = Quaternion.Lerp(transform.rotation,
    desiredRotation, SmoothSpeedRotation);

    Vector3 smoothedPosition = Vector3.Lerp(transform.position, desiredPosition,
    SmoothSpeedPosition);

    transform.rotation = smoothedRotation;
    transform.position = smoothedPosition;
}
```

The aim of *Game A* was to shoot as many targets as possible within a given time frame and can be categorized as third-person shooter game. Although the targets, visualized in the form of glowing blue planets, were occurring one after another and there was no pause between the appearance of individual planets, the `TargetSpawner` class refrained from showing new targets whenever the player performed a severe turn with the spaceship. This prevented that targets appeared outside of the player's view while the camera was rotating. Furthermore, the targets never appeared near the position of the spaceship or close to the players gaze, to avoid that they were hidden by the carcass of the spaceship or too easy to shoot. Targets themselves appeared with half their initial scale and then were animated to scale up to their original size. They then remained on the screen for a certain amount of time (0.6 to 0.8 seconds in the game tests) and started to blink in a different color shortly before disappearing. To ensure that the three individual gaze direction techniques were able to show their visual cues,

Figure 5.7: *Game B* was set in medieval times, in the house of an alchemist. Player could navigate through the environment using mouse and keyboard.

the `TargetSpawner` class also allowed for a time threshold during which the position of the planet about to be spawned was already determined and enough time was available to draw the attention of the player to the target's position before it appeared. Furthermore, to ensure that players could not fly into objects or targets within the scene, the `KeepAtDistance` class continuously repositioned all objects to keep their original distance to the player.

Finally, to shoot a target players merely had to position the mouse cursor (that was replaced by an image of a crosshair) over the target and click the left mouse button. The game then immediately marked a target as shot. To provide visual feedback, bullets appeared in the form of a glowing green laser that formed an infinite line from the spaceship's muzzle to where the player clicked on the screen and quickly decreased in length to create the illusion of an actual bullet that travelled towards where the player had clicked (see Figure 5.6). Immediately after the bullet appeared, the planet was removed from the game scene, visualized as an explosion. In case the player missed the target, the planet simply decreased in size until it disappeared from the view.

5.4.2 Game B

Game B was designed to be in stark contrast with *Game A* and realized as a first-person exploration game (see Figure 5.7). While the former game scenario was an action-heavy third-person space shooter where targets had to be shot as quickly as possible before they disappeared, the latter game scenario did not have any time constraints imposed on the game mechanics whatsoever. To the contrary, players could freely explore the game Game And were asked to thoroughly investigate the scenery to find as many hidden objects as possible.

Set in medieval times, the protagonist of the game was placed in an alchemist's house (see Figure 5.7). Players, however, never got to see the protagonist but were able to maneuver the character by controlling a first-person camera. By using the "W, A, S, D" keys, the camera could be moved forwards, left, backwards or right to navigate through the scene. Players were able to go and see every room in the house but had no chance of leaving the house at any point in the game. Additionally, players could crouch, thus

(a)

(b)

Figure 5.8: The goal of *Game B* was to find as many hidden objects as possible. These objects were in the form of a golden coin that can be seen in figure (a). Figure (b) depicts a typical example of how the coins were hidden throughout the game.

lowering the camera, by pressing the shift key. A core mechanic of many first-person games, jumping, was not implemented, as it was not needed throughout game play. Furthermore, to recreate natural head movements of the character, the `SmoothRotation` class allowed for an effortless rotation of the camera. Instead of directly taking the values of the mouse input, the `SmoothRotation` class collected a list of input values over an adjustable length of time, determined as number of frames. The values were then averaged to obtain a stutter-free camera rotation.

The goal of *Game B* was to find as many hidden objects as possible. These objects were represented as golden coins (see Figure 5.8) that needed to be stolen from the house the protagonist found himself/herself in. To pick up a coin, players simply had to be close enough to the coin and select it with the left mouse button. However, the coins had to be in the player's field of view and directly clicked on to prevent players from picking up coins by accident. Overall, there were 33 hideouts throughout the game, where the coins could be hidden. To accomplish their task, players were given no information as to how many objects were hidden in the scene and could end the game whenever they felt that they had found all coins. To promote a thorough search for the collectible objects, the time was not shown on the screen and no time limit was given.

Both games, *Game A* and *Game B*, also were accompanied with rudimentary background music and essential sounds that contributed to the atmosphere of the respective game setting. While *Game A* had an electronic melody playing in the background and synthesized noises for the menu, shots, explosions and engines, *Game B* included background music reminiscent of medieval times and realistic sounds like picking up coins and opening doors. All of these measures were used to create a more realistic atmosphere and a more engaging game setting, to keep players motivated.

The purpose of implementing the two test games and the additional two gaze guid-

Figure 5.9: Heat maps are commonly used in static footage to visualize gaze behaviour. Depending on the eyes' time spent observing specific areas of an image, the according pixels are colored differently.

ance techniques, was to evaluate *PlayGuide* and to investigate the capabilities and shortcomings of the system. Besides assessing metrics like flow, immersion, tension and challenge through the *Game Experience Questionnaire* developed by IJsselsteijn et al. [17], data was also captured in form of events that occurred in the games during the testing phase. The next section will explain how the data was captured to analyze the behaviour of the players and the efficiency and effectivity of *PlayGuide*.

5.5 Data Collection

An essential addition to developing the test games was to provide a way of logging in-game events, which ensured a thorough evaluation of *PlayGuide*. Data, such as when the player's gaze fixated on certain game objects, allowed for a fact-based analysis of the guidance system. To provide and record data in form of in-game events, two essential systems were introduced. Firstly, a *gaze-to-object mapping* algorithm that mapped the player's gaze to game objects in the scene. Secondly, a system that could store in-game events to persist the data collected during the game tests.

5.5.1 Gaze-to-Object Mapping

Traditional ways to interpret the viewer's gaze behaviour include *scan paths* (mentioned in Section 4) or heat maps (depicted in Figure 5.9) that simply accumulate the time spent observing specific areas of an image and are approaches that are suited for static images, where the content of the image does not change. However, in computer games, where the camera exhibits constant and mostly rapid movements, each image to be analyzed within a sequence of images drastically differs from the one before and the gaze cannot simply be associated with pixels on the screen. *Gaze-to-object mapping* (GTOM) represents a practical approach for games that solves this problem by mapping the gaze of the player to game objects in the scene.

Although there are multiple GTOM techniques, for example by Bernhard et al. [3] and Sundstedt et al. [43], many are computation-intensive and thus not suitable for real-time analysis of the player's gaze. The GTOM technique used to evaluate the player's gaze in the test games was recommended by the the leading manufacturer of eye-tracking devices, *Tobii Technology*[6] and is called *Ray Casting Shotgun* [38]. This simple technique has low performance costs while maintaining relatively exact results. To map a game object to the player's gaze, the `GazeToObjectMapping` class casts an adjustable number of rays (15 in the test games) into the game scene, within a certain radius. The game object that is intersected by the most rays is then considered to be the fixated object. The origin of the rays was calculated by adding the value $v = \mathbb{N} \cap [0, 1]$ to the x-axis of the point $P(x, y)$ located at the origin $(0, 0)$, multiplying the point by the radius of the foveal vision converted into pixels on the screen, rotating it by using

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}, \tag{5.4}$$

where $\theta \in \mathbb{N} \cap [0, 360]$ and translating the position of the ray by the position of the gaze, which ensured a denser distribution of the rays towards the center of the gaze and emulates the decrease of visual acuity in the fovea. Finally, the game object that is intersected by the most rays is considered to be the fixated object.

However, not all game objects were tracked throughout the testing of the games. Only the ones that were interesting for later analysis and marked as point of interest, such as the planets in *Game A* and the coins in *Game B* were analyzed by the GTOM algorithm and triggered an in-game event that was recorded. How these events were recorded and saved for later analysis is detailed in the following subsection.

5.5.2 Logging In-Game Events

As a means to persist the data that was collected throughout testing *PlayGuide*, a class called `DataRecorder` provided methods to record in-game events and save them as soon as the game was exited. To be accessed anywhere in the implementation, the `DataRecorder` class was realized as a *singleton* class that managed a list of in-game events. An in-game event was represented by a `DataSet` object that contained detailed information about the event: the time the event occurred in seconds since the game has started, the name and a unique identifier of the game object the event was associated with and an enumerator value describing the action. The latter was defined as the `Action` enumerator, which contained predefined values to describe different in-game events. An example `DataSet` would contain the following values: "1.12", "Coin1", "289712", "Collected". To add a new entry to the list of game events stored in the `DataRecorder`, the `AddNewDataSet` method was be called the following way:

```
DataRecorder.Instance.AddNewDataSet(Time.time, gameObject, Action.Collected);
```

[6] https://www.tobii.com/

Program 5.4: To persist the data that was collected during the test games, the data was stored as a "comma-separated values" file that could be easily read by many other programs.

```
public void WriteDataToCsv() {

    List<DataSetNew> sortedList = _dataSets.OrderBy(o=>o.Time).ToList();

    if (sortedList.Count > 0) {
        string filePath = CreateNewCSVFile();
        StreamWriter writer = new StreamWriter (filePath);

        writer.WriteLine (sortedList[0].GetDataSetHeader());

        for (int i = 0; i < sortedList.Count; i++) {
            writer.WriteLine (sortedList[i].GetDataSetData());
        }

        writer.Flush();
        writer.Close();
    }
}
```

Table 5.1: A typical example how the collected data was organized in a spreadsheet.

Time (ms)	*GameObject*	*ID*	*Action*
0.00000	GameManager	9172	GameStarted
1.12458	Kitchen	9187	RoomEntered
2.20385	Coin1	9149	GazeAttendedObject
3.93643	Coin1	9149	Collected
5.28476	Kitchen	9187	RoomExited
8.93751	GameManager	9172	GameExited

To save the list of `DataSet` objects, the stored events were ordered by the time of their appearance and written to a comma-separated values (CSV) file. CSV files represent a simple file format to save or exchange data and have the advantage that it can be easily read and imported into many programs, including *Microsoft Excel*[7] or *SPSS*[8], which was needed for the data analysis. The method `WriteDataToCsv`, responsible for writing the data to a CSV file, is detailed in Program 5.4. After collecting and persisting the data, the CSV files were imported into the analytics program *SPSS*, where the data was structured and easily readable as in the example shown in Table 5.1.

In summary, the implementation of *PlayGuide* did not merely encompass the system itself but a multitude of other systems and components needed to evaluate the system.

[7] https://products.office.com/de-at/excel

[8] https://www.ibm.com/analytics/data-science/predictive-analytics/spss-statistical-software

Besides *PlayGuide*, two additional gaze direction techniques have been implemented as well as two test games to thoroughly test the gaze direction techniques within different game genres. Furthermore, a system was developed to collect and save in-game events that were needed to ensure a fact-based analysis in addition to the qualitative and quantitative interviews conducted with the test subjects.

Chapter 6

Evaluation

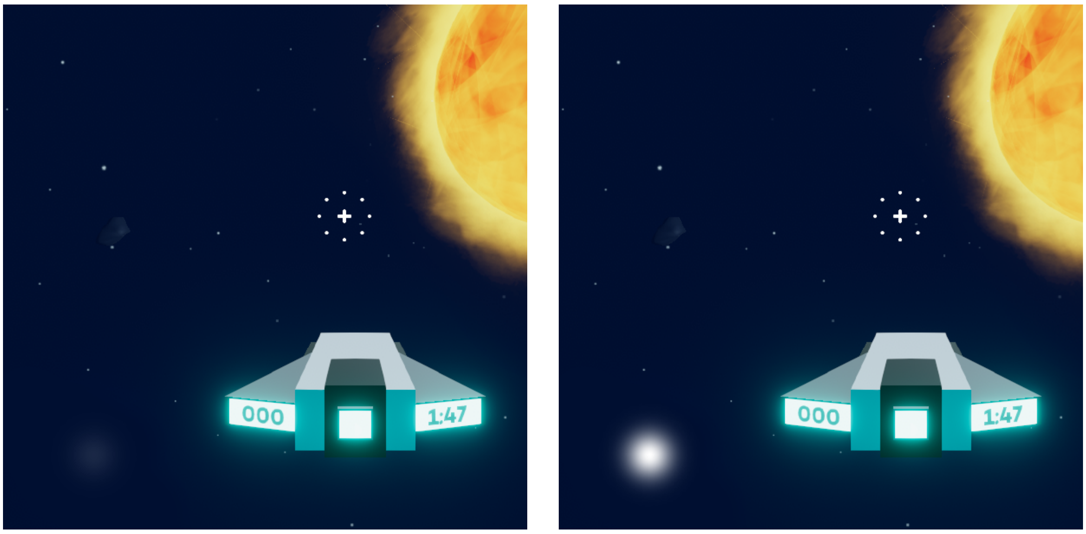
As a means to provide a comprehensive evaluation of *PlayGuide*, the visual gaze-based player guidance system was tested thoroughly in two different game contexts (*Game A* and *Game B*) and compared to two additionally implemented *gaze direction* techniques. First, to get elaborate insights into the evaluation of *PlayGuide* and the consequentially gained results, the conditions and test scenarios are detailed in this chapter. Furthermore, the study setup and testing procedure is explained. Finally, the participants, data analysis and results are also presented in this chapter.

6.1 Conditions and Test Scenarios

Besides *PlayGuide* itself, two additional *gaze direction* techniques were implemented and tested to determine the impact of *PlayGuide* compared to other existing techniques. This resulted in a total of three conditions, differentiated by their individual approaches to direct the player's gaze. However, all three conditions were used to direct the player's gaze towards important objects within the games that were tested.

6.1.1 Condition 1 – Traditional Gaze Direction

The first condition represented *traditional gaze direction* (TGD) techniques that are independent of the player's gaze, already popularized in computer games. Traditional visual cues, like lights, colors or image overlays that already are a standard in computer games and used, just like *PlayGuide*, to guide the players' attention towards important objects in a scene. To fit the style of the two game scenarios that were tested, the TGD condition had to be individually adapted to the games' environment, targets and mechanics. In the first game scenario players could steer a spaceship and had to shoot targets, in the form of small blue planets. As a means to indicate the position of a planet that was about to appear, the game displayed an image overlay, visualized as white markers centered on the planet's position (as can be seen in Figure 5.4). This is a common approach in the genre of space shooters. For example, the game *Everspace* [65], an action-focused single-player space shooter, uses similar image overlays to tag enemies, show important information and indicate objects of significance to the player. The second game scenario was set in medieval times and realized as a first-person exploration game. To indicate the positions of the coins that needed to be collected,



(a) (b)

Figure 6.1: Figure (a) shows an example of the image space modulation applied by *Condition 2* (*PlayGuide*), while Figure (b) shows the image space modulation applied by *Condition 3* (overt gaze direction).

a green glowing three-dimensional arrow appeared and hovered above the coin, clearly standing out from the rest of the scene as its bright green color and slow movements stood in stark contrast to the plain static interior of the game (see Figure 5.5). This technique is also often used as a visual cue in first-person games such as *Planetside 2* [68] or *Dishonored 2* [67] to tag enemies, allies or targets by displaying a colorful arrow above the target.

6.1.2 Condition 2 – PlayGuide

The second condition was represented by *PlayGuide*, the actual focus of this thesis, and can be categorized as a *subtle gaze direction* technique. It uses information about the player's gaze to subtly apply image-space modulations at specific points of interest in the peripheral vision of the player to unobtrusively attract the player's gaze. However, as soon as the eyes relocate towards the point of interest, the modulations caused by *PlayGuide* are terminated leaving the player in the unknown about what guided his or her gaze in the first place. Also, it has to be noted that *PlayGuide* did not guide the player's gaze directly to the point of interest but only roughly into the right direction, which is due to the *subtle gaze direction* technique *PlayGuide* is based on (more on which can be read in Section 5.2).

6.1.3 Condition 3 – Overt Gaze Direction

The third condition was an *overt gaze direction* (OGD) technique, which also uses information about the players gaze. However, instead of subtly applying image space modulations (like *PlayGuide*), the OGD condition relied on clearly visible image modu-

Figure 6.2: The test subjects played two different games with keyboard and mouse. To track eye movements, the eye-tracking device was mounted below the monitor.

lations to direct the gaze of the player. Furthermore, unlike *PlayGuide*, the modulations were not terminated when the player's gaze was near the point of interest. The difference in the visibility of the modulations between *PlayGuide* and *Condition 3* can be seen in Figure 6.1.

6.1.4 Test Scenarios

For this thesis' comparative study, all three conditions listed above were tested in two different test scenarios. First, a third-person shooter set in space (*Game A*), in which as many targets had to be shot within a given time frame. Secondly, a first-person exploration game set in medieval times (*Game B*), in which hidden objects had to be found – as many as possible, no time frame given. The details of the according implementations can be read in Section 5.4.

The two test scenarios used the conditions mentioned in this section in different and unique ways. While *Game A* used the *gaze direction* techniques to steer the gaze towards relevant objects that were going to appear (much like an early warning system), *Game B* directed the gaze towards objects that were already in the game, but hidden from the player's view by other game objects. The following section will detail how the two games and the three *gaze direction* conditions were tested.

6.2 Participants and Procedure

The experiment, which was solely conducted at the *University of Applied Sciences Upper Austria Campus Hagenberg*, used the same hardware to provide identical testing conditions for every participant. All game scenarios were tested with a keyboard and mouse, to gather and analyze the gaze input of the participants, the eye tracker *Tobii*

EyeX[1] was used. Additional hardware was comprised of a standard desktop PC, a 4K 29-inch monitor and a stereo-headset. The use of the headset during the testing procedure was mandatory, as the games featured sound effects and background music to further convey the games' atmosphere.

A total of 20 people participated in testing procedure of *PlayGuide*, 11 male and 9 female participants between the ages of 16 and 54 (M = 25.9, SD = 8.50). The structure of the testing procedure was as follows: First, the participants of the study were welcomed and the eye tracker was calibrated to fit the eye movements of the individual. After the initial calibration phase, participants were given a questionnaire to fill in their age and level of gaming experience (ranging from novice to expert). Subsequently, participants were introduced to the subject matter and the scope of the experiments, which included that different *gaze direction* techniques and two different games were tested. Also participants were informed that each of the games had to be played three times in order to test each of the three conditions. Details about the individual conditions, however, were not shared. After that, the intensity of the modulations to be applied by *PlayGuide* was calibrated for each participant and game. Finally, participants were allowed to play each of the two games with each of the three gaze direction techniques (i.e., TGD, *PlayGuide* or OGD) applied. Thus, all participants had to play a total of six times. To avoid biases, the order of the test games and conditions was randomized.

Before playing any of the games, participants were reminded that *Game A* had a time limit and a high score that needed to be achieved, whereas *Game B* was about exploration and had no time constraints whatsoever. Participants were also instructed to end *Game B* whenever they felt they had reached the game's target. When the target of a game scenario was reached, players were asked to fill in the *Game Experience Questionnaire* by IJsselsteijn et al. [17]. After all games and conditions had been played, a qualitative interview was held with the participants to asses their opinion about the different *gaze direction* techniques in the context of the two different test games. In total, the procedure was about 50 to 60 minutes per participant.

6.3 Evaluation Methods

The two test games and the three *gaze direction* conditions were tested using different evaluation methods. Firstly, after each condition the participants of the study had to fill in a questionnaire called *Game Experience Questionnaire* by IJsselsteijn et al. [17]. It uses a total of 33 statements to measure the seven different factors competence, sensory and imaginative immersion, flow, tension/annoyance, challenge, negative affect and positive affect. The following list contains an example statement for each factor that was assessed by the *Game Experience Questionnaire*, rated on a five-point Likert scale ranging from "not at all" to "extremely":

- Competence (*Com*): "I felt skilful".
- Sensory and imaginative immersion (*Sen*): "It was aesthetically pleasing".
- Flow (*Flo*): "I was fully occupied with the game".
- Tension/Annoyance (*Ten*): "I felt annoyed".

[1] https://help.tobii.com/hc/en-us/articles/212818309-Specifications-for-EyeX

- Challenge (*Cha*): "I thought it was hard".
- Negative affect (*Neg*): "It gave me a bad mood".
- Positive affect (*Pos*): "I enjoyed it".

A list of all statements to assess the 7 different factors can be found in Appendix B.

Secondly, during game play data was collected in the form of in-game events that were comprised of a timestamp, name, unique identifier and action that described the event. As the *Game Experience Questionnaire* only captures subjectively perceived impressions by the players, such as competence, immersion and flow, the in-game events provided objective data on performance (effectivity and efficiency) of the conditions that could be compared and used to validate if the evaluation of the games by the participants of the study actually coincided with the data that was collected or if the subjectively perceived impressions of the players stood in contrast to the actual performance of the individual conditions. The actions were not universally used but individually adapted to the games and the information needed to properly evaluate the games. In *Game A* (detailed in Section 5.4) the actions included the following: "marked as point of interest", which described the event whenever a target was marked as point of interest in a game and thus the moment when the conditions started to show visual cues to alert the player of a target that was going to appear. "Appeared" described the time the target actually appeared on the screen and "attended" the time the player's gaze fixated on a target. Furthermore, the event "shot" indicated the time the players clicked on a target and thus initiated their shot. Finally, for each player the modulation intensity was measured by the "modulation intensity" event. In *Game B* (detailed in Section 5.4) the actions included the following: "time spent in game", which described the time the player spent in the game and "coins", which was the number of coins the player found throughout the game. "times attended" counted the total number of times a player attended the coins and "rooms visited" the number of times a player visited the rooms in the game. As in *Game A*, in *Game B* the modulation intensity was also measured for each player by the "Modulation intensity" event.

Thirdly, after a participant had tested all the test games and conditions, a qualitative interview was held to assess his or her opinion about the individual conditions in the games, initiated by the question: "The games that you played tested different ways of guiding the player's gaze. Which gaze guiding technique, in which game, did you like best?". The answers from the participants to this question can be found in Appendix A.

6.4 Results

To compare the results of the three conditions obtained through the *Game Experience Questionnaire* and the in-game events, the analyses were performed using the repeated-measures analysis of variance (rANOVA) after validating that the condition of sphericity was satisfied. The results of Mauchly's sphericity tests for the *Game Experience Questionnaire* from *Game A* and *Game B* can be found in Table 6.1. All Mauchly's sphericity tests conducted for the collected game events met the condition of sphericity. Furthermore, the results of Mauchly's sphericity tests for *Game A* and *Game B* regarding the measures that were calculated with the help of the collected in-game values proved to meet the condition for sphericity. The according values can be seen in Table 6.2 for

Table 6.1: This table contains the results of Mauchly's sphericity tests by citing the values Mauchly-W(2) and significance p for each factor of the *Game Experience Questionnaire*: competence (*Com*), sensory and imaginative immersion (*Sen*), flow (*Flo*), tension/annoyance (*Ten*), challenge (*Cha*), negative affect (*Neg*) and positive affect (*Pos*).

Factor	*A - Mauchly-W(2) (p)*	*B - Mauchly-W(2) (p)*
Com	0.991 (0.645)	0.871 (0.150)
Sen	0.994 (0.159)	0.992 (0.691)
Flo	0.858 (0.100)	0.987 (0.536)
Ten	0.680 (0.927)	0.914 (0.074)
Cha	1.000 (0.983)	0.937 (0.061)
Neg	0.820 (0.722)	0.963 (0.233)
Pos	0.886 (0.085)	0.965 (0.173)

Game A and *Game B*. To adjust the probability values of the statistical tests, the pairwise comparisons used the Bonferroni correction. This method and all aforementioned and following statistical calculations were carried out by using the statistics tool SPSS[2], with the significance α set to 0.05 for all tests. The next subsections will share the results of the conducted experiments split into the individual evaluations methods, the *Game Experience Questionnaire* and the collected in-game data.

6.4.1 Game Experience Questionnaire

In this subsection the results of the *Game Experience Questionnaire* are discussed, which will provide the data on how the different conditions (TGD, *PlayGuide* or OGD) influenced the player in terms of the seven different factors assessed by the questionnaire: competence (*Com*), sensory and imaginative immersion (*Sen*), flow (*Flo*), tension/annoyance (*Ten*), challenge (*Cha*), negative affect (*Neg*) and positive affect (*Pos*). The participants study had to fill in 33 questions that allowed answers on a five-point Likert scale, with 1 being the lowest and 5 the highest score that can be achieved.

Game A

In the first game, *Game A*, players had to maneuver a spaceship and shoot at targets that appeared at random positions in the field-of-view of the player. The *gaze direction* techniques in this scenario indicated the spawn points of the targets 0.5 seconds before they appeared. For the first factor, *Com*, the results of the rANOVA showed a significant effect on competence ($F_{2,198} = 17.57$, $p = 0.00$, $\eta^2 = 0.151$). The Bonferroni-corrected pairwise comparisons showed that the difference between *PlayGuide* (M = 3.11, SD = 0.92), which had the lowest score on competence, and TGD (M = 3.76, SD = 0.84) was significant ($p = 0.00$). Also, the difference between *PlayGuide* and OGD (M = 3.55, SD = 0.89) was significant ($p = 0.00$). However, it was also shown that the condition TGD

[2] https://www.ibm.com/analytics/data-science/predictive-analytics/spss-statistical-software

Table 6.2: This table contains the results of Mauchly's sphericity tests by citing the values Mauchly-W(2) and significance p for each measures that were calculated with the help of the collected in-game values. For *Game A*: The time between when a game object was marked as point of interest and when it was attended (*Measure A1*), the time between when a game object was marked as marked as point of interest and when it was shot (*Measure A2*), the time between when a game object was marked as attended and when it was shot (*Measure A3*), the number of targets that appeared (*Measure A4*), the number of targets that were attended by the player (*Measure A5*) and the number of targets that were actually shot (*Measure A6*). For *Game B*: The total time spent in game (*Measure B1*), the total number of coins that were found (*Measure B2*) the total number of rooms visited (*Measure B3*), the average number of times one room was visited (*Measure B4*), the average time to find a coin (*Measure B5*) and the average number of times a coin was attended before it was found (*Measure B6*).

Measure	*A - Mauchly-W(2) (p)*	*B - Mauchly-W(2) (p)*
A1/B1	0.539 (0.104)	0.931 (0.526)
A2/B2	0.991 (0.919)	0.703 (0.082)
A3/B3	0.926 (0.501)	0.819 (0.166)
A4/B4	0.983 (0.854)	0.819 (0.166)
A5/B5	0.989 (0.904)	0.342 (0.713)
A6/B6	0.832 (0.872)	0.808 (0.147)

Table 6.3: *Game A*: Means and standard deviation for competence (*Com*), sensory and imaginative immersion (*Sen*), flow (*Flo*), tension/annoyance (*Ten*), challenge (*Cha*), negative affect (*Neg*) and positive affect (*Pos*) on a scale from 1 to 5 per condition.

Factor	*Traditional*	*PlayGuide*	*Overt*
Com	3.76 (0.842)	3.11 (0.920)	3.55 (0.892)
Sen	3.02 (1.189)	2.99 (1.176)	3.02 (1.197)
Flo	3.37 (1.203)	3.44 (1.149)	3.37 (1.261)
Ten	1.33 (0.510)	1.77 (0.890)	1.37 (0.486)
Cha	2.65 (1.048)	3.27 (1.171)	2.67 (1.055)
Neg	1.46 (0.711)	1.58 (0.965)	1.51 (0.763)
Pos	4.04 (0.790)	3.78 (0.864)	4.06 (0.789)

did not differ significantly from the condition OGD ($p = 0.186$). For the second factor and third factor, *Sen* and *Flo*, the results of the rANOVA for sensory and imaginative immersion ($F_{2,198} = 0.91$, $p = 0.91$, $\eta^2 = 0.001$) and flow ($F_{2,198} = 0.33$, $p = 0.72$, $\eta^2 = 0.003$) did not show any significant results. However, tension/annoyance (*Ten*) did yield significant results ($F_{2,118} = 12.99$, $p = 0.00$, $\eta^2 = 0.18$) and Bonferroni-corrected pairwise comparisons revealed that *PlayGuide* (M = 1.77, SD = 0.89) received the highest score, thus contributing most to the perceived tension/annoyance, compared

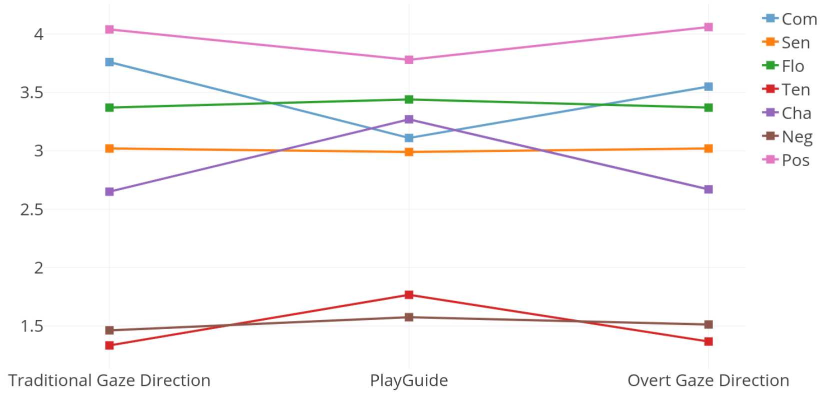


Figure 6.3: Game *Game A*: Means for the factors competence (*Com*), sensory and imaginative immersion (*Sen*), flow (*Flo*), tension/annoyance (*Ten*), challenge (*Cha*), negative affect (*Neg*) and positive affect (*Pos*). Includes all conditions (x-axis: *Traditional Gaze Direction*, *PlayGuide*, *Overt Gaze direction*) on a scale from 1 to 5.

to OGD (M = 1.37, SD = 0.49, $p = 0.001$) and TGD (M = 1.33, SD = 0.51, $p = 0.001$). Again, no significant effect could be shown between TGD and OGD ($p = 1.00$). Furthermore, the rANOVA test for the factor *Cha* revealed a significant effect ($F_{2,198} = 22.38$, $p = 0.00$, $\eta^2 = 0.18$) on challenge. Pairwise comparisons using the Bonferroni method showed significant results between *PlayGuide* (M = 3.27, SD = 0.17) and TGD (M = 2.65, SD = 0.05, $p = 0.00$) as well as *PlayGuide* and OGD (M = 2.67, SD = 0.06, $p = 0.00$). Yet, again no significant difference between TGD and OGD ($p = 1.00$). Although the factor negative effect did not show any significant differences between the three conditions using the rANOVA test ($F_{2,158} = 0.66$, $p = 0.52$, $\eta^2 = 0.008$), the factor positive affect indeed showed significant results ($F_{2,198} = 7.06$, $p = 0.001$, $\eta^2 = 0.067$). The additionally conducted Bonferroni-corrected pairwise comparisons unveiled that *PlayGuide* (M = 3.78, SD = 0.89) was significantly different compared to TGD (M = 4.04, SD = 0.79, $p = 0.024$) and OGD (M = 4.06, SD = 0.79, $p = 0.002$), while showing the lowest mean score of the three conditions. Again, TGD and OGD did not show any significant differences ($p = 1.00$). For a detailed overview of the mean values see Figure 6.3 and Table 6.3).

Game B

In the second game, *Game B*, players had to find as many hidden objects as possible. These objects were represented as coins and placed at predefined locations throughout the environment. The *gaze direction* techniques in this game indicated the locations of the hidden coins to the player. Regarding the first factor, *Com*, results of the rANOVA indicated a significant effect of the conditions on the competence ($F_{2,198} = 9.259$, $p = 0.00$, $\eta^2 = 0.086$). Bonferroni-corrected pairwise comparisons further showed that TGD (M = 3.53, SD = 0.904) differed significantly from *PlayGuide* (M = 3.25, SD = 1.019, $p = 0.04$) and that *PlayGuide* also differed significantly from OGD (M = 3.81, SD = 1.089,

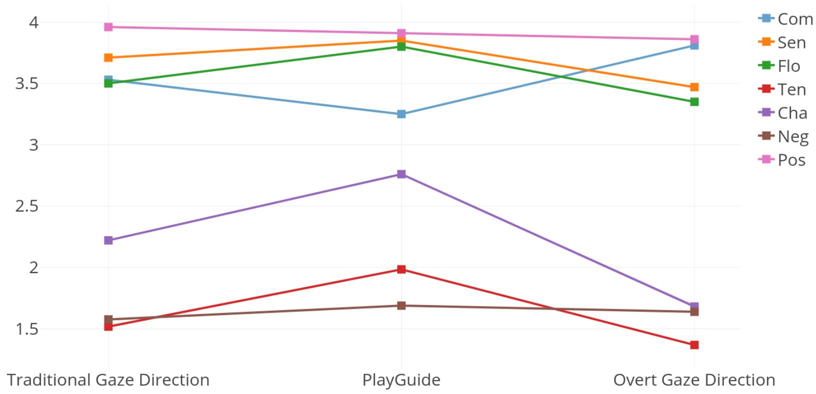


Figure 6.4: Game *Game B*: Means for competence (*Com*), sensory and imaginative immersion (*Sen*), flow (*Flo*), tension/annoyance (*Ten*), challenge (*Cha*), negative affect (*Neg*) and positive affect (*Pos*). Includes all conditions (x-axis: *Traditional Gaze Direction*, *PlayGuide*, *Overt Gaze direction*) on a scale from 1 to 5.

Table 6.4: *Game B*: Means and standard deviation for competence (*Com*), sensory and imaginative immersion (*Sen*), flow (*Flo*), tension/annoyance (*Ten*), challenge (*Cha*), negative affect (*Neg*) and positive affect (*Pos*) on a scale from 1 to 5 per condition.

Factor	*Traditional*	*PlayGuide*	*Overt*
Com	3.53 (0.904)	3.25 (1.019)	3.81 (1.089)
Sen	3.71 (1.094)	3.85 (0.978)	3.47 (1.167)
Flo	3.50 (1.133)	3.80 (1.119)	3.35 (1.226)
Ten	1.52 (0.676)	1.98 (1.142)	1.37 (0.688)
Cha	2.22 (1.186)	2.76 (1.429)	1.68 (0.839)
Neg	1.57 (0.839)	1.69 (0.894)	1.64 (0.846)
Pos	3.96 (0.790)	3.91 (0.854)	3.86 (0.921)

$p = 0.001$). However, OGD did not differ from TGD enough to be significant ($p = 0.084$). Participants clearly evaluated *PlayGuide* with the lowest scores out of all three gaze direction techniques. Also, the sensory and imaginative immersion, *Sen*, factor proved to be significant examined with the rANOVA test ($F_{2,198} = 7.173$, $p = 0.001$, $\eta^2 = 0.068$). A closer examination of the pairwise differences between the conditions revealed that TGD (M = 3.71, SD = 1.094) did not significantly differentiate itself from *PlayGuide* (M = 3.85, SD = 0.978, $p = 0.513$). However, participants scored TGD and OGD (M = 3.47, SD = 1.167) differently, with a significant difference ($p = 0.047$). The same holds true for the difference between *PlayGuide* and OGD ($p = 0.001$). According to the rANOVA tests conducted, the factor flow (*Flo*) also exhibited significant differences ($F_{2,198} = 12.860$, $p = 0.000$, $\eta^2 = 0.115$). For this factor participants of the study attributed the highest overall value to *PlayGuide*. While *PlayGuide* (M = 3.80, SD = 1.119) differed significantly from TGD (M = 3.50, SD = 1.133, $p = 0.002$) and OGD (M = 3.35, SD = 1.226, $p = 0.000$), TGD and OGD were not exhibiting any significant differences ($p = 0.300$). In terms of tension/annoyance (*Ten*), the rANOVA tests also reported significant effect of the conditions on the measured factor ($F_{2,118} = 10.523$, $p = 0.000$, $\eta^2 = 0.115$). The Bonferroni-corrected pairwise comparisons then reported that *PlayGuide* (M = 1.98, SD = 1.142) again differed significantly from TGD (M = 1.52, SD = 0.676, $p = 0.013$) and OGD (M = 1.37, SD = 0.688, $p = 0.688$). Also, like the last factor, *Flo*, TGD and OGD did not differ significantly from each other ($p = 0.658$). The last factor in *Game B* that showed significant differences using the rANOVA tests was challenge, *Cha* ($F_{2,198} = 42.554$, $p = 0.000$, $\eta^2 = 0.301$). The Bonferroni-corrected pairwise comparisons revealed that all of the three conditions differed significantly from each other. TGD (M = 2.22, SD = 1.186) from *PlayGuide* (M = 2.76, SD = 1.429, $p = 0.000$), *PlayGuide* from OGD (M = 1.68, SD = 0.839, $p = 0.000$) and TGD from OGD ($p = 0.000$). The highest score was received by *PlayGuide*, followed by TGD and then OGD. The two factors negative affect (*Neg*, $F_{2,158} = 0.477$, $p = 0.621$, $\eta^2 = 0.006$) and positive affect (*Pos*, $F_{2,198} = 0.455$, $p = 0.635$, $\eta^2 = 0.005$) were not significantly different as indicated by the rANOVA tests. A detailed overview of the mean values can be seen in Figure 6.4 and Table 6.4). The next subsection will share the results of the collected in-game data.

6.4.2 In-Game Data

The in-game data collected during the play tests provides objective results that can be analyzed to evaluate the behaviour of the players depending on the applied conditions. Unlike the *Game Experience Questionnaire* that was the same for both scenarios, the data itself represented events that were individually adapted to each of the two test scenarios, thus the used measures are unique to each game (for further details see Section 6.3). Furthermore, since the games were meant to contrast each other, there was no point of comparing them side-by-side but rather focus on the difference between the conditions.

Game A

The first measure in *Game A* was the time between the moment a target was marked as point of interest and the moment the player looked at the position of the target (within a given radius) (*Measure A1*). This means, how fast were the individual conditions (either TGD, *PlayGuide* or OGD) to attract the player's attention. Using the rANOVA test, the results indicated a significant effect of the condition on the time needed by the player to attend the object ($F_{2,38} = 54.615$, $p = 0.000$, $\eta^2 = 0.742$). The Bonferroni-corrected pairwise comparisons further indicated that although *PlayGuide* (M = 0.78, SD = 1.429) was significantly different to TGD (M = 0.51, SD = 0.839, $p = 0.000$) and OGD (M = 0.55, SD = 0.839, $p = 0.000$), the difference between TGD and OGD was insignificant ($p = 0.552$). Figure 6.5 depicts the times according to the individual participants. The second measure, *Measure A2*, describes the time that was needed between the moment a target was marked as point of interest and the moment the player

Table 6.5: *Game A*: Means and standard deviation for the measures calculated with the help of the collected in-game events.

Measure	*Traditional*	*PlayGuide*	*Overt*
A1	0.51 (0.061)	0.78 (1.073)	0.55 (1.131)
A2	0.97 (0.063)	1.12 (0.061)	1.01 (0.067)
A3	0.52 (0.069)	0.42 (0.072)	0.52 (0.056)
A4	42.85 (1.872)	39.40 (2.683)	41.40 (2.583)
A5	36.30 (6.490)	28.05 (6.763)	35.25 (4.876)
A6	43.85 (0.933)	42.65 (1.663)	43.05 (1.099)

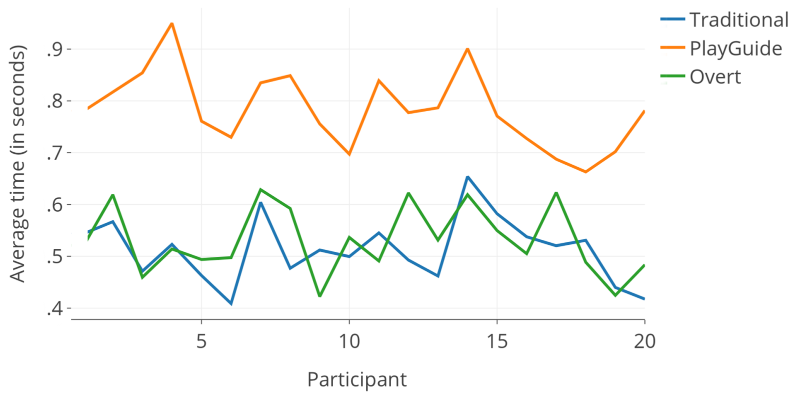


Figure 6.5: This graph depicts the average time between the moment a target was marked as point of interest and the moment the player attended the target.

actually triggered the shot that hit the target. As in the first measure the rANOVA test showed a significant difference ($F_{2,38} = 72.053$, $p = 0.000$, $\eta^2 = 0.791$). Unlike the first measure, the Bonferroni-corrected pairwise comparisons showed that all conditions differed significantly from each other. TGD (M = 0.97, SD = 0.063) to *PlayGuide* (M = 1.12, SD = 0.061, $p = 0.000$), OGD (M = 1.01, SD = 0.067) to TGD ($p = 0.040$) and *PlayGuide* to OGD ($p = 0.000$). However, in both measures *PlayGuide* boasts the highest mean values. For third measure (*Measure A3*), again, the rANOVA test showed a significant result between the time a target was attended and the player triggered a shot ($F_{2,38} = 51.323$, $p = 0.000$, $\eta^2 = 0.730$). Interestingly, although TGD (M = 0.52, SD = 0.069) and OGD (M = 0.52, SD = 0.056) did not differ significantly using the Bonferroni-corrected pairwise comparisons ($p = 1.000$), the difference between *PlayGuide* (M = 0.42, SD = 0.072) and TGD ($p = 0.000$) as well as OGD ($p = 0.000$) was significant, attributing an overall faster reaction time that was needed by the players from attending a target to triggering a shot to *PlayGuide*. The fourth measure (*Measure A4*) was the number of targets that were actually attended (looked at) by the player. The rANOVA

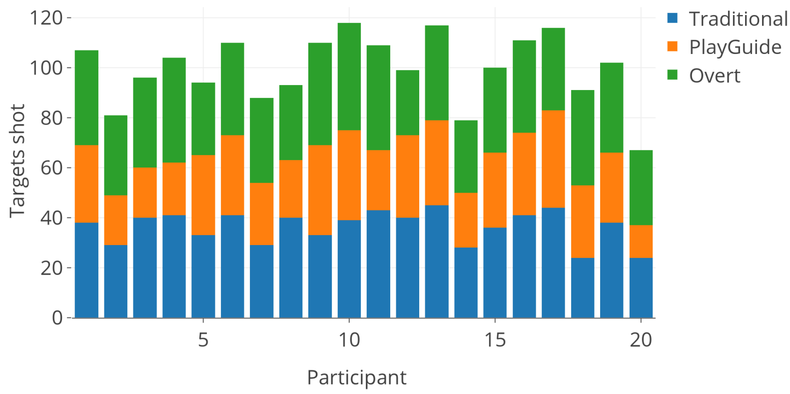


Figure 6.6: Participants overall shot less targets with *PlayGuide*. This graph shows the number of targets shot according to the individual participants and conditions.

test yielded significant results ($F_{2,38} = 9.748$, $p = 0.000$, $\eta^2 = 0.339$). The Bonferroni-corrected pairwise comparisons revealed that only one pair of conditions was significantly different, namely TGD (M = 42.85, SD = 1.872) and *PlayGuide* (M = 39.40, SD = 2.683, $p = 0.001$). The difference between TGD and OGD (M = 41.40, SD = 2.583, $p = 0.225$) and the difference between *PlayGuide* and OGD ($p = 0.080$) was insignificant, meaning that while with *PlayGuide* players shot the least number of targets, only the TGD could significantly improve on that number. A much ampler difference was found in the fifth measure *Measure A5*, which was the number of targets shot. Thus, for this measure the rANOVA also showed a significant difference ($F_{2,38} = 9.748$, $p = 0.000$, $\eta^2 = 0.339$). Although, the difference between TGD (M = 36.30, SD = 6.490) and OGD (M = 35.25, SD = 4.876, $p = 1.000$) was insignificant, *PlayGuide* (M = 28.05, SD = 6.763) did differ significantly from TGD ($p = 0.000$) and OGD ($p = 0.001$). Figure 6.6 depicts the targets shot according to the individual participants. The sixth and last measure for *Game A*, *Measure A6*, was the number of targets that were shown to the player. Although the lifetime of the targets was the same in each condition, the number was influenced by how fast players shot the targets. The numbers proved to be significant measured with the rANOVA test ($F_{2,38} = 4.361$, $p = 0.020$, $\eta^2 = 0.187$), yet they did not vary by much. The Bonferroni-corrected pairwise comparisons showed that *PlayGuide* (M = 42.65, SD = 1.663) did not differ significantly from TGD (M = 43.85, SD = 0.933, $p = 0.053$) or OGD (M = 43.05, SD = 1.099, $p = 1.000$), unlike TGD and OGD ($p = 0.023$). The mean and standard deviation for all measures are sorted and listed in Table 6.5.

Game B

As in the first game scenario, in *Game B* six measures were taken. The first one (*Measure B1*) was the time the participants spent in the game. Using the rANOVA test this measure yielded significant results ($F_{2,38} = 27.344$, $p = 0.000$, $\eta^2 = 0.590$). As in the preceding statistical analyses, the differences between the conditions were assessed with the Bonferroni-corrected pairwise comparisons. For the first measure, these comparisons

Table 6.6: *Game B*: Means and standard deviation for the measures calculated with the help of the collected in-game events.

Measure	*Traditional*	*PlayGuide*	*Overt*
B1	293.70 (101.380)	385.85 (135.44)	199.48 (81.33)
B2	8.35 (1.309)	7.50 (2.328)	10.65 (0.671)
B3	8.20 (4.753)	8.95 (3.364)	6.90 (1.832)
B4	2.05 (1.188)	2.24 (0.841)	1.73 (0.458)
B5	35.65 (13.481)	58.94 (35.500)	18.59 (7.068)
B6	3.07 (1.191)	3.51 (1.222)	2.27 (0.569)

show that all conditions differed significantly from each other. TGD (M = 293.70, SD = 101.380) from *PlayGuide* (M = 385.85, SD = 135.44, $p = 0.008$), OGD (M = 199.48, SD = 81.33) from TGD ($p = 0.001$) and *PlayGuide* from OGD ($p = 0.000$). The second measure, *Measure B2*, was the number of coins found by the players and, like the first measure, also showed significant results using the rANOVA tests ($F_{2,38} = 24.289$, $p = 0.000$, $\eta^2 = 0.561$). In total, 11 coins could be found per condition. The Bonferroni-corrected pairwise comparisons indicated that the differences between TGD (M = 8.35, SD = 1.309) and *PlayGuide* (M = 7.50, SD = 2.328) were insignificant ($p = 0.345$). Nevertheless, OGD (M = 10.65, SD = 0.671) showed significant differences compared to *PlayGuide* ($p = 0.000$) and TGD ($p = 0.000$). Furthermore, none of the participants found all coins with *PlayGuide*, whereas only one participant found all coins with TGD. The third (*Measure B3*) and fourth measure (*Measure B4*) yielded no significant results using the rANOVA test. The former counted the overall number of times players visited the rooms in the game ($F_{2,38} = 1.977$, $p = 0.153$, $\eta^2 = 0.094$), while the latter calculated the average times a room was visited ($F_{2,38} = 1,831$, $p = 0.13$, $\eta^2 = 0.034$). Nevertheless, the fifth measure, *Measure B5*, which described the average time to find a coin, did show significant results ($F_{2,38} = 18.681$, $p = 0.000$, $\eta^2 = 0.496$) and the Bonferroni-corrected pairwise comparisons confirmed that all three conditions were significantly different. TGD (M = 35.65, SD = 13.481) from *PlayGuide* (M = 58.94, SD = 35.500, $p = 0.026$), OGD (M = 18.59, SD = 7.068) from *PlayGuide* ($p = 0.000$) and TGD from OGD ($p = 0.000$). Lastly, the average number of times a coin was attended before it was found represented the sixth measure (*Measure B6*) and also proved to be significant when validated with the rANOVA test ($F_{2,38} = 8.522$, $p = 0.001$, $\eta^2 = 0.310$). This time, the Bonferroni-corrected pairwise comparisons indicated significant results between the conditions OGD (M = 2.27, SD = 0.569) and TGD (M = 3.07, SD = 1.191, $p = 0.021$) as well as OGD and *PlayGuide* (M = 3.51, SD = 1.222, $p = 0.001$). Insignificant results were indicated between TGD and *PlayGuide* ($p = 0.709$). As in the previous results, the mean and standard deviation for all measures are sorted and listed in Table 6.6. The results are further discussed in the following chapter to share the insights gained throughout the testing phase of *PlayGuide*.

Chapter 7

Discussion

The evaluations presented in the preceding chapter show promising results for *PlayGuide*, with the exception of a few caveats. To anticipate the outcome of the user tests: *PlayGuide* did perform very differently in both game scenarios, which is reflected in the data collected through the *Game Experience Questionnaire*, the collected in-game data as well as the qualitative interviews. Besides the detailed analysis of *PlayGuide*, this chapter will also contain the challenges and limitations that were encountered during the development and testing of the player guidance system. Additionally, feasible improvements and applications are shared, the latter of which reveal the true potential of *PlayGuide*. Although the application of *PlayGuide* for this thesis was to test the rudimentary functionality, there is a lot more *PlayGuide* can be used for, functioning as a practical tool for game developers and designers alike.

7.1 Analysis of PlayGuide

In the previous chapter, the outcomes of the user tests have been listed in detail. However, the implications of the gathered data has not yet been conferred. The following subsections are divided into the test games *Game A* and *Game B*, in which the three individual conditions will be compared and analyzed.

7.1.1 Game A

For *Game A* the experiments showed that *PlayGuide* did not perform as well as the other conditions, receiving overall worse scores in the *Game Experience Questionnaire* than TGD (TGD) or *Overt Gaze Direction* (OGD), which can be attributed to the instructions of the game. At the beginning, players were ordered to shoot as many targets as possible. As can be learned through the qualitative interviews (see Appendix A), participants of the study preferred a clear visualization of where the targets were going to appear and attributed low importance as to how their gaze was guided towards the target. In other words, whatever means helped the players to achieve their target was preferred the most. *PlayGuide*, however, only applies subtle visual cues that take more time to be noticed and directs the gaze only roughly into the direction of the target, which is not as helpful as the clearly visible – thus effective – visual cues applied by the other two conditions. This also reflects in the Bonferroni-corrected pairwise comparisons between TGD and OGD, which showed no significant differences in any of the factors that were tested by the *Game Experience Questionnaire*, meaning that they performed equally well. Similarly, players felt the most annoyed when using *PlayGuide*. Of course the factor tension/annoyance is tightly coupled with how well the players performed in the game, clarified by one participant who echoed the general sentiment of the players: "I didn't even notice it and honestly, it was frustrating. I barely shot anything." (see Appendix A). Nevertheless, the scores were generally low in this category, since some people stated that they actually liked the challenge posed by *PlayGuide*. Especially participants that rated themselves as "expert gamers". The fact that players performed worst (shooting the least amount of targets) with *PlayGuide* also reflected both in the factor challenge, which scored the highest of all three conditions and in the factor positive affect, where *PlayGuide* received the lowest score out of all three conditions.

The subjective impression of the players self-assessment regarding their low performance using *PlayGuide* is clearly vindicated and becomes apparent when looking at the in-game data. While the number of targets shot was insignificantly different between TGD and OGD and almost identical with respective means of roughly 36 and 35 targets shot per game, *PlayGuide* decreased this success rate by a lot. About 20% less targets were shot with the player guidance system applied. A matter that worsened the results of *PlayGuide* in the *Game Experience Questionnaire* is that despite the fact that players were not able to shoot as many targets, they were still seen and did not go unnoticed by the player. With *PlayGuide* out of an average of 39 targets that were noticed only were 28 shot, which is a success rate of 72%, compared to TGD where out of an average of 43 targets that were attended, 36 were shot, resulting in a far higher success rate of 83%.

Investigating *PlayGuide* in terms of efficiency, there is one interesting key factor that needs to be examined. The average time between the moment a player looked at a target and the moment the target was shot is significantly less compared to TGD and OGD. Logic dictates that when a player identifies the position of a target, the time needed to aim and shoot at that target must be the same throughout all three conditions. However, the in-game data corroborates that with the use of *PlayGuide* players were faster at the aforementioned task. Unfortunately, this advantage in speed is nullified by the fact that players needed considerably more time to actually notice the visual cue applied by *PlayGuide* compared to the other conditions, which were faster and performed equally as well in attracting the players attention.

Regarding the aesthetics of the visual cues applied by the different conditions, participants stated that the TGD was the most pleasing and fit best for the game genre. Due to the subtle nature of the visual cues applied by *PlayGuide*, people did not comment negatively nor positively on the visual cues of the player guidance system. However, OGD was categorized by some interviewees as "too extreme".

Summarizing, it can be said that although *PlayGuide* overall performed worse than the other two conditions, it does have its advantages. Firstly, although data suggests that people were annoyed, performed worse at the game's target and felt less competent, players with a higher skill level in gaming seemed to prefer the challenge that came with the player guidance system. Secondly, although the average number of shot targets was lower compared to the alternative conditions, *PlayGuide* still managed to guide the players' gaze towards the target's location in a comparable time frame and did not drastically differ from TGD nor OGD. Thirdly, *PlayGuide* increased the aim efficiency, with the lowest average time needed for the players to shoot and aim at the targets. Still, the disadvantages clearly outweigh the advantages and there is an important factor that needs to be considered: The overall target plays a huge role in how *PlayGuide* is perceived. Players followed the instructions of the game closely and usually wanted to perform well. This meant that any means to achieve the player's goal as fast as possible was appreciated, which is also why there is no significant difference between sensory and imaginative immersion or flow throughout any of the three conditions despite the fact that participants labelled OGD as too stark, extreme or intense. In the case of *Game A* the goal was to shoot as many targets as possible and this was easiest with the TGD or OGD condition. Thus, *PlayGuide*, compared to the other two conditions, was subjectively categorized as the worst option to most players, although it also guided the player's gaze albeit not as effective nor efficient. All in all, this makes TGD and OGD better candidates for action-based games, where time and performance are key components for the player's success. Nevertheless, the successful application of *PlayGuide* in an entirely different game genre is presented in the following subsection.

7.1.2 Game B

Contrary to *Game A*, which was a third-person action-heavy shooter, *Game B* was a first-person game where players could freely explore the environment and had to find coins that were hidden throughout the level. Also, *Game B* did not have any time constraints and the players were asked to take as much time as they needed to fulfill the task. This liberation from the time component and the introduction of the explorative gameplay yielded vastly different results than the game scenario discussed in the previous subsection. Much like in the former game scenario, *PlayGuide* scored the lowest value for the factor competence, assessed through the *Game Experience Questionnaire*. Although players were unaware about the number of coins hidden, nor how many they had already found – unless they counted themselves – players assessed their competence according to the following considerations. During the qualitative interviews (see Appendix A) players reported that, subjectively, OGD posed the best condition to find all hidden objects, as they simply needed to enter a room and pay attention to where a clearly visible visual cue appeared. It was also mentioned multiple times that TGD, while being hard to understand for some, was immediately understood by most players and the green glowing arrows served as a clearly visible indicator that quickly led to the hidden objects. These circumstances made players more confident in the fact that they indeed had found all the hidden objects as soon as no visual cue was visible anymore. Of course, this was not as easy with the subtle visual cues applied by *PlayGuide*, which was reflected in the number of hidden coins that were found. Out of 11 coins that were hidden throughout the level, none of the players found all 11 coins with the help of *PlayGuide*. Unsurprisingly, this inherently meant that *PlayGuide* received the highest values in the factor challenge, followed by TGD, where the cues were less subtle, and OGD, where the cues were almost instantly visible and could hardly be overlooked. In terms of sensory and imaginative immersion, *PlayGuide* as well as TGD scored the highest values. As reported in the qualitative interviews, the visual cues applied by OGD simply were too intense for an enjoyable gaming experience which resulted in

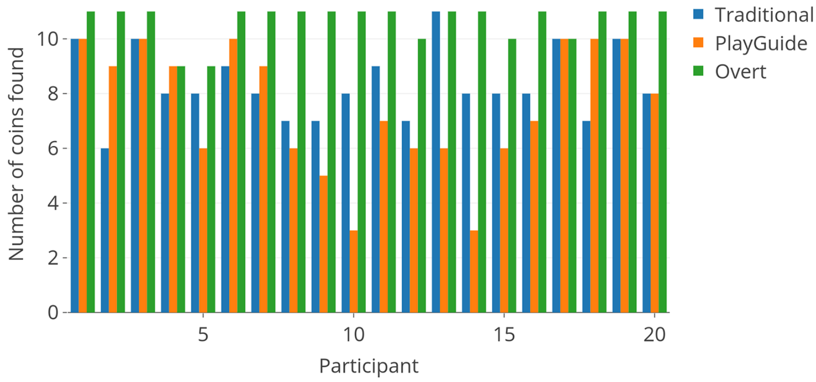


Figure 7.1: This graph shows the number of hidden objects (in the form of coins) that were found according to each condition and participant.

the overall lowest score for immersion. Interestingly, most players were not bothered by the intensity of the visual cues in *Game A* although it inherited the same values in *Game B*. Also in terms of flow, *PlayGuide* received significantly higher scores than the other two conditions, which was expressed by an interviewee as "search effect" (see Appendix A). In other words, *PlayGuide* was the only condition that did not explicitly show the position of the hidden objects but merely led the player's gaze into the right direction. Thus, the players actually were occupied with searching for the coins instead of scanning their surroundings for visual cues, ultimately leading to a stronger feeling of flow and immersion. Surprisingly, despite these promising results, *PlayGuide* also received the highest score for the factor tension/annoyance. Something that might have been the reason to irritate some players and also made them feel less competent, is that with *PlayGuide* and TGD players sometimes overlooked coins and then found them at a different point in time during the game, which is reflected by the fact that with the aforementioned conditions on average players looked at a coin three times before they found them.

The most drastic difference between observed in-game values was the time players spent in the game. At the beginning of *Game B* players were reminded that they could take as much time as they needed to accomplish the task, which lead to the fact that with *PlayGuide* on average players spent almost twice as much time in the game than with the lowest scoring condition, OGD, all while scoring *PlayGuide* with the highest values in sensory and imaginative immersion and flow. Furthermore, none of the players negatively mentioned or complained about the prolonged time spent in the game. Thus, *PlayGuide* effectively extended the game time by a factor of two without losing the players' interest. Also, the resulting average time to find a coin was clearly the lowest with OGD, while TGD doubled that value and *PlayGuide* even tripled it.

As in terms of how many coins were found throughout the game, *PlayGuide* did not score well. As already mentioned, none of the players found all 11 hidden coins with the help of the player guidance system. Nevertheless, the subtle image cues applied by *PlayGuide* were as effective as the more apparent TGD cues, albeit the latter received lower scores on sensory and imaginative immersion and flow. The most effective solution was OGD with an average of 10.65 coins found per player and 15 out of 20 test players that found all coins (see Figure 7.1). The aggressive nature of the visual cues of OGD was extremely effective in guiding the player's gaze and led to this success rate. This also becomes apparent when looking at average number of times a coin was attended before it was found. While players on average looked at a coin 3.5 times with the *PlayGuide* condition applied and 3 times with the TGD condition applied before they found it, this number drops down to 2 with the OGD condition applied.

To sum up, unlike in *Game A*, *PlayGuide* is an excellent candidate for exploration games. Restating the hypothesis deducted from *Game A*, the overall target of the game drastically changes the way *PlayGuide* is received by the players. In *Game B* exploration was the main focus of the game. It did not matter whether players found all coins or not and, moreover, there was no indicator on how many coins had already been found, leaving the player in the unknown about the actual number of hidden objects already found. Players also gave the highest scores to *PlayGuide* in terms of sensory and imaginative immersion and flow, albeit necessitating double the amount of time that was needed in the fastest condition and tripling the average time to find a coin, indicating that players actually had to spend more time to explore the environment without loosing the feeling of immersion or flow. Moreover, interviewees stated that with *PlayGuide*, it felt more rewarding to find a coin "almost" on your own, as the player guidance system only indicated the position of the hidden objects. For genres akin to explorative games (e.g., role playing games), *PlayGuide* poses a compelling alternative to TGD and OGD techniques. The system succeeded when neither time nor completeness was of importance but rather the exploration of the game environment. The following section will share further insights on challenges and limitations that were encountered during the development and testing of *PlayGuide*.

7.2 Challenges and Limitations

Albeit the fact that *PlayGuide* showed promising results during the testing phase, the conducted experiments also exposed the shortcomings and limitations of the system and that *PlayGuide* is not without flaws. For future research these limitations pose interesting challenges that improve upon the current version of the player guidance system.

First and foremost it needs to be mentioned that although consumer versions of eye trackers have gotten cheaper over the last recent years, the adoption rate of the eye-tracking hardware has increased [46] and *Tobii*, the company that is the leading manufacturer of consumer eye-tracking devices [38], is pushing to broaden the market, the technology is still not widely adopted. However, *PlayGuide* relies on the gaze to analyze the behaviour of the player and this automatically excludes the use of *PlayGuide* for all gamers who are not yet equipped with such a device. Another caveat of the eye-tracking technology is that most inexpensive consumer versions of eye-tracking devices deliver mostly reliable information about the player's gaze, albeit the data is not always accurate. This is especially true when the gaze of the player shifts towards the edges of the screen and during the user tests this proved to be a weak spot of the eye tracker used. Furthermore, the hardware used to evaluate *PlayGuide* heavily depends on the fact that players remain in their current position. Vast movements of the torso or the head are reflected in displaced gaze positions of the player's gaze on the screen. Unfortunately it turned out that during game play people do move considerably, changing their posture to a more comfortable one after a while. While *PlayGuide* is capable of handling occasional outlines, such displacements of the gaze position result in a dysfunction of *PlayGuide*, which is dependent on the correct gaze samples to calculate the appropriate moment in which to hide the modulations so the player will not see them.

Unfortunately, unlike the newer model *Tobii 4C*[1], the used eye-tracking device (*Tobii EyeX*[2]) only supplies the gaze data of the viewer and does not provide any additional meta information, like the distance of the player's eyes to the screen, which is needed for the calculations done by *PlayGuide* to categorize the eye movements of the player. This meant that the information had to be manually set before the start of each game. However, as mentioned in the preceding paragraph, people rarely sit still in front of the computer over a longer period of time and although *PlayGuide* can deal with some inaccuracy, it does influence the precision of the calculations and occasionally eye movements will not be categorized correctly.

Apart from that *PlayGuide* itself works independently and does not need any other input from the player than setting the minimum and maximum intensity of the modulations with which it guides the player's gaze. This is due to the fact that each individual's peripheral vision has a different sensitivity towards motion and luminance and this needs to be addressed. Although an average value could be used to satisfy most of the players, this would mean that a small minority of the people who rely on *PlayGuide* will either not notice the modulations at all or will have a disrupted gaming experience, clearly noticing the modulations.

Considering all of what was mentioned so far, some test users did notice the subtle image space modulations applied by *PlayGuide*, which contradicts the aim of the system to unobtrusively guide the players towards important objects in the game scene. The fact that the eye tracker does deliver inaccurate data at times and that players moved during the testing phase of the game prototypes caused that sometimes *PlayGuide* did not terminate the modulations quickly enough to be hidden from the player's foveal vision. In such cases, the subtle image space modulations were seen and could be identified by the players. Other times test users outright ignored the clues given by *PlayGuide*. Most of the time, these players were in a hurry and moved the camera hastily, rushing through the game without stopping to to see whether there was something interesting or important that needed to be considered. This implies that *PlayGuide* operates best in situations where the player is looking for something or moves the camera in a more slow and deliberate manner. Otherwise the subtle image cues will be hidden by the motion of the camera itself and not detected by the player. Also, viewers cannot be guided towards objects that are not in the camera's field-of-view. The guidance system merely directs the gaze towards points of interest the camera is already facing. This means that important parts of the game that are not deemed as interesting in the first place and the player does not move the camera to, cannot be considered by *PlayGuide*. Hence, by no means the player guidance system attempts to replace traditional visual cues. To the contrary, traditional visual cues will always play an important role in game

[1] https://help.tobii.com/hc/en-us/articles/213414285-Specifications-for-the-Tobii-Eye-Tracker-4C
[2] https://help.tobii.com/hc/en-us/articles/212818309-Specifications-for-EyeX

development and must be used adequately to provide an enjoyable gaming experience.

Another shortcoming of the visual gaze-based player guidance system is that it was developed in *Unity*[3] and thus can only be used in *Unity* since it relies on essential classes and application programming interfaces that are provided by the popular game development engine. Although this does not pose a major problem, it does mean that *PlayGuide* can only be implemented in games that are made with *Unity* and cannot be used outside the platform at this time. Furthermore, another downside that came with the choice to use *Unity* as a development platform was that the *gaze-to-object mapping* (GTOM) algorithm used to evaluate *PlayGuide* is a relatively simple one. GTOM techniques are responsible for mapping the player's gaze to objects in a scene, more on which can be read in Section 3.2. Although there do exist complex and very accurate solutions, as proposed in [3, 43], most of them are not performant nor suitable for games where performance is a key factor. The method first used for the evaluation of *PlayGuide* required the implementation of a compute shader due to its complexity and delivered accurate results. A compute shader enables simple calculations to be run much faster on the graphics processing unit (GPU) of the computer. However, *Unity* does not provide a way to get the data back to the core processing unit (CPU) asynchronously, which stalls any other calculations on the CPU until the data sent from the GPU is received by the CPU, vastly influencing the performance of the game. This eventually led to the compromise of using a more performant but less exact technique called *Ray Casting Shotgun* already used in games and detailed in Section 5.5.1.

During the time of the development and the testing of *PlayGuide* a lot has been learned. The following section will share possible ways to improve on the current limitations of *PlayGuide*.

7.3 Improvements

In the last section the challenges and limitations of *PlayGuide* were mentioned, which became apparent during the implementation and evaluation of the player guidance system. This chapter will focus on possible solutions to overcome the shortcomings currently exhibited by the system.

Firstly, some factors were caused by the hardware that was used to test *PlayGuide*. The *Tobii EyeX* eye tracker is an eye-tracker used by consumers and thus not as accurate as eye trackers used for research. However, the use of *PlayGuide* with a high-end eye tracking system would not be adequate, since the player guidance system is supposed to be used in a typical use case environment and with hardware that can be found in the homes of the end consumer. This, in turn, means that tracking errors caused by the movement of the player or misinterpretations towards the edge of the screen need to be addressed by *PlayGuide* itself. A possible solution would be a more aggressive termination of the subtle image space modulations that guide the gaze of the player to prevent errors. Furthermore, to automatically obtain and update the distance of the player's eyes to the screen, there is the possibility to upgrade to a more expensive consumer version eye-tracking device like the *Tobii 4C*[4], which provides additional meta informa-

[3] https://unity3d.com/

[4] https://help.tobii.com/hc/en-us/articles/213414285-Specifications-for-the-Tobii-Eye-Tracker-4C

tion that has to be set manually in the current version of *PlayGuide*. Furthermore, as mentioned in the previous section, the intensity of the subtle image space modulation applied by *PlayGuide* needs to be manually set to fit the needs of the individual player. While this is currently done in a very unexciting calibration test scene by changing the intensity of the modulations until they are noticed and require the active participation of the player, this calibration process could very well be integrated into the game itself without the player actually realizing. As the information about the player's gaze is inherently available, the gaze patterns could be continuously checked to assess whether the player is responding to the visual cues or not. The intensity could then be adjusted accordingly by *PlayGuide*.

A possible solution to the problem that *PlayGuide* cannot guide players towards objects that are not in the camera's field-of-view was proposed by Eli Ben-Joseph and Eric Greenstein [59]. In a virtual reality setting they used the same basic gaze direction technique as *PlayGuide* and sound to turn the head of players towards the right direction. However, instead of only modulating a small part of the image, they displayed illumination modulations that approximately covered a quarter of the right or left side of the screen, depending on where they wanted the player to look at [59]. The experiment yielded promising results and might pose a solution to relocate the player's camera to face the point of interest.

Nevertheless, the biggest step towards improving *PlayGuide* would be to make the system independent of any game development platform (e.g., *Unity*) and make it accessible for game developers working with their own proprietary game development platforms. In addition, providing application programming interfaces would allow for the use of *PlayGuide* across many platforms and programming languages. This major evolution of the player guidance system would facilitate many potentially very exciting applications, some of which are mentioned in the following section.

7.4 Potential Applications

Using *PlayGuide*, not only did the results show that the approach worked in certain situations but also that it was more effective in maintaining a feeling of immersion and a state of flow during game play. Although the player guidance system was applied to a very basic use case for evaluation, there are multiple potential applications for using *PlayGuide* in games besides leading the player towards important game objects within the scene. An excerpt of the manifold areas of application is given in this section.

One potentially beneficial area of application is *gamification*. Although many consider this term to be overused and a hype that is exaggerated compared to its benefits [64], the process of applying game-design elements and game principles to non-game contexts is hugely popular and represents a worldwide business market, whose value is reported to grow from 4.91 billion dollars to in 2016 to almost 12 billion in the year 2021 [69]. In the field of educational or serious games, *PlayGuide* could train and correct students to look into the right direction or at the right objects. Additionally, new gaze patterns could be trained to form a habit of adhering to specific gaze patterns.

Another potential area, in which *PlayGuide* could be of use is the training and aid of novice computer gamers. Expert gamers have the advantage of their experience that has been well-trained over the course of many hours, days, months or even years in a specific game, which can leave novice players seemingly overwhelmed when playing new games [6]. To counteract this disadvantage, *PlayGuide* could aid inexperienced players by directing their gaze towards the right target. This might be the enemy controlled by a pro-gamer, user interface actions that need to be triggered or simply the right direction to face. Games could individually adapt the gaze direction technique by adjusting the intensity of *PlayGuide* or refraining from showing the visual aid at all, resulting in matches that feel more natural and balanced and a difficulty level that assimilates to the individual skill level of the player.

Contrary to what *PlayGuide* has been used for in this thesis, it can also serve as a distraction technique instead of a guidance technique, to increase the difficulty of a game. Also, if so desired, this technique could be used to confuse and bewilder the player from reaching his or her actual target. A prime example of such a scenario is *The Stanley Parable*, where the confusion of the player is not only a part of the game but also the core game mechanic [62]. Furthermore, this would make the player guidance system a perfect candidate for horror games. To exacerbate the scare of the player, the eye could be guided away from wherever it is currently focused on, to distract the player while the startling moment is imminent.

Another area where *PlayGuide* seems to be an interesting candidate is a technique called *perceptually adaptive rendering* proposed by Bailey et al. [1]. To provide a longer time frame for computing complex objects and models in a scene, the player can be guided away from parts of the scene that are more computationally intensive than others. Such areas can then be progressively updated while the player is occupied looking at different parts of the image. Of course, as mentioned by Bailey et al., this also requires the knowledge of how long a scene takes to render and how long a player can be distracted by the modulations in the image. A field of research where a similar approach is already used is virtual reality games. Modern virtual reality goggles allow for the tracking of the wearer's gaze. This information can then be used to decrease the rendering quality with increasing distance to the viewer's center of the gaze. As the capability of sensing image details decreases rapidly towards the perimeter of the fovea and even more substantially in the peripheral vision, the decrease of render quality will most likely not be noticed by the player, while effectively saving render time and performance costs [32].

Kids might also profit from the use of *PlayGuide* in a game. While the controls are usually understood very quickly, grasping the granularity of the game mechanics is significantly harder to achieve. Much like the novice player, who could be guided during the first few hours of a game, kids could also be trained to look for the right objects in a game. Depending on the skill level, the intensity and tenacity in which *PlayGuide* is showing the visual cues could be modified. This approach might be very successful, since the user tests for *PlayGuide* showed that people did not need any instructions to understand how *PlayGuide* works or what it does. Due to its inherent logic, players immediately understood that where the attention is guided, there must be something that is interesting or important to the game. This kind of simplicity might also be grasped by young kids that are completely new to playing games which, of course, would need a thorough investigation and was not examined in the scope of this master's thesis.

Lastly, it needs to be mentioned that the recently very popular, emerging and ever improving virtual reality (VR) technology poses a huge potential application for *PlayGuide*. By this time, some advanced VR headsets are capable of tracking the wearer's gaze positions and the player would not have to buy additional equipment but could profit from a technology that is innately available with VR [73]. Thus, this would enable *PlayGuide* to be used within a VR context to help players who are overwhelmed with the surrounding flood of information (and especially players that experience VR for the first time) to orient themselves in a 360° environment and to focus on important image areas. Of course, all of the aforementioned potential applications for *PlayGuide* could be a use case for VR as well. Furthermore, while it would pose hardly any effort to transfer the visual gaze-based player guidance system to a VR context, the advantages for players might be immense – especially in the case of VR, where immersion is the key factor to an enjoyable gaming experience.

Summarizing, this chapter discussed the results of evaluation of the user tests and it was concluded that the successful application of *PlayGuide* compared to TGD and OGD techniques heavily depends on the type of game genre and overall target of the game. Furthermore, improvements of *PlayGuide* are possible in terms of performance, calibration and the independence of the player guidance system from game development platforms. Also, potential applications of *PlayGuide* were detailed, including *gamification*, the aid of novice computer gamers, the use of *PlayGuide* as a distraction technique and as a game training aid for young kids. Lastly, the following chapter will conclude this master's thesis.

Chapter 8

Conclusions

In this master's thesis a novel technique of directing a player's gaze to specific points of interest has been presented and evaluated. With the help of the game development platform *Unity*[1] and the eye-tracking device *Tobii EyeX*[2], the player's gaze is guided towards game objects that might be of interest to the player by using subtle image cues that are almost invisible to the viewer's eye. The visual gaze-based player guidance system for three-dimensional computer games, named *PlayGuide*, is based on a *subtle gaze direction* technique for static images proposed by Bailey et al. [1] and adopts their method of unobtrusively directing the viewer's gaze to be used in a highly dynamic context, namely games. Compared to other *gaze direction* techniques, such as *traditional* and *overt gaze direction* techniques, which are already used in games, *subtle gaze direction* uses subtle image modulations in the peripheral vision to attract the attention of the viewer. This method of guiding the viewers' gaze without them noticing poses a huge possibility to help gamers during visual search without robbing them of the sensation that comes with accomplishing the tasks on their own, thus benefiting the feeling of immersion and flow. The development of two game prototypes was essential for evaluating *PlayGuide* and to assess whether the player guidance system can be used to replace, and maybe even outperform, *traditional* and *overt gaze direction* techniques in terms of factors such as immersion and flow, assessed with the help of the *Game Experience Questionnaire* by IJsselsteijn et al. [17]. Furthermore, in-game events were captured to collect objective data about the performance of *PlayGuide* and qualitative interviews were held to assess the players' opinions about the player guidance system. *PlayGuide* was intended to pose an all-in-one solution for game developers that want to provide a non-intrusive solution to guide the player's gaze and to provide help in situations were the player might be stuck.

Twenty participants partook in the studies and tested *PlayGuide* in the context of two different game genres. Each of the two test game scenarios implemented *PlayGuide*, a *traditional* and *overt gaze direction* technique. The evaluation of the *Game Experience Questionnaire*, the in-game events and the qualitative interviews showed promising results. While in the first game scenario, a fast paced third-person shooter set in space, *PlayGuide* was received well by the participants, they actually preferred the alternative

[1] https://unity3d.com/

[2] https://tobiigaming.com/product/tobii-eyex/

gaze direction techniques. The *Game Experience Questionnaire* further revealed that players felt the least competent and the most challenged with the help of *PlayGuide*. However, in the other game scenario, a first-person exploration game set in medieval times, *PlayGuide* was preferred over the two additionally tested gaze direction techniques and scored the highest values in terms of immersion and flow, while offering the biggest challenge and nearly doubling the time players spent exploring the environment. In conclusion, *PlayGuide* was successful in guiding the player's gaze and poses a viable option to *traditional* and *overt gaze direction* techniques. Nevertheless, factors such as immersion and flow are highly dependent on the game genre and the game's overall target. Thus, players preferred a *gaze direction* technique that helped them fulfill their task as thorough as possible. When time and performance were a priority, players preferred the *gaze direction* techniques that clearly marked the target's position and that helped them to achieve a higher score. However, when the overall aim was to explore the environment without any time limit, players chose *PlayGuide* as their preferred *gaze direction* technique as it guided their gaze only towards important game objects rather than directly onto them, creating the subjective sensation that players were able to explore the environment on their own.

Nevertheless, due to the fact that the adoption of eye-trackers is progressing slowly, there are not many actual applications of this technology to games and blockbuster titles, such as *Dying Light* [63], are only slowly incorporating gaze as an opportunity to engage players. However, eye trackers pose a huge potential to enrich the gaming experience and there are many areas of application apart from using it as a means to control the game. This research proposed a novel technique that analyzes and uses the gaze of the player to provide a more enjoyable and overall better gaming experience in terms of flow, immersion and challenge. Up until now games have relied on obvious *pull cues* that attract the attention of the player. However, the potential to guide the player's gaze by seemingly invisible image cues can hugely benefit the gameplay for both novice and experienced players. The development of *PlayGuide* introduced a new tool that can be used by game developers and, moreover, is only the first advance in an otherwise emergent field of research.

For future research, not only the limitations of *PlayGuide* offer compelling opportunities to improve on the player guidance system, but also the application to more complex situations than the ones presented in this thesis. For example, *PlayGuide* is now only capable of directing the player's gaze towards game objects that are already within the player's field-of-view. An important potential area of research could be to guide the players into the right direction, thus not towards an object but a general direction that needs to be faced. Furthermore, *PlayGuide* has proven to be effective in the genre of exploration games but not in the context of a shooter game, when compared to other *gaze direction* techniques. Thus, a very promising area of research could be the application of *PlayGuide* to assist only novice players during their first encounter with a new game, hence closing the skill gap between novice and expert players. As the overwhelming amount of information that is displayed in a game can be irritating, almost intimidating for beginning players who are confronted with a complex user interface or task for the first time, naturally more experienced players will be superior to beginners. By providing *subtle gaze direction*, novice players could benefit from *PlayGuide*, which is only roughly guiding the player's gaze towards the right direction, without giving them too much of an advantage over expert players. Of course, the modus operandi of *PlayGuide* itself could be refined as well by improving on the *subtle gaze direction* method proposed by Bailey et al. [1]. Although their method mostly functions without any problems, it seems to work best with eye-tracking devices that are very accurate but unfortunately mostly only found in research labs. Eye trackers used by end consumers are far less accurate and more error prone due to the fact that players might not calibrate their devices properly or move their head considerably during game play. Thus, *PlayGuide* would benefit greatly from an algorithm that is more resistant to outliers and inaccurate gaze samples.

Nevertheless, despite its flaws, the visual gaze-based player guidance system for three-dimensional computer games, *PlayGuide*, evolved from a simple idea to guide the player's attention into a universal tool that directs the player's gaze and can even replace traditional *gaze direction* techniques used in computer games. Additionally, research showed that factors such as immersion and flow hugely benefited from the subtlety of its cues, while serving as a helpful tool to assist players during game play.

Appendix A

Qualitative Interviews

A.1 General Remarks

This appendix presents a transcript of the qualitative interviews that were held at the end of each of the conducted user tests. Thirteen out of twenty participants agreed to share their opinion on *PlayGuide*, the visual gaze-based player guidance system for three-dimensional computer player, presented in this paper. For better readability, from this point on the interviewer will be denoted with the abbreviation "I" and the interviewed test user as "T", for reasons of anonymity. Both the questions as well as the answers are in German, the language the interviews were held in.

A.2 Interview Transcripts

Test User 1

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Generell war das starke Blinken zu viel, es hat beim Spielen gestört. Das schwache Blinken war genau richtig.

I: Und im Bezug auf die einzelnen Spiele, welche Führungsarten haben dir da am besten gefallen?

T: Also in dem Spiel mit dem Raumschiff habe ich keinen Unterschied gemerkt, ich war, glaube ich, drei mal gleich gut beziehungsweise schlecht.

I: Und in dem Spiel mit den Münzen?

T: Da hat mich bei den grünen Pfeilen die Distanz am meisten gestört, dass man überall hin musste, damit der Pfeil auftaucht. So im Gesamten hat mir das Spiel besser gefallen als das erste Spiel im Weltall, die Ästhetik war viel besser. Also, die Grafik.

I: Okay, guter Input! Das wars erst einmal. Danke für deine Zeit und danke für's Testen!

Test User 2

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Bei dem ersten Spiel, das Münzenspiel, wo man die Münzen sammeln muss, da war der grüne Pfeil sehr cool. Das kennt man auch aus anderen Spielen. Das hat mir gefallen. Das Blinken, also das ganz leichte Blinken, das war eine Herausforderung. Das hat mir aber am besten gefallen. Man muss einfach mehr acht geben! Das kann aber irgendwie auch ein Nachteil sein, weil man sich mehr konzentrieren muss. Beim Pfeil zum Beispiel oder beim krassen Blinken, da geht das halt einfach. Da glaube ich hab ich auch alles gefunden.

I: Und bei dem anderen Spiel?

T: Also beim zweiten Spiel, dem mit dem Schießen, da war das leichte Blinken am schwierigsten, weil man es kaum gesehen hat. Das starke Blinken war leicht, da habe ich sofort gewusst wo ich hinsehen musste.

I: Super, ich denke das reicht erstmal! Danke noch einmal für's testen!

Test User 3

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Mmmh. . .

I: Beim ersten Spiel zum Beispiel, dem mit dem Raumschiff im Weltall.

T: Ah, okay. Meinst du was mir generell gefallen hat?

I: Naja, es hat drei Arten gegeben, wie dein Blick geführt wurde. Das Visier, wo sich die weißen Balken zusammenziehen, das leichte Blinken und so ein starkes Blinken.

T: Ja.

I: Welche haben dir da gefallen? In den einzelnen Spielen.

T: Ah! Okay! Jetzt verstehe ich. Beim Spiel mit dem Raumschiff hat mir der leichte Hinweis gar nicht gefallen. Das ist mir auch nicht wirklich aufgefallen und war, ganz ehrlich, frustrierend. Da habe ich ja fast nichts getroffen.

I: Und beim starken Blinken?

T: Das starke Blinken hat nicht gepasst. Das war für das Spiel nicht passend. Da war das Visier schon viel besser, das hat zum Stil gepasst.

I: Und beim anderen Spiel, dem mit den Münzen?

T: Da war das leichte Blinken eindeutig am besten. Da hab ich suchen können, so ein Sucheffekt war das. Das war bei dem Pfeil und dem starken Blinken nicht so, das war als Hinweis zu stark. Da wusste man gleich, wo die Münzen waren. Irgendwie langweilig.

I: Okay, ja. Das haben die anderen auch so ähnlich gesehen wie du! Danke auf jeden Fall für deine Meinung und deine Zeit!

Test User 4

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Generell zu den Spielen, mir hat das Spiel wo man suchen musste gefallen. Da waren die Assets sehr schön, das macht auch das Spielerlebnis irgendwie interessanter. Die Assets haben fast wie die in World of Warcraft ausgesehen, das mag ich. Auch das Setting und so.

I: Und die Blickführungsarten?

T: Ja, da war der grüne Hinweis, der Pfeil, am ästhetischsten, das hat schon gut ins Spiel gepasst. Dafür hat der leichte Hinweis das Spielerlebnis nicht beeinträchtigt, finde ich. Man fühlt sich irgendwie mehr skilful wenn man die Münzen selbst findet und nicht direkt darauf hingewiesen wird. Der starke Hinweis reißt aus dem Spielerlebnis, aber gibt gute Aufklärung wie weit man im Game schon ist. Also, ob man fertig ist oder nicht. Das ist auch irgendwie gut.

I: Wie war das beim anderen Spiel?

T: Das im All?

I: Ja genau.

T: Hm. Was war da noch mal? Ah ja. Das erste Dings, das Visier. Das war am ästhetischsten, das hat am besten für das Spielszenario gepasst, das kennt man irgendwie auch aus anderen Games. Das starke Blinken, war aber am besten um den Highscore nach oben zu treiben, da hab ich einfach alles getroffen. Auch nicht so schlecht. Es ist einfach die Motivation anders als beim Suchspiel. Es geht um Geschwindigkeit, da, finde ich, dürfen die Hinweise schon offensichtlich sein. Ich will ja den Highscore haben.

I: Sehr gut. Das stimmt! Ich glaube, das war's dann für's Erste. Danke!

Test User 5

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Der Audio Cue war sehr hilfreich. Da wusste man, dass ein Planet bald auftaucht. Das war super.

I: Der Audio Cue?

T: Ja, da war so ein "Klack", bevor der Planet aufgetaucht ist. Das war bei den anderen auch? Ich weiß nicht, auf jeden Fall ist es mir da aufgefallen. Beim subtilen Hinweis, das war einfach zu schwer, da hab ich mich glaub ich auf das Zielen selbst zu sehr konzentriert.

I: Und der starke visuelle Hinweis?

T: Da war das mit dem Audio Cue und dem Ziel besser.

I: Okay. Wie war das Spiel für dich, wo man die Münzen suchen muss?

T: Äh. Gut. Ich weiß halt nicht wie viele Münzen ich gefunden hab. Also weiß ich nicht wie gut ich war. Der subtile Hinweis war da aber cool. Der hat sich am besten angefühlt, man konnte die Münzen mehr oder weniger selber finden. Da war beim dem starken Blinken, da hab ich mich gefragt, irgendwie, warum spiel ich das? Verstehst du? Es war einfach eh alles vorgegeben und auch irgendwie anstrengend für das Auge. Da war der Pfeil besser. Das war angenehm, weil die aufgetaucht sind, wenn man nahe war. Das hat optisch nicht so gestört. Und generell, hat das ganz gut ausgesehen.

Test User 6

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Wie meinst du?

I: Also beim Spiel wo man die Münzen suchen muss, zum Beispiel. Was ist dir da aufgefallen?

T: Ähm. Also einmal war so ein starkes Blinken, das war am leichtesten.

I: Ja?

T: Ja. Und ich glaube einmal war so ein Pfeil. Aber da habe ich nicht ganz verstanden, wann der eingeblendet wird. Die Distanz? Oder die Höhe vielleicht?

I: Der ist nach einer gewissen Distanz aufgetaucht.

T: Achso. Das habe ich nicht verstanden.

I: Einmal war so ein leichtes Flackern. Wie hat dir das gefallen.

T: Eher nicht so gut. Da war das starke Flackern besser, das war am leichtesten.

I: Und im All Spiel? Wie war das da?

T: Da war es irgendwie ein Nachteil, wenn man sich bewegt hat. Ich will ja einen hohen Score erreichen, da bewege ich mich einfach nicht und schieße nur auf die Planeten.

I: Okay. Und von den Hinweisen her?

T: Da war das Ziel am besten, das mit den weißen Linien. Das war sofort sichtbar, die anderen Hinweise hat man gesehen, wenn es sowieso schon zu spät war.

I: Hm. Interessant! Danke auf für deinen Input, das hat mir sehr geholfen! Und danke für's Testen!

Test User 7

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Okay, also, hm. Das im All, da haben die subtilen Hinweise ganz gut funktioniert. Das starke Zeug, das Blinken, das war zu übertrieben. Da habe ich zwar den höchsten Score erreicht, aber das war nicht spannend. Komisch war das HUD, die Hinweise, die sich beim Planeten zentriert haben. Ich denke, da war ich schlecht. Das war einfach schwer. Vielleicht habe ich es aber auch ständig mit dem Crosshair verwechselt.

I: Und das Spiel mit den Münzen?

T: Die Pfeile waren zach, nicht interessant. Ich spiele solche Suchspiele aber auch echt ungern. Da sind die Pfeile langweilig, standard halt. Das kennt man aus anderen Spielen. Da war der subtile Hinweis schon cool, das war eine Challenge, die Münzen selbst zu suchen. Auch wenn mir das Suchen normal nicht so Spaß macht, das war ganz cool.

I: Das starke Blinken?

T: Das war irritierend.

I: Okay. Herzlichen Dank für deinen Input!

Test User 8

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Hm, naja, mir hat die Challenge gefehlt bei dem Hinweis mit dem Pfeil.

I: Wie? Bei welchem Spiel?

T: Bei dem Sammelspiel, das im Mittelalter mit der coolen Musik. Da war die Challenge nicht da. Da hat man nur die grünen Pfeile gesucht, das war offensichtlich.

I: Und die anderen Durchgänge?

T: Die anderen Hinweise? Ja. Das Blinken war besser als der Pfeil. Das war nicht ganz so "in-your-face". Es hat sich auch weniger geführt angefühlt.

I: Das starke oder so ein leichtes Blinken?

T: Das was so stark geblinkt hat. Das leichte war am besten. Das war die meiste Challenge. Das hat man glaube ich nicht immer gesehen. Da hat man Münzen entdecken können, ohne dass man direkt darauf Hingewiesen wird. Manchmal war halt Hilfe da, wenn man sie gebraucht hat.

I: Und im All?

T: Da war der "Target-Lock" am besten. Das war cool mit der Spaceship-Ästhetik. Aber das leichte Blinken hat mir schon am besten gefallen. Da muss man sich konzentrieren, das geht nicht so nebenbei. Beim starken Blinken, da habe ich die Maus schon immer bevor der Planet da war dort gehabt. Das war ja auch langweilig irgendwie.

I: Super! Danke!

Test User 9

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Das Weltall, da war das Zielvisier vom Design her am passend. Diese leichte Blickhilfe hat mehr geholfen irgendwie und war auch für den Spielfluss besser, finde ich. Das starke Blinken war einfach übertrieben. Das mochte ich nicht. Wo man die Münzen sammeln muss, da habe ich die leichte Blickhilfe fast nicht mitbekommen, das müsste stärker sein. Die Pfeile waren da am besten, da musste man sich bewegen um die zu finden. Das starke Blinken war einfach "in-your-face". Zu krass.

I: Wow. Okay. Danke für die schnelle Antwort! Das war super! Und danke noch einmal für das Testen.

Test User 10

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Welche Blickführungsarten gab es denn?

I: Also in jedem Spiel hat es drei Blickführungsarten gegeben, ein traditioneller visueller Hinweis, das war der grüne Pfeil und das anvisieren, einen subtilen Hinweis und ein starkes Blinken.

T: Ah, stimmt! In dem einen Spiel, das Weltraum Spiel. Da war mir der Unterschied zwischen dem starken und dem subtilen Hinweis nicht klar. Irgendwie habe ich beides nicht sehr gut gesehen.

I: Auch das starke Blinken nicht?

T: Nein, also nicht wirklich, ich habe das Anvisieren gut gefunden. Das schaut gut aus und ist leicht zu erfassen. Das passt im Kontext von einem Spiel, in dem man ein Spaceshuttle steuert.

I: Und in dem Spiel, in dem du die Münzen sammeln musstest?

T: Da habe ich das starke Blinken gut gesehen.

I: Ja?

T: Ja. Fast schon zu gut. Da war das Suchen keine Herausforderung mehr. Also das beste Verhältnis witschen Herausforderung und Erfolgsgefühl hatte der subtile Hinweis. Aber das braucht irgendwie auch eine Eingewöhnungsphase.

I: Und die grünen Pfeile?

T: Da geht man nur überall hin und sieht nach ob ein Pfeil da ist oder nicht. Das ist keine Herausforderung. Hm... Ich glaube, das waren alle Arten wie du den Blick geführt hast oder? Sonst wären mir keine mehr aufgefallen.

I: Das stimmt. Das waren alle! Drei hat es insgesamt gegeben und zwei Spiele! Danke auf jeden Fall für's testen!

Test User 11

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Mir hat das Spiel, wo die Münzen versteckt waren, echt gut gefallen. Bei den Pfeilen hab ich aber ein paar übersehen denke ich. Vielleicht auch nicht. Die anderen zwei Blickführungsarten waren auf jeden Fall eindeutiger, da hat man auch gleich gewusst was das bedeuten soll.

I: Wie meinst du das?

T: Naja, da wusste ich gleich, das bedeutet, dass ich dort hinsehen muss.

I: Okay und welche Art hat dir am besten Gefallen?

T: Das was so stark geblinkt hat. Das wäre super in einem RPG. Da könnte man Dinge echt einfach finden. Da war das andere, das leicht geblinkt hat, nicht so super. Besser als die Pfeile irgendwie, aber das stark blinkende, das war am genail. In Rollenspielen kann ich mir das gut vorstellen.

I: Stimmt, da könnte man es super einsetzen! Und was sagst du zum Space-Game?

T: Das war ganz cool, mit den Partikeleffekten. Die haben mir gefallen. Da waren mir aber die starken hinweise zu einfach. Ich meine, da habe ich einfach alle getroffen! Da war das leichte blinkende wieder super, das war mehr Action. Die anderen zwei, naja, da hat man halt immer getroffen.

I: Okay. Interessant! Vielen Dank für deinen Input!

Test User 12

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Ich wollte noch vorweg etwas sagen, also die Steuerung war etwas gewöhnungsbedürftig. Das war echt arg schnell eingestellt alles. Da könnte man ruhig etwas mehr die Geschwindigkeit herausnehmen.

I: Okay, das werde ich mir notieren.

T: Sonst hat mir alles ganz gut gefallen. Beim ersten Spiel, das mit dem Raumschiff, da war das erste perfekt, das anzielen. Das war super schnell und eigentlich wie eine Zielhilfe. Richtig angenehm zu spielen. Das zweite mit dem leichten Blinken, das war eher actionlastig, man muss mehr arbeiten und fragt sich ob man es überhaupt erwischt. Das starke Blinken war halt gut um sich vorzubereiten. Da hat man sich Zeit lassen können.

I: Obwohl die Zeit, in der die Hinweise vorher angezeigt wurden genau gleich war.

T: Okay. Ja, dann war das ein subjektiver Eindruck.

I: Das passt schon! Dafür teste ich ja schließlich.

T: Beim Spiel wo man suchen musste war das eh anders. Da war halt kein Zeitdruck oder so. Wenn es ein Zeitlimit gegeben hätte, dann wäre die Lösung mit den Pfeilen sicher gut gewesen. Aber weil das ja egal war, hab ich das Andere gut gefunden.

I: Das Andere?

T: Das man nicht wirklich gesehen hat. Ich weiß nicht was das war. Aber das hat immer nur so ungefähr zu den Coins gezeigt. Das war mehr Freiheit und man kann entdecken. Es ist aber fast gar nicht aufgefallen, da muss man schon mehr aufpassen.

I: Ja, das stimmt, das war natürlich bewusst so gewählt.

T: Achso. Ja, dann glaub ich hat es eh nur mehr das Blinken gegeben. Das war zu arg. Irgendwie einfach. Vielleicht wäre das eher gut für ein Kinderspiel. Ich glaube Kinder würden das sofort verstehen

I: Super Input. Das könnte auf jeden Fall eine Möglichkeit sein! Danke noch einmal für deine Zeit und Geduld!

Test User 13

I: Die Spiele, die du gerade gespielt hast, testen verschiedene Arten den Blick des Spielers zu führen. Welche Blickführungsart hat dir, in welchem Spiel, am besten gefallen?

T: Das zweite hat mir am besten gefallen. Das wo im All der Overlay war.

I: Was meinst du genau?

T: Ja, das wo die weißen Marker gekommen sind, das war am besten.

I: Und sonst?

T: Sonst hat mir das Leichte besser gefallen als das Starke.

I: Okay. Wie war das beim Exploration Game, wo du die Münze hast finden müssen?

T: Also die grünen Lichter waren zu spät. Das war nicht so leicht sichtbar. Irgendwie waren sie schon gut erkennbar aber oft habe ich sie nicht gesehen und dann habe ich immer zwei mal schauen müssen.

I: War das mit dem starken Blinken besser?

T: Ja schon, aber das war ja langweilig. Ich mein, das hat man gleich gesehen.

I: Und der subtile Hinweis?

T: Das war am besten.

I: Super! Herzlichen Dank für deine Infos. Ach ja und danke für das Testen noch einmal, das war super.

Appendix B

The Game Experience Questionnaire

The *Game Experience Questionnaire* by IJsselsteijn et al. [17] uses a total of 33 statements, which need to be evaluated on a five-point Likert scale ranging from "not at all" to "extremely". These statements measure the seven different factors *competence*, *sensory and imaginative immersion*, *flow*, *tension/annoyance* and *challenge*, as well as *negative affect* and *positive affect*. The following list provides all statements included in the questionnaire, sorted according to the the factor they are measuring:

1. Competence:
   - I felt skilful
   - I felt competent
   - I was good at it
   - I felt successful
   - I was fast at reaching the game's target
2. Sensory and imaginative immersion:
   - It was aesthetically pleasing
   - I felt imaginative
   - I felt that I could explore things
   - I found it impressive
   - It felt like a rich experience
3. Flow:
   - I was fully occupied with the game
   - I forgot everything around me
   - I lost track of time
   - I was deeply concentrated in the game
   - I lost connection with the outside world
4. Tension/annoyance:
   - I felt annoyed
   - I felt irritable
   - I felt frustrated

5. Challenge:
    - I thought it was hard
    - I felt pressured
    - I felt challenged
    - I felt time pressure
    - I had to put a lot of effort into it
6. Negative affect:
    - It gave me a bad mood
    - I thought about other things
    - I found it tiresome
    - I felt bored
7. Positive affect:
    - I felt content
    - I thought it was fun
    - I felt happy
    - I felt good
    - I enjoyed it

Appendix C

CD-ROM/DVD Contents

C.1 Evaluation

- **Evaluation:** The original evaluation files
  - **Evaluation/GameExperienceQuestionnaireResults.pdf:** Complete results of the *Game Experience Questionnaire* saved as PDF
  - **Evaluation/GeneralNotes.pdf:** Complete collection of the gathered age data and testing order saved as PDF
  - **Evaluation/InGameResults.pdf:** Complete results for the collected in-game data saved as PDF

C.2 Project

- **Project:** The binary and executable files for *PlayGuide* and the test games
  - **Project/Bin:** Folder that contains the project files for *PlayGuide* (Unity Version 2017.4.3)
  - **Project/Exec:** Folder that contains the executables for *PlayGuide* (Unity Version 2017.4.3)

C.3 Thesis

- **Thesis:** The master's thesis and its contents
  - **Thesis/Images:** Folder that contains the original images used for this master's thesis
  - **Thesis/Pdf:** Folder that contains this master's thesis saved as a PDF

References

Literature

[1] Reynold Bailey et al. “Subtle Gaze Direction”. *ACM Transactions on Graphics* 28.4 (2009), 100:1–100:14 (cit. on pp. 1–3, 21–23, 26, 27, 29, 61, 63, 65).

[2] Erhardt Barth et al. “Guiding the Mind’s Eye: Improving Communication and Vision by External Control of the Scanpath”. In: *Proceedings of SPIE Conference on Human Vision and Electronic Imaging XI*. Vol. 6057. International Society for Optics and Photonics. San Francisco, 2006, p. 60570D (cit. on p. 22).

[3] Matthias Bernhard et al. “Gaze-to-Object Mapping During Visual Search in 3D Virtual Environments”. *ACM Transactions on Applied Perception* 11.3 (2014), 14:1–14:17 (cit. on pp. 14, 15, 37, 59).

[4] H Collewijn, AJ Martins, and RM Steinman. “Compensatory Eye Movements during Active and Passive Head Movements: Fast Adaptation to Changes in Visual Magnification.” *The Journal of Physiology* 340.1 (1983), pp. 259–286 (cit. on p. 6).

[5] Charles E. Connor, Howard E. Egeth, and Steven Yantis. “Visual Attention: Bottom-Up Versus Top-Down”. *Current Biology* 14.19 (2004), R850–R852 (cit. on pp. 17, 18).

[6] Anna Cox et al. “Not Doing but Thinking: The Role of Challenge in the Gaming Experience”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. Montreal, 2012, pp. 79–88 (cit. on p. 61).

[7] Andrew T. Duchowski. *Eye Tracking Methodology. Theory and Practice*. Heidelberg: Springer-Verlag, 2007 (cit. on pp. 5–10, 12, 17, 18).

[8] James T. Enns and Sarah C. MacDonald. “The Role of Clarity and Blur in Guiding Visual Attention in Photographs”. *Journal of Experimental Psychology: Human Perception and Performance* 39.2 (2013), p. 568 (cit. on p. 19).

[9] Joseph H. Goldberg and Anna M. Wichansky. “Eye Tracking in Usability Evaluation: A Practitioner’s Guide”. In: *The Mind’s Eye*. Elsevier, 2003, pp. 493–516 (cit. on p. 10).

[10] Elizabeth R. Grant and Michael J. Spivey. “Eye Movements and Problem Solving: Guiding Attention Guides Thought”. *Psychological Science* 14.5 (2003), pp. 462–466 (cit. on p. 20).

[11] Steve Grogorick et al. “Subtle Gaze Guidance for Immersive Environments”. In: *Proceedings of the ACM Symposium on Applied Perception*. Cottbus: ACM, 2017, 4:1–4:7 (cit. on pp. 19, 20).

[12] John E. Hall. *Guyton and Hall Textbook of Medical Physiology*. Kidlington: Elsevier Health Sciences, 2015 (cit. on p. 6).

[13] Hajime Hata, Hideki Koike, and Yoichi Sato. “Visual Guidance with Unnoticed Blur Effect”. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. Bari: ACM, 2016, pp. 28–35 (cit. on p. 20).

[14] Angelique W. Hendriks. “Vergence eye movements during fixations in reading”. *Acta Psychologica* 92.2 (1996), pp. 131–151 (cit. on p. 6).

[15] Anthony Hornof, Anna Cavender, and Rob Hoselton. “Eyedraw: A System for Drawing Pictures with Eye Movements”. In: *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility*. Atlanta: ACM, 2004, pp. 86–93 (cit. on p. 7).

[16] David H. Hubel. *Eye, Brain, and Vision*. New York: Scientific American Library/-Scientific American Books, 1995 (cit. on p. 4).

[17] Wijnand A. IJsselsteijn, Yvonne A. W. de Kort, and Karolien Poels. *The Game Experience Questionnaire*. Eindhoven: Technische Universiteit Eindhoven. 2013 (cit. on pp. 3, 36, 43, 63, 74).

[18] Robert J.K. Jacob and Keith S. Karn. “Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises”. In: *The Mind's Eye*. Elsevier, 2003, pp. 573–605 (cit. on p. 18).

[19] John Jonides. “Voluntary Versus Automatic Control Over the Mind's Eye's Movement”. *Attention and Performance* (1981), pp. 187–203 (cit. on p. 19).

[20] Azam Khan et al. “Spotlight: Directing Users' Attention on Large Displays”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Portland: ACM, 2005, pp. 791–798 (cit. on p. 20).

[21] Oleg V. Komogortsev and Alex Karpov. “Automated Classification and Scoring of Smooth Pursuit Eye Movements in the Presence of Fixations and Saccades”. *Behavior research Methods* 45.1 (2013), pp. 203–215 (cit. on p. 13).

[22] Manu Kumar et al. “Improving the Accuracy of Gaze Input for Interaction”. In: *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*. Savannah: ACM, 2008, pp. 65–68 (cit. on p. 13).

[23] Michael Land, Neil Mennie, and Jennifer Rusted. “The Roles of Vision and Eye Movements in the Control of Activities of Daily Living”. *Perception* 28.11 (Feb. 1999), pp. 1311–1328 (cit. on pp. 18–20).

[24] Yen-Chen Lin et al. “Tell Me Where to Look: Investigating Ways for Assisting Focus in 360° Video”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Denver: ACM, 2017, pp. 2535–2545 (cit. on pp. 20, 21).

[25] Robert Gabriel Lupu and Florina Ungureanu. "A Survey of Eye Tracking Methods and Applications". *The Bulletin of the Polytechnic Institute of Jassy* (2013), pp. 71–86 (cit. on p. 6).

[26] Ann McNamara, Reynold Bailey, and Cindy Grimm. "Search Task Performance Using Subtle Gaze Direction with the Presence of Distractions". *ACM Transactions on Applied Perception* 6.3 (Aug. 2009), 17:1–17:19 (cit. on p. 20).

[27] Ann McNamara et al. "Directing Gaze in Narrative Art". In: *Proceedings of the ACM Symposium on Applied Perception*. Los Angeles: ACM, Aug. 2012, pp. 63–70 (cit. on pp. 3, 19).

[28] Carlos H. Morimoto and Marcio R.M. Mimica. "Eye Gaze Tracking Techniques for Interactive Applications". *Computer Vision and Image Understanding* 98.1 (2005), pp. 4–24 (cit. on pp. 4, 5).

[29] Thomas E. Ogden and Robert F. Miller. "Studies of the Optic Nerve of the Rhesus Monkey: Nerve Fiber Spectrum and Physiological Properties". *Vision Research* 6.9-10 (1966) (cit. on p. 26).

[30] Robert P. O'Shea. "Thumb's Rule Tested: Visual Angle of Thumb's Width is About 2 Degrees". *Perception* 20.3 (1991), pp. 415–418 (cit. on p. 5).

[31] Stephen E. Palmer. *Vision Science: Photons to Phenomenology*. Cambridge: MIT Press, 1999 (cit. on pp. 17, 18).

[32] Anjul Patney et al. "Towards Foveated Rendering for Gaze-tracked Virtual Reality". *ACM Transactions on Graphics* 35.6 (Nov. 2016), 179:1–179:12 (cit. on p. 61).

[33] Alex Poole and Linden J. Ball. "Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects". In: *Encyclopedia of Human Computer Interaction*. IGI Publishing Hershey, Jan. 2006, pp. 211–219 (cit. on p. 7).

[34] Patrick Renner and Thies Pfeiffer. "Attention Guiding Techniques Using Peripheral Vision and Eye Tracking for Feedback in Augmented-Reality-Based Assistance Systems". In: *Proceedings of the 2017 IEEE Symposium on 3D User Interfaces*. IEEE. Los Angeles, Mar. 2017, pp. 186–194 (cit. on p. 19).

[35] Richard Rouse and Steve Illustrator-Ogden. *Game Design Theory and Practice*. Burlington: Jones & Bartlett Learning, 2000 (cit. on p. 30).

[36] Sajay Sadasivan et al. "Use of Eye Movements As Feedforward Training for a Synthetic Aircraft Inspection Task". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Portland: ACM, 2005, pp. 141–149 (cit. on p. 20).

[37] Dario D. Salvucci and Joseph H. Goldberg. "Identifying Fixations and Saccades in Eye-tracking Protocols". In: *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*. Palm Beach Gardens: ACM, 2000, pp. 71–78 (cit. on pp. 11, 13, 25).

[38] Martin Schön. “An Evaluation of Interactors’ Gaze-to-Object Mapping Performance in 3D Virtual Environments”. MA thesis. Stockholm: KTH Royal Institute of Technology, School of Computer Science and Communication, Aug. 2016 (cit. on pp. 15, 37, 57).

[39] Alia Sheikh et al. “Directing Attention in 360-Degree Video”. *Proceedings of the International Broadcasting Convention* 9 (2016), pp. 29–36 (cit. on pp. 2, 3).

[40] Robert F. Spencer and John D. Porter. “Structural Organization of the Extraocular Muscles”. *Reviews of Oculomotor Research* 2 (1988), pp. 33–79 (cit. on pp. 5, 6).

[41] Srinivas Sridharan et al. “Subtle Gaze Manipulation for Improved Mammography Training”. In: *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*. Toulouse: ACM, 2012, pp. 112–112 (cit. on pp. 3, 19, 20).

[42] Veronica Sundstedt et al. “A Psychophysical Study of Fixation Behavior in a Computer Game”. In: *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*. Los Angeles: ACM, 2008, pp. 43–50 (cit. on p. 14).

[43] Veronica Sundstedt et al. “Visual Attention and Gaze Behavior in Games: An Object-Based Approach”. In: *Game Analytics*. Springer, Apr. 2013, pp. 543–583 (cit. on pp. 2, 7, 11, 14, 15, 18, 37, 59).

[44] Andreas Traschütz, Wolf Zinke, and Detlef Wegener. “Speed Change Detection in Foveal and Peripheral Vision”. *Vision Research* 72 (2012), pp. 1–13 (cit. on pp. 2, 5).

[45] Anne Treisman. “Features and Objects in Visual Processing”. *Scientific American* 255.5 (Nov. 1986), pp. 114–125 (cit. on p. 17).

[46] Eduardo Velloso and Marcus Carter. “The Emergence of EyePlay: A Survey of Eye Interaction in Games”. In: *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. Austin: ACM, 2016, pp. 171–185 (cit. on pp. 11, 23, 57).

[47] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. “Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets”. In: *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Zurich: ACM, 2013, pp. 439–448 (cit. on p. 15).

[48] Eleonora Vig, Michael Dorr, and Erhardt Barth. “Learned Saliency Transformations for Gaze Guidance”. In: *Proceedings of SPIE Conference on Human Vision and Electronic Imaging XVI*. Vol. 7865. International Society for Optics and Photonics. San Francisco, Mar. 2011, pp. 152–162 (cit. on p. 20).

[49] Alfred L.L Yarbus. “Eye Movements During Perception of Complex Objects”. In: *Eye Movements and Vision*. Springer-Verlag, 1967, pp. 171–211 (cit. on p. 18).

[50] Laurence R. Young and David Sheena. “Survey of Eye Movement Recording Methods”. *Behavior Research Methods & Instrumentation* 7.5 (Sept. 1975), pp. 397–429 (cit. on pp. 8, 9).

Audio-visual media

[51] Tobii AB. *Tobii Pro Glasses 2 hero shot*. May 2013. URL: https://www.tobii.com/imagevault/publishedmedia/xcixwlpg0l3t5uiflgsw/TobiiPro_Glasses_2.jpg (cit. on p. 8).

[52] Tobii AB. *Tobii Pro Glasses 2 hero shot*. May 2013. URL: https://www.tobii.com/imagevault/publishedmedia/yli14947afs2id8e0gzj/TobiiPro_TX300_clinical_reasearch_eye_movements_st.jpg (cit. on p. 9).

[53] Buzo. *Simple Pair of Teapots*. Apr. 2014. URL: http://www.blendswap.com/blends/view/75962 (cit. on p. 14).

[54] OpenStax College. *Structure of the Eye*. May 2013. URL: https://commons.wikimedia.org/wiki/File:1413_Structure_of_the_Eye.jpg (cit. on p. 5).

[55] Ra'anan Gefen and Nano-Retina Inc Leonid Yanovitz. *Saccades and Microsaccades*. Apr. 2016. URL: https://commons.wikimedia.org/wiki/File:Saccades_and_Microsaccades.jpg (cit. on p. 12).

[56] *OpenArena-Gauntlet*. Dec. 2006. URL: https://commons.wikimedia.org/wiki/File:OpenArena-Gauntlet.jpg (cit. on p. 18).

Online sources

[57] Tobii AB. *Specifications for the Tobii Eye Tracker 4C*. 2017. URL: https://help.tobii.com/hc/en-us/articles/213414285-Specifications-for-the-Tobii-Eye-Tracker-4C (visited on 04/19/2018) (cit. on p. 2).

[58] Tobii AB. *Specifications for Tobii EyeX*. 2017. URL: https://help.tobii.com/hc/en-us/articles/212818309-Specifications-for-EyeX (visited on 04/09/2018) (cit. on pp. 7, 13).

[59] Eli Ben-Joseph and Eric Greenstein. *Gaze Direction in Virtual Reality Using Illumination Modulation and Sound*. 2016. (Visited on 03/12/2017) (cit. on pp. 2, 3, 23, 60).

[60] Carl Bianco. *How Vision Works*. 2000. URL: https://health.howstuffworks.com/mental-health/human-nature/perception/eye.htm (visited on 02/28/2018) (cit. on p. 5).

[61] Zoya Bylinskii et al. *MIT Saliency Benchmark*. 2016. URL: http://saliency.mit.edu/ (visited on 05/01/2018) (cit. on p. 19).

[62] Galactic Cafe. *The Stanley Parable*. 2018. URL: https://www.stanleyparable.com/ (visited on 05/23/2018) (cit. on p. 61).

[63] *Dying Light Game*. 2015. URL: https://dyinglightgame.com (visited on 04/08/2018) (cit. on p. 64).

[64] Inc. Gartner. *Hype Cycle for Emerging Technologies, 2014*. 2014. URL: https://www.gartner.com/newsroom/id/2819918 (visited on 05/23/2018) (cit. on p. 60).

[65] ROCKFISH Games GmbH. *Everspace*. 2018. URL: https://everspace-game.com (visited on 05/11/2018) (cit. on pp. 31, 40).

[66] Adam Henry. *Visual Cues In Level Design*. 2018. URL: https://www.gamasutra.com/blogs/AdamHenry/20150713/248186/Visual_Cues_In_Level_Design.php (visited on 04/14/2017) (cit. on pp. 2, 23, 30).

[67] Bethesda Softworks LLC. *Dishonored 2*. 2018. URL: dishonored.bethesda.net (visited on 05/11/2018) (cit. on pp. 31, 41).

[68] Daybreak Game Company LLC. *Planetside 2 Website*. 2018. URL: https://www.planetside2.com (visited on 05/11/2018) (cit. on pp. 31, 41).

[69] Statista. *Value of the Gamification Market Worldwide in 2016 and 2021 (in Billion U.S. Dollars)*. 2018. URL: https://www.statista.com/statistics/608824/gamification-market-value-worldwide/ (visited on 05/23/2018) (cit. on p. 60).

[70] Statista. *Value of the Global video Games Market from 2011 to 2020*. 2018. URL: https://www.statista.com/statistics/246888/value-of-the-global-video-game-market/ (visited on 04/14/2017) (cit. on p. 23).

[71] Unity Technologies. *Unity Editor System Requirements*. 2000. URL: https://unity3d.com/de/unity/system-requirements (visited on 04/05/2018) (cit. on p. 24).

[72] Tobii Technology. *Tobii Technology Website*. 2018. URL: https://www.tobii.com (visited on 02/28/2018) (cit. on p. 2).

[73] Tobii Technology. *Tobii Technology Website*. 2018. URL: https://www.tobii.com/tech/products/vr/ (visited on 06/09/2018) (cit. on p. 62).

[74] Thales S. Teixeira. *The Rising Cost of Consumer Attention: Why You Should Care, and What You Can Do About It*. 2014. (Visited on 03/18/2017) (cit. on p. 1).