Universität Regensburg

Deep Learning-based Latency Compensation: From Understanding to Application

Inaugural-Dissertation zur Erlangung der Doktorwürde der Fakultät für Informatik und Data Science der Universität Regensburg

Vorgelegt von
David Halbhuber
aus Obertraubling
2024

Erstgutachter:	Prof. Dr. Niels Henze
Zweitgutachter:	Prof. Dr. Enrico Rukzio
Tag der mündlichen Prüfung:	11.12.2024

Abstract

Humans interact with computers in continuous feedback loops. To start an interaction, a user provides an input to the computer, which the computer processes. The results of this procession are then delivered to the user by the computer. Subsequently, the user can either end the interaction if they achieved their desired goal or start the next loop iteration by communicating a new input to the computer. Since computers require time to process the user's input, to calculate responses, and to produce outputs, the user does not receive feedback on their input immediately. This time between the user's input to a system and the user's perception of the system's generated output is called latency.

Latency is inherently part of every interactive system. Regardless of the shrinking sizes of the latest transistors, updated networking architecture, and novel developments in software engineering, transmitting and processing user input will always require time. A latency that is too high, however, negatively influences user experience and performance. When interacting with a high latency system, users are less accurate. For example, when selecting elements in a user interface, they require more time to complete specific tasks and derive less satisfaction from the interaction than interacting with a low-latency system. In general, the faster and more interactive a system or application is, and thus, the tighter the interaction deadline is, the more pronounced the negative effects of latency are. Hence, fast-paced video games stand out as one category of interactive systems significantly impaired by latency. In video games, a high latency leads to players scoring fewer points, needing more time to complete in-game tasks, or even being unable to complete a given task at all. Depending on the game, the internal game mechanics, and various other factors, the threshold before latency starts affecting gameplay varies. Previous works report latency thresholds between 25 ms and 1 000 ms.

However, particularly in fast-paced games that require split-second decision-making, such as shooting games, latency can make the difference between virtual life and death. To overcome and compensate for the negative effects of latency, researchers, developers, and publishers proposed and investigated different latency compensation approaches. Previous work, for instance, evaluated methods based on manipulating the geometrical dimension of video games to account for latency, communicating latency to the players via in-game objects, allowing them to adjust their gaming behavior to the current latency, or predicting future game states which allow the game to calculate game events before the actual user input which reduces the overall perceived latency. Although a large body of work researched latency and its compensation in video games, the methods used are typically highly specialized, perceived to be unfair by players, or unsuitable for generalization. Thus, there are still open challenges and questions about the effects of latency in video games and its compensation that require investigation to provide players with a high level of game experience and performance while accounting for latency.

The overarching goal of this dissertation is to enhance our understanding of the effects of latency on video game players and to use this knowledge to design and develop better latency compensation methods. To achieve this, we conducted nine empirical user studies investigating different aspects of latency in video games and its compensation. In the first two studies, we investigated how latency variation, a change in latency, influences player performance and experience. We show that a small-term latency variation, a rapid change in latency, is not as negatively influential as a long-term switch between two levels. Players in our studies could cope with small-term variations but were affected if latency switched for a longer duration. In the subsequent four studies, we investigated how the perceptual channel, which is how the players perceive latency, modulates its effects. We show that standalone auditory latency only affects highly skilled players, that the in-game perspective, how players visually perceive the game, does not alter the effects of latency, and that the mere display of latency induces an expectation-based performance and experience degradation in players.

In the last three studies presented in this dissertation, we use the knowledge gained within this thesis to develop novel latency compensation methods. We evaluate artificial neural network-based latency compensation techniques based on our investigation and previous work. First, we showcase that artificial neural networks can compensate for the negative effects of latency in a custom video game using game-internal information about the game world and player behavior by predicting player movement in the game, effectively alleviating playing in a high latency system to a low latency level. Next, we applied the developed methods in a slow-paced commercial video game without accessing game-internal data. In the last study, we further refined our approach and predicted player inputs for latency compensation in a fast-paced commercial shooting game using a multi-model approach. Overall, the presented studies demonstrate that artificial neural networks predicting player behavior and input are well-suited methods for latency compensation in video games.

Building on the empirical findings of our studies, this thesis concludes with actionable design guidelines for future latency compensation development and latency research for games and beyond.

Acknowledgments

This work would not have been possible without the love, support, and help of a great many people. I want to dedicate the next pages to that people. Please know that I am eternally grateful for all of you. Thank you.

First, I thank my advisor Prof. Dr. Niels Henze. Thank you for all the last minute revisions, for all the fruitful discussions, for always having an open ear, for always having time to discuss my work, for always supporting all my endeavors, and for all the talks we had about sciences, research, and all the other stuff. Thank you so much.

I also thank Prof. Dr. Enrico Rukzio for being my second examiner. Thank you for taking the time and for a great and insightful discussion at my defense.

My deepest thanks also goes to Prof. Dr. Christian Wolff, who supported me from the start to the end of my work, with invaluable feedback and countless interesting discussions. Also, thank you deeply, for providing me the opportunity to work in you lab. I really appreciate all your investment.

I also want to express my gratitude to Dr. Johanna Bogon, this work would have not been possible without you. I really enjoyed all our talks, all the projects we did together, all the conferences we went, and all the lectures we gave together. Please know, from the bottom of my heart, that I am deeply thankful for getting to know you and to having had the opportunity to work with you.

I thank Prof. Dr. Nele Rußwinkel, Prof. Dr. Katrin Wolf, and Prof. Dr. Valentin Schwind for being on my thesis advisory board.

This work, of course, would have not been possible without a great set of colleagues. I thank Susanne Klinger for always having an open door about any issue (or just to have a chat) and for supporting me in all the organizational challenges that come with working at an university. Thank you very much. I, also, thank Jakob Fehle for being a great colleague and friend. Thank you for all the insightful discussions about research, working at the university, Schnitzel and all the other stuff. I also would like to thank Michael Achmann-Denkler, with whom I had the pleasure to share an office for almost four years. Furthermore, I thank all the other amazing people of which I had the pleasure of working with at the media informatics lab Dr. Martin Kocur, Dr. Raphael Wimmer, Andreas Schmid, Thomas Schmidt, Elisa Valetta, Alexander Kalus, Vitus Maierhöfer, Marie Sautmann, and Martina Emmert. Also, I want to thank all of the amazing students (and of which there are simply to many to be mentioned individually here) – Thank you!

A big thank you is owed to my family: Thank you so much to all my sisters; Marie Halbhuber, Isabelle Halbhuber, Jana Simmel, and Maike Halbhuber. The level of support and safety (independent of my doctoral work) you provide is unimaginable. I feel nothing but gratitude, pride, and love for each of you. I know that there were times where it was not easy and all the more grateful I am today, for today we will manage, together. I also want to thank my mother Edith Halbhuber, and my grandparents Christine and Jörg Halbhuber for always supporting me. Thank you to Daniel Simmel for being a great friend and the best husband for my sister I could imagine. I want to thank Roman Schmal, Michael Kelly, and Christoph Kopp, without you my time at the university (as well as the time before and after) would have only been half (if even) the fun. My deepest gratitude is also owed to my in-laws. Thank you to Anna-Elise and Ludwig Dechant for being so supportive and for all the help over the past years. My gratitude also goes to the Wettstein-family Julia, Christian, Noah, and Emma, simply for being amazing people and great friends. Also, I thank Sarah and Roxy, the two best dogs in the world.

Last but not least, I want to say thank you to my beautiful wife Lisa Halbhuber. The day we met was the greatest day of my life. You are my best friend and the kindest, smartest, and most beautiful person I have ever gotten to know. I cannot express how deeply thankful I am for all your help, your patience, your advice, and your love. Me would have not been possible without You. I love you.

Table of Contents

1 Introduction **15**

1.1 Latency . . . 22

1.1.1 Latency Definition . . . 22

1.1.2 Local and Network Latency . . . 23

1.1.3 Latency Thresholds in HCI and Video Games . . . 24

1.2 The Human-Computer Feedback Loop . . . 26

1.2.1 Overview . . . 26

1.2.2 The Cycle of Execution, Evaluation, and Latency . . . 26

1.2.3 The Model Human Processor . . . 28

1.3 A brief History of Video Games . . . 30

1.4 Research Questions . . . 32

1.5 Methodological Approach and Contributions . . . 35

1.6 Publications and Work Distribution . . . 37

1.7 Thesis Outline . . . 41

2 Background and Related Work **43**

2.1 Latency in Interactive Systems . . . 43

2.1.1 Search Strategy . . . 44

2.1.2 Results of the Review on Latency in Interactive Systems . . . 44

2.1.3 Summary . . . 51

2.2 Latency in Video Games . . . 52

2.2.1 Method . . . 52

2.2.2 Results . . . 56

2.2.3 Summary 68
2.3 Latency Compensation Methods 69
2.3.1 Search Strategy 70
2.3.2 Game-internal Latency Compensation 71
2.3.3 Game-external Latency Compensation 86
2.3.4 Summary 91
2.4 Synthesis of Related Work 91

3 Understanding Latency Variation 95
3.1 Understanding Small-Term Variation of Latency (Study I) 96
3.1.1 Background and Research Rationale 96
3.1.2 Method 97
3.1.3 Results 104
3.1.4 Discussion 109
3.1.5 Conclusion 114
3.2 Understanding Long-Term Variation of Latency (Study II) 114
3.2.1 Background and Research Rationale 115
3.2.2 Apparatus and Prestudy 116
3.2.3 Method 120
3.2.4 Results 123
3.2.5 Discussion 130
3.2.6 Conclusion 136

4 Understanding Latency Perception 139
4.1 Understanding Audio Latency in Custom Video Games (Study III) 141
4.1.1 Method 143
4.1.2 Results 147
4.1.3 Discussion 151
4.1.4 Conclusion 153
4.2 Understanding Audio Latency in Commercial Video Games (Study IV) 155
4.2.1 Method 155
4.2.2 Results 158
4.2.3 Discussion 164
4.2.4 Conclusion 169

4.3 Understanding In-Game Perspective and Latency (Study V) 170
4.3.1 Background and Research Rationale 171
4.3.2 Apparatus 172
4.3.3 Method 175
4.3.4 Results 179
4.3.5 Discussion 187
4.3.6 Conclusion 193
4.4 Understanding Latency Expectancy (Study VI) 194
4.4.1 Background and Research Rationale 196
4.4.2 Method 196
4.4.3 Results 200
4.4.4 Discussion 204
4.4.5 Conclusion 208

5 Compensating Latency via Artificial Neural Networks 211
5.1 Artificial Neural Networks in Custom Video Games (Study VII) 214
5.1.1 Background and Research Rationale 215
5.1.2 Data Collection and Apparatus 215
5.1.3 Model Development 219
5.1.4 Method 222
5.1.5 Results 224
5.1.6 Discussion 227
5.1.7 Limitation and Future Work 231
5.1.8 Conclusion 232
5.2 Artificial Neural Networks in Slow-Paced Games (Study VIII) 233
5.2.1 Background and Research Rationale 234
5.2.2 Data Collection and Apparatus 234
5.2.3 Method 240
5.2.4 Results 242
5.2.5 Discussion 243
5.2.6 Conclusion 248
5.3 Artificial Neural Networks in Fast-Paced Games (Study IX) 248
5.3.1 Background and Research Rationale 249
5.3.2 Data Collection and Apparatus 250

5.3.3 Method 257
5.3.4 Results 260
5.3.5 Discussion 268
5.3.6 Conclusion 271
5.4 Synthesis Compensating Latency 272

6 Conclusion **273**
6.1 Summary of Contributions 273
6.2 Design Guidelines and Implications 275
6.3 Limitations and Future Work 280
6.4 Latency Compensation beyond Games 285

Bibliography **289**

List of Figures **321**

List of Tables **325**

List of Acronyms **327**

1

Introduction

Over the last 80 years, digital computers have been constantly developed further. From inert, room-sized, and mechanically operated computing machines to today's mobile, pocket-sized jacks of all trades – computers have evolved rapidly. This evolution also transformed the way we interact with computers. While it was perfectly normal to wait for several days for the results of a, at that time, complex mathematical calculation in the age of tape-operated computers, the same is unimaginable today. Today, the world is at everyone's fingertips. Enabled by smartphones and the Internet, we are able to obtain almost any information nearly instantaneously, outsource significant computational endeavors to cloud-based servers anywhere in the world, and connect and stay in touch with friends, family, and colleagues independent of location. All this is made accessible through a small device that fits in our pocket. While 80 years ago, it was reserved for a selected and specifically trained elite to operate computers, nowadays, almost everyone has access to one and uses it daily. Our lives and our society shifted to a universalization of the computer.

With this shift to a computer-centered society, new possibilities and opportunities as well as new challenges and questions arose. For example, it was crucial to establish reliable ways for humans to communicate with computers. And likewise, for the computer, it was paramount to present its results in a humanly understandable way. Dedicated research disciplines, such as psychology, human factors, and ergonomics, committed tremendous resources to investigate these open questions on how the interaction between humans and computers can be optimized. About 40 years ago, these efforts led to the emergence of human-computer interaction (HCI) as a dedicated research field (MacKenzie, 2013). Although research investigating the interaction between computers and humans dates back much longer, it was only then that HCI unambiguously was established as a dedicated discipline, arguably with the first installment of the Conference on Human Factors in Computing Systems (CHI) in 1982 (*CHI '92: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 1992).

Researchers in HCI investigate and observe how humans interact with computers, they design and evaluate novel interfaces for humans to interact with computers, and establish input-output paradigms that provide frameworks for future developments. A key challenge of HCI research is to provide users with the best possible way to interact with a computer. The quality of interaction between humans and computers can be measured in two constructs: user experience (UX) and user performance (UP).

UX is a highly subjective measure and describes the qualia of an interaction. It loosely relates to how users feel while, before, and after interacting with a system, if they enjoyed it or not, or if they are satisfied with the result of an interaction. As it is highly subjective, it is influenced by each user. The user's personal preference, previous experiences, and the current use context represent only a few possible influences on UX (Hassenzahl & Tractinsky, 2006). In an attempt to make UX more approachable and quantifiable by research and developers, there have been several different definitions over the years. For example, the DIN ISO 9241-210 defines UX as "A person's perceptions and responses that result from the use and/or anticipated use of a product, system or service" (DIN, 2022).

UP, on the other hand, is a hard objective measure. It is often described as a holistic measure of sub-performances, such as task completion time (TCT) – the amount of time required by a user to complete a specific task using a particular system, error rate – the frequency of errors made by users while performing tasks, an error is typically defined as a user action that deviates from the intended or correct sequence of steps to complete a task, or task completion rate – the percentage of successfully completed tasks out of the total number of attempted tasks, a task is considered successful if the user accomplishes the goal without significant errors or failures.

Evidently, quantifying the quality of a user's interaction with a computer is neither easily done nor perfectly definable. A broad spectrum of aspects shape and influence the interaction, such as the device, the task, or the user's prior knowledge. However, there are even more fundamental components that influence the communication between humans and computers. One particular influential element is the interaction's temporality.

Fundamental and early work in HCI, such as Card et al.'s model human processor (MHP) (1983), demonstrates that humans and computers interact in iterative and continuous feedback loops. The human provides an input to the computer, which the computer processes. The results of this processing can be, in turn, sent back to the user. Then, the user can start another iteration by providing a new input or building on the previous one. This human-computer feedback loop is temporally limited. The human, on the one hand, needs time to formulate an input and to understand the computer's output. On the other hand, the computer needs time to calculate the results of the user's input. The time between the user's input to the system and the computer starting to display the results of the input is called latency, which in HCI is often also called system response time (SRT) or system delay. Similar to the evolution of computers, the acceptable latency of an interaction also evolved over the years. While latency played almost no role in the early days of computing, it constantly became more and more critical. The establishment of crucial interaction paradigms, such as windows, icons, menus, and pointers (WIMP) and the graphical user interface (GUI), marked tipping points of latency's influence. With increasing interactivity, latency's impact increased as well. While early work, such as Card et al.'s (1983), proposed that the optimal latency of an interaction is about 100 ms, more novel work (Kaaresoja & Brewster, 2010) showed that latency already starts to affect the interaction between humans and computers at 30 ms. Because of its relevance, a large body of work investigates the effects of latency on UX and UP.

No matter the device used; a touch-operated mobile device, a head-mounted virtual reality (VR) goggle, or a stationary computer, latency always played a crucial role when interacting with computer systems (MacKenzie & Ware, 1993). A high latency affects UX and UP for a wide range of task and interaction modalities such as VR settings (Meehan et al., 2003; Kasahara et al., 2017; Akiduki et al., 2003), mobile devices (Henze et al., 2016; Henze et al., 2017), classical workstations, and augmented reality (AR) (Nabiyouni et al., 2017; Lee et al., 2010). Specific examples are the works by Jota et al. (2013) and Annett et al. (2014), in which they showed that latency of more than 25 ms leads to reduced UP when interacting with a mobile device. In other work, Ng et al. (2012) found that users perceive latency starting at 2 ms. Building on this, Ng et al. (2014) showed in later work that users can even detect discrepancies between 1 ms latency and 2 ms latency in some tasks. While previous work demonstrates that the effects of latency negatively alter a variety of tasks and interaction modalities, there is one HCI domain in which the adverse effects of latency manifest especially prominent – the domain of video games.

In video games, a latency that is too high significantly and negatively influences the gaming session and its players. The adverse effects of latency manifest in various ways, such as decreased player scores, longer completion times for in-game tasks, or even failure to complete specific tasks at all (Eg et al., 2018; Beigbeder et al., 2004; Claypool & Finkel, 2014). However, not all types of games are equally affected by latency. Work by Claypool and Claypool (2006), for example, indicates that first-person shooter (FPS) games are particularly susceptible to the effects of latency since these games rely on timely user inputs and split-second decision-making. In general, the tolerance threshold for latency in fast-paced video games has been estimated to be at 150 ms in some work (Long & Gutwin, 2018), while other work reported decreased player performance (PP) starting at 100 ms latency (Quax et al., 2004). While a clear latency threshold for PP is unattainable yet, previous work also highlighted that latency affects the players' gaming experience. Game experience (GX) describes the overall enjoyment, fun, or pleasantness experienced while playing a game. Similarly to UX, it is influenced by various aspects, such as how the players perceive the game world, how visual and auditory information is provided, what video game genre the player usually plays and likes, or if the players perceive the game's challenge as appropriate. Previous work investigating the effects of latency on GX also found that latency negatively impacts video game's quality of experience (QoE), with QoE being a holistic, single-valued measure describing the overall experience. For example, Liu et al. (2021) showed that a latency of 150 ms leads to a 25 % decrease in QoE. In related work, Liu et al. (2021) found a linear decline in QoE by 20 % as latency increases from 25 ms to 125 ms. Previous work also aimed to capture the effects of latency on GX using more in-depth and multidimensional instruments, such as the Player Experience Inventory (PXI) (Abeele et al., 2020). Durnez et al. (2021), for example, found that latency decreases the experienced flow, with flow being a state of complete immersion, effortless concentration, and enjoyment (Csikszentmihalyi, 1990, p. 4). The authors also found that increasing latency decreases the player's interest in and enjoyment of the game. In summary, the effects of latency on PP and GX are substantial and multifaceted. The impaired responsiveness, compromised immersion, gameplay imbalances, and negative emotional and cognitive impacts collectively highlight the destructive nature of latency in games.

Researchers and game developers, thus, proposed different latency compensation techniques to reduce or even overcome these adverse effects. The goal of any latency compensation method in video games is to allow players to achieve low latency levels of PP and GX despite playing in a high latency setting. Holistically, we can categorize these compensation techniques into two categories: (1) game-internal compensation and (2) game-external compensation. Research in the first category investigated how the adverse effects of latency can be reduced by changing properties or the behavior of the game world. For example, Lee et al. (2019) proposed a method in which the geometrical dimensions of in-game objects are altered to account for the delayed user input. Specifically, the authors reduced the size of obstacles in a game similar to Flappy Bird[1] and found that the latency-induced performance degradation is effectively compensated by their method. Other work, for example, by Gutwin et al. (2004), uses unique game objects to signalize the presence, magnitude, and effects of latency in the game. The authors found that the adverse effects are not as pronounced if the current latency is displayed to the players. The authors argue that the players can manually adapt to the reduced responsiveness by using the latency-indicating game objects and changing their behavior accordingly. However, modifying or adapting the game world is not the only way to minimize latency's effect on PP and GX. Other work investigated how latency can be reduced by reducing the system's overall latency independent of a specific game. Henze et al. (2016), for instance, used artificial neural networks (ANNs) for latency compensation and demonstrated an approach in which they reduced latency by predicting user input on a touch device. By generating this prediction, the supported system does not have to wait for the actual input and can start computing an anticipated output or action earlier. Compared to a system without latency compensation, this creates a temporal advantage for the system utilizing user predictions. Conversely, this reduces the latency perceived by the user.

Even with the combined efforts of game developers and researchers, handling latency in video games is still a challenge. On the contrary, rapid advances in networked technology are bringing paradigm-shifting methods such as cloud gaming to the fore. In cloud gaming, the entire game is rendered on dedicated remote servers and sent to the player as a video stream over the Internet. The player's device, in turn, sends the player's input back to the game server via the Internet. Ultimately, this requirement for communication over the Internet increases the overall latency (Claypool & Finkel, 2014;

[1] https://flappybird.io/

Claypool et al., 2014). Hence, from an HCI perspective, it is vital to understand how latency impacts PP and GX to ensure an optimal level of experience and performance potential for all video game players. Furthermore, utilizing this knowledge to inform the development of effective latency compensation techniques is likewise pivotal. To better our understanding of the effects of latency in video games and to be able to compensate for these negative effects, this thesis's work builds on two pillars:

The first pillar includes investigations on how different aspects of latency influence players to gain a deeper understanding of its effects. Specifically, we study how latency variation, a sudden change in latency, alters performance and experience and how players can adapt to such changes. We hypothesize that changing latency while players are in an ongoing gaming session is detrimental. Knowledge about how changing latency affects GX and PP and how players adapt to it is crucial to inform the behavior of future latency compensation techniques. In the next step, we investigate how players perceive latency and what perceptual channels are more strongly affected by latency. Humans typically rely more strongly on visual information (San Roque et al., 2015), so we hypothesize that auditory latency has less intense effects than visual latency. Then, we investigate how the visual in-game perspective, which reflects how the players perceive the game world visually, alters the effects of latency. Previous work indicated that playing with a first-person view (Claypool & Claypool, 2006), in which the player sees the game world through the eyes of the avatar, increases the effects of latency. Hence, we hypothesize that latency's effects depend on the in-game perspective. The first pillar concludes with an investigation into how the expectancy of latency, i.e., how the knowledge of the latency level while playing, alters its effects. The overarching goal of the first pillar is to inform the design and development of novel latency compensation techniques and how they should behave to provide an ideal performance potential and GX.

The second pillar of this dissertation aims to use the knowledge gained about latency variation and perception to apprise the behavior of novel latency compensation methods. Building on previous work, for example, work by Henze et al. (2016), we use different ANNs to compensate for latency by predicting player behavior in custom and commercial video games. We hypothesize that using ANNs for player prediction compensates effectively for the adverse effects of latency and thus increases PP and GX.

In summary, this thesis derives guidelines on the effects of latency and its compensation for game developers and researchers to alleviate latency-inflicted games and scenarios to a level of optimal experience and performance potential. Additionally, we briefly look beyond this thesis's scope and showcase methods for latency compensation based on ANN and other methods outside of video games. Figure 1.1 provides a structural and conceptual overview of the work presented in this thesis.

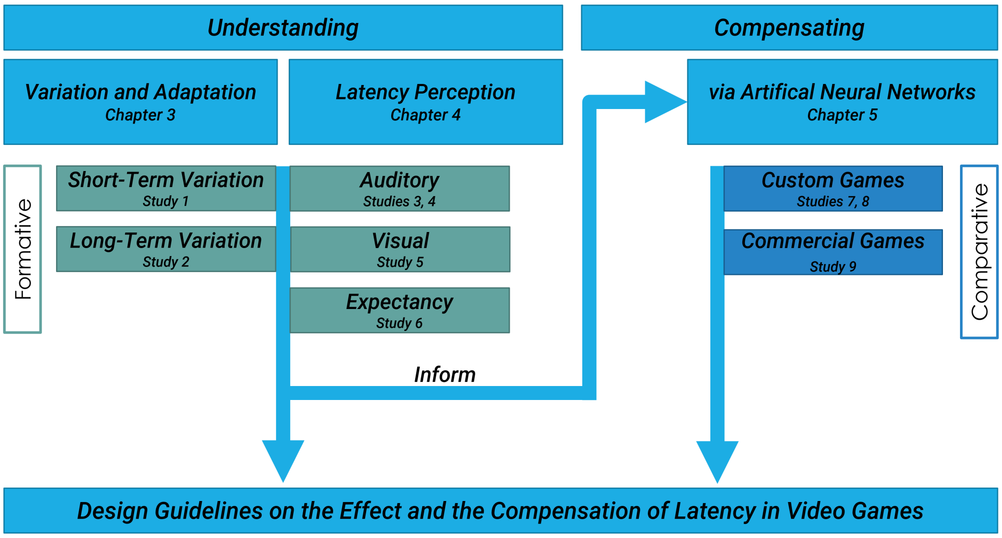


Figure 1.1: Depicts the structure of the work conducted within this thesis. The two pillars, understanding and compensating, each have distinct goals. The first pillar uses formative studies to investigate how different aspects of latency, such as its variation, the perceptual channel it is delivered by, and how the expectancy of latency changes game experience and player performance. The knowledge gained is used in the second part of this thesis, which evaluates different artificial neural network-based latency compensation in comparative experiments. In conclusion, the understanding of latency and its compensation accumulates in actionable design guidelines.

This chapter first clarifies the term latency and demarcates it from similar concepts to create a shared understanding for this thesis. Next, the human-computer feedback loop, the cycle of execution and evaluation (Hutchins et al., 1985), and the MHP (Card et al., 1983, pp. 1-35; Card et al., 1983, pp. 24-44) are introduced to visualize the psychological foundation to why latency affects users and their interaction with computers. Afterward, a brief overview of the background of video games is provided, elucidating their impact on society and research. Then, this section presents the research questions (RQs) of this thesis, the methodology approach for all conducted experiments, and the final synthesized contributions of this work. Lastly, all publications and publications in preparation are listed, and the author's contributions are declared before closing this section with an outline of this dissertation by listing all chapters with a brief summarization.

1.1 Latency

In this section, we define and explain latency to create a shared understanding for the further course of this thesis. We then continue to distinguish between local and network latency since both types of latency may affect video game players differently. This section closes with a brief discussion on what latency levels are acceptable when using interactive systems in general and video games in particular.

1.1.1 Latency Definition

Latency is considered the delay between a user's input and a noticeable output of a system (MacKenzie & Ware, 1993). Wimmer et al. (2019) elaborate that end-to-end latency, the latency that the user ultimately perceives, is mainly formed by three partial latencies: (1) input latency, (2) processing latency, and (3) output latency. Input latency is the delay between a user's input and the conclusive reception at a target system. Processing latency is the delay between the system receiving the input, processing it, and passing it on for display. It includes sub-latencies such as network latency or disk latency. Lastly, output latency consists of the passed time between the finished processing of an input and its actual display to users. These different latencies have different origins. Input and output latency are primarily caused by external equipment, such as keyboards, computer mice, and displays. Processing latency is made up of communication between the in- and output of the target system, as well as the target system's processing performance.

While assessing latency's origins is essential to improve individual bottlenecks on a technical level, the practical and actionable implications and opportunities for action on the player side are limited. In the end, it is not highly relevant to the player if the delay is caused by a slowly updating monitor or computer mouse with a low sampling rate; what matters is the magnitude of latency perceived by the player. However, while this rule of thumb generally holds, one important distinction influences how latency affects video game players: The distinction between local and network latency.

1.1.2 Local and Network Latency

The differentiation between local and network latency is particularly relevant since previous work showed that both types of latency may affect players to varying degrees (Liu et al., 2021d). Local latency is caused by the locally used hardware. This hardware includes but is not limited to the used computer mouse, the keyboard, the monitor, the platform running the game itself, such as a gaming workstation or a gaming console, and any other peripheries used by the players (Wimmer et al., 2019; Ivkovic et al., 2015). Previous work demonstrated that local latency values in the wild can reach values up to 283 ms (Ivkovic et al., 2015).

Network latency, conversely, is caused by the transmission of data packages over some sort of network. While there are different types of network architectures (Smed et al., 2002), the most common technique in video games are so-called server-client architectures (Smed et al., 2002). In a server-client setting, the server assumes an authoritative instance to which the clients connect. The server receives client information, such as player input, calculates reaction to the provided information, and propagates the information to all connected clients. A fundamental role of the server is to synchronize the current game state to all connected clients. The clients in this scenario are the players' devices on which the game is locally running. A server-client architecture is typically deployed in online video games where players play against or with other humans. While server-client settings were, in the past, exclusively used in multiplayer games, nowadays' video games increasingly require players to connect to a server, even to play alone. These games are referred to as always-on video games. Always-on video games are, although often played in single-player mode, also influenced by network latency. The actual network latency perceived by each player in turn is individually built by different components, such as the physical location of the clients and the game server and their distance to each other, the connection speed of the used Internet up-link, or the number of simultaneously connected clients to the game server. As various factors influence network latency, it is difficult to reliably determine its range in ecologically valid settings. For this reason, previous work investigating the effects of network latency has used a range of latency values up to 2000 ms (Claypool & Claypool, 2006).

Network and local latency affect different elements of video games and hence manifest their effects on players differently as well. While local latency affects all processes calculated and rendered locally, for example, the movement of the player's crosshair in an FPS game, network latency affects every action processed by the game server or transmitted to other players in a multiplayer setting. These actions are often crucial game elements, such as hit recognition or synchronization of avatar positions in a shared virtual game world. In a multiplayer setting, network latency can theoretically reach higher values than local latency. However, the impact of local latency on PP and GX seems to be higher. Previous work (Liu et al., 2021d) compared playing with the same level of local and network latency and found that the negative effects of local latency are much more pronounced than those of network latency. Playing with 125 ms of local latency led to a 25 % decrease in PP and GX compared to playing with the same level of network latency. While this work demonstrated that different types of latency affect players differently, it also raises the question of at what value latency starts to affect users of interactive computing systems.

1.1.3 Latency Thresholds in HCI and Video Games

Although a large body of work examines the effects of latency in interactive systems, it is difficult to determine precise thresholds above which latency has a significant negative impact. Early work in the field of HCI, such as the work of MacKenzie and Ware (1993), argue that a latency below 100 ms is optimal. The authors discuss their claim in light of humans' reduced information processing capacities. They conclude that lower latency is unnecessary because the information presented cannot be processed faster by the user. More recent work, however, refuted this hypothesis and showed that users can perceive much lower latency values. Moreover, users cannot only perceive these lower latency values, but previous work also demonstrated that even extremely low latency values negatively alter UX and performance. For example, Jota et al. (2013) and Annett et al. (2014) showed that 25 ms of latency reduces UX and performance in a direct-touch task. In their work, they asked users to perform different direct-touch tasks on a mobile device using a stylus. They found that users had an increased TCT when operating the mobile device with 25 ms of latency. In later work, Ng et al. (2012) even found that although UP seems not to improve below 10 ms of latency, users can distinguish between 1 ms and 2 ms of latency. Summarizing previous work paints a rather vague picture of what latency values are acceptable for users of interactive systems. However, it generally indicates that less latency is better. Nevertheless, there are potential edge cases, for example, in human-robot interaction (HRI), particularly in joint or collaborative action between robots and humans. A too-fast reaction by the robot can induce a feeling of uncertainty and can, in the worst case, even harm the interaction (van der Wel et al., 2021). Although a too-low latency in HRI is not always ideal, these interactions are typically highly specialized and not representative of the classical communication between humans and computers.

In video games, the picture is even more opaque. Driven by the rapid development of the video game market, especially in the last two decades, the latency requirements for video games have also fundamentally evolved. If 18 years ago, a latency of 1000 ms was still tolerable in real-time strategy (RTS) games as proposed by Claypool and Claypool (2006), today, such values are neither realistic nor acceptable for gamers. Complicating matters further is that there is an innumerable variety of game types, all placing different demands on system responsiveness. A fast-paced shooter, where every millisecond counts and potentially decides over virtual life and death, is much more sensitive to latency than a slow-moving turn-based strategy game that thrives on players strategically planning their next moves. Nevertheless, previous work aimed to define the latency sensitivity of video games using different approaches. For example, Claypool and Claypool (2010) proposed a latency framework containing two variables: deadline and precision. Deadline refers to how fast in-game action needs to be conducted by the player. Precision refers to how accurately those actions need to be. Claypool and Claypool argue that the shorter the deadline for in-game actions and the higher the requirement of precision of this action, the more sensitive the game action is to latency. Using this framework, they categorize games into sensitivity brackets based on the game's genre. Later work by Sabet et al. (2020) extends this framework and proposes additional variables that define a game's latency sensitivity, such as temporal and spatial accuracy, the impact of player action, or the urgency of player action. Although these works provide a theoretical foundation for the latency thresholds acceptable in video games, they do not offer a clear-cut latency value for each type of video game. Hence, applied work tested different latency levels and video games in controlled studies. Results from these studies typically do not converge. While they all agree that a too-high latency has negative effects, they do not agree on the actual latency value. Eg et al. (2018), for example, found that latency of 83 ms is optimal for players of FPS games, while Liu et al. (2021; 2021) found that latency starts to degrade GX and PP in the same type of game already at 25 ms. Overall, previous work indicates that games behave similarly to other interactive systems regarding latency sensitivity: Less latency is better.

1.2 The Human-Computer Feedback Loop

In the previous sections, we showed that latency affects the communication between humans and computers. However, to comprehend why latency has negative effects, we need to understand how humans interact with computers. Hence, this section first provides a general understanding of feedback loops to visualize human-computer communication before diving into a theoretical assessment of Hutchins et al.'s (1985) seminal work on the gulfs of execution and evaluation. Lastly, we contextualize latency within Card et al.'s (1983, pp. 1-44) MHP to gain a deep psychological understanding of the effects of latency.

1.2.1 Overview

On a high level, the human-computer feedback loop refers to the iterative and continuous process of interaction and information exchange between humans and computers. On the human side, the feedback loop involves at least two steps: Providing an input of any kind, such as a button press or a text entry, to a computer and observing and interpreting the input results. On the other hand, the computer first receives an input, such as a button press, calculates its effects, and provides the user the input results as output. After receiving the output, the users can start a new cycle by providing input to the computer again. This process applies to every interaction between humans and computers, independent of the task's complexity. Whether a user simply visits a website or performs highly complex mathematical computer simulations – the human-computer feedback loop paradigm persists.

The human-computer feedback loop provides a general framework highly applicable to HCI and various adjacent fields, such as learning environment adaption (Könings et al., 2005), artificial intelligence (AI) (Schank, 1991), or autonomous cars with humans in the loop (Flemisch et al., 2011). Thus, this general framework has been refined, extended, and contextualized from different angles over the last decades. One example is the work on the gulfs of execution and evaluation by Hutchins et al. (1985).

1.2.2 The Cycle of Execution, Evaluation, and Latency

In their work, Hutchins et al. (1985) demonstrate that a successful interaction between humans and computers comprises two crucial phases on the user side: execution and evaluation. The user needs to be able to provide input to the system and they need to be able to execute a task. After the user-initiated execution of the task, the computer generates an output. The output, in turn, is then evaluated by the user. Figure 1.2 depicts this cycle of execution and evaluation.

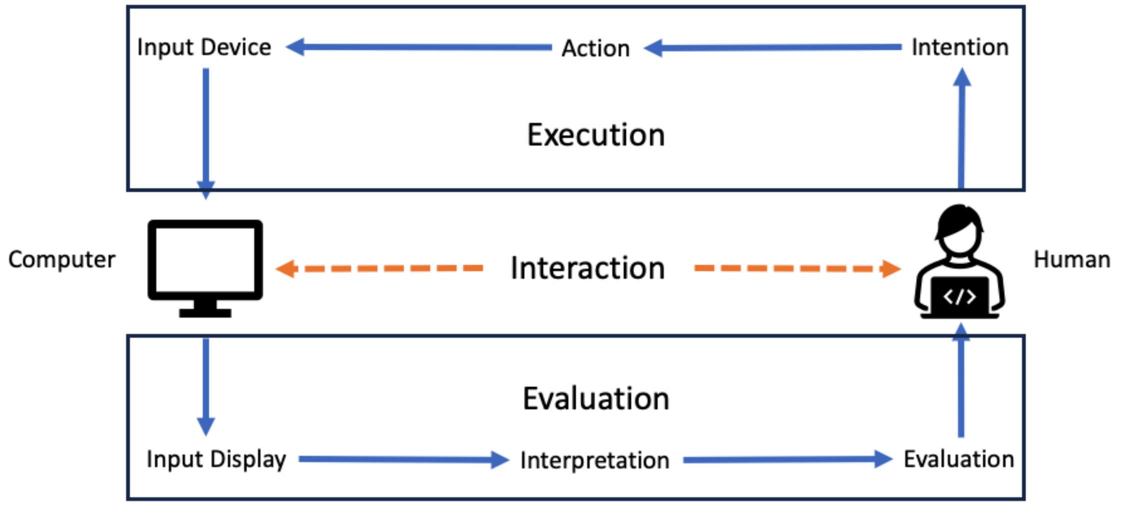


Figure 1.2: Depicts the cycle of execution and evaluation adapted and extended from Hutchins et al.'s (1985) original work, which demonstrates the interaction between humans and computers. A user forms an intention and goal and then performs actions to achieve the goal using provided input devices. The computer displays the results of the input, which has to be interpreted by the user. The user now needs to evaluate if the response fits the original intention and if they achieved their desired goal.

Both phases can be further subdivided into smaller phases. On the execution side, the users first need to identify their intent. They need to think of their goals in the context of the used system. Secondly, the users must determine the appropriate action to achieve their goal. In the last step, users must execute the action using the provided interface, such as pressing a button in a GUI or providing a text entry.

In the evaluation phase, on the other hand, the user receives the output to the provided action. Now, the user has to evaluate if their action produced the desired outcome. Norman (1991) argues that the evaluation phase is similar to the degree to which the system or artifacts provide representations that can be directly perceived and interpreted in terms of user expectations. In other words, the evaluation phase is directly influenced by the difficulty of the user interpreting the output. To provide an accessible way for the user to evaluate the system's output, the feedback should be provided clearly, constantly, and with as little delay as possible, Norman argues (1991).

In principle, latency affects both phases directly and indirectly. In the execution phase, it hinders the last step and the users' action execution. On the other hand, latency also increases the time required to display the result of an execution. This increased delay between input and output increases the difficulty of the evaluation since users cannot directly perceive the impact of their actions. Hence, inspecting the communication between humans and computers using the cycle of execution and evaluation, it becomes evident why latency affects UX and UP.

The cycle of execution and evaluation provides an actionable understanding of why and how latency affects the interaction between humans and computers. However, to further deepen this understanding, we can refer to fundamental work investigating the user's cognitive process when interacting with computers from a psychological angle, such as Card's MHP (1983, pp. 1-44).

1.2.3 The Model Human Processor

The MHP, proposed by Card, Newell & Simon (1983, pp. 24ff), is a theoretical framework that describes the human-computer feedback loop in terms of a series of information processing stages. On the user side, it consists of three main components: perception, cognition, and motor response.

At the heart of the feedback loop is the perception stage. When humans interact with computers, they receive output through visual, auditory, or haptic feedback provided by the computer's interface. The human perceives this output and then interprets it, forming a mental representation of the information. This stage involves the human's ability to sense and understand the computer's output. Once the human has perceived the computer's output, they enter the cognition stage. Cognition encompasses higher-level mental processes such as thinking, decision-making, and problem-solving. In this stage, humans engage in cognitive processing to analyze and interpret the information received from the computer. They may compare the output to their mental model or goals, make inferences, and form judgments. The cognition stage is crucial in how humans understand and derive meaning from the computer's output. Following cognition, the human enters the motor response stage. Here, the actions taken by the human in response to their cognitive processing come into play. They provide input to the computer through various modalities such as keyboards, computer mice, touchscreens, or voice commands. The human's actions are intended to modify the input to the computer, conveying their intentions or desired outcomes. The motor response stage bridges the cognitive processes of the human with the physical interaction required to communicate with the computer effectively.

Simultaneously, the computer processes the input received from the human. It follows its programming and algorithms to execute specific operations, perform calculations, access databases, or run software applications. The computer's processing stage transforms the human's input into meaningful output, which becomes the basis for the feedback provided to the human. The feedback stage involves the computer generating output based on the input received from the human. This output can be visual displays, text, sounds, or any other modality through which the computer communicates with the human. The output serves as new input for the human, initiating the next iteration of the feedback loop.

Latency affects all stages of the MHP. It affects how humans perceive and interpret the computer's output. A noticeable delay between an action performed by the user and the resulting feedback from the computer disrupts the perception stage. This delay can lead to a disjointed UX, as the expected real-time feedback is not aligned with the user's actions. Latency also influences cognitive processing. A delay in receiving feedback from the computer affects the human's ability to make immediate judgments or decisions based on that feedback. The longer the latency, the more challenging it becomes for users to maintain a coherent mental model of the task, potentially hindering their ability to accurately interpret the information and make informed decisions. Latency influences the motor response stage as well. A delay between the user's input and the computer's response can introduce a sense of disconnectedness or frustration. Users may need to wait for the computer to catch up before proceeding with their intended actions. This delay can affect the flow and efficiency of interaction, potentially slowing down task completion and reducing user satisfaction. Lastly, latency in generating feedback from the computer can also disrupt the feedback stage. If the computer takes significant time to generate output based on the user's input, it impedes the seamless progression of the feedback loop. Users may have to wait for the computer's response, leading to a slower and less dynamic interaction.

Overall, latency introduces a sense of sluggishness, disrupts the natural flow of interaction, and affects the perception, cognition, motor response, and feedback stages in the MHP, increasing the gulf between evaluation and execution. Minimizing latency and striving for real-time responsiveness in computer systems is paramount to create a smooth and efficient interaction, enabling users to effectively engage with computers and accomplish their goals in a seamless manner.

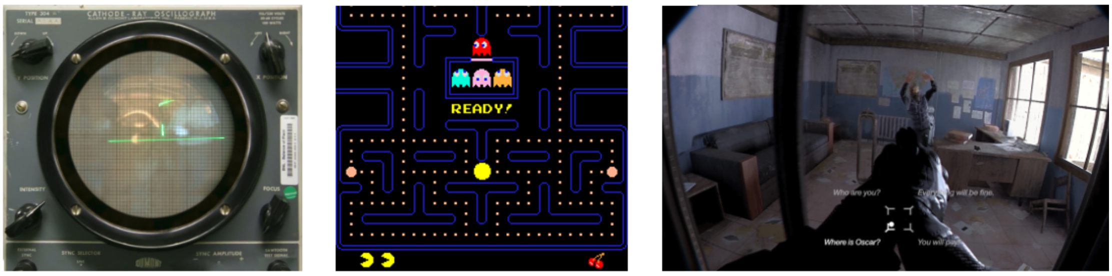
1.3 A brief History of Video Games

Since video games are interactive systems, the interaction between players and games, the player-game interaction (PGI), is also represented by the MHP and Hutchins et al.'s (1985) gulfs of execution and evaluation. Hence, the effects of latency follow the same systematization as in non-video game interactive systems. However, since video games typically require more timely interaction than, for example, writing an email, they are particularly affected by latency. Video games have evolved significantly since their inception to become a ubiquitous and influential medium in contemporary society. This subsection traces their evolution from the early days of simple electronic games to the complex and immersive experiences of today to contextualize them within this thesis.

The roots of video games can be traced back to the early 1950s when scientists began experimenting with electronic computers. In 1958, physicist William Higinbotham created Tennis for Two, a rudimentary tennis simulation displayed on an oscilloscope (Kent, 2010, p. 27). However, it was not until the 1970s that video games gained widespread recognition with the advent of arcade games and home consoles. The release of Pong by Atari in 1972 marked a turning point in the history of video games, as Kent argues (2010, pp. 116 – 141). This simple tennis-like game gained immense popularity and launched the arcade gaming phenomenon. It was followed by a wave of arcade hits such as Space Invaders (1978) and Pac-Man (1980) (Kent, 2010, pp. 287 – 323), which became cultural phenomena and helped to establish the video game industry. In the mid-1970s, home consoles such as the Atari 2600 and Intellivision brought video games into people's living rooms. This led to the explosion of the home video game industry and the creation of iconic games such as Super Mario Bros (1985) or The Legend of Zelda (1986) (Kent, 2010).

The 1990s saw significant advances in video game technology. The release of the Super Nintendo Entertainment System (SNES) and the Sega Genesis brought 16-bit gaming to the forefront. This era also ushered in the transition from 2D to 3D games with the introduction of consoles such as the Sony PlayStation (1994) (Wikipedia, 2023f) and the Nintendo 64 (1996) (Wikipedia, 2023d). Games such as Super Mario 64 (1996) and Tomb Raider (1996) redefined the industry with their immersive 3D worlds. With the widespread use of the Internet, online gaming became increasingly popular. The release of massively multiplayer online role-playing games (MMORPG), such as Ultima Online (1997) and EverQuest (1999), allowed players to connect and interact in persistent virtual worlds. The 2000s also witnessed the rise of competitive e-sports, with games such as Counter-Strike (2000) and StarCraft II (2010) laying the groundwork for professional gaming (Fish, 2021).

Today's video game market is permeated by countless large development studios and indie developers, generating bigger revenue than the music and film industry combined (Statista, 2023b). The history of video games showcases the rapid evolution of the medium. From the humble beginnings of Tennis for Two, over the first widespread success of arcade games such as Pac-Man, to the highly sophisticated and diverse landscape of today with hyper-realistic video games such as Unrecorded (see Figure 1.3). Video games have become a powerful tool for entertainment, storytelling, and social interaction, with implications that extend beyond the realm of gaming itself. To advance the development of games, game developers, publishers, and researchers strive to optimize the gaming experience for players. Achieving this goal entails minimizing factors, such as latency, that negatively influence gameplay and games. Previous work showed that latency causes players to score fewer points, take longer to complete in-game tasks, or be unable to complete a given task at all (for example, Eg et al., 2018; Beigbeder et al., 2004; Claypool & Finkel, 2014). In addition, latency also greatly reduces the perceived gaming experience of players (for example, Durnez et al., 2021; Liu et al., 2023).



a) Tennis for Two, 1958 b) Pac-Man, 1980 c) Unrecord, 2023

Figure 1.3: Shows the evolution of video games. Starting with Higinbotham's Tennis for Two in 1953 (image source: Brookhaven National Laboratory (2013)), over arcade gaming success Pac-Man in 1980 (image source: Wikipedia (2023e)), to today's photo-realistic games as Unrecord (image source: Studios Drama (2022)).

1.4 Research Questions

To achieve the overarching goal of this thesis of deepening our understanding of latency and to use this knowledge to develop novel latency compensation techniques, we identify several RQs. These RQs are categorized into three topics (c.f. Figure 1.1 and Table 1.1) and guide through this dissertation. In the following, we outline the RQs followed through this work.

Topic	Research Question	Section
VA	RQ1: Does small-term latency variation affect player performance and game experience?	3.1
	RQ2: Does long-term latency variation affect player performance and game experience?	3.2
LP	RQ3: Does auditory latency affect player performance and game experience?	4.1, 4.2
	RQ4: How does the visual in-game perspective alter the effects of latency?	4.3
	RQ5: How does the expectation of latency alter player performance and game experience?	4.4
ANN	RQ6: Can artificial neural networks be used to compensate for the negative effects of latency in custom video games?	5.1
	RQ7: Can artificial neural networks be used to compensate for the negative effects of latency in commercial slow-paced video games?	5.2
	RQ8: Can artificial neural networks be used to adaptively compensate for the negative effects of latency in commercial fast-paced video games	5.3

Table 1.1: Summary of research questions (RQs) of this thesis. RQs are categorized into three topics; Variation and adaptation (VA), latency perception (LP), and compensation via artificial neural networks (ANN). See Figure 1.1 for an overview of how the topics related to each other.

Since the overall perceived latency by players in interactive systems is built by different sub-latency, such as processing latency, network latency, or input latency, it never really is constant (Casiez et al., 2017). In ecologically valid gaming settings, it rather varies between different values. However, previous work investigating the effects of latency in video games often treated it as a constant (Liu et al., 2021d; Beigbeder et al., 2004; Beigbeder et al., 2004). Hence, it is unclear how (and if) latency variation affects PP and GX in video games. Furthermore, understanding how players deal with latency variation is also paramount in light of latency compensation since compensation systems may suddenly change the current level of latency. Building on this, we formulate our two first RQs as follows:

RQ1: *Does small-term latency variation affect player performance and game experience?*

RQ2: *Does long-term latency variation affect player performance and game experience?*

Similarly, previous work investigated how different types of latency alter PP and GX. Although audio elements in games fundamentally shape a player's performance and experience (Grimshaw et al., 2008; Gormanley, 2013), surprisingly, the effects of auditory latency in video games have not been investigated. Hence, we formulate our third RQ as follows:

RQ3: *Does auditory latency affect player performance and game experience?*

In the same vein, previous work researched how different game characteristics, such as its pacing (Sabet et al., 2019b), its rules (Claypool & Claypool, 2010), or its in-game task complexity (Schmidt et al., 2017) alter the effects of latency. However, while a crucial part of every game, it is unclear how the in-game perspective (how players perceive the game world) alters the effects of latency. The in-game perspective is a crucial aspect of the game and defines how the player interacts with the game. Hence, to investigate the interaction between the in-game perspective and latency, we formulate our fourth RQ as follows:

RQ4: *How does the in-game perspective alter the effects of latency?*

Previous work also highlighted that how players perceive latency, or how it is communicated via the game interface, may fundamentally alter their experience (Kosch et al., 2022; Michalco et al., 2015). However, it is unclear if this effect is purely technical, i.e., a real effect of latency or an expectation-based effect such as a Placebo (Beecher, 1955; Arnstein et al., 2011) or Nocebo (Colloca & Barsky, 2020) effect. Hence, to investigate how the expectation of latency alters PP and GX, we formulate our fifth RQ as follows:

RQ5: *How does the expectation of latency alter player performance and game experience?*

Building on the knowledge gained from answering the previous RQ, we design novel latency compensation methods to counteract the negative influence of latency. Previous work showcased several different methods to account for latency directly in games, such as time warp (K. Lee & C. Chang, 2017; Bernier, 2001), geometrical manipulation (Lee et al., 2019), or dead reckoning (Pantel & Wolf, 2002). However, most of these methods have disadvantages, such as being specific for one game or genre (Pantel & Wolf, 2002), being perceived as unfair (K. Lee & C. Chang, 2017; Bernier, 2001), or potentially producing in-game inconsistency (Liu et al., 2022). Thus, previous work also investigated game-external latency compensation techniques, such as optimizing network (Sun, 2019) or memory usage (Cheng et al., 2010). One particular promising technique is to use an ANN for user input prediction (Le et al., 2017). Previous work showed remarkable success in predicting user input and subsequently in reducing the perceived latency (Henze et al., 2016; Le et al., 2017). However, although ANNs have shown promising results, they have not been used in video games to compensate for the negative effects of latency. Hence, we formulate three RQs to investigate the use of ANN-based latency compensation in video games. For each RQ, we gradually increase the complexity of the problem. While the first RQ aims to answer if full access to game-internal states allows an ANN to compensate for latency, the second RQ already increases the prediction complexity since it limits the ANN to use only game-external information of a slow-paced video game. The last RQ accumulates all previous work, in which we investigate if ANNs can be used to adaptatively compensate for latency in a fast-paced shooting game while only using information readily available to the player. Hence, we formulate the last three RQs of this thesis as follows:

RQ6: *Can artificial neural networks be used to compensate for the negative effects of latency in a custom video game?*

RQ7: *Can artificial neural networks be used to compensate for the negative effects of latency in a commercial slow-paced video game?*

RQ8: *Can artificial neural networks be used to adaptively compensate for the negative effects of latency in a fast-paced commercial video game?*

Ultimately, this thesis accumulates in actionable design guidelines on how to compensate for latency in video games for researchers and developers.

1.5 Methodological Approach and Contributions

The research presented in this thesis was conducted based on well-established research methodologies in HCI. In each of the presented subsections, and hence for each of the presented RQs (c.f., 1.1), we aim to identify and understand one specific problem from a user's perspective. We built this problem statement on a thorough literature review. After establishing the RQ, we brainstorm potential ideas and methods to answer the postulated question. These ideas result in hard- or software prototypes, which are evaluated in either controlled laboratory studies or in-the-wild studies with high ecological validity. In this thesis, we mainly collected quantitative data, which allows us to answer our RQs in a frequentist analysis. Nevertheless, we also collect qualitative data to gain in-depth insight into the effects of latency and our latency compensation prototypes. After collecting data in the user studies, we analyze the data using well-established statistical tests and qualitative methods.

This thesis makes several contributions to video game and latency research and the design of latency compensation methods. We categorize these contributions using Wobbrock's and Kientz's (2016) framework for research contributions in HCI. The authors propose seven research contribution types in HCI, of which this dissertation provides five: empirical, artifact, dataset, methodological, and theoretical contribution.

First and foremost, this dissertation provides empirical contributions from nine user studies conducted over three years. The presented work demonstrates that video game players are mostly unaffected by short-term latency variations (RQ1) but that PP and GX are negatively influenced if latency varies on a long-term basis (RQ2). Hence, we learned that a short latency fluctuation is less negatively influential to PP and GX than exposure to high latency for a longer time. These results indicate that latency compensation techniques should always try to compensate for latency. A small error in the compensation (leading to a small-term fluctuation) is not as harmful as playing with high latency to provide a consistent and stable gaming experience.

We also investigated how players perceive latency and found that auditory latency generally does not affect PP and GX but may be relevant if players are highly experienced (RQ3). Furthermore, we highlight that the adverse effects of latency in video games are independent of the used in-game perspective (RQ4): Previous work revealed contrary findings and argued that the in-game, at least partly, perspective dictates the game's latency sensitivity (Claypool & Claypool, 2010). While auditory perception and in-game visual perspective in video games do not change the effects of latency, we found that how latency is presented in the game is crucial. Our work shows that the mere expectation of latency, for example, displayed in a video game, influences PP and GX (RQ5).

Based on the understanding and knowledge gained about latency variation and perception, we built different methods to compensate for latency, which also contributes to the empirical findings of this thesis. We show that ANNs can be used to compensate for the adverse effects of latency in custom video games (RQ6) and in slow-paced commercial video games (RQ7). In our last study, we even show that ANNs can be used to adaptively compensate for latency in a fast-paced commercial FPS game (RQ8). Summarizing the contribution of our investigation in latency compensation, we show that ANNs predicting player behavior can counteract the reduced responsiveness induced by latency. Players using the developed systems achieved better scores, better accuracy, and an increased gaming experience overall.

Second, this thesis also provides artifact and dataset contributions. For the research in this thesis, we designed and developed different video games. The source code for all video games is obtainable in the accompanying Open Science Framework (OSF) repository [1]. Furthermore, we used our games or commercial video games to collect data from 1 113 gaming sessions of 617 players. Other researchers can use the collected data and the developed games to investigate further RQs or to shed new light on existing problems. Additionally, we developed several ANNs for latency compensation. The final and trained models can also be found in the repository.

Third, this thesis creates new knowledge on how to carry out latency research (methodological contribution). We show that the mere suggestion of latency – the expectation of latency – can bias participants. The study setting fundamentally alters participants' behaviors, performances, and experiences. In our work, we show that if participants know they are playing with latency, this knowledge changes the study's outcome. We, therefore, argue that future latency research should be careful in briefing participants beforehand about the fact that they are playing video games with latency or latency compensation methods to prevent a bias induced by the participants' expectations.

Lastly, this work bears theoretical contributions in the form of design guidelines for future latency researchers and video game developers. In this thesis, we learned about how latency variation affects players, how different perceptual channels are affected by

[1] https://osf.io/e5rvc/?view_only=457caeddb7b7471b9c0007f1b4ce50bb

latency, how latency needs to be communicated to players, and how to compensate for the adverse effects of latency effectively. All this understanding and knowledge accumulates a theoretical foundation for actionable design guidelines.

1.6 Publications and Work Distribution

The main impetus for this thesis's ideas, data, and findings originate from scientific publications, publications in submission, or manuscripts in preparation, which are referenced below. Certain sections of previously published work are incorporated into this thesis. The author's contributions to each publication or unpublished manuscript are acknowledged using the CRediT taxonomy[1].

- **To Lag or Not to Lag: Understanding and Compensating Latency in Video Games (Halbhuber, 2022)**

  The main ideas of this thesis are outlined in this publication. The author wrote the article, published in the *Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY'22)*. Parts of Chapter 1 are based on this publication.

  Personal Contribution: Conceptualization, project administration, and manuscript writing.

- **The Effects of Latency in Video Games: A Systematic Review and Thematic Analysis (Halbhuber et al., n.d.(a))**

  The author collected and analyzed the data, initiated and supervised the project, analyzed the data, and wrote the manuscript with valuable input from Niels Henze and Christian Wolff. The manuscript is currently in preparation. Section 2.2 is based on this manuscript.

  Personal Contribution: Conceptualization, study supervision, data collection and curation, formal analysis, methodology design, project administration, supervision, validation, visualization, and manuscript writing.

- **Small Latency Variations do not Affect Player Performance in First-Person Shooters (Schmid et al., 2023)**

[1] https://niso.org/standards-committees/credit

This publication is based on the Master thesis by Thomas Fischer. The student created the hard- and software prototypes and collected the data in a study. The author initiated and supervised the project. The author analyzed the data. The article was written by the author in cooperation with Andreas Schmid, with valuable input from Niels Henze and Raphael Wimmer. Section 3.1 is based on this publication, which was published in the *Proceedings of the 2023 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY'23)*.

Personal Contribution: Conceptualization, data curation, formal analysis, contribution to methodology, project administration, supervision, validation, visualization, and writing the manuscript.

- **Don't Break my Flow: Effects of Switching Latency in Shooting Video Games (Halbhuber et al., 2022e)**

  The author created the hard- and software prototypes and collected the data. The author analyzed the data and wrote the article with valuable input from Niels Henze and Valentin Schwind. The article was published in the *Proceedings of the 2022 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY'22)*. Section 3.2 is based on this publication.

  Personal Contribution: Conceptualization, data collection, study supervision, data curation, formal analysis, methodology design, project administration, supervision, validation, visualization, and manuscript writing.

- **Understanding Player Performance and Gaming Experience while Playing a First-Person Shooter with Auditory Latency (Halbhuber et al., 2022b)**

  This publication is based on the Bachelor Thesis by Maximilian Huber. The software prototype was developed by the author and adapted by the student. The student collected the data. The author initiated, supervised the project, analyzed the data, and wrote the article with valuable feedback by Niels Henze and Valentin Schwind, which was published in the *Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY'22)*. Parts of Section 4.1 are based on this article.

  Personal Contribution: Conceptualization, data collection, study supervision, data curation, formal analysis, methodology design, project administration, supervision, validation, visualization, and manuscript writing.

- **The Effects of Auditory Latency on Experienced First-Person Shooter Players (Halbhuber et al., 2022c)**

  This publication is based on project work by Annika Köhler, Markus Schmidbauer, and Jannik Wiese. The students adapted the provided hard- and software prototypes and collected the data. The author initiated and supervised the project. The author analyzed the data and wrote the article with valuable input from Niels Henze. The article was published in the *Proceedings of Mensch und Computer 2022 (MuC'22)*. Parts of Section 4.2 are based on this article.

  Personal Contribution: Conceptualization, study supervision, data curation, formal analysis, methodology design, project administration, supervision, validation, visualization, and manuscript writing.

- **The Effects of Latency and In-Game Perspective on Player Performance and Game Experience (Halbhuber et al., 2023b)**

  This publication is based on the Master thesis by Philipp Schauhuber. The student developed the software prototype and collected the data. The author initiated and supervised the project. The author analyzed the data and wrote the manuscript with valuable input from Niels Henze and Valentin Schwind. Section 4.3 is based on this publication, which was published in the *Proceedings of the 2023 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY'23)*.

  Personal Contribution: Conceptualization, study supervision, data curation, formal analysis, methodology design, project administration, supervision, validation, visualization, and manuscript writing.

- **Better be quiet about it! The Effects of Phantom Latency on Experienced First-Person Shooter Players (Halbhuber et al., 2022d)**

  This publication received a Best Paper award and is based on the Bachelor thesis by Maximilian Schlenczek. The student adapted a software prototype and collected the data. The author initiated and supervised the project. The author analyzed the data and wrote the manuscript with valuable input from Niels Henze and Johanna Bogon. The article was published in the *Proceedings of the 21st International Conference on Mobile and Ubiquitous Multimedia (MUM'22)*. Section 4.4 is based on this article.

Personal Contribution: Conceptualization, study supervision, data curation, formal analysis, methodology design, project administration, supervision, validation, visualization, and manuscript writing.

- **Increasing Player Performance and Gaming Experience in High Latency Setups (Halbhuber et al., 2021)**

  This publication is based on the author's Master's thesis. The author developed the software prototype, collected the data, and wrote the article. Niels Henze and Valentin Schwind provided valuable input for writing the article. The article was published in the *Proceedings of the 2021 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY'21)*. Section 5.1 is based on this article.

  Personal Contribution: Conceptualization, data collection, study supervision, data curation, formal analysis, methodology design, project administration, supervision, validation, visualization, and manuscript writing.

- **Using Artificial Neural Networks to Compensate Negative Effects of Latency in Commercial Real-Time Strategy Games (Halbhuber et al., 2022f)**

  This publication is based on project work by Maximilian Seewald, Fabian Schiller, and Mathias Götz. The students adapted the provided hard- and software prototypes and collected the data. The author initiated and supervised the project. The author analyzed the data and wrote the article with valuable input from Niels Henze and Jakob Fehle. The article was published in the *Proceedings of Mensch und Computer 2022 (MuC'22)*. Section 5.2 is based on this article.

  Personal Contribution: Conceptualization, study supervision, data curation, formal analysis, methodology design, project administration, supervision, validation, visualization, and manuscript writing.

- **Leveling the Playing Field: Adaptive Latency Compensation in First-Person Shooters using Artificial Neural Networks (Halbhuber et al., n.d.(b))**

  This publication in preparation is based on the Master thesis by Maximillian Seewald. The student developed soft- and hardware prototypes and collected data. The author initiated and supervised the project. The author analyzed the data and wrote the manuscript with valuable input from Niels Henze and Valentin Schwind. The manuscript is currently in preparation. Section 5.3 is based on this manuscript.

Personal Contribution: Conceptualization, study supervision, data curation, formal analysis, methodology design, project administration, supervision, validation, visualization, and manuscript writing.

- The author of this thesis was also involved in further publications that are beyond the scope of this dissertation. These publication address topics such as emotions in video games (Halbhuber et al., 2019), sentiment analysis in movies (Schmidt et al., 2021b), dataset generation for video games (Halbhuber et al., 2022a), a workshop contribution on negative latency (Henze et al., 2023), and time and time perception in HCI (Riemer et al., 2023; Bogon & Halbhuber, 2023; Halbhuber et al., 2023c). Furthermore, the author was involved in articles investigating the effects of latency in full-body motion-tracked VR (Halbhuber et al., 2023a).

1.7 Thesis Outline

This thesis presents the results and evaluations of nine user studies, a literature review on the effects of latency in HCI, a systematic literature review following the PRISMA protocol on the effects of latency in video games, and a thorough analysis of work investigating latency compensation. Furthermore, we discuss and summarize our findings and derive implications through design guidelines. This thesis closes with a conclusion and an outlook into future research directions. The dissertation is structured as follows:

Chapter 1 - Introduction provides a brief background and elucidates the research motivation before highlighting the RQs, presenting the research methodology, and the author's contributions.

Chapter 2 - Background and Related Work summarizes previous work and presents fundamentals on how latency affects users in interactive systems, how latency affects players in video game players, and how previous work aimed to compensate for it. This chapter ends with a summary which motivates and contextualizes this thesis's RQs.

Chapter 3 - Understanding Variation and Adaptation presents two studies investigating how small-term and long-term latency variations influence PP and GX in video games.

Chapter 4 - Understanding Latency Perception describes and evaluates the results of three studies investigating if auditory latency alters player behavior, how the in-game perspective alters the effects of latency, and how latency expectations modulate its effect.

Chapter 5 - Compensating Latency via Artificial Neural Networks applies the understanding from previous chapters to design, develop and evaluate novel latency compensation methods based on ANNs for custom video game, slow-paced commercial video games, and fast-paced commercial video games. The developed compensation techniques are tested and evaluated in three separate user studies.

Chapter 6 - Conclusion summarizes the previous chapters and derives implications and actionable design guidelines. This chapter also highlights the limitations of this dissertation and potential directions for further research.

This chapter is partly based on to the following publication that introduces the ideas of this thesis:

Halbhuber, D. (2022). "To Lag or Not to Lag: Understanding and Compensating Latency in Video Games." In: *Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '22. Bremen, Germany: Association for Computing Machinery, pp. 370–373. ISBN: 9781450392112. DOI: 10.1145/3505270.3558364.

2

Background and Related Work

Research on latency and its effects on humans has a long tradition in HCI. Previous work such as Miller's "*Response time in man-computer conversational transactions*" (1968) or Sheridan and Ferrell's "*Remote Manipulative Control with Transmission Delay*" (1963) laid the foundation of today's latency research. These cornerstones of latency research have been refined, extended, and revisited for the last thirty years. Hence, this chapter presents an overview of foundational research investigating latency in different contexts. We start by elucidating latency and its effect on interactive systems in general before continuing to analyze its impact on video games. After profoundly understanding latency's effects in video games, we highlight different latency compensation methods. Finally, we summarize and synthesize all findings in the last part of this chapter.

Since the focus of the presented topics – latency in interactive systems, video games, and latency compensation methods – vary in direction, we report the search strategy for each topic in the corresponding section.

2.1 Latency in Interactive Systems

Every interactive system is affected by latency. These systems can take different forms, including personal computers running desktop or web-based applications, mobile phones, VR or AR systems, and smart devices. Typically, interactive systems incorporate a wide range of input and output modalities such as text, graphics, images, audio, touch, gestures, or voice commands. Crucially, they all allow users to accomplish a specific goal, such as opening a website, by providing input and receiving the system's output. Based on the output, the users can, in an iterative manner, in real-time, and interactively refine the input to the system to accomplish even complex tasks. The development of interactive systems dates back to 1963 and Ivan Sutherland's seminal and influential PhD thesis on the Sketchpad (Sutherland, 1964; Booth, 2004[1], p.5). The Sketchpad was groundbreaking at the time since it allowed users to create drawings and interact with graphical objects using a light pen and a display screen. It introduced interactivity by enabling users to continuously manipulate the drawings using the light pen, whose output was detected by the screen. However, this interactivity also emphasizes the negative effects of latency, as users now continuously interact with systems in open input-output loops.

2.1.1 Search Strategy

We conducted a literature review to learn how latency affects interactive systems and their users. We queried relevant research database and search engines (ACM Digital Library[2], Google Scholar[3], IEEE Xplore[4], BASE[5]) using the following query (coded as pseudo search term): (("Interactive System" OR "Computer" OR "Machine") AND ("Latency" OR "System Response Time" OR "Delay" OR "Lag") AND ("Effects" OR "Experience" OR "Performance"). We excluded articles that focus only on technical aspects of latency without accounting for its influence on users directly, such as optimizing random-access memory (RAM) latency (Lee et al., 2013). We also excluded medical and biological publications that use the term latency more broadly to describe a temporal offset between subsequent events, for example, sleep latency (Higuchi et al., 2005). To round out the results of the search term searches across databases, we performed a bottom-up approach in which we manually selected highly relevant and influential papers investigating latency.

2.1.2 Results of the Review on Latency in Interactive Systems

In the following, we present the results of the review on latency in interactive systems.

[1]Reprint of Sutherland's work.
[2]https://dl.acm.org/
[3]https://scholar.google.de/
[4]https://ieeexplore.ieee.org/Xplore/home.jsp
[5]https://www.base-search.net/

2.1.2.1 The Effect of Latency on User Performance

In their seminal work, MacKenzie and Ware (1993) investigated how latency alters movement times and error rates in a target acquisition task. Target acquisition is a fundamental operation when interacting with computer systems, it involves moving an input representation, such as a mouse cursor, from a defined starting position to a defined target position, such as a desktop icon. The Fitts' law paradigm describes moving and selecting targets on a computer screen, which MacKenzie and Ware build their work on (Fitts, 1954). Fitts' law classifies the difficulty of target acquisition tasks in so-called bits. The initially proposed metric is formulated as follows:

$$ID = log_2(\frac{2D}{W})$$

In the formula, ID describes the index of difficulty, D the distance to the target, and W the width of the target. However, Mackenzie and Ware used a slightly adapted variation of Fitts' Law called the Shannon formulation (1993), which is called this way because of its resemblance with the seminal Shannon-Hartly theorem of information theory (Shannon, 1948). The Shannon formulation of Fitts' Law is formulated as follows:

$$ID = log_2(\frac{A}{W} + 1)$$

Again, ID describes the index of difficulty, A is the distance or amplitude to the target, and W is the width or size of the target. Using the Shannon formulation, MacKenzie and Ware were able to systematically design six target acquisition tasks with increasing difficulty. Next, the authors tested eight participants in a study utilizing the designed tasks. In the study, the authors introduced four levels of latency (ranging between 8.3 ms and 225 ms). The participants' goal in each task was to select a fixed number of lateral targets on a computer screen using a mouse. After each target selection, the mouse cursor was repositioned to the center of the screen to ensure comparability between trials. Using this method, MacKenzie and Ware found that 225 ms latency increased movement time by 63.9 % and error rate by 214 % compared to 8.3 ms of latency. However, the authors did not find a significant difference between other latency levels, although a general trend of performance degradation with increasing latency was recognizable. Furthermore, the authors discuss that latency's adverse effects go beyond the mere increase of movement time, as shown by the increased error rate. Most importantly, the authors also highlight that task complexity or difficulty, as measured by the Shannon formulation, influences the task's latency sensitivity. A more difficult task is more strongly affected by latency than a less arduous task.

While MacKenzie and Ware's work builds the foundation for today's research on the effects of latency in interactive systems, the authors themselves grounded their investigation in earlier fundamental work in Human Factors such as the work by Sheridan and Ferrell (1963). In their work, Sheridan and Ferrell present one of the first investigations into how feedback delay (latency) alters the user's performance in a direct manipulation task. To achieve this, the authors build a custom two-parted apparatus. The first part recorded the user's hand motion using a sled attached to a rail. The rail is equipped with several potentiometers that can track the sled's position on the rail. This information is transmitted to the second part of the apparatus – the receiver. The receiver gets the movement information from the user's hand motion and mimics it on its rail and sled using a set of servo motors. Using a combination of magneto-conductive tapes, Sheridan and Ferrell induced a specific and controlled delay between sender and receiver. Figure 2.1 shows a schematic diagram of the apparatus. In an experiment, Sheridan and Ferrell used the apparatus to test four levels of latency (0.0 s, 1.0 s, 2.1 s, and 3.2 s) in a simple task. The user's task was manipulating the sender's sled to align the receiver's sled with a small box. The task replicates a remote grasping task. Sheridan and Ferrell motivated the task by drawing parallels between the task and exploring and maintaining equipment in space, farming and mining the ocean, and carrying out experiments in noxious chemical and radioactive environments. To measure the user's performance, they obtained the average TCT and correlated it with the modification of Fitts' law's index of difficulty. Overall, with their work, the authors show that the introduced latency increases the amount of open-loop movements by the user and their TCT. This means that users more often needed to wait to close a feedback loop (to receive the feedback to their action) when interacting with the apparatus and delay. Consequently, this also increased the required time for an interaction.

More recent work investigating latency reinforces and deepens early findings on the effects of latency in HCI. Friston et al. (2016), for example, researched the impact of small latency on standardized Fitts' Law and Steering Law (Accot & Zhai, 1997) tasks. The authors tested six levels of latency: (1) 6 ms, (2) 16 ms, (3) 26 ms (4) 36 ms, (5) 56 ms, and (6) 86 ms. Contrary to earlier work, Friston et al. were able to test minimal latency values (6 ms) using a specialized low latency apparatus. In their experiment,

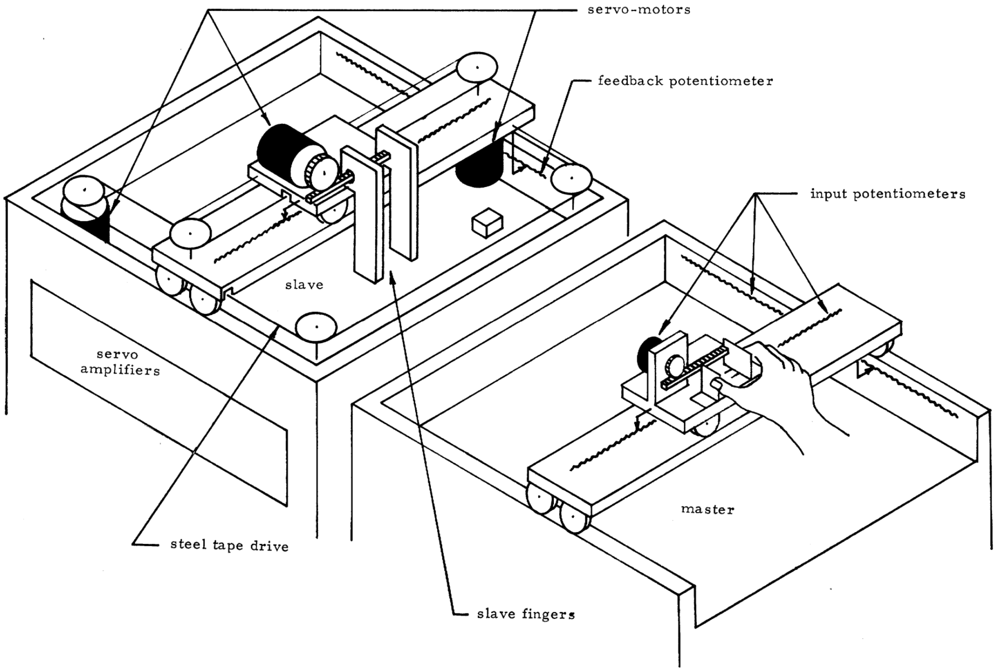


Figure 2.1: Schematic diagram of Sheridan and Feller's minimal manipulator apparatus. Figure adapted from original work by Sheridan and Feller (1963). User could manipulate the sled's movement on the sender side, this motion than is transferred to the receiver, which mimics the movement. Using this apparatus Sheridan and Feller showed that latency, a delay between sender and receiver, negatively influences task completion time.

the authors asked participants to perform several tasks ajar the Fitts' Law or Steering Law paradigm, with several levels of latency induced. For the investigation, the authors seated the participants in front of a classical workstation operated with a keyboard and computer mouse. Overall, the authors found that latency begins affecting UP starting at 16 ms. The authors found no effect of the lowest level of latency on UP. Crucially, the authors found that the impact of latency is non-linear, which shows that a higher level has a more severe influence. Furthermore, the authors argue that not every motion of an interaction via the computer mouse is equally affected by latency. Hence, they divided the classical target-acquisition task into three phases: (1) acceleration, (2) acquisition, and (3) correction. The authors' results highlight that the correction phase is most strongly affected by latency. This is directly relatable to previous work, for example, Sheridan and Ferrell, since correction of one's own action requires timely feedback by the system and timely loop closing to allow the user to adjust. Latency prevents this. Overall, the authors argue that the effects of latency on performance are multi-faceted, and measuring in a single variable, such as time completion time, may be misleading. Other work by Pavlovych and Stuerzlinger (2009), as well as the work by Thether et al. (2009), comes to a similar conclusion.

Previous work (Pavlovych & Stuerzlinger, 2009; Teather et al., 2009; Sheridan & Ferrell, 1963; MacKenzie & Ware, 1993) consistently shows that latency has a negative effect on UP in various different tasks. However, the work by Friston et al. (2016) raises the important question at which level or threshold latency's negative effect starts to be perceived by users.

2.1.2.2 The Perceivable Threshold for Latency

Previous work aimed to answer the question at what thresholds latency is perceived and potentially influential to an interaction. One hundred milliseconds have long been regarded as the minimum latency that is necessary to provide a direct interaction that is not influenced by latency (MacKenzie et al., 1991). Overall, this work demonstrates that the thresholds for latency are strongly dependent on the input modality and task (MacKenzie et al., 1991). However, more recent work by Jota et al. (2013), for example, investigates at what values latency produces a just noticeable difference (JND) (Stern & Johnson, 2010) in touch interactions and comes to different thresholds. To achieve this, the authors built a specialized apparatus implementing touch interaction. In two experiments, the authors instructed participants to use the apparatus to perform a touch-dragging task. The task was based on the ISO 9241-9 one-direction tapping task (now ISO 9241-400:2007 (International Organization for Standardization, 2007)), and they asked participants to drag an object from one on-screen position to another by moving their finger over the apparatus's surface. Participants were asked to place their finger on a cursor, drag the cursor across a path to the target, and release it once they felt the cursor was positioned correctly. Using this setup, Jota et al. artificially added latency to the experiments. Latency was varied between 2 ms (the practical minimum obtainable using the authors' system) and 120 ms. The authors found that the JND in touch interaction is, on average, at 64 ms. However, simultaneously, the authors also show that although latency reliably is detected at 64 ms by most users, their performance can start to deteriorate even earlier at about 40 ms of latency. Hence, performance decreases, although users do not explicitly notice latency. The authors argue that 20 ms of latency is the threshold that ensures that latency does not influence performance and users do not perceive it.

While the work by Jota et al. (2013) fundamentally shows that latency perception and its effect on the users' performances are decoupled, it also highlights latency's dependence on input modality and task. Other work investigating latency thresholds comes to different – although similar – conclusions about when latency starts affecting UP. Annett et al. (2014) investigated in their work the input quality of digital touch styli. Although they did not quantify the effects of latency of the stylus tested in their experiment, they obtained qualitative data from participants using the stylus with two different levels of latency. In their qualitative analysis, they found that participants appreciated a more direct manipulation, were concerned with the delayed interaction in general, and preferred a more natural feel. Building on Annett et al.'s work, Ng. et al. (2014) conducted a study to quantify the effects of latency on stylus-based touch interaction. To do so, they built a high-performance stylus system with a theoretical latency of 1 ms. In the study, they followed a similar approach as Jota et al. (2013) and asked participants to perform several one-direction dragging tasks. The participant's goal was to select one object from one location and move it to another location using the provided stylus. Building on previous work, in the experiment, the authors controlled for a continuous level of latency (Ng & Dietz, 2013). Overall, the authors found that users in the study were able to discriminate between 1 ms and 2 ms while dragging with the stylus. However, when using the stylus to scribble, the authors found that participants could only distinguish between 7 ms and 40 ms of latency. In conclusion, the authors consolidate previous findings regarding latency but extend them by showing that even extremely low latency differences of 1 ms are discernible by users.

Reviewing previous work investigating acceptable latency thresholds paints a diverse picture. A clear-cut recommendation for a latency value using a specific input modality and a particular task is difficult. Thus, to better contextualize the large body of work, previous work also aimed to summarize this work and provide systematic latency guidelines. In their work, Attig et al. (2017) combined previous works investigating latency thresholds in an attempt to unify them. The authors show that earlier work by Miller (1968), Sheiderman and Plaisant (2010), Card et al. (1983), and even more recent work by Seow (2008) argue that the acceptable threshold of latency lays between 50 ms and 200 ms of latency for the interaction to be perceived as instantaneous. While some of the work is based on expert estimations (Miller, 1968) or empirical data (Shneiderman & Plaisant, 2010), others are based on generalized psychophysical experiments and defend their estimation with the perceptual limitations of humans (Card et al., 1983). More recent work, including Attig et al.'s review (2017), comes to less conservative thresholds. Kaaresoja (2010) argues, based on empirical experiments, that the threshold depends on the perceptual channel the humans perceive the feedback with. Hence, the authors state that a visual latency is acceptable between 30 and 85 ms, auditory latency is permissible between 20 - 70 ms, and lastly, tactile feedback (such as input vibration) needs to be between 5 ms and 50 ms to not affect UP.

In summary, defining latency thresholds for the wide variety of input modalities available in HCI is difficult. Previous work offers various references and possibilities, which allow us to form a general assumption about acceptable latency thresholds. However, one crucial aspect of latency that is not directly investigated in previous work is the influence of latency variability. Although the presented investigations often highlight that latency is variable – often called jitter in that work – none of the work so far directly accounts for latency variability in their experiment.

2.1.2.3 On the Variability of Latency

Since an interaction's latency, the time between user input and system response is comprised of several partial latency types (Casiez et al., 2017; Casiez et al., 2015) it is never really a constant. When an input event is triggered, for example, by physically closing the contacts of a mouse button, an event is transferred from the input device to the computer via USB. However, the input device itself contributes to end-to-end latency as it takes time to scan and de-bounce buttons and since USB polling rates are limited (Wimmer et al., 2019). The input event is registered by the operating system's kernel and passed on to the user space, where input callbacks of application toolkits are triggered. Task scheduling, high system load caused by background applications, as well as input handling of the application toolkit, can delay this process (Casiez et al., 2015) and vary the overall perceived latency by the user.

In network applications, such as multiplayer games, events also have to be transferred to a server that sends back a response. Depending on the type of connection, bandwidth, and physical distance to the server, network round-trip times can have a significant impact on latency and its variability.

Most applications update their state in a loop and re-paint regions if necessary. The latency added by re-painting depends on the used graphics toolkit or game engine, as well as the complexity of the rendered content (Schmid & Wimmer, 2023). This step is highly resource-intensive for complex applications, such as modern video games, because of computationally intensive calculations and high-resolution representation. Once an image is rendered to a frame buffer, it is sent to a monitor and displayed to the user. In addition to the time it takes to transfer an image to the monitor (cf. display response time (Stadler et al., 2020; Dossena & Trentini, 2022)), the monitor's refresh rate also contributes to the overall latency. A monitor continuously updates its content at a fixed rate.

Overall, actual latency in the wild, due to all these individual factors influencing it, is highly volatile and can vary even between magnitudes. However, most of the previous work assumed a constant latency in their experiments. Although it is possible that variable latency, as it occurs in real-world scenarios, has a different effect on the user's perception and performance. Only a small number of studies investigate variable latency in controlled environments. Davis et al. (2010), for example, investigated the effects of fixed and variable latency on driving performance and mental load in a driving simulator study. They first gathered baseline data in a pre-study where participants drove without added latency. For their main study, Davis et al. compared high constant latency (700 ms) to varying latency (400 – 1100 ms, mean: 700 ms). Latency variation was generated with a sinusoidal function. In both latency conditions, participants performed significantly worse than the baseline regarding driving performance. With varying latency, lane offset was significantly higher than with constant latency. Latency had no main effect on average velocity, task load, and motion sickness. However, it is worth noting that Davis et al. investigated latency values that are far beyond the proposed guidelines in the previous section and are even higher in comparison to a real-world setting.

2.1.3 Summary

Previous work consistently demonstrates that latency affects UP in a variety of input modalities, such as using a mouse and keyboard on a classical workstation (MacKenzie et al., 1991), using a stylus on a touch-sensitive graphic pad (Jota et al., 2013) or even using touch interaction on a smartphone (Ng et al., 2014), in a variety of task. Most work argues that the latency-induced performance degradation is caused by a lack of control-feedback (MacKenzie & Ware, 1993). Users require fast and responsive feedback to perform ideally. While there is a large body of work investigating latency, a general conclusion at what level latency starts to affect user interaction is not obtainable. Different results postulate different thresholds based on different evaluation methods such as expert's assumptions (Miller, 1968), empirical data (Shneiderman & Plaisant, 2010), or human perceptual limits (Card et al., 1983). Nevertheless, it is safe to assume, based on the presented work, that lower is better. One crucial aspect that is often not directly investigated in previous work is latency's variability. Since the overall perceived latency is comprised of several sub-latency, such as input latency, processing latency, and transmission latency (Casiez et al., 2017; Casiez et al., 2015), it is never really a constant. Currently, it is unclear if latency variability does indeed affect user interaction more strongly than constant latency.

2.2 Latency in Video Games

The previously presented work consistently showed that latency affects users and their communication with interactive systems. As they are interactive systems, video games are also affected by latency. Hence, we conducted a systematic literature review and a thematic analysis to deepen our knowledge about the effects of latency in video games. Fundamentally, this review aims to answer three questions: (1) What effects on PP and GX does previous research unravel? (2) How can we systematically categorize these effects? And (3) what characteristics define a game's latency sensitivity? Fundamentally, answering these questions allows us also to extrapolate gaps, missing knowledge, and yet investigated phenomena regarding latency. This is crucial to prevent designing latency compensation systems that oversee certain aspects of latency that otherwise substantially alter experience and performance.

This section will be part of the following manuscript in preparation for submission:

Halbhuber, D., Wolff, C., & Henze, N. (n.d.[a]). "The Effects of Latency in Video Games: A Systematic Review and Thematic Analysis." In: *Currently in preparation*.

2.2.1 Method

We designed, conducted, and report this review according to a review protocol developed to comply with the specifications of the PRISMA 2020 Statement (Page et al., 2021) and the PRISMA-S checklist (Rethlefsen et al., 2021). PRISMA provides rigorous and widely-used reporting guidelines for systematic reviews. The protocol originated in the health field but is now commonly used in various other domains, including HCI (MacArthur et al., 2021) and PGI (Formosa et al., 2022). The PRISMA protocol streamlines the systematic review process for researchers by identifying critical focus areas. According to the PRISMA 2020 statement and its guidelines, we have utilized the recommended headings in the following section, excluding non-applicable sections such as meta-analysis. The reporting of methods used in the analysis follows afterward.

2.2.1.1 PRISMA-P Checklist

In the following, we report the required PRISMA-P checklist (Page et al., 2021; Rethlefsen et al., 2021) to our study.

Eligibility Criteria

The following inclusion and exclusion criteria were applied to the retrieved studies in this work:

Inclusion criteria:

- Peer-reviewed (including conference papers, review articles, journal articles, letters, and editorials in press)
- Full-Papers (including full conference papers) and extended abstracts
- Published in 1990 or later
- Discusses the effects of latency in video games

Exclusion criteria:

- Books, theses, and unpublished material
- Publications not written in English
- Grey literature (newspaper, magazine articles, non-peer-reviewed conference papers, and reports.)
- Published before 1990
- Does not discuss the effects of latency in video games

Database	Number of articles identified
ACM Digital Library	539
Google Scholar	522
IEEE Xplore	997
BASE	644
Semantic Scholar	347
Manual bottom-up search	20
Total	**3069** (or 2812 excluding duplicates)

Table 2.1: Number of papers retrieved from each information source.

2.2.1.2 Information Sources

We utilized several scientific electronic databases and search engines as information sources. Additionally, we performed a manual bottom-up approach, starting at papers that are either highly cited or known to us to be highly relevant (i.e., they directly investigate or discuss the effects of latency on video game players). We conducted the initial searches using the electronic databases between 11th November and 18th November 2022. Furthermore, we conducted an updated search between 19th February to 25th February, 2023 and between 1st September to 12th September, 2023. Table 2.1 shows the number of articles retrieved by information sources.

Search Strategy

We developed and tested different search terms across the presented information sources to gather all relevant articles meeting our inclusion criteria. We started developing the search terms by reviewing highly pertinent articles that investigate and discuss the effects of latency on video game players. This step allowed us to formulate different commonly used identifiers for latency (for example, "lag", "delay, or "system response time") and for video games in general (such as "gaming", "video game", or "digital game"). These initial search terms were tested and refined over multiple searches conducted for all information sources. We used the following search terms (coded as a pseudo search term) for all databases: (("Game" OR "Gaming" OR "Video Game" OR "Digital Game") AND ("Latency" OR "Lag" OR "Delay" OR "System Response Time" OR "Jitter" OR "SRT")).

Selection Process

We screened all papers (n = 3049 + 20 manual search) retrieved utilizing the presented search terms by title and abstract for relevance. One author conducted an initial screening. Next, all relevant papers (n = 113) received a full-text assessment for eligibility by one author. After the first round, the involved authors discussed the results, and the final set of papers was established. We used no automation tools throughout this process. All articles were manually retrieved and assessed. Lastly, after the full-text assessment for eligibility, we established a final set of 67 papers eligible for our systematic review.

Data Collection Process

We analyzed data from 67 articles and extracted the data to a specifically designed spreadsheet. We did not contact the authors of the reviewed articles for further information since all data required is provided by either the article itself or the information source. All data was collected and extracted by one author. In case of uncertainty about the systematization of the effects of latency, all authors discussed and resolved them.

2.2.1.3 Data Items

We extracted the following data items from each article: author/s, year of publication, citation count, journal or conference, the discussed effects of latency on video game players, and, if applicable, other relevant information related to latency and video games.

Quality Assessment

We did not deploy a dedicated quality assessment procedure since the study design of the surveyed articles is not the focus of our review.

Synthesis Method

We did not perform a meta-analysis of the extracted information since the final set of papers also includes qualitative work not focused on empirical findings. Thus, our work does not include the statistical necessities to calculate effect sizes or statistical power values across articles. However, we synthesize our findings by discussing a systematization approach to the effects of latency on video game players.

2.2.1.4 Thematic Analysis

Since the main goal of our review is to provide a theoretically informed systematization of the effects of latency on video game players, we used a semi-reflexive inductive thematic analysis to generate themes categorizing the effects of latency (Braun & Clarke, 2021; Braun & Clarke, 2006). Hence, all presented categories are coded using thematic analysis. In our work, we followed the widely used (Formosa et al., 2022; Adams et al., 2008; Cooper et al., 2022; Barrera & Shah, 2023; MacArthur et al., 2021) guidelines to thematic analysis established by Braun and Clarke (2021), which structures a thematic analysis in six, actionable steps: (1) familiarizing yourself with your data, (2) generating initial codes, (3) searching for themes, (4) reviewing themes, (5) defining and naming themes, and (6) producing the report.

After the initial screening of the title and abstract, all remaining articles received a full-text assessment by one author. We used an inductive approach to openly develop the first set of codes categorizing the effects of latency. During the code and theme development, we collaboratively reflected on the established codes to refine and revise them. Lastly, we established the final set of codes and themes in a synchronous discussion. We used MAXQDA as dedicated coding software for the thematic analysis; codes and themes were produced using MAXQDA's coding function.

2.2.2 Results

In the following, we describe the results of our systematic literature review. We first explore the study's selection and characteristics before we continue to elucidate on the conducted thematic analysis. Lastly, we discuss our findings in the context of the studies presented RQs and elaborate on this review's limitations.

2.2.2.1 Study Selection

Three-thousand and forty-nine (3 049) papers fit the initial queries, and an additional 20 papers were manually identified as relevant in a bottom-up approach (see Figure 2.2). This initial baseline of papers decreased to 113 papers after we removed papers written in a language other than English (n = 1), irrelevant papers (n = 2 698), and duplicates (n = 277 papers) based on a screening of the abstracts and titles of the papers. One hundred thirteen (113) papers received a full-text assessment for eligibility. Via the full-text assessment, we removed papers not matching our inclusion criteria for publication type

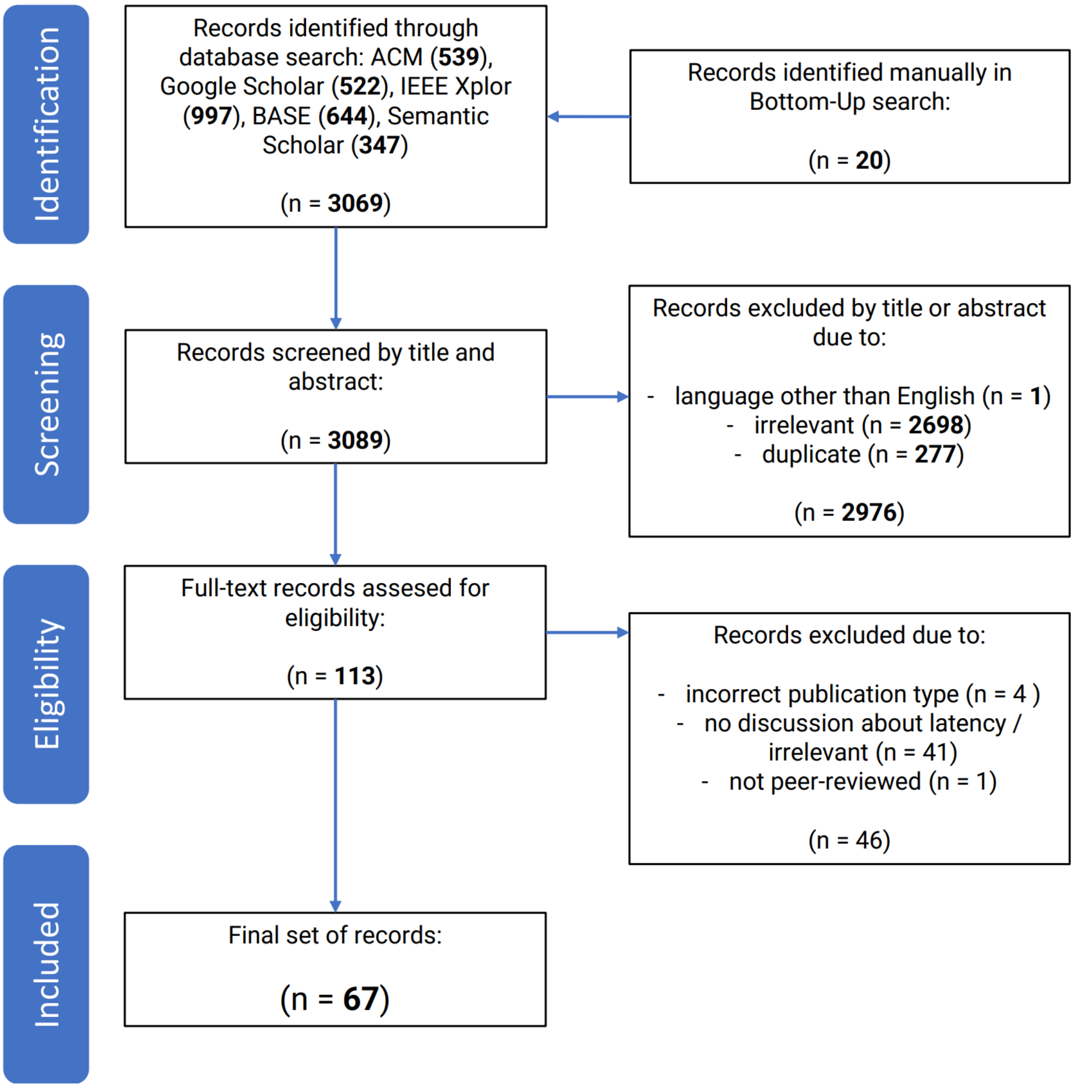


Figure 2.2: Prisma flow chart of the systematic literature review conducted within the context of this thesis.

(n = 4), papers not discussing the effects of latency in video games (n = 41), and papers that were not processed through a rigorous peer-review process (n = 1). A total of 67 papers were deemed eligible for inclusion in our review.

2.2.2.2 Study Characteristics

There were 67 eligible articles (articles that discussed the effects of latency in video games). Table 6.1 and Table 6.2 depict an overview of all articles included in the review. The majority of these articles focus on video games played on a classical local workstation (60/67, 90 %), and the remainder focuses on cloud-based gaming (6/67, 9 %) and VR games (1/67, 1 %). 93 % (62/67) of the articles motivate their investigation in a multiplayer scenario, while only five articles (7 %) explicitly investigate a local single-player gaming session. Only eleven papers (16 %) provide a reference about what latency players encounter in real-world gaming sessions. However, 22 articles (33 %) discuss what latency is acceptable in video games before it affects players. Latency values tested in the respective studies range from 0 ms (Wahab et al., 2021) up to 1000 ms (Claypool & Claypool, 2010).

The majority of included articles (46/67, 68 %) were published in the last ten years (2013 - 2023), the remaining papers were published before 2013 (21/67, 32 %), while the oldest article was published in 1999 (Vaghi et al., 1999). Unsurprisingly, the number of publications investigating latency and its effects increased ten years ago and remained somewhat constant over the last decade. This shows that latency in video games, even today, remains an open problem. Only three articles (4 %) had no citation yet, the remaining articles were cited 72.5 times (*SD* = 122.3 times), with the highest cited article being cited 708 times (Claypool & Claypool, 2006). Figure 2.3 depicts the number of publications (left) by publication year included in this review. The Figure also shows the articles' citation counts by year of publication (left).

2.2.2.3 Thematic Analysis

The findings about the effects of latency in video games from all 67 articles were coded using thematic analysis. The final dimensions identified indicate the main effects of latency. Overall, two overarching themes emerged: PP and GX. Through our thematic analysis, we further delineated four dimensions within PP: Task completion time, accuracy, target tracking and selection, and score. Analogously, GX has five dimensions: Unfairness and fairness, flow, immersion, agency and loss of control and, enjoyment and QoE. It is important to note that a considerable number of papers (for example, (Liu et al., 2021d; Liu et al., 2021c)), use single questions to assess GX. While this approach is valid in the context of the respective work, it does not allow for an in-depth analysis

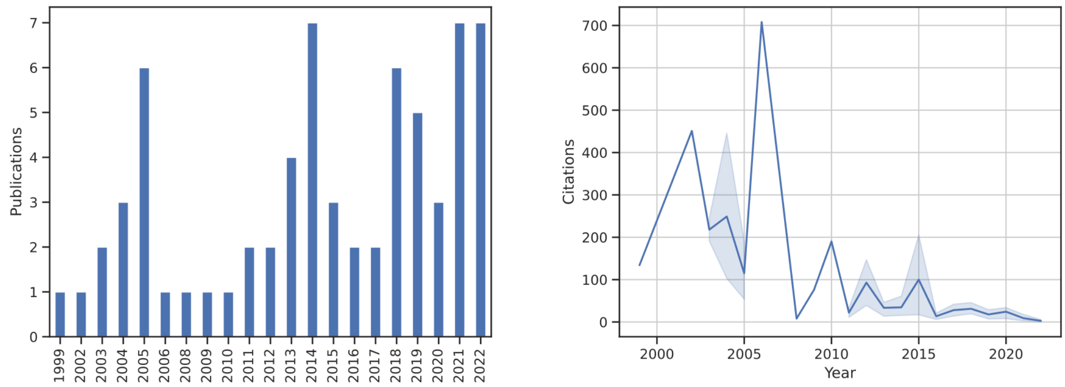


Figure 2.3: Shows number of publications (left) and number of citations by year for all articles included in the systematic literature review.

nor a targeted coding of latency's effects within the scope of this review. Thus, articles that use single questions to assess GX are unified in the enjoyment and QoE dimension. Furthermore, since some effects of latency are only discussed once, for example, the effects of latency on the navigational ability of players, we also report effects that did not fit into one of the proposed categories in a separate category (other effects).

2.2.2.4 Effects of Latency on Player Performance

In the following, we discuss our findings about the effects of latency on PP in video games. We structure this discussion by the previously established dimension of our thematic analysis.

Task Completion Time

A large body of work, for example, Beznosyk et al. (2011) and Claypool et al. (2020), measure and discuss the effects of latency in video games on TCT. In HCI, TCT refers to the duration it takes for a user to complete a specific task or activity using a specified interface or input modality. This metric is often used to evaluate the efficiency and usability of interfaces and interaction methods (MacKenzie, 2013). Shorter TCTs generally indicate a more efficient and user-friendly design, while longer times indicate that the user faced difficulties interacting in this setting. Specifically in gaming, TCT refers to the time a player needs to accomplish a specific objective, mission, or task within the game. These tasks can either be atomic, such as selecting a single element in a game UI, or contain a chain of interaction, such as commanding multiple units in an RTS.

The effects of latency on TCT are not surprising in light of classical HCI models such as Card et al.'s (1983) MHP. Latency increases the time between consecutive control loops, making it harder for players to concatenate input-output loops coherently. Since players have to wait longer for feedback, producing the next input takes them longer. This effect, however, goes beyond just the current latency. Playing with 100 ms of latency does not mean that each input-output pair is delayed by 100 ms. Instead, the offset, induced by latency, can grow even bigger, making it continuously harder for players to seemingly interact with the game, as discussed by Beznosyk (2011). Although it is not clear from the research how much latency increases TCT, it is clear that it does. Players in high latency settings take longer to complete in-game tasks or, in the worst case, cannot complete them at all (Beigbeder et al., 2004).

Accuracy

The negative impact of latency on the player's accuracy is one of the most discussed consequences of latency on PP. More than half of the reviewed articles (37/67), for example, Normoyle et al. (2014) or Lee and Chang (2018), discuss the negative effects of latency on how accurately players are aiming in video games.

While it is safe to say that part of the accuracy degradation is caused by a lack of control-feedback, similar to the increase of TCT – players move their computer mice to aim for a target, however, caused by latency, the confirmation of the movement is delay – there are other causes discussed in prior work as well. One such cause is the lack of temporal predictability induced by latency. In fast-paced games, timing is crucial. Latency can cause inconsistency in the timing of actions, making it challenging for players to predict and react to in-game events accurately. This unpredictability can lead to missed opportunities and decreased accuracy (Long & Gutwin, 2018).

Previous work comes to different conclusions on how much and at what thresholds latency affects accuracy. For example, Quax et al. (2004), found that 60 ms significantly reduces the accuracy of players playing fast-paced FPS games compared to 20 ms and 40 ms of latency. Liu et al. (2021c) showed that an increase from 25 ms to 125 ms can entail a loss of accuracy by 25 %. Claypool et al. (2019) investigated the effects of latency between 50 ms and 300 ms of latency. They revealed a clear degradation of accuracy with increasing latency. The authors argue that turn frequency and turn angle of moving targets can alter aiming accuracy in video games as well, which results in an altered TCT (with the task being to shoot the target).

Target Tracking and Selection

Target tracking and selection are fundamental gameplay elements in various video game genres, such as FPSs, role-playing games (RPGs), and strategy games. Target tracking involves the process of monitoring and following specific in-game objects or characters, which can be enemies, non-player character (NPC), or even virtual items. This tracking can be automatic, where the game system assists the player by automatically locking onto nearby targets, or manual, requiring players to manually select or lock onto targets, as seen in FPS games. Additionally, games often provide feedback to players, such as health bars or visual indicators, to help them keep track of their chosen target. On the other hand, target selection involves choosing a particular in-game object or character as the primary focus of the player's attention and actions. This selection can be crucial in combat-oriented games, where players strategically decide whom to attack based on threat level or vulnerability, or in RPGs and adventure games, where players choose which NPC to interact with, what object to use, or which quest to pursue. Target tracking and selection varies across games, input devices such as keyboards, computer mice, gamepads, and in-game mechanics.

Twenty-one articles in our reviewing body (21/67) discuss the negative effects of latency on target tracking and selection in video games. These articles discuss latency's effects in light of accuracy or TCT. It is evident that if latency affects the players' accuracy and TCTs, the overachieving goal, for example, selecting an NPC character to interact with, is also negatively influenced. Claypool et al. (2019) further discuss that the latency's negative effects on target selection depend on player skill, with more skilled players being less affected by latency than less skilled players. Lee et al. (2019) argue that moving-target selection is one of the most representative tasks in real-time games. The authors illustrate an example in which a player has a certain time frame available to react to a game action, for example, while a moving target passes over a certain area of the player's view. Players need to anticipate the correct time for the input; latency, the authors argue, makes this anticipation more difficult.

Score

Almost all articles discussed within the context of this review (46/67), such as Ries et al. (2008), Liu et al. (2021a) or Sabet et al. (2020b), highlight negative effects of latency on the players' scores. In video games, a score is a numerical representation of a player's performance and achievements. It plays a crucial role in quantifying a player's success and progress, with its meaning and purpose varying based on the game's genre and mechanics. Scores are typically earned by completing specific objectives, missions, or tasks, such as collecting items in platforming games or achieving fast lap times in racing games. Additionally, they can reflect a player's accuracy and precision in shooting or targeting-based games, where higher accuracy translates to a higher score. Time-based scoring is another common approach, where players earn points for completing tasks swiftly and efficiently. Some games incorporate combo or multiplier systems to reward players for consecutive successes, and scores often fuel competition among players through leaderboards, motivating players to strive for higher scores and greater achievements.

Latency decreases the game score by decreasing the players' accuracy, TCT, or selection and tracking ability. Since the score is a holistic measure of the overall game performance, it is influenced by all other negative effects discussed so far. Similar to previous effects, it is hard to quantify the effects of latency on the players' scores. While some work, such as Liu et al. (2021c), found that a 20 % higher latency decreases scores by 20 %, others found a 50 % reduction (Quax et al., 2004). In the worst case, some work even reports that a high latency leads to players being unable to complete certain in-game tasks (such as shooting a target), which also prevents them from earning points. Subsequently, a too-high latency can even reduce scores by 100 %.

2.2.2.5 Effects of Latency on Game Experience

Latency not only affects the objective PP but also has negative consequences on the subjective gaming experience. In the following, we discuss our findings about the effects of latency on GX in video games. We structure this discussion by the previously established dimension of our thematic analysis. However, since GX is an elusive and inherently subjective concept, categorizing it is much more difficult than categorizing the effects of latency on PP. This circumstance is further exacerbated because GX is not uniformly surveyed. Some researchers use evaluated and standardized questionnaires. Other researchers use qualitative methods such as post-experience surveys, and still other works reduce GX to single questions, such as "How much did you like this game round?".

Unfairness and Fairness

Twenty-six articles (26/67) in our review discuss the concept of (un-)fairness, such as the work by Lee and Chang (2017), Sabet et al. (2020b), or Zander et al. (2005). Fairness in video games refers to the concept that the game's rules and mechanics should create an environment where all players have an equal and reasonable opportunity to succeed or win and where outcomes are determined primarily by skill, strategy, and decision-making rather than by external factors or unfair advantages. For example, Ravindran et al. (2008) describe fairness in video games by a construct they call cohesiveness. They argue that a player's perceived consistency is directly tied to the state of the game the player observes at a given time. Absolut cohesiveness means that all players (in a multiplayer setting) see the same copy of the game world simultaneously. This would lead to cohesiveness among players, as every player knows that the game enforces the same state for every other player. However, latency disrupts this perfect synchronicity. It can create an uneven playing field, where players with lower latency may have a competitive advantage over those with higher latency (Zander et al., 2005). When a player's actions register more quickly due to lower latency, they can react faster to in-game events, giving them an edge in competitive situations (Schmidt et al., 2021a).

Overall, latency induces a feeling of unfairness – the thought of being disadvantaged compared to other players or the game itself – and, hence, reduces the subjective gaming experience.

Flow

Almost a third of the reviewed articles (21/67) discuss the effect of latency on the feeling of flow, such as the work by Durnez et al. (2021) and Carlson et al. (2021). Flow, a construct first introduced by Csikszentmihalyi (1990), in the context of video games refers to a state of deep immersion and engagement that players experience when they are fully absorbed in a game. It's a mental state where players become so engrossed in the gameplay that they lose track of time, forget their surroundings, and feel a sense of euphoria and satisfaction. Flow is often characterized by a combination of different elements such as (1) clear goals, which entails players having a clear objective and goals to achieve within the game, providing a sense of purpose and direction, (2) immediate feedback, which means that the game provides constant and immediate feedback to the player's action, allowing them adjust their strategies and decisions accordingly, and (3) effortless action, which means players feel a sense of effortless and automatic action, where they instinctively know what to do without conscious thought. In his original publication, Csikszentmihalyi discusses even more elements of an optimal flow state, such as a balanced challenge, a high concentration level, a loss of time, and intrinsic motivation.

Latency disrupts the flow experience since it delays feedback loops, breaks concentration, leads to unpredictable gameplay and even reduces the sense of control. For example, Durnez et al. (2021) found that players of a 3D exergame reported a significantly lower flow rating when playing with 500 ms than 175 ms of latency. Other work reported similar effects (Ries et al., 2008; Schmidt et al., 2021a).

Immersion

Sixteen articles of the reviewed corpus (16/67) discuss how latency alters the player's subjective feeling of immersion. For example, Beyer and Möller (2014) found a slight decrease in perceived immersion with rising latency. Claypool (2018) discusses that a player's feeling of immersion partly modulates latency's effects, while Sabet et al. (2022) argues that fast-paced, immersive, and temporal-sensitive VR experiences are strongly impacted by latency.

Immersion in the context of video games refers to the degree to which a player feels deeply involved, engaged, and absorbed in the virtual world and the ongoing experience. It is a sense of "being there" within the game, where the real world fades into the background, and the player feels a strong connection to the game environment and its narrative. Increasing immersion is often considered a key goal in game design because it enhances the overall enjoyment and gaming experience (Michailidis et al., 2018; Cairns et al., 2014; Sanders & Cairns, 2010).

Latency can affect immersion in several ways. High latency can result in noticeable delays between a player's actions and the corresponding reactions in the game world. This delay can break the sense of realism, making it feel like the game world is not responding naturally to the player's inputs. This can disrupt the immersion as players become aware of the technical limitations of the game. Furthermore, latency can impact the fluidity and responsiveness of player interactions with the game world. When actions feel sluggish or unresponsive because of latency, players may become frustrated and less engaged, diminishing their immersion. Since immersion often relies on players willingly suspending their disbelief and accepting the game world as real. High latency can break this suspension of disbelief when players notice delays or inconsistencies in the game's responses, making it more difficult to fully immerse themselves in the virtual world.

Agency and Loss of Control

Even though only four articles in our review directly discuss the concept of agency or a feeling of losing control in light of latency, it is a recurring theme in the broader context of literature investigating latency's effect in interactive systems, video games, and even latency compensation methods. Hence, since it is such a ubiquitous concept, it is necessary to discuss it.

In the context of video games and interactive media, agency refers to the degree of control, choice, and influence a player or user has within the virtual environment or interactive experience (Winkler et al., 2020). Individuals feel a sense of empowerment and autonomy when they can make meaningful decisions and see the consequences of their actions reflected in the game world. These reflections of actions are delayed by latency.

While our perceptual system can discern differences between a motor action (such as mouse button click) and the visual response (such as the firing of a weapon), well below 100 ms (Raaen & Eg, 2015; Keetels & Vroomen, 2012; Rohde & Ernst, 2013), Eg et al. (2018) argue that simply perceiving a delay not necessarily infers with the interaction. However, if too much time passes, for example, between clicking the left mouse button to fire a virtual weapon and the visual response of this action, players lose the sense of being the agent of the response (Farrer et al., 2013). It is unclear how much time between action and response negatively affects the agency. While some work investigated values as low as 50 ms (Caserman et al., 2019) and found no effect, other work reported negative effects of latency on agency between 125 ms (Waltemate et al., 2016) and 700 ms (Ebert & Wegner, 2010) of latency. Wahab et al. (2021) investigated latency values between 0 ms and 300 ms and found that participants reported a loss of control in certain latency conditions. However, the authors do not further elaborate on these findings. Durnez et al. (2021) used the feeling of control or agency only as a proxy for flow. However, as reported above, they did find significant negative effects of latency on flow. Subsequently, a negative effect of agency and the feeling of control can be assumed.

Enjoyment and Quality of Experience

While previous paragraphs on the effects of GX, such as immersion, flow, or agency, often are subtle proxies to a player's experience, previous work also followed a more direct approach in simply asking players how much they enjoyed the previous gaming round. Enjoyment refers to experiencing pleasure, satisfaction, or delight in something.

People feel it is a subjective and emotional response when they find an activity, experience, or situation pleasing, gratifying, or fulfilling. In empirical instruments such as questionnaires assessing GX, enjoyment often is obfuscated behind emotions, player engagement, or satisfaction (IJsselsteijn et al., 2013).

QoE is similar to enjoyment, a holistic metric describing the whole interaction (typically answered on a Likert item, from bad experience to great experience), without an in-depth assessment of reasons for the experienced qualia. Nevertheless, assessing the QoE is commonly done in research with video games and latency since it can be done in a single question and does not require participants to take part in time-consuming questionnaires.

Forty-two of all reviewed papers (42/67) discuss the negative effects of latency on the players' subjective levels of enjoyment or QoE. For example, Joerg et al. (2012) found that a player's level of enjoyment directly correlates with the level of delay or latency. They further revealed that a less responsive player avatar in a setting with more than 100 ms latency affects the player's enjoyment particularly. Quax et al. (2013) found in an evaluation of gaming experience that the effects of latency on the enjoyment of a game depend on the game's genre. In a study, they showed that fast-paced games, in their case, racing games, are more negatively affected compared to strategy and puzzle games. Normoyle et al. (2014) demonstrated that game enjoyment was stable up to a level of 300 ms of constant latency. Surprisingly, Saverly et al. (2014) investigated different levels of latency and latency compensation techniques and found that some players enjoyed the uncertainty induced by latency as it added some more challenge to an otherwise relatively dull test apparatus. However, the findings of Saverly et al. are an isolated case. No other work in our review corpus identified an enjoyment gain with increasing latency. On the contrary, Liu et al. (2021b; 2021c; 2021a) repeatedly showed that latency decreases the QoE. They showed that the QoE linearly decreases by 25 % from 25 ms to 125 ms and that more experienced players are more strongly affected by latency's effects.

Overall, latency – often as a secondary effect – reduces a player's enjoyment and quality of gaming session experience. Both are holistic concepts inherently influenced by more atomic elements of GX, such as flow, immersion, and agency. Nevertheless, even when using enjoyment or QoE as a proxy for the multi-faceted construct GX, it is evident that latency has negative effects.

2.2.2.6 Other Effects and Games' Latency Sensibilities

While our review aimed to categorize all reported effects on PP and GX in the presented categories, this was sometimes not feasible. Some works investigate highly specialized edge cases. Other works report effects that other publications have not discussed. However, to still be able to provide an extensive picture in this review, we briefly elucidate these works in the following.

For example, Xu et al. (2022) investigated how latency alters player movement in CS:GO. They found that a higher latency reduced overall player movement. They argue that this also entails that players are surviving for less time since players have a harder time to avoid being shot and a harder time to position them to shoot opponents. Beigbeder et al. (2004) followed a similar approach and let players complete a running course in the FPS Unreal Tournament 2003. In their study, players had to run along a predefined track while exposed to different latency levels. Beigbeder et al. found that TCT (the time players required to finish the track) remained stable up to 300 ms. However, after 300 ms a negative upward trend in completion time was detectable.

Tseng et al. (2011) investigated how players perceive latency, what players think the cause for latency is, and how they react to latency. The authors found that players often need help with latency as they fail to identify the root of the delay. Tseng et al. illustrate that players frequently have to resort to a trial-and-error approach to resolve their latency issues. Ultimately, the authors argue that players blame game providers and developers to cope with latency.

Eg et al. (2018) explore a concept called performance-enjoyment link (Weiner, 1985), which postulates that enjoyment is a secondary effect to a good performance. Generally, humans enjoy doing something they are good at. Hence, in video games, performing at high levels promises a higher gaming experience. However, Eg. et al. found that PP, measured as the participants' self-reported skills, does not interact with GX and latency. This aligns with other work by Klimmt et al. (2009), which also reported that GX is not necessarily dependent on success or performance.

Sabet et al. (2022) found that the effects of latency may be gender-specific. In a VR study, they found that male participants are more critical or sensitive to high latency (50 ms vs. 120 ms) compared to female participants. Although no other work makes similar conclusions, Sabet et al. highlight that latency perception is subject to variability. Hence, other work aimed to specify what factors define the latency sensitivity of a video game. Durenz et al. (2021), Wahab et al. (2021), and Schmidt et al. (2017) discuss that task complexity and the game's pacing fundamentally alter the game's latency sensitivity with more complex and faster games stronger affected by latency. Raaen and Grønli (2014), Xu et al. (2022), and Suznjevic et al. (2013) consider that player perception is key, and that it is crucial how games communicate latency to the player as it can change their perception. Other work found that the skill level (Liu et al., 2021a), the weapon in a FPS game (Xu et al., 2022), the game's rule (Claypool & Claypool, 2010) or specific game characteristics (Sabet et al., 2020a) such as feedback frequency, importance of actions, or the number of required actions modify a game's latency sensitivity.

2.2.2.7 Limitation

In this review, we investigated the effects of latency in video games on PP and GX. Although we followed a systematic approach to obtain the data investigated in this study, it has limitations. First, this review only includes articles written in English. However, it is possible that there is work published investigating the effects of latency in video games in other languages, which we had to omit. Second, we tailored the search terms and used databases extremely carefully and in multiple iterations. Nevertheless, it is possible that our search terms and the database unintentionally lead to the exclusion of articles of relevance. Lastly, while we categorized GX successfully into five different dimensions, it needs to be noted that GX is a highly elusive concept. It is subject to each player's subjectivity, feelings, emotions, and preferences. Furthermore, GX is assessed very diversely in previous work. Some research uses validated instruments, while others are satisfied with single-item questionnaires. The nature of subjectivity and how GX is measured makes it hard to clearly define it. Ultimately, this also impacted our work and its presentation. Although we designed them over multiple iterations, it is possible, however unlikely, that our categories are subject to annotator bias.

2.2.3 Summary

Previous work consistently demonstrates that latency in video games negatively influences PP and GX. In this work, we conducted a systematic literature review following the PRISMA (Page et al., 2021) protocol to unravel and categorize those effects. For our systematization, we used a semi-reflexive inductive thematic analysis. We generated four dimensions on how latency affects PP and five dimensions on how it affects GX. Additionally, we illustrated other effects that are not unambiguously classifiable into the presented categories. While previous work investigated extremely wide latency ranges between 0 ms (Wahab et al., 2021) up to 1000 ms (Claypool & Claypool, 2010), it is not clear at what level latency starts to affect a gaming session. However, the consensus in prior work is that less latency is better.

Our work highlights that latency affects TCT (Beznosyk et al., 2011; Claypool et al., 2020), accuracy (Normoyle et al., 2014; Lee & Chang, 2018; Quax et al., 2004), tracking and selection performance (Liu et al., 2021c; Claypool et al., 2019; Lee et al., 2019), as well as holistic performance metrics such as the score (Ries et al., 2008; Liu et al., 2021a; Sabet et al., 2020b) in video games. Latency also alters the subjective experience of a gaming session and leads to an increased feeling of unfairness (Lee & Chang, 2017; Zander et al., 2005), a reduced flow (Durnez et al., 2021; Carlson et al., 2021), a worsened immersion (Beyer & Möller, 2014; Sabet et al., 2022), a loss of agency and feeling of control (Eg et al., 2018; Durnez et al., 2021), and overall reduces the general enjoyment and QoE (Jörg et al., 2012; Liu et al., 2021b; Liu et al., 2021c; Liu et al., 2021a). Further related work also investigated how latency alters player movement (Xu et al., 2022), how players perceive and coop with latency (Tseng et al., 2011), and that its effect may be gender-specific (Sabet et al., 2022). Lastly, previous work also aimed to systematize what game characteristics, such as its task complexity (Schmidt et al., 2017), its pace (Wahab et al., 2021), or its rules (Claypool & Claypool, 2010) alter the game's latency sensitivity without coming to a clear-cut conclusion.

2.3 Latency Compensation Methods

The previous section shows that latency negatively influences PP and gaming experience. Hence, to overcome these adverse effects, previous work proposes different methods. Latency compensation methods can be categorized into two main approaches: Game-internal latency compensation, which accounts for latency directly in the game or game-external methods, which aim to compensate or reduce the underlying reasons of latency such as processing time of the machine running a game. These two main approaches can be further divided. Game-internal compensation techniques are, for example, frequently categorized by how they aim to compensate for the effects of latency. One classification by Liu et al. (2022) structures game-internal latency compensation techniques in four classes: (1) feedback, (2) time manipulation, (3) world adjustment, and (4) prediction. Latency compensation methods from the feedback class provide auditory or visual feedback to the players based on the current latency. These techniques do not alter the actual latency in the game. For example, Gutwin et al. (2004) proposed a method in which specialized game objects signalize the presence, magnitude, and effects of latency in the game. Methods from the categories time manipulation (2) and world adjustment (3) both directly alter the game world. Time manipulation methods slow down, speed up, or stop game time entirely to synchronize game events. One example of such a method is time warp (K. Lee & C. Chang, 2017; Bernier, 2001). Time warp is widely used in online multiplayer video games but is often considered unfair by gamers since it always favors the actor (K. Lee & C. Chang, 2017). World adjustment methods, such as geometrical manipulation, manipulate game objects depending on the amount of latency, for example, making targets bigger and thus easier to hit in a high latency setting (Lee et al., 2019). The last category, prediction, contains methods using the available game state data, such as the gamer's position in the game world or other objects in the game world, to interpolate or extrapolate a future game state. Proactively calculating a future game state potentially decreases the perceived latency.

Another approach to latency compensation is to reduce the latency game-externally by reducing the latency of the system running the game. These methods are more independent than game-internal compensation methods since they are not tailored to a specific game but aim to reduce the latency of the input-output loop more directly.

In the following, we report on previous work investigating game-dependent latency compensation in video games before continuing to game-independent compensation methods.

2.3.1 Search Strategy

To learn about game-dependent and -independent latency compensation methods, we reviewed the literature and queried the same research databases reported in Section 2.1.1. We used the following query (coded as a pseudo search term) to retrieve articles investigating latency compensation in video games: (("Video Games" OR "Gamels") AND ("Latency" OR "System Response Time" OR "Delay" OR "Lag") AND ("Compensation" OR "Compensating" OR "Reduction" OR "Reducing" OR "Lowering" OR "Improving")). And the following query to retrieve articles investigating latency compensation methods: (("Latency" OR "System Response Time" OR "Delay" OR "Lag") AND ("Compensation" OR "Compensating" OR "Reduction" OR "Reducing" OR "Lowering" OR "Improving")). However, instead of conducting a top-down literature review as reported in Section 2.2.1.2, we more strongly relied on a bottom-up approach. This was necessary since the general nature of the used keywords provided an extremely high number of results. A comprehensive top-down review of all articles hence was not feasible. For example, a search in Google Scholar with the keywords "Latency Compensation" results in 243 000 articles as of September 2023. Refining this query to "Latency Compensation Video Games" still yielded 18 900 results. Hence, we opted for a bottom-up approach instead of conducting a top-down literature review. To start our review, we used highly influential, highly cited articles that proposed, discussed, and evaluated latency compensation techniques, such as Liu et al.'s taxonomy of latency compensation methods (2022), Gutwin's et al.'s (2004) latency signalization method and Salay and Claypool's comparison of automatic and manual gameworld alterations (2020). Starting from a pool of about 20 papers, we carefully expanded our review to include more relevant articles and investigations. In the following, we report the results of this review. We start with game-internal latency compensation methods.

2.3.2 Game-internal Latency Compensation

To structure our review of game-internal latency compensation methods, we follow the categories and definitions proposed by Liu et al. (2022), starting with compensation methods of the feedback category.

2.3.2.1 Latency Compensation via Feedback

Latency compensation via feedback entails providing the player with information about the current latency. This information can either conceal the current level of latency from the player or provide the player with indicators of its level (such as bars or numerical values in the game UI) (Liu et al., 2022). Crucially, latency compensation based on feedback does not actually change the game's current state; it only communicates or conceals the current level of latency.

To expose latency to the users, previous work suggests different methods. For example, Gutwin et al. (2004) propose an approach to dealing with latency using so-called decorators. Decorators visually show the presence, magnitude, and possibly even the effect of latency on the users. The authors argue that the visual representation allows users to understand the consequences of latency better and enables them to employ their natural coping strategies. The authors based this assumption on earlier work by Vaghi et al. (1999) that discusses that communicating the presence of latency to the players of a game may be a major factor in allowing players to adapt to it. Hence, in two studies, Gutwin et al. investigated if two different types of decorators can compensate for the negative effects of latency. In the first study with 16 participants, the authors faded the mouse cursor from white to black for different levels of latency (white = 0 ms to black = 1400 ms). Overall, they tested four levels of delay; 0 ms, 800 ms, 1100 ms, and 1400 ms. Using this fading cursor, participants had to perform several prediction tasks. In detail, the users were shown pre-recorded cursor movements and were asked to predict how far the cursor would move. In the second study, Gutwin et al. tested a decoration approach called future state decorator. The decorator in this study shows the potential future position of the mouse cursor based on its current position and velocity. Using these decorators, twelve participants took part in a collaborative task. Two participants had to work collaboratively in each session to achieve a common goal. They had to move a virtual box from one location to another. Typically, this task would lead to errors in a non-compensated scenario since both participants cannot anticipate their partner's latency-affected mouse movement. In their study, Gutwin et al. artificially added four levels of latency (0 ms, 200 ms, 400 ms, and 800 ms). Both studies show that decorators can reduce the negative effects of latency. In the first study, participants had fewer prediction errors when the fading cursor showed the current level of delay. Similarly, in the second study, the future state decorated reduced the errors in the collaborative environment. The authors discuss their findings and conclude that the decorators in their studies worked because they provided answers to questions that were crucial to the task at hand. The first study allowed participants to estimate if the cursor had already stopped. The second study showed if the partner in the collaborative environment had already or was about to pick up the virtual box. Hence, both decorators reduced the overall error rate.

Other work, by Wikstrand et al. (2004), for example, followed a similar approach as Gutwin et al. (2004) to expose the current level of latency to the players of a video game. The authors developed a custom version of the seminal Pong game for their investigation. They coupled it with a specifically designed method to communicate the current level of latency to the player. The current level of latency was depicted as a shadow tracking the game paddle. Hence, players using this method could estimate and anticipate the influence of latency on their gaming session. In a study with 24 participants, the authors tested 100 ms and 250 ms of latency with the feedback system enabled and disabled. While participants played the game, Wikstrand et al. measured their in-game movement and behavior. Furthermore, they collected data on the subjectively experienced enjoyment and mental load. Their study showed that playing with a higher level of latency led to increased mental load, less fun, reduced performance, and less in-game movement. However, their analysis also shows that using their feedback method, players could adapt to the higher level of latency, which consequently required a reduced mental effort to play the game.

While latency exposure methods communicate – or expose – the current latency, latency concealment techniques aim to obfuscate – or conceal – latency from the players. This is often done visually. For example, in a complex RTS game, players might command their units to move from one location to another. The game could respond as fast as possible, creating a visual effect, such as the units starting their move animation. However, the actual command, the movement of the units, is not immediately executed but only after the command and its effects have been approved by the game's logic. The visual effect when sending the command conceals that the actual command is affected by latency (in this case, processing latency, input latency, and rendering latency) and provides the player with more responsive feedback. Liu et al. (2022) argue that latency concealment includes image warping techniques, which are often used in VR head-mounted displays (HMDs) to reduce the latency of head motions (Smit et al., 2009). Image warping is an image-based technique that updates the current view by re-projecting a previous image or set of images. This approach allows, for instance, in VR, for a constant and smooth image update rate, which reduces the negative effects of latency (in this case, processing latency between the workstation and the used HMD) drastically (So & Griffin, 1992), since the HMD has not to wait for the next rendering. The same approach of concealing latency by warping images was investigated by Kim et al. (2020) in FPS games. They developed a custom FPS game and implemented three warping methods for their investigation. In a study with nine participants, the authors tested five conditions: (1) 105 ms latency and no warp applied, (2) 105 ms latency and rotational-naive warping, (3) 105 ms latency and rotational-aware warping, (4) 105 ms latency and rotational- and translation-aware warping, and lastly (5) 25 ms latency and no warp applied. Using this setup, they found that even the simplest warping method (rotational-naive warping) effectively reduced the negative effects of latency on in-game performance. The author concluded that warping could be a computationally inexpensive solution to the high latency problem of video games, particularly games played using cloud streaming approaches. To further validate Kim et al.'s approach, later work implemented an openly accessible web-based implementation of Kim et al.'s latency concealment method (Boudaoud et al., 2021).

While latency exposure and concealment are promising approaches to reduce the negative effects of latency, other work proposes different methods, such as compensation for latency via time manipulation.

2.3.2.2 Latency Compensation via Time Manipulation

Time manipulation in latency compensation refers to altering the virtual time in the game. Time manipulation approaches are typically exclusively used in multiplayer scenarios in which multiple players from different locations play in a joint gaming instance. Most commonly, a dedicated server hosts this instance in a data center. However, there are other architectures for multiplayer games, such as peer-to-peer or client-hosting connections. Time manipulation compensation aims to synchronize game events for all players. Since not all players connect to the server from the same location, with the same hardware and internet quality, they all experience different levels of latency (in this case, a combination of all types of latency can lead to highly different levels for each player). By synchronizing all game events, independently of when they arrive, on a shared time stream, the effects of latency theoretically are reduced.

The most commonly used time manipulation compensation method in commercial video games is time warp (Liu et al., 2022). Time warp rolls back a joint game state to the point in time at which the user performs the input rather than applying it when the server receives it. To illustrate how time warp works, consider the following example: Two players – players A and B – play a fast-paced FPS against each other. Each player is connected from their respective home to the central gaming server. Since player A lives further away from the data center hosting the server, they have an overall higher latency than player B. For the sake of simplicity, let us assume that player A plays with 500 ms and player B with 50 ms of latency. If player A now takes a shot at player B with their virtual weapon it takes up 500 ms for the central server to register and process the shot (in a real-world scenario, latency comprises the round-trip time between client and server, and would since be lower but still high). The server uses time warp and rolls the game state 500 ms back, to check if player A hit player B with their shot. Regardless of the outcome, the server propagates the result of this action to all players. If player A indeed hit player B, player B now – 500 ms after the actual event– receives damage from getting hit. In the worst case, this could be the difference between virtual life and death. However, while the event was sent from player A to the server, the time for player B did not stand still, which means that player B could have moved to a safe location. However, since the server propagates a time-warped game state to player B, player B may get hit by a shot given by player A even though player A no longer sees player B in the game world. This phenomenon is often expressed by time warp compensation methods and is called "shot around the corner" or "always favor the actor." This behavior is not ideal since it punishes the player with the lower latency. Despite these deficits, early fast-paced games such as Half-Life, Overwatch, or Mortal Kombat used time warp to reduce the negative effects of latency. However, even highly recent games such as Valorant, which was released in 2020, use time warp mechanics, as the developers revealed (Liu et al., 2022). Since time warp is still so frequently used in commercial games, it is investigated in several research publications.

For example, Sun and Claypool (2019) investigated latency compensation via time warp in a cloud-based gaming setting. The authors used a custom cloud-based game system and a self-developed arcade-like top-down shooting game to do this. They deployed the game to the streaming service and implemented a timewarp algorithm. Figure 2.4 depicts a schematic of their time warp manipulation deployed by Sun and Claypool. In a study with 30 participants, Sun and Claypool evaluated the efficiency of their time warp algorithm. The authors tested five levels of latency (0 ms, 100 ms, 200 ms, 400 ms, and 800 ms) with time warp enabled and disabled. While participants played the game, the authors recorded the in-game score to measure PP. Additionally, the authors asked participants about their subjective experience with the responsiveness and consistency of the game after each played round. While their results show that time warp reduced some of the negative effects of latency on the player's performance, players also rated the game with a lower level of consistency. The authors argue that the degradation in subjectively experienced game consistency was caused by the shooting mechanic implemented in the game. The game was based on a projectile-based shooting mechanic, meaning the player and enemy entities shoot projectiles. These projectiles flew for a certain time and then hit or did not hit a target object. However, since time warp rolls back game states, players stated that projectiles behaved unnaturally, reducing game consistency.

The time warp-induced inconsistency and potential solutions to this problem have also been the focus of prior work. For example, Mauve et al. (2004) proposed to

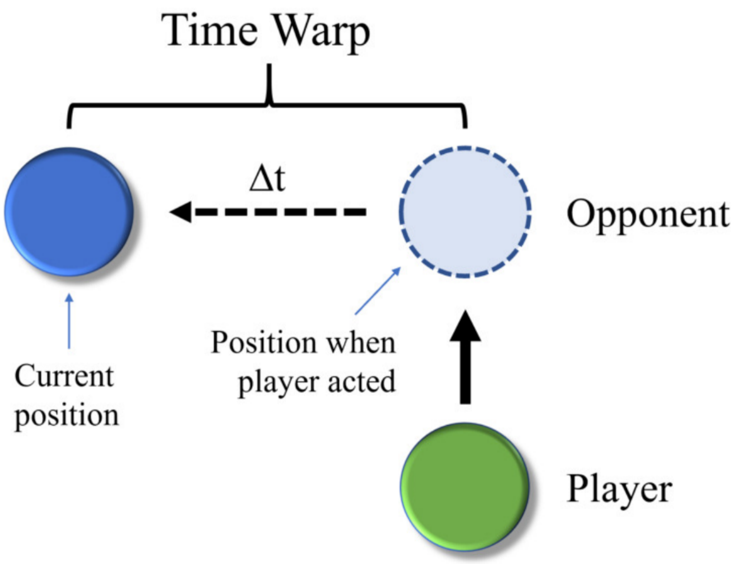


Figure 2.4: Schematic diagram Sun and Claypool's time warp implementation. Figure from original work (Sun & Claypool, 2019). With the implemented algorithm, the player acts, for example, shoots at an opponent based on the opponent's current position displayed to them. However, the central gaming server receives the input $\vec{\Delta}$t later caused by the player's latency. When receiving the input, the opponent potentially has already moved. To compensate for this, the server warps time by $\vec{\Delta}$t, calculates if the player hit the opponent, rolls the game world back to real-time, and propagates the results. This behavior can result in unrealistic game behavior, such as being hit by an enemy at a (assumed) safe position.

artificially delay time warp propagation to the clients to allow for consistent distribution. In their work, they call this delay local lag, albeit not being the same as real, local latency (as produced by input methods, monitors, and workstations). The authors suggest artificially increasing the local delay of interconnected systems, for example, players' gaming systems, to align the time difference to a central server unit. Using such an approach could, in theory, make compensation systems such as time warp more consistent. However, they also artificially increase the latency of players playing with low latency. In later work, Liang and Boustead (2006) developed a time warp algorithm and coupled it to an artificial delay as proposed by Mauve et al. (2004). They customized the game Quake III to achieve this, allowing for distributed multiplayer gameplay. In this apparatus, the authors integrated a time warp algorithm and offset its propagation client-sided to align latency levels between all players. In a study with computer-controlled bots, the authors found that performance degradation was effectively compensated by their time warp-based method. However, since the authors conducted the study without real participants, they could not assess or evaluate the effects of their methods on GX or game consistency.

Time warp is not the only approach using time manipulation for latency compensation. Other works, similar to work presented by Mauve et al. (2004) and Liang and Boustead (2006), aimed to limit the incoming or outgoing events by certain time frames to align latency levels across players in networked multiplayer settings. One of the earliest works investigating latency synchronization using incoming and outgoing delay is the work by Diot and Gautier in 1999 (1999). In their work, they developed a serverless distributed multiplayer game, MiMaze, in which players could solve mazes collaboratively. The game was provided and played via the Internet. To synchronize game events from multiple players with different levels of latency and to provide a consistent gaming session for all players, Diot and Gautier used so-called bucket synchronization. In general terms, a bucket synchronization approach collects all events from players for a certain timeframe, stores them, and propagates the data to all players involved at the end of the defined time frame. Effectively, bucket synchronization can be considered single-sided incoming delay compensation since the player's incoming events are delayed. However, since the algorithm does not account for the individual latency of each player, it is still possible to introduce some variation and inconsistency when propagating the collected events. However, Diot and Gautier found in a study with 25 participants playing with varying latency between 0 and 300 ms of latency that their algorithm significantly increased PP and subjectively experienced consistency.

Other work, such as by Li et al. (2018), expanded on Diot and Gautier's approach and delayed incoming and outgoing events. To achieve this, the authors developed a custom server-client gaming scenario. In their work, the server is informed about latency to the clients by the clients. Accordingly, the server limits both upstream and downstream of the clients to either increase or decrease their latency with the overall goal of creating an even latency environment for all players. In a small study with 10 participants, Li et al. replicated previous findings regarding the effects of latency on PP. Additionally, they showed that this reduction in performance is minimized using their compensation system. Furthermore, and more crucially, the authors found that their system alleviated the perceived fairness of each player. Without the compensation system activated, players felt they were unfairly disadvantaged by latency. However, with an activated latency compensation system, players, on average, reported a higher feeling of fairness. Although

Li et al. do not discuss it directly, it is possible that this increase in perceived fairness was not solely produced by the latency compensation system but also partly caused by the mere feeling of players that such a compensation system was active. Previous work investigating user expectations when communicating with an interactive system showed that what a user expects significantly influences how the user feels and behaves in an interaction (Kosch et al., 2022).

2.3.2.3 Latency Compensation via World Adjustment

While compensation via exposure or time manipulation does not change the game, latency compensation via world adjustment starts exactly there. The overarching impetus of world adjustment methods is to change the actual game world to accommodate latency. For example, this could mean increasing target sizes in an FPS or widening a street in a racing game, to counteract the reduced accuracy of playing with a high latency. Hence, world adjustment aims to decrease the game's difficulty to enable players to perform and experience the game similar to a low-level latency setting (Bateman et al., 2011). Other methods of world adjustments, for example, are implementing so-called sticky targets, target magnetism, or magnetic bullets in FPS games. Sticky target refers to targets in FPS games that slow down the cursor velocity while near them, allowing the players to perform more precise target acquisition. Target magnetism or magnetic bullets combine similar concepts, although applied in different parts of the game. Target magnetism refers to projectiles, such as bullets, being automatically dragged to the target. On the other hand, magnetic bullets are supposed to automatically fly towards a target (contrary to being dragged by a magnetic target). All these approaches aim to make atomic game actions, such as target acquisition or target selection, easier to counteract the effects of latency on these actions. Several previous works investigated different world adjustment methods.

Ivkovic et al. (2015), for example, investigated two tasks (target acquisition and target tracking) highly relevant to almost any kind of video game. In the target acquisition task, players were instructed to only aim the mouse cursor at a spherical target and click the left mouse button. In the target tracking task, the target appeared directly in the middle of the player's view. Players were instructed to click the target, which, after clicking, started moving horizontally. Players were asked to follow the target with the mouse cursor by keeping the cursor on the target. Using the two tasks, Zenja et al. tested five levels of latency (11 ms, 41 ms, 74 ms, 114 ms, and 164 ms) in combination with three latency compensation techniques based on world adjustments. The tested compensation methods were sticky targets and aim dragging. While sticky target refers to reducing the gain of the mouse in the vicinity of the target (as described above), aim dragging refers to the mouse cursor partially being dragged in the direction the target is moving. In a study with 18 participants, the authors found that their compensation method significantly reduced the negative effects of latency in target acquisition. Regarding the target tracking task, the authors even found that aim dragging removed the negative effects of latency completely. They argue that aim dragging enabled players to achieve the same performance as in the baseline (11 ms) condition. Furthermore, the authors revealed that their world adjustments lead to increased experience quality and subjectively more responsive interaction.

While sticky target and aim dragging is a promising approach, as presented by Ivkovic et al. (2015), it is limited to games that actively use a mouse cursor and incorporate atomic in-game actions such as target tracking or target selection. However, a large number of games, such as fighting games, racing games, or jump-and-run games, do not rely on such actions for their core gameplay. Other work implemented world adjustments to overcome this challenge, focusing on other game elements. For instance, Xu and Wah (2013) adjusted the possible time frame for player action in a fighting game. If latency was higher, the authors granted the players more time to perform an in-game action (such as throwing a punch with a virtual avatar). If latency was lower, this time frame was reduced. The authors applied their adjustment methods in a study with 20 participants and found that using their adaptive world adjustment led to players being significantly less aware of the delay. Overall, the author argues that their compensation technique is well suited to counteract the negative effects of latency.

However, since Xu and Wah (2013) neither assessed PP nor experience, further work is necessary. A different approach takes the work by Lee et al. (2019). In their work, the authors developed a custom version of the game Flappy Bird. In the original Flappy Bird, players must navigate a small bird through a maze of pipes. Players could push the bird a bit up by tapping the screen on their smartphones while simulated gravity dragged the bird back down. Players progressed in the game by skillful and well-timed tapping of the screen, pushing the bird up and letting it fall down, to navigate through the pipes. However, the game was over if the bird touched a pipe or fell to the bottom of the screen. In the adapted version by Lee et al., the authors integrated a latency compensation method based on world adjustment, which changed the size of pipes in correlation to the

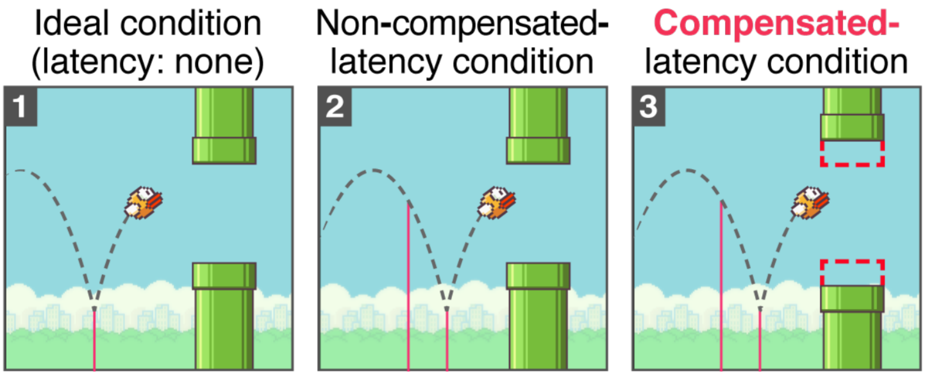


Figure 2.5: Illustrates the latency compensation approach based on world alternation developed by Lee et al. (2019). The left (1) shows an ideal no-latency scenario in Flappy Bird. The user provides an input (red line), and the bird immediately jumps up. The middle shows a non-compensated latency scenario. The user provides an input (first red line). However, the input is registered with delay (second red line). This potentially leads to a game over, as the bird must not touch the pipes. The right shows a compensated latency scenario, in which Lee et al.'s approach is depicted. Via world alternation, the authors scale the size of the pipes down to account for the delayed processing of the latency-induced input. This gives the player more leeway to navigate the bird through the pipes and may even prevent a game over caused by latency. Figure adapted from original work by Lee et al. (2019).

current latency. By decreasing the pipe size and increasing the gap between the pipes' ends, the authors provide more leeway to the player to navigate the bird, making the game easier. In a study with 12 participants, the authors tested three levels of latency (40 ms, 80 ms, and 120 ms) with their geometrical compensation enabled or disabled. The results show that players achieved similar error rates when performing with latency and compensation enabled compared to baseline latency (40 ms). Effectively, Lee et al. show that their world alternation approach can reduce some negative aspects of latency. Later work by Salay and Claypool (2020) replicated Lee et al.'s findings using the same approach in the same game.

Overall, previous work shows that latency compensation via world alternation is a valid approach to compensate for latency in certain cases. However, all the presented work investigating this compensation approach is highly specialized to either one certain game or one certain video game genre. The next paragraph describes latency compensation via prediction, which promises a more generalized approach.

2.3.2.4 Latency Compensation via Prediction

The last category of Liu et al.'s (2022) taxonomy discussed in this thesis investigates latency compensation via predictions. According to Liu et al., a prediction-based latency compensation technique estimates a future game state without receiving a joint game state from an authoritative server in a multiplayer scenario. While most of the discussed approaches in the following discuss predictions in the context of compensating for network latency, in principle, they, as it is with all previously presented compensation methods, also work to compensate for local latency. In the broader context of this thesis, we conclude that a strict distinction between network and local compensation is unnecessary since they can be applied for both types of latency. However, in the case of prediction-based latency compensation, certain methods are only applicable for multiplayer games played in a server-client setting, such as server-sided prediction. While we still discuss these methods to provide a complete picture, we will specifically note such prediction techniques in the following.

There are different ways in which prediction-based latency compensation can function. The most tangible mode is predicting a future game state based on local user input, a client-side prediction. However, this is predominantly relevant for multiplayer games where every action, such as shooting an enemy player, must be validated and confirmed by a central server before it gets propagated to all players in the session. In a client-sided prediction, the client applies changes to the game state before confirming the game state change with the authoritative server. This drastically reduces the latency perceived by the user since the game does not have to wait for a server to confirm and allow the action. However, this can also lead to game inconsistency, in case the server does not confirm the action and the game has to revert the predicted game state change (Burgess & Katchabaw, 2006). Although client-sided prediction is usually used in client-server architecture-based video games to compensate for network latency, the same approach could theoretically be applied to compensate for local latency. Instead of a server confirming a predicted game state change induced by the user, the reception of the actual user input achieves the affirmation. Since client-sided prediction is highly speculative in a multiplayer setting and potentially leads to game inconsistency, it is rarely used as a standalone latency compensation technique. Subsequently, no publications are investigating it in an ecologically valid gaming setting. Nevertheless, some work is researching its applicability in a more general HCI context, such as the work by Chen et al. (2007). In their work, the authors implemented a system called Echo. Echo predicts and depicts the effects of a user input based on previous states. Chen et al. integrated Echo in a collaborative virtual environment where two participants had to work together in a virtual object control task (Park & Kenyon, 1999). In detail, participants had to move a virtual frame from a start point along a virtual rod to an endpoint without the two objects touching each other. The task was considered successful if the participants completed it without objects colliding. In this scenario, Echo predicted the future position of the frame. In a study with 18 participants, the authors used this task to test the efficiency of Echo. To achieve this, they inflicted the collaborative virtual environment with 0 ms to 900 ms of artificial latency and enabled or disabled Echo. Chen et al. found that if latency exceeded 600 ms participants were no longer able to complete the task without Echo. In general, they found that if latency exceeded 100 ms Echo significantly improved the participants' performances, such as decreased TCTs and error rates. While Chen et al.'s (2007) results are promising, they are, as are other compensation methods, highly specialized. It is unclear if the methods proposed by the authors generalize to other use cases in general and video games in particular.

Since client-sided self-predictive compensation methods are highly speculative, caused by the nature of uncertainty of future events, previous work also investigated other predictive methods for latency compensation. These methods either work by interpolation or extrapolation. Interpolation provides an estimation of new data based on the range of known data. The simplest method of interpolation is linear interpolation. This is often called lerp in computer sciences, particularly in computer vision and game engineering. Lerp assumes a linear distribution of a given range of data to estimate new data within this range. For example, if two known points are given by the coordinates (x_0, y_0) and (x_1, y_1), the linear interpolant (the connection between both points) is a straight line. Any new point x between (x_0, y_0) and (x_1, y_1) is given by the linear interpolant. Figure 2.6 demonstrates this process. Points (x_0, y_0) and (x_1, y_1) are illustrated in red, the interpolant, and the estimated data point in blue.

Interpolation in video games is typically used to visually smooth the gameplay in multiplayer games when game state updates from an authoritative server are delayed by latency and the visual update rate of the client occurs more often than server-sided updates (Liu et al., 2022). Since interpolation requires at least two known game states to interpolate a previous state between them, strictly speaking, it does not reduce latency. However, it can compensate for the negative effects of it, such as jerky enemy movement

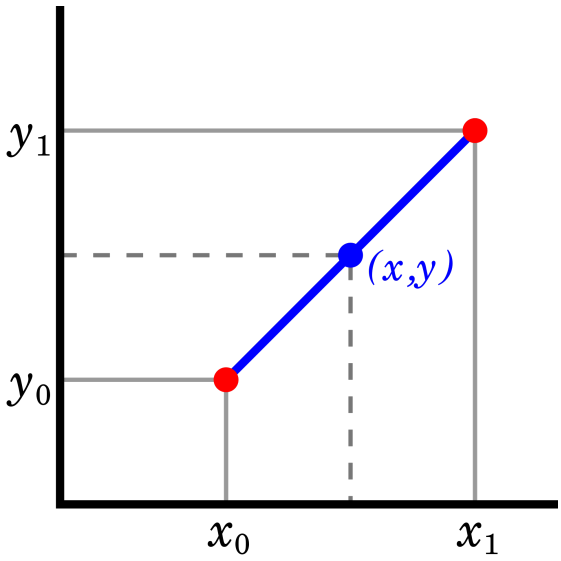


Figure 2.6: Illustrates a linear interpolation (lerp) process. Two points are given by the coordinates (x_0, y_0) and (x_1, y_1) (red). The interpolant (the connection between both points) is blue. The estimated value of new point x in the interval of $(x_0,\ x_1)$ is y and given by the interpolant. Original figure from Wikipedia (2023c).

in a multiplayer game. Previous work by Lee and Chang (2015) investigated if interpolation can compensate for the negative effects of latency in the FPS game CS:GO. In their work, the authors interpolated between two visual game frames. They argue that without interpolation, a player in the game may see a moving entity in one position for several frames before it is later, delayed by latency, teleported to an updated location. Successfully hitting such a target is hard to achieve, the authors continue. To investigate if interpolation can counteract this effect on the players' accuracy, they conducted a small study with 4 participants. All participants played with two levels of latency (0 ms and 150 ms) and interpolation, either activated or deactivated. The authors found that, on average, players performed more accurately with interpolation than without it.

In a similar work, Savery et al. (2014) investigate if local game inconsistency, induced by client-sided predictions, can be reduced using interpolation. In a study with 26 participants playing different games, the authors found that interpolation indeed can increase the subjectively perceived consistency of the game. Surprisingly, the authors did not find an evident correlation between consistency, PP, and gaming experience. Although players rated the game more consistently, they performed worse and did not necessarily have the highest level of GX. Savery et al.'s work is crucial since they show that latency compensation, in this case interpolation, is not always beneficial. Particularly, compensation methods based on predictive models can make errors and can lead to unnatural or inconsistent game behavior, affecting performance and experience. Interpolation, in general, is only one possible approach to predict game states to compensate for latency. Other methods do not interpolate but extrapolate.

Extrapolation does not estimate unknown data between known data (as interpolation does). Instead, it aims to fit a function on given data to estimate new data beyond the original data used for the extrapolation. For example, assume known data points ($x_0 = 0$, $y_0 = 0$ / $x_1 = 1$, $y_1 = 1$ / $x_2 = 2$, $y_2 = 2$). Now, using the simplest method of extrapolation, linear extrapolation, would assume a linear correlation between all points, resulting in a straight line (extrapolant) through all points. Using the extrapolant, extending the data beyond the original data set, for example, to a point ($x_3 = 3$, $y_3 = 3$) is easily done. Latency compensation methods based on extrapolation use the same approach. However, instead of extrapolating between just points, they extrapolate the position, rotation, or translation of game objects, player or enemy movement, or even the whole game frame. Since extrapolation methods generate new data, they are prone to some degree of uncertainty. Although extrapolation is prone to uncertainty, it is the most widely used latency compensation method in commercial video games. This is because extrapolation actually reduces the perceived latency by players since predicting game states in the future provides a temporal advantage to the system.

One commonly used extrapolation method is dead reckoning (Wikipedia, 2023a; Pantel & Wolf, 2002). Dead reckoning is a physics-based approach that calculates an actor's current or future position (such as a player-controlled avatar) using previously determined positions and kinematics states. For example, we can assume that an avatar's position P_0 is given for T_0 and that the avatar is moving with a constant velocity V_0. Using this, we can calculate a future position P_x using the following formula:

$$P_x = P_0 + V_0 * T$$

Extending this simple formula with basic linear physics allows dead reckoning methods to account for acceleration. Instead of predicting the position of an avatar running with constant velocity, one could also predict the position of a car with a known acceleration A in a racing game by using:

$$P_x = P_0 + V_0 * T + \frac{1}{2} A_0 * T^2$$

Since dead reckoning is widely used in commercial video games, it is also subject to more investigations. Li et al. (2007) used a dead reckoning approach to compensate for latency in navigation path prediction in video games. The authors elaborate that the personal habits of players influence their choice of paths when encountering multiple possible future paths. In their work, they aimed to incorporate the personal preference of players for certain paths in their dead reckoning prediction. They used a custom-developed 2D game to evaluate their method, where players navigate small tanks through different levels. In a study with 37 players and three latency levels (1 s, 2 s, and 3 s), the authors found that their compensation method performed significantly better than classical dead reckoning approaches (which do not account for personal player preferences). Hence, their algorithm could more accurately predict a future game state and player decision than previous approaches, effectively counteracting the delayed responsiveness induced by the tested latency.

Duarte et al. (2020) proposed another dead reckoning approach coupled with a machine learning-based classification, which aims to classify whether a classical dead reckoning algorithm can predict a future game state. If the classification system concludes that the dead reckoning prediction is not possible, not efficient, or not accurate enough, the system falls back to another compensation method. The authors call this method smart reckoning and aim to reduce in-game inconsistency caused by the uncertainty entangled with extrapolation methods. To achieve this, the author gathered public data from the MMORPG World of Warcraft, which was used to train the classification system. The classification system, the authors argue, detects abrupt changes in the player characters' velocities, directions, or accelerations. Since dead reckoning extrapolates future data based on previous data it is not well-suited to deal with sudden nondeterministic changes in kinematic states. Using this method, Duarte et al. were able to significantly improve dead reckoning prediction accuracy. Consequently, a better prediction improves game consistency (Zhou et al., 2004). However, Duarte et al. did not evaluate their method with real-time gameplay instead simulated player movement using public data sets to come to their results. Other work also investigates different variations of dead reckoning, such as Mc Coy et al.'s neuro reckoning (2007) or Dong's (2013) adaptation using target prediction to improve dead reckoning results.

Overall, latency compensation via extrapolation suffers from similar disadvantages as other in-game latency compensation methods. Uncertainty in a prediction may lead to game inconsistency, and a reduction in PP and experience follows suit. While extrapolation methods promise a more generalizable latency compensation approach – dead reckoning, for example, is used in most commercial FPS games– they are still limited to certain games or genres. Although they potentially work well in games with motion, such as racing games, they may not work at all in games without game object motion, such as point-and-click adventures. Previous work, thus, proposed different methods to compensate for latency independently of the actual game to overcome this challenge of missing generalizability. We will elaborate on these methods in the following.

2.3.3 Game-external Latency Compensation

Contrary to a game-internal latency compensation system, external compensation is not specifically tailored to one game or type of game but instead aims to reduce latency by altering or optimizing the origin of different types of latency. Such systems could, for example, aim to optimize the network utilization between client and server in a multiplayer setting to reduce network latency (Sun, 2019) or to reduce local processing latency caused by the central processing unit (CPU) accessing data from the RAM by intelligently scheduling, dividing, and throttling memory requests (Cheng et al., 2010). However, since these approaches typically are not directly integrated into games, developing, implementing, and evaluating such systems is often out of reach for game developers. This is also shown in the low number of previous works investigating game-external latency compensation systems in a gaming setting. Nevertheless, there are prior publications investigating game-external compensation methods, which we will discuss in the following.

Sun and Claypool (2019), for example, used a custom cloud-based game streaming system to investigate different types of streaming technologies. As already briefly discussed in Chapter 1, cloud gaming refers to a technology in which a video game is running on a remote server and accessed and played by a client via the Internet. The server is responsible for processing inputs, calculating game events, and rendering the game. All computationally expensive operations, such as physics, pathfinding, or AI behavior, are outsourced to the server. After calculating the current state of the game, the server sends it back to the player using a continuous video stream. The player receives the stream on their local device and interacts with the game as with a locally played video game. The streaming technology has the major advantage that players do not require high-end gaming equipment (such as GPUs, CPUs, or RAM) to play the latest game and do not need to download and install video games (which is becoming an ever-increasing concern with recent games being as large as 250 GB (Orland, 2022)) since they are already pre-installed on the server (Chen et al., 2013). However, since the game is practically played via the Internet, cloud gaming scenarios are heavily affected by latency. Thus, to minimize this latency, Sun and Claypool evaluated the interplay between the required bitrate and the perceived responsiveness by players of different streaming approaches. The authors implemented three streaming approaches: (1) Image-based streaming – in which the stream is based on single images, (2) video-based streaming – in which the stream is based on compressed continuous images, and (3) graphics-based streaming – in which no visual material is sent from the server to client, but rather rendering instructions. In graphics-based streaming, the client itself is responsible for implementing the instructions. This approach still limits the computational load of the client compared to a classical rendering pipeline since the server provides pre-calculated information such as details on lighting, rasterization, and viewport-transformation (cf., for examples Wikipedia, (2023b)), but is at the same time more demanding than just displaying an image or video. In a study in which 30 participants played a streamed game with different streaming technologies, the author found that graphics-based streaming requires the least bitrate and, consequently, was most robust for latency. The authors discuss their results in light of current commercial game streaming servers. They argue that most services nowadays use video-based streaming, which is still considerably more resource-friendly than image-based streaming, but could improve their services if they switched to a graphics-based approach. However, as described, graphics-based rendering is still computationally demanding and, thus, negates one of cloud gaming's key advantages of playing modern video games on almost any device. Nevertheless, since latency is the number one reason for low adaption rates of cloud gaming services, providers should consider finding a middle ground.

Other work in the same line of optimization approaches by Petlund et al. (2008) investigated, based on similar work by Griwods and Halvorsen (2006), how the standardized transmission control protocol (TCP) (Force, 2020) protocol can be specifically tailored to accommodate modern multiplayer games. In their work, the authors modified the TCP with several adjustments. Specifically, they reworked TCP's timeout mechanism and bundled network packages. In its original implementation, the TCP protocol sends packages and waits for a confirmation from the receiver. If the confirmation is not received within a certain timeframe – this is called a timeout – TCP re-sends the package. Petlund et al. reduced this timeframe by about three-thirds and thus enabled a much faster and more responsive re-transmission of timed-out packages. Additionally, the authors bundle small network packages, which all would require individual handshakes between sender and receiver (transmission and confirmation), into larger bundles that only require a single transmission/confirmation pair. To test their TCP variation, they run multiple simulations with AI bots playing a small game with 130 ms of latency. Overall, the authors found that their TCP optimization performed significantly better in the latency setting than the standard TCP protocol. Although Petlund et al. found significant improvements in network performance, measured in data loss and transmission times of network packages, they did not evaluate their approach with human players in an ecological-valid setting. Hence, it is unclear how latency compensation, as proposed by the authors, would affect PP and experience.

Other work investigating game-external latency compensation is highly diverse. For example, Ababsa et al. (2004) compared different methods to reduce the latency of head-mounted AR headsets, while Jerald et al. (2007) discuss and evaluated various latency compensation methods for VR headsets. However, there is one approach that is of particular interest in the context of this thesis, which is latency compensation via ANNs.

2.3.3.1 Latency Compensation via Artificial Neural Networks

ANNs are a type of machine learning model inspired by the structure and function of the human brain (Hayman, 1999; Goodfellow et al., 2016). They are used for various tasks in HCI and are a fundamental component of deep learning. ANNs consist of interconnected nodes (neurons) organized into so-called layers. Neurons are the basic building blocks of ANNs. Each neuron receives input (data), processes it, and produces an output. These neurons are organized in layers. An ANN typically has several layers: (1) an input layer, which summarizes all neurons receiving raw data input, (2) a variable number of hidden layers, which are responsible for performing intermediate processing, and (3) an output layer, which produces the final result or prediction. If an ANN has more than one hidden layer, the network is considered to be deep (hence, deep learning), albeit a clear definition of deep learning does not exist. To learn, ANNs utilize a process called training. In the training, several algorithms adjust the network weights and biases based on the error between a predicted output and the actual target values. Weights and biases are a fundamental part of each neuron. A weight parameter quantifies the connection strength between neurons, while biases adjust the neuron's activation function. The variable and adaptable structure of ANNs allows them to learn complex and non-linear relationships in data. In recent years ANNs have been successfully used for a range of tasks; For example, to classify handwritten letters (Cohen et al., 2017), to differentiate between cloths (Xiao et al., 2017), or to detect objects in general (He et al., 2017), to alleviate autonomous driving (Fridman et al., 2018) or to help classify skin cancer (Jaleel et al., 2012). In HCI, ANNs have been used to predict users' hand motions based on EMG signals (Ahsan et al., 2011), to detect facial emotions (Chowdary et al., 2021), or to identify users via their voice (Bae et al., 2016).

Since ANNs are highly adaptive, extremely flexible, and showed remarkable success in other tasks, there is also work investigating their use as a means to compensate for latency. Buker et al. (2012), for example, used ANNs to compensate for the latency of head-worn displays used for AR. The authors aimed to reduce latency-induced simulator sickness. To achieve this, Buker et al. trained several ANNs to predict each axis of head motion – x, y, and z (yaw, pitch, and roll). In two experiments, the authors evaluated their system. In the first experiment, Buker et al. simulated participants using the head-worn display with different levels of latency. The results of this first experiment showed that head motion, head motion frequency, and latency are related to each other. The authors used this knowledge to parameterize stimuli in the second experiment with 48 participants. In the follow-up experiment, Buker et al. tested different levels of head motion frequency (which causes different levels of latency), and their ANN-based prediction system either enabled or disabled. The authors found that their predictive latency compensation significantly reduced the amount of simulator sickness participants reported. They concluded that ANNs, and predictive systems in general, are well-suited to counteract the negative effects of latency in head-worn interactive systems.

Other work, inspired by Buker et al.'s work (2012), investigated a similar approach to reduce latency of touchscreen-operated smartphones. Henze et al., for example, discusses that current touchscreen devices have latency of up to 100 ms and that this latency potentially has negative effects on UP. Hence, to overcome the adverse influence of latency, the authors propose a predictive method, which anticipates where a user's finger on the touchscreen will be in the future. This prediction can be used by the system –

in this case, a smartphone – to increase the responsiveness of the touchscreen. To achieve this, Henze et al. developed and published an application that was able to record the users' touch interactions, such as touch taps and strokes. Using the application, the authors gather a large amount of ecology-valid touch interactions. This data was then used to train the ANN to predict touch interaction in 33,33 ms, 66,67 ms, and 100 ms. In a set of two studies, the authors evaluated their predictive model. In the first study, participants had to perform three different tasks while Henze et al. recorded the interaction. This recorded information was later used to evaluate the performance of their previously trained ANN. The authors show that their ANN outperforms classical prediction methods such as extrapolation (cf. Section 2.3.2.4), meaning that their approach produces a smaller error for all three prediction frames (33,33 ms, 66,67 ms, and 100 ms). Hence, Henze et al. used the ANN-based prediction model in a second user study to evaluate its performance with real participants. In the study, 14 participants had to perform two tasks, a dragging task ajar to a standardized Fitts' Law task and a drawing task, with three levels of ANN-based prediction enabled (0 ms, 33,33 ms, and 66,67 ms). In the experiment, Henze et al. measured TCT for the Fitts' Law task and subjective feedback for the drawing task. Overall, the authors found that their predictive model was able to significantly increase UP in the Fitts' Law task. However, they also show that participants perceived a significantly higher level of jitter when using the 66,66 ms prediction. The authors conclude that since their latency compensation approach is software-based only, it is easily integrable in other smartphones with touchscreen. Hence, it is an easy and effective way to reduce the touch latency of smartphones. In later work, Henze et al. (2017) improved on their prior work by using more complex neural networks such as a long short-term memory (LSTM) architecture.

Building on Henze et al.'s work (2016; 2017), later work by Le et al. (2017) improved ANN-based touchscreen latency compensation even further. In their work, Le et al. gathered data from wrist-worn internal measurement units (IMUs) of users interacting with a touchscreen. Using the IMU data and touch interaction data (similar to Henze et al. (2016)), they trained different ANNs to predict a touch interaction's trajectory. In two studies, the authors show that an ANN-based latency compensation enriched with IMU data produces an even better prediction than the previous system relying exclusively on software-based reductions. Le et al. demonstrate that participants reached a significantly higher throughput while the ANN's prediction error was significantly smaller than previous approaches.

In summary, previous work highlights several different areas and use cases for ANN-based systems. Specifically for latency compensation, prior work illustrates great applicability and success when used to decrease latency in AR, VR, or touchscreen-operated devices. Despite this success, ANNs have not yet been used to compensate for the effects of latency in video games.

2.3.4 Summary

In this section, we presented and discussed several approaches to latency compensation. Starting with Liu et al.'s (2022) taxonomy on latency compensation, we illustrated different methods for game-internal latency compensation. We highlighted approaches such as latency exposure (Gutwin et al., 2004; Vaghi et al., 1999; Smit et al., 2009), time manipulation (Sun & Claypool, 2019; Mauve et al., 2004; Liang & Boustead, 2006), world adjustment (Bateman et al., 2011; Ivkovic et al., 2015; Lee et al., 2019), and prediction-based methods (Burgess & Katchabaw, 2006; Chen et al., 2007; Lee & Chang, 2015; Savery et al., 2014). We continued by elucidating game-external methods which aim to optimize the underlying reason for latency, such as network traffic (Petlund et al., 2008; Griwodz & Halvorsen, 2006) or memory optimization (Cheng et al., 2010). We closed this review of game-external latency compensation methods by shedding light on the use of ANN-based systems for latency compensation (Buker et al., 2012; Henze et al., 2016; Le et al., 2017).

In summary, the presented approaches to compensate for the negative effects of latency are as diverse as the effects of latency on players and the gaming session itself. While there are many promising approaches, they all come with caveats, such as being perceived as unfair (K. Lee & C. Chang, 2017; Bernier, 2001), producing a high level of uncertainty (Wikipedia, 2023a; Pantel & Wolf, 2002) or having to wear additional hardware (Le et al., 2017). Although ANN-based systems are promising (Henze et al., 2016; Henze et al., 2017; Buker et al., 2012; Ahsan et al., 2011), they have not yet been used to compensate for latency in video games.

2.4 Synthesis of Related Work

In this chapter, we illustrated and discussed previous work investigating latency and its effects in interactive systems and video games. Building on this, we showed how previous work aimed to compensate for the negative effects of latency in video games and interactive systems more generally. In the following, we synthesize the gained understanding to build a foundation for this thesis's RQs. The goal of the proposed RQs is to deepen our understanding of latency to develop novel latency compensation techniques. To achieve this, we first need an understanding of how latency variation and perception modulate the effects of latency.

Since overall perceived latency by players in interactive systems is built by various different sub-latency, such as processing latency, network latency, or input latency, it never really is constant (Casiez et al., 2017). In ecologically valid gaming settings, it rather varies between different values. However, previous work investigating the effects of latency in video games often treated it as a constant (Liu et al., 2021d; Beigbeder et al., 2004; Beigbeder et al., 2004). Hence, it is unclear how (and if) latency variation affects PP and GX in video games. Building on this, we formulate our two first RQs as follows:

RQ1: *"Does small-term latency variation affect Player Performance and Game Experience?"*

RQ2: *"Does long-term latency variation affect Player Performance and Game Experience?"*

Similarly, previous work investigated how different types of latency alter PP and GX. Although auditory elements fundamentally shape a player's performance and experience in video games (Grimshaw et al., 2008; Gormanley, 2013), surprisingly, the effects of auditory latency in video games have not been investigated. Hence, we formulate our third RQ as follows:

RQ3: *"Does auditory latency affect Player Performance and Game Experience?"*

In the same vein, previous work researched how different game characteristics, such as its pacing (Sabet et al., 2019b), its rules (Claypool & Claypool, 2010), or its in-game task complexity (Schmidt et al., 2017) alter the effects of latency. However, while the in-game perspective is a crucial part of every game and shapes how players visually perceive the game, it is unclear how the in-game perspective alters the effects of latency. Hence, to investigate the interaction between the in-game perspective and latency, we formulate our fourth RQ as follows:

RQ4: *"How does the in-game perspective alter the effects of latency?"*

Previous work also highlighted that how players perceive latency, or how it is communicated to them via the game interface, may fundamentally alters their experience (Kosch et al., 2022; Michalco et al., 2015). This circumstance is even used in compensation methods based on latency exposure or latency concealment, as we learned in Chapter 2. However, it is unclear if this effect is of a purely technical nature, i.e., a real effect of latency or an expectation-based effect such as a Placebo (Beecher, 1955; Arnstein et al., 2011) or Nocebo (Colloca & Barsky, 2020) effect. Hence, to investigate how the expectation of latency alters PP and GX, we formulate our fifth RQ as follows:

RQ5: *"How does the expectation of latency alter Player Performance and Game Experience?"*

Building on the knowledge gained from answering the previous RQ, we aim to design novel latency compensation methods to counteract the negative influence of latency. Previous work showcased several different methods to account for latency directly in games, such as time warp (K. Lee & C. Chang, 2017; Bernier, 2001), geometrical manipulation (Lee et al., 2019), or dead reckoning (Pantel & Wolf, 2002). However, most of these methods have disadvantages, such as being specific for one game or genre (Pantel & Wolf, 2002), being perceived as unfair (K. Lee & C. Chang, 2017; Bernier, 2001), or potentially producing in-game inconsistency (Liu et al., 2022). Thus, previous work also investigated game-external latency compensation techniques, such as optimizing network (Sun, 2019) or memory usage (Cheng et al., 2010). One particular promising technique is to use ANNs for user input prediction (Le et al., 2017). Previous work showed remarkable success in predicting user input and subsequently in reducing the perceived latency (Henze et al., 2016; Le et al., 2017). However, although ANNs have shown promising results, they have not been used in video games to compensate for the negative effects of latency. Hence, we formulate three RQs to investigate the use of ANN-based latency compensation in video games. For each RQ, we gradually increase the complexity of the problem. While the first RQ aims to answer if full access to game-internal states allows an ANN to compensate for latency in custom video games, the second RQ already increases the prediction complexity since it limits the ANN to use only game-external information of a commercial and slow-paced video game. The last RQ accumulates all previous work, in which we investigate if ANNs can be used to adaptatively compensate for latency in a commercial, fast-paced shooting game while only using information readily available to the player. Hence, we formulate the last three RQs of this thesis as follows:

RQ6: *"Can Artificial Neural Networks be used to compensate for the negative effects of latency in a custom video game?"*

RQ7: *"Can Artificial Neural Networks be used to compensate for the negative effects of latency in a commercial slow-paced video game?"*

RQ8: *"Can Artificial Neural Networks be used to adaptively compensate for the negative effects of latency in a fast-paced commercial video game?"*

3

Understanding Latency Variation

To design novel latency compensation systems which improve PP and PX in high latency settings, we require a thorough understanding of how latency variation affects players. As we learned in the previous chapter, latency is never really a constant as various factors, such as the local gaming setting, the network connection, or used peripheral devices, influence it. Hence, we conducted two user studies to investigate how changing latency affects players and their gaming session. In the first study we artificially induced ephemeral latency variation in to a FPS game. In this controlled setting latency changed its value within milliseconds to mimic real-world local latency jittering. We found that short-term latency variation neither affected PP nor PX (RQ1). After establishing that small-term latency variation does not affect the gaming session, we conducted a second user study investigating how long-term latency switches alter PP and PX. In this study, latency stayed at a pre-defined value for up-to forty seconds before abruptly switching to another level. Latency stays constant on the second level, again for up-to 40 seconds before switching back to its initial level. Similarly to the first study, it is paramount to learn how players are influenced by switching latency to derive guidelines on how future latency compensation methods should behave. The results of the second study show that long-term latency switches significantly impaired the participants' feelings of flow. Additionally, we found negative effects on the perceived tension, the experienced challenge, and the players' performance (*RQ2*).

This chapter presents, the experimental setting, the procedure, and the results of both studies. All findings are discussed in detail within the context of the conducted study. Additionally, we elucidate on more generalizable implications and conclusions.

3.1 Understanding Small-Term Variation of Latency (Study I)

Research focusing on latency has a long history, for example, MacKenzie and Ware's seminal work (1993), in which the authors showed that adding latency increases movement time and error rate in pointing tasks. Since then, scientific community and gaming enthusiasts developed a number of methods for measuring the latency of different systems such as desktop PCs (He et al., 2000; Kaaresoja & Brewster, 2010; Schmid & Wimmer, 2021; Casiez et al., 2015), smartphones (Deber et al., 2016; Kämäräinen et al., 2017), VR systems (Di Luca, 2010), or input devices (Wimmer et al., 2019). In practice, however, the base latency of a system is not constant. It varies depending on the polling rate of the USB connection (Wimmer et al., 2019), game loops, processing times, and vertical display synchronization. However, in most studies that investigate the effect of latency on performance, a constant amount of latency is added to the system response for each condition, and only these constant values are reported. Rarely do authors measure or report how large the latency variations of their setup are.

This might be a problem. In case small-term latency variation has a noticeable effect on players' performances in video games, consequences would be significant for the development of novel latency compensation techniques and for the validity of previous findings. However, currently, it is unknown if small-term latency variation affects PP and GX (RQ1, cf. Table 1.1).

This section is partly based on the following article:

Schmid, A., Halbhuber, D., Fischer, T., Wimmer, R., & Henze, N. (Oct. 2023). "Small Latency Variations Do Not Affect Player Performance in First-Person Shooters." In: *Proc. ACM Hum.-Comput. Interact.* 7.CHI PLAY, pp. 197–216. DOI: 10.1145/3611027.

3.1.1 Background and Research Rationale

In this work, we investigate how small variations of latency influence PP and PX. We call this small-term variation latency jitter and it is the variation of local latency that occurs with each input, as opposed to network jitter, which is caused by buffering of packages in network communication (Zheng et al., 2001). We conducted a within-subjects study ($n = 28$) to investigate how latency jitter influences PX and PP when playing a fast-paced FPS. We utilized an FPS since previous work showed that they are particularly negatively affected by latency (Claypool & Claypool, 2006). To operationalize latency jitter, we varied the level of mean base latency (low = 50 ms vs. high = 150 ms) and the level of variation (low = ± 0 ms vs. high = ± 50 ms). To maximize internal validity, we used a system optimized for extremely low and constant end-to-end latency for our study apparatus.

The results of our data analysis consolidate previous findings and show that a high latency affects game performance and experience. However, we found no effect of small-term latency variation on neither the players' performances nor their experiences (RQ1). Furthermore, investigating the interaction between base latency and its variation, we found no effect on most of our measures. However, we found that players derived a greater sense of meaning when playing the game with low base latency and high variation compared to a high latency with low variation.

Our findings are crucial to latency and video games research since we show that small-term latency variation does, generally, not affect performance and experience, thus, validating previous work defining latency as a constant. Furthermore, in light of latency compensation methods, our work shows that small uncertainties and errors in the compensation are acceptable.

3.1.2 Method

To investigate how local latency jitter affects experience and performance in video games, we conducted a within-subjects study with 28 participants playing a modified version of Cube 2: Sauerbraten[1]. Participants played with two levels of mean base latency and two levels of latency variation.

3.1.2.1 Apparatus

In our work, we used the open-source game Cube 2: Sauerbraten, a fast-paced first-person arena shooter developed in 2004. In Sauerbraten, players control an avatar equipped with a virtual weapon. The game's goal is to navigate the avatar through different levels

[1] `http://sauerbraten.org/`

and shoot other entities, such as other players or AI-controlled bots, to survive and gain points. Despite its age, Sauerbraten enjoys a lively and consistent community, which recently even launched an official Steam[1] fork of Sauerbraten called Tomatenquark[2]. The rationale for using Sauerbraten in our work is threefold: First, FPS games, such as Sauerbraten, are susceptible to latency as demonstrated by previous work (Claypool & Claypool, 2006; Liu et al., 2021d; Liu et al., 2021c). In FPSs, latency leads to players being less accurate, scoring fewer points, and having a reduced gaming experience. Second, Sauerbraten is an open-source project that allows us to modify and control every aspect of the gaming session, such as what weapons players are allowed to use, how AI-controlled bots behave, and which maps are played. Furthermore, in contrast to proprietary video games, we have direct access to the game's source code, which makes low-level logging of game events straightforward. Third, preliminary tests have shown that Sauerbraten is highly performant and has a very low impact on the system's end-to-end latency. High game performance is crucial in our work since we aim to investigate low amounts of latency variation. Hence, every fluctuation, for example, induced by a game with high demand on system resources, may potentially bias our work.

In the following section, we first highlight our modification to the game to make it fit to be used in a study with high internal validity. Then, we elaborate on how we measured the local latency of our setup since the local latency needs to be factored in all future investigations.

Modification of Cube 2: Sauerbraten

Sauerbraten is typically played against other humans or multiple bots. However, to prevent differing player skills and play styles to confound our work, we modified the game so one player only faces one bot at a time. We used Sauerbraten's built-in level 75 bots which corresponds to a medium to hard difficulty. We set the difficulty level of the bot in a way that players are neither under- nor over-challenged by it.

As the map and the player's weapon fundamentally change how the game is played, we restricted both for our study to prevent confounding effects. We restricted rounds to

[1] `https://store.steampowered.com/`
[2] `https://store.steampowered.com/app/1274540/Tomatenquark/`

the in-game map Teahupoo. Furthermore, we disabled all virtual weapons except the standard pistol, which we modified to have unlimited ammunition, and prevented players from changing or picking up new weapons during the gaming session.

In the next step we removed all keyboard shortcuts that were not essential for our work, such as using an in-game object or opening menus such as the map overlay. Lastly, we added custom logging functions to track different game events. We logged the number of shots, hits, misses, players deaths, and bot deaths. Figure 4.5 shows the in-game view of the player (left) an an aerial view of the game map Teahupoo (right).

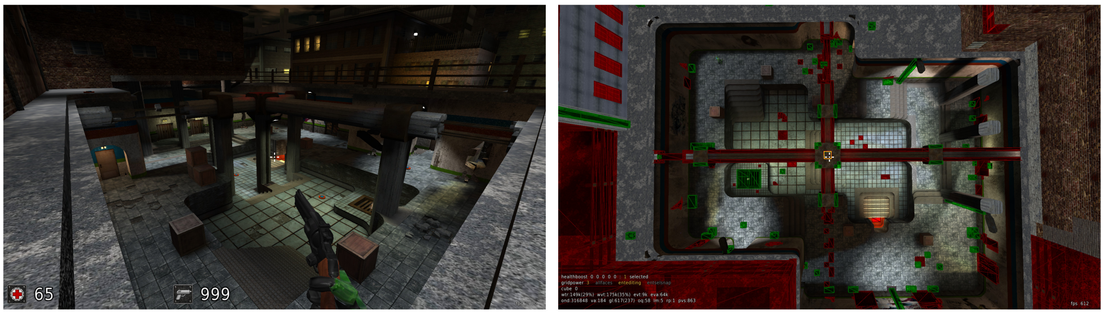


Figure 3.1: Shows two screenshots from the first-person shooter game Cube 2: Sauerbraten. The left depicts the player's viewport while playing. The screenshot shows the player's weapon, health, and ammunition. The right shows an aerial view of the in-game map Teahupoo, which was used for all gaming rounds.

Local Latency

To investigate the effects of local latency jitter on players of our modified FPS game, we needed a reliable method to add latency to the test system and a system with minimal and constant end-to-end latency.

To add latency to the system, we used a C program using the evdev[1] library to capture and block input events from physical input devices, similar to Liu and Claypool's EvLag (2021). For each input event from a mouse or keyboard, the program creates a thread that waits for a specified amount of time before invoking the input event with a virtual input device provided by evdev. The program allows for adding constant or varying delays with uniform or normal distribution.

[1] https://linux.die.net/man/4/evdev

Additionally, the program can be controlled via inter-process communication using a first in, first out (FIFO) queue so added latency can be adjusted between conditions without needing to restart the program.

For our study, we used a HP Pavillon Gaming 790 desktop PC with an Intel i7-8700 (3.2 GHz), Nvidia GTX 1080, 16 GB DDR4 RAM running Debian Buster 5.10 with proprietary Nvidia graphics drivers (version 470.103.01). As periphery, we used an ASUS ROG Strix XG248Q at 1920×1080 pixels with 240 Hz, a Logitech G15 gaming mouse with an input device latency of 2.17 ms (SD = 0.3 ms) as reported by Wimmer et al. (2019) and a Logitech G213 gaming keyboard with a input device latency of 2.55 ms (SD = 0.34 ms).

We measured the end-to-end latency of our system with Schmid and Wimmer's proposed method (2021). An Arduino-based device electrically triggers a button click on an input device and measures the time until a brightness change on the computer's monitor is detected with a photo sensor. We used this setup to trigger a mouse click leading to a gun shot in Sauerbraten and attached a photo diode at the center of the screen so a visible muzzle flash would stop the latency measurement. The measured mean end-to-end latency of our system running Sauerbraten is 9.11 ms (SD = 1.4 ms, range = 6.2 ms - 15.5 ms). Detailed measurement results are depicted in Figure 3.2. All latency values reported in the remainder of the study incorporate this local latency without explicitly mentioning it.

3.1.2.2 Study Design

We used two independent variables (IVs) in a 2×2 within-design to control for mean base latency and latency variation: BASE (1) refers to the mean baseline latency participants played with. BASE has two levels: (I) *low* which refers to playing with 50 ms, and (II) *high* which refers to playing with 150 ms of base mean latency. The second IV is VARIATION (2) and defines how much latency varied around the mean base latency BASE. VARIATION has also two levels: The first level (I) *low* refers to no variation. The second level *high* (II) refers to a variation of ±50 ms. This entails that the actual latency, for example, in a *high* BASE / *high* VARIATION round varied from 100 ms to 200 ms (150 ms ± 50 ms). Latency was applied to each input of the computer mouse (movement and clicks) and keyboard following a uniform distribution. Therefore, during conditions without VARIATION, all input events are delayed by a constant amount. During conditions with VARIATION, random delays are added to each event, resulting in jittering mouse

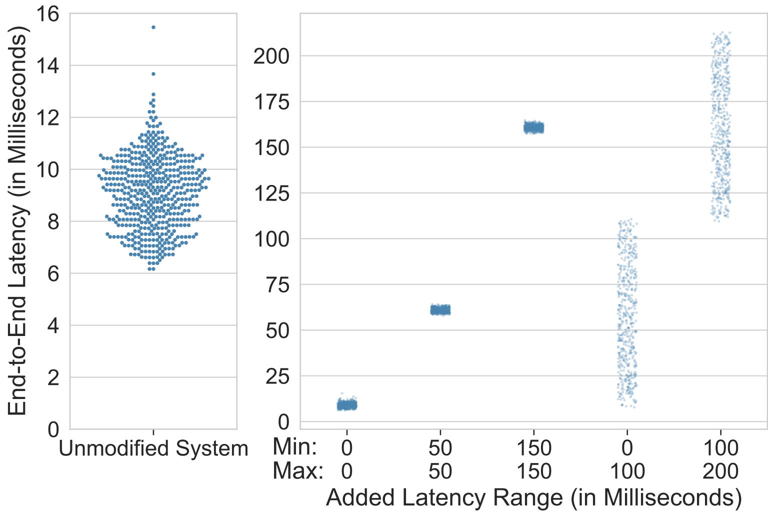


Figure 3.2: Results of end-to-end latency measurements for different latency conditions. The plot on the left depicts the system's end-to-end latency running Cube 2: Sauerbraten without any added latency. The plot on the right shows the system's end-to-end latency for the different conditions used in our study. Each measurement series consists of 500 individual measurements with random delays in between. The measuring probe was attached to the top left corner of the monitor and vertical synchronization was disabled.

movement. In those conditions, it is also possible that the order of rapid consecutive input events changes if high latency is applied to the first event and low latency is applied to the second event. The levels of BASE are in line with previous work, which shows that latency in the wild reaches values up to 150 ms (Ivkovic et al., 2015). However, the levels of VARIATION are constrained by the chosen levels of BASE since we are not able to decrease latency below 0 ms. Thus, the lower bound of BASE defines the upper bound of VARIATION.

As our method for measuring the system's end-to-end latency requires measuring probes to be attached to the input devices' circuit boards, as well as bright flashing regions on the screen, we did not measure the system's latency during the study. Therefore, we validated all latency conditions beforehand with the method described in section 3.1.2.1. Results can be seen in Figure 3.2.

To measure PP, we utilized a range of dependent variables (DVs). In line with previous work (Liu et al., 2021a), PP is measured with three DVs: (1) *Hitrate* – which quantifies the ratio of total shots to successful hits, (2) *KD-Ratio* – which refers to the ratio of player kills and deaths, and lastly (3) *TotalKills* – which corresponds to the total amount of enemy kills per round.

To evaluate GX, we used the 30-item PXI (Abeele et al., 2020). We used the PXI, since the instrument was rigorously validated and tested in multiple studies (Abeele et al., 2020). Given its multi-dimensionality, the instrument allows for an in-depth analysis of player experience, contrary to other work, which, for example, uses single-item questionnaires to assess GX. The PXI is divided into two categories: (1) functional consequences and (2) psycho-social consequences. The functional consequences dimension is built by five subscales: Ease of Control, Progress Feedback, Audiovisual Appeal, Clarity of Goals, and Challenge and encompasses essential game aspects such as gameplay mechanics, controls, and audio-visual elements. Latency substantially impacts these functional components. For instance, a delay in player actions being registered due to high latency can result in a diminished sense of control (Winkler et al., 2020), reduced responsiveness (Beigbeder et al., 2004), or a feeling of an inappropriate challenge, ultimately leading to a less satisfying gameplay experience.

The psycho-social consequence dimension of the PXI delves into the social and psychological ramifications of gaming. The dimension describes second-order emotional experiences derived from playing and it also contains five subscales: Mastery, Curiosity, Immersion, Autonomy, and Meaning. Potentially, latency affects each of the psycho-social subscales individually and differently as each of dimension shapes one crucial aspect of the overall gaming experience.

3.1.2.3 Procedure

Participants were met and greeted at the laboratory by an experimenter. They were not informed about the exact details of the study (to investigate the effects of latency jitter), to prevent a bias induced by the participants' expectations (Kosch et al., 2022). Hence, participants were just told to test a game. Subsequently, participants gave informed consent to our data collection and were briefed on the further course of the study. After we explained the controls and the objective of the game, each participant played six rounds of Cube 2: Sauerbraten. Each round lasted for five minutes. The first and last rounds of the study were always played without artificially added latency (BASE) or variation (VARIATION) to control for exhaustion-induced performance degradation. In the remaining four rounds, we altered BASE and VARIATION. Each of the four rounds represents one of the unique combinations of BASE and VARIATION. The rounds with changing BASE and VARIATION were counterbalanced using a balanced Latin Square to prevent sequencing effects. After each round, participants filled out the PXI on a separate device and had a short break, which allowed us to alter the game for the next round. Upon finishing all six rounds, participants filled out a demographic questionnaire and the study was concluded. In a short debriefing, we informed participants about the exact purpose of the study. We estimated a total duration of one hour for participation. The study was designed, conducted, and analyzed following the research ethics policy issued by the University of Regensburg and received clearance per this policy.

3.1.2.4 Apparatus and Task

As apparatus, we utilized the low-latency hardware setup described in section 3.1.2.1. Our modified version of Sauerbraten was executed in full-screen mode.

In each of the six rounds, participants controlled an avatar equipped with a virtual weapon. In the game world, participants were free to roam and fought against one AI-controlled bot. The participants' objective in the game was to shoot the adverse bot as often as possible without getting shot by the bot. After shooting a bot three times, the bot died and respawned at a random location in the game world. If the bot hits the player's character three times, the player character died as well, and also respawned at a random location in the game world. Players obtained points for successfully killing an enemy bot. However, they did not lose points if they did not hit the bot or got killed themselves. This overall procedure was repeated six times (four times for each unique combination of BASE and VARIATION and two times with an unaltered game version).

3.1.2.5 Participants

Since previous work showed that the effects of latency are reliably detectable with a relatively small number of participants (Long and Gutwin (2018): 20 and 18 participants; Liu et al. (2021): 25 participants), we recruited 28 participants (24 male, four female) using our institution's mailing list and advertisement in a local gaming club. Participant age ranged from 20 years to 33 years with a mean age of 24.6 years (SD = 3.4 years). Participants' prior experiences with FPS games ranged from 10 hours to 18 000 hours, with 2081 hours (SD = 525 hours) on average. Their self-reported skill level on a 10-point

Likert-item ranged from 2 points to 9 points (M = 5.2 points, SD = 2.1 points). Students participating in our study were eligible to obtain course credits as compensation for their participation.

3.1.3 Results

In this section, we report the results of our data analysis. We structure this section by IVs instead of DVs for better readability. Additionally, we only report p-values in body text. However, full inferential statistical data can be found in Table 3.1. The collected data were screened for normal distribution using Shapiro-Wilk tests (Gaussian distribution assumed if $p > 0.05$). All measures, except the autonomy subscale of the PXI ($p = 0.632$), showed a violation of normal distribution (all $p < 0.05$). For inferential assessment of non-parametric data for BASE (*low* vs. *high*) and VARIATION (*low* vs. *high*), we thus applied a rank-aligned 2×2 ART-ANOVA (Wobbrock et al., 2011) with repeated measures on both factors. Analogously, we used a conventional 2×2 ANOVA for the analysis of parametric data. Participants' IDs were entered as error term in both ANOVAs to account for random variation induced by individual participants. Effect sizes (η_p^2) are interpreted following the recommendation by Field (2013). Post-hoc tests are alpha-corrected.

3.1.3.1 Null Hypothesis Testing

Base

ART-ANOVA revealed significant main effects of BASE on *Hitrate*, *KD-Ratio*, and *TotalKills* (all $p < 0.024$). Participants performed significantly better when playing with *low* LATENCY than with *high* LATENCY. They were more accurate, had a better kill-to-death ratio, and killed more bots overall. Figure 3.3 depicts *Hitrate* (left), *KD-Ratio* (center), and *TotalKills* (right) grouped by unique combinations of BASE and VARIATION.

ART-ANOVA and ANOVA showed significant main effects of BASE on all subscales of the PXI (all $p < 0.015$). Overall, participants had a significantly better GX when playing with *low* than *high* BASE. Participants rated the game as easier to control, were more satisfied with the progress feedback provided by the game, found the game to be more appealing on an audiovisual level, had an easier time grasping the game's goal, and

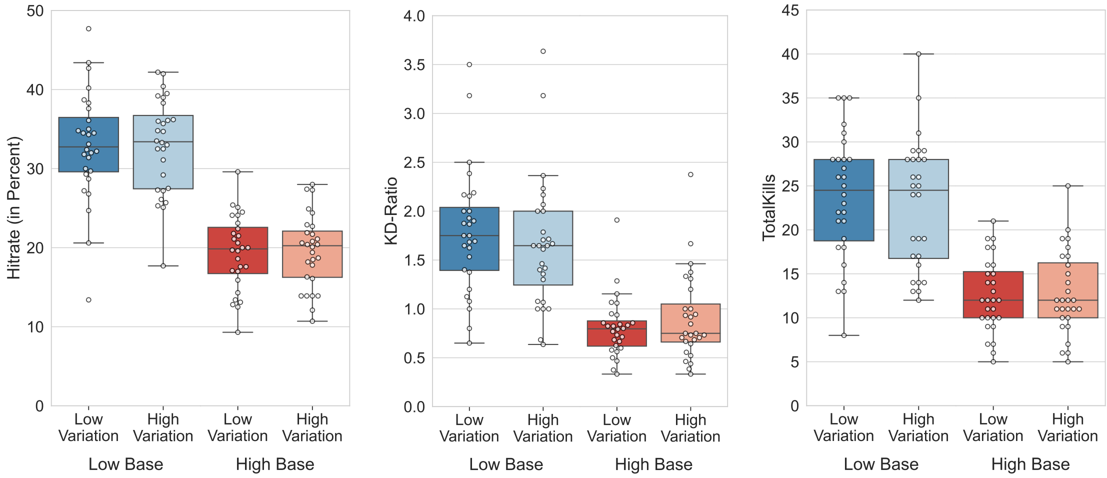


Figure 3.3: Depicts boxplots of *Hitrate* (left), *KD-Ratio* (center), and *TotalKills* (right) for each combination of BASE and VARIATION. Participants performed significantly better when playing with *low* LATENCY than with *high* LATENCY. They were more accurate, had a better kill-to-death ratio, and killed more bots overall. There was no effect of latency variation on any of the dependent variables in both BASE latency conditions.

found the challenge provided by the game to be more appropriate when playing with the lower level of BASE. Furthermore, players derived a greater extent of mastery, meaning, autonomy, and immersion in the *low* BASE conditions.

Variation

We analysed the impact of VARIATION with the same systematic as BASE. ART-ANOVA revealed no significant main effect of VARIATION on *Hitrate*, *KD-Ratio* and *TotalKills* (all $p > 0.365$). Furthermore, ART-ANOVA and ANOVA showed no significant main effects of BASE on any subscale of the PXI (all $p > 0.131$). We found no significant effect of VARIATION. Hence, VARIATION as an isolated factor did not alter players' performances or GXs.

Base × Variation

Lastly, we analysed the interaction BASE × VARIATION and its effects on our measures. ART-ANOVA showed no interaction effect on *Hitrate*, *KD-Ratio*, and *TotalKills* (all $p >$ 0.157). Similarly, ART-ANOVA and ANOVA revealed no significant interaction effect on all subscales of the PXI (all $p > 0.174$) except on the meaning dimension.

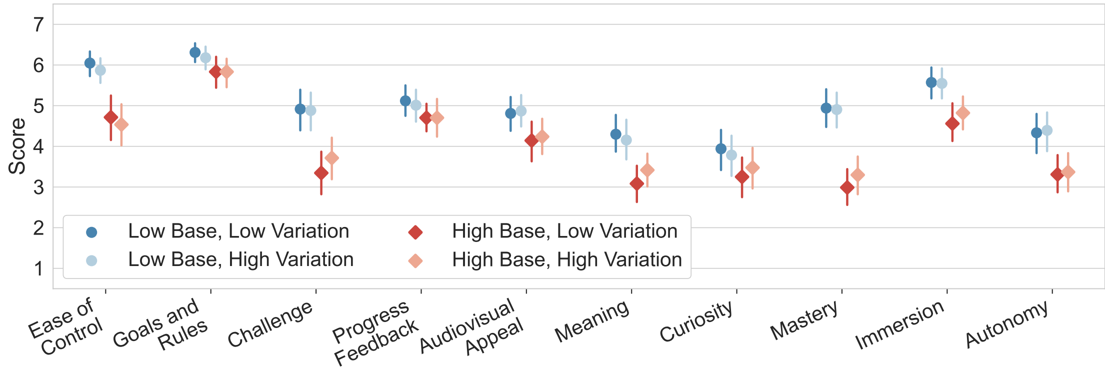


Figure 3.4: Depicts the results of *Ease of Control*, *Goals and Rules*, *Challenge*, *Progress Feedback*, *Audiovisual Appeal*, *Meaning*, *Curiosity*, *Mastery*, *Immersion*, and *Autonomy* for each combination of BASE and VARIATION. Participants rated the game as easier to control, were more satisfied with the progress feedback provided by the game, found the game to be more appealing on an audiovisual level, had an easier time grasping the game's goal, and found the challenge provided by the game to be more appropriate when playing with the lower level of BASE. Furthermore, players derived a greater extent of mastery, autonomy, meaning, autonomy, and immersion in the low BASE conditions.

ART-ANOVA found a significant interaction effect (BASE × VARIATION) on meaning ($F(1, 27) = 5.585$, $p = 0.025$, $\eta_p^2 = 0.17$ / large). To entangle the interaction effect between BASE and VARIATION, we used paired alpha-corrected (Bonferroni) t-tests. T-tests revealed a significant difference between playing with *low* LATENCY / *high* VARIATION and *high* LATENCY / *low* VARIATION (t(27 = 4.117, p < 0.001, CI95[0.538, 1.605], d_{cohens} = 0.62 / medium (Cohen, 2013)), but no effect between *low* LATENCY / *low* VARIATION and *low* LATENCY / *high* VARIATION (t(27 = 1.317, p = 0.198, CI95[-0.099, 0.465], d_{cohens} = 0.06] / small) or between *high* LATENCY / *low* VARIATION and *high* LATENCY / *high* VARIATION (t(27 = 2.352, p = 0.158, CI95[-0.598, -0.067], d_{cohens} = 0.02] / small).

Participants playing with *low* LATENCY / *high* VARIATION derived a significantly greater level of meaning from playing the game ($M = 4.154$, $SD = 1.306$) compared to participants playing with *high* LATENCY / *low* VARIATION ($M = 3.083$, $SD = 1.239$) (Figure 3.4).

	Base			Variation			Base × Variation		
Performance Measures	$F(1, 27)$	p	η_p^2	$F(1, 27)$	p	η_p^2	$F(3, 81)$	p	η_p^2
Hitrate	257.091	**< 0.001**	0.90	0.085	0.772	< 0.01	0.001	0.951	< 0.01
KD-Ratio	5.671	**0.024**	0.17	0.846	0.365	0.03	0.632	0.433	0.02
TotalKills	211.101	**< 0.001**	0.88	0.842	0.371	0.02	2.115	0.157	0.07
PXI Subscale									
Ease of Control	40.280	**< 0.001**	0.59	2.431	0.131	0.08	0.026	0.872	0.01
Progress Feedback	6.712	**0.015**	0.19	< 0.001	0.991	< 0.91	0.839	0.367	0.01
Audiovisual Appeal	25.470	**< 0.001**	0.48	0.662	0.442	0.02	0.211	0.649	0.01
Goals and Rules	17.602	**< 0.001**	0.39	0.126	0.725	< 0.01	0.026	0.872	0.01
Challenge	39.463	**< 0.001**	0.59	1.068	0.311	0.03	2.087	0.160	0.07
Mastery	69.694	**< 0.001**	0.72	0.221	0.641	< 0.01	1.946	0.174	0.06
Curiosity	8.326	**0.007**	0.23	0.017	0.896	< 0.01	0.958	0.336	0.03
Immersion	31.721	**< 0.001**	0.54	0.675	0.418	0.02	0.775	0.386	0.02
Meaning	17.214	**< 0.001**	0.39	1.991	0.169	0.06	5.585	**0.025**	0.17
Autonomy*	23.471	**< 0.001**	0.93	0.099	0.698	< 0.01	< 0.001	> 0.999	0.01

Table 3.1: Results of the BASE and VARIATION ART-ANOVA and ANOVA (*) analysis. Each row represents one dependent variable and its analysis for either main effects of BASE and VARIATION or the interaction effect BASE × VARIATION. We found significant main effects of BASE on all measures and no effect of VARIATION. Investigating the interaction, we found that players derived a significant greater level of meaning from playing with low BASE / high VARIATION than playing with high BASE / low VARIATION.

3.1.3.2 Bayesian Inference

To further examine the effects of latency and its variation, we performed multiple Bayesian 2×2 RM-ANOVAs with BASE (*low* vs. *high*) and VARIATION (*low* vs. *high*) as factors. Null hypothesis significance testing (NHST) detects differences between distributions, thus, accepting or rejecting a null hypothesis. While useful for detecting statistical differences in data, NHST cannot determine if an insignificant difference indicates similarity between the studied data. To explore similarity in our data, we utilized a Bayesian analysis, which estimates the probability that the null hypothesis (i.e., no differences in distribution) is true, instead of rejecting it, as NHST does (Cleophas & Zwinderman, 2018; Wagenmakers et al., 2018). Unlike NHST, Bayesian inference calculates probabilities for both H_0 and H_1.

We used *JASP* (Wagenmakers, 2022) and followed the default prior probability distribution recommended by Wagenmakers et al. (2018) for Bayesian inference. For post-hoc testing, we used Bayesian t-tests, and corrected the posterior odds for multiplicity using Westfall's approach (Westfall et al., 1997; de Jong, 2019). To interpret Bayes factors (Lavine & Schervish, 1999; Kass & Raftery, 1995), which indicate the strength of evidence for H_0 over H_1, we followed the guideline of Lee and Wagenmakers (2014).

Base

A Bayesian 2 x 2 RM-ANOVA found extreme evidence ($0 < BF_{01} < 0.01$, *error* = 0.647 %) for a model that supports a true effect of BASE on *Hitrate*, on *KD-Ratio* ($0 < BF_{01} < 0.01$, *error* = 0.771 %) and *TotalKills* ($0 < BF_{01} < 0.01$, *error* = 1.184 %), which indicates that the gathered performance data is at least a hundred times more likely in support of a distribution in which BASE alters *Hitrate*, *KD-Ratio*, and *TotalKills*.

Similarly, we found extreme evidence for accepting a hypothesis that postulates a true effect of BASE on the subscales of the PXI, *Ease of Control* ($0 < BF_{01}$) < 0.01, *error* = 0.754 %), Audiovisual Appeal ($0 < BF_{01} < 0.01$, *error* = 1.317 %), Goals and Rules ($0 < BF_{01} < 0.01$, *error* = 1.272 %), Challenge ($0 < BF_{01} < 0.01$, *error* = 2.068 %), Mastery ($0 < BF_{01}$) < 0.01, *error* = 13.441 %), Immersion ($0 < BF_{01}$) < 0.01, *error* = 1.137 %), Autonomy ($0 < BF_{01} < 0.01$, *error* = 2.772 %), and Meaning ($0 < BF_{01} < 0.01$, *error* < 0.01 %) and very strong evidence on the subscales Curiosity ($BF_{01} = 0.029$, *error* = 0.961 %) and Progress Feedback ($BF_{01} = 0.061$, *error* = 0.842 %).

In summary the Bayesian inference of the effects of LATENCY consolidate our previous findings, which demonstrated that the mean latency – BASE – fundamentally altered PP and GX.

Variation

A Bayesian RM-ANOVA found moderate evidence for a model that implies no true effect of VARIATION on *Hitrate* ($BF_{01} = 5.143$, *error* = 0.781 %), on *KD-Ratio* ($BF_{01} = 5.061$, *error* = 1.195 %) and *TotalKills* ($BF_{01} = 5.061$, *error* = 1.021 %). Post-hoc investigations using Bayesian t-tests corrected for multiple testing within VARIATION (*low* vs. *high*) found moderate evidence that the achieved *Hitrate* ($BF_{01} = 6.758$, *error* < 0.001 %) and *KD-Ratio* ($BF_{01} = 6.556$, *error* < 0.001 %), and the amount of total kills *TotalKills* ($BF_{01} = 5.059$, *error* = 0.843 %) is not altered by the levels of VARIATION.

Investigating the effects of VARIATION on GX, we found extreme evidence against a model that supports a true effect of VARIATION on nine of the ten PXI subscales (all $BF_{01} > 100$, 1.419 < *error* < 6.733 %), only the investigation of Progress Feedback differed but still yielded very strong evidence in favor of H_0 ($BF_{01} = 68.138$), *error* =

2.101 %). Post-hoc investigations using Bayesian t-tests corrected for multiple testing within VARIATION (*low* vs. *high*) revealed weak to moderate evidence that the given scores in the subscales - Ease of Control (BF_{01} = 1.828, *error* = 0.012 %), Progress Feedback (BF_{01} = 6.049, *error* < 0.001 %), Audiovisual Appeal (BF_{01} = 5.403, *error* < 0.001 %), Goals and Rules (BF_{01} = 5.050, *error* = 0.052 %), Challenge (BF_{01} = 3.760, *error* = 0.013 %), Mastery (BF_{01} = 4.705, *error* = 0.040 %), Curiosity (BF_{01} = 6.653, *error* < 0.001 %), Immersion (BF_{01} = 4.226, *error* = 0.025 %), Autonomy (BF_{01} = 6.248, *error* < 0.001 %), and Meaning (BF_{01} = 4.202, *error* = 0.024 %) - are influenced by the levels of VARIATION.

Base × Variation

Lastly, we investigated the interaction between BASE and VARIATION. We found extreme evidence that the interaction affects *Hitrate* (BF_{01} < 0.01, *error* = 1.740 %), *KD-Ratio* (BF_{01} < 0.01, *error* = 8.277 %) or *TotalKills* (BF_{01} < 0.01, *error* = 7.857 %). We found the same symptomatic investigating the effects of BASE X VARIATION on the GX. We found extreme evidence for accepting a hypothesis that postulates a true effect of BASE X VARIATION on the subscales of the PXI, Ease of Control (BF_{01} < 0.01, *error* = 1.951 %), Audiovisual Appeal (BF_{01} < 0.01, *error* = 18.441 %), Goals and Rules (BF_{01} < 0.01, *error* = 5.486 %), Challenge (BF_{01} < 0.01, *error* = 1.849 %), Mastery (BF_{01} < 0.01, *error* = 3.452 %), Immersion (BF_{01} < 0.01, *error* = 1.604 %), Autonomy (BF_{01} < 0.01, *error* = 2.317 %), and strong evidence on the subscales Curiosity (BF_{01} = 0.257, *error* = 2.772 %) and Progress Feedback (BF_{01} = 0.292, *error* = 2.162 %). However, investigating the interaction effect on Meaning we found equal support for H_0 and H_1 (BF_{01} = 1, *error* = 4.016 %).

3.1.4 Discussion

The results of our NHST showed that a high base latency significantly affects PP and GX. However, we found no significant main effect of latency variation on either PP or GX. While most of our measures were unaffected by the interaction between latency and its variation, inferential analysis showed that players rated playing the game with low base latency and high variation better on the PXI's meaning subscale than playing with high latency and low variation. The meaning subscale, in general, refers to the level of meaning derived from playing the game. It quantifies how well players connected with the game, and how well they were able to resonate with what is important while playing the game on a psychosocial level (Abeele et al., 2020).

We consolidated our findings in regards to base latency using a Bayesian analysis, which showed that our data is highly in favor of a model that acknowledges latency as a true effector on PP and GX (all $0 < BF_{01} <= 0.061$). Furthermore, the analysis revealed moderate evidence that latency variation (RQ1, cf. Table 1.1) does not alter PP ($5.062 < BF_{01} < 6.758$) and very strong to extreme evidence in support of a model that postulates no effect of latency variation on the GX ($68.138 < BF_{01} < 100$). However, investigating the interaction between latency and its variation revealed that our data is in favor of a model that supports a true effect of the interaction on all measures (all $BF_{01} < 0.292$) except on the *Meaning* subscale of the PXI ($BF_{01} = 1$).

In this section, we first discuss and contextualize our findings about the effects of high base latency and latency variation on PP and GX, shed light on how base latency and its variation interact, and elucidate on the contrary findings revealed. We conclude by discussing our study's limitations and possible future work.

3.1.4.1 Effects of Base Latency, Latency Variation and the Interaction

Regarding the influence of constant latency on players, our findings brace previous work, demonstrating that a high latency leads to a decrease in PP and GX. Liu et al. (2021c), for example, showed that linearly increasing latency from 25 ms to 125 ms in the Counter-Strike: Global Offensive (CS:GO) decreases player accuracy and overall score. Other work, for instance, by Sabet et al. (2020b), illustrated that high latency also has negative effects on the subjective GX. Our work is in line with both findings regarding performance and experience. In our work, players had a significantly reduced accuracy, total amount of bot kills, and a worse kill-to-death ratio, while simultaneously deriving a significantly lower quality of GX when playing with high base latency. In line with previous work, we argue that the lack of responsiveness of the game induces the degradation of performance and experience. Playing with a high base latency led to a discrepancy between in- and output and, thus, to a decreased performance and experience.

To answer this thesis's first RQ (RQ1, cf. Table 1.1), our analysis of the influence of small-term latency variation yielded no significant results. On the contrary, using a Bayesian approach, we found up-to extreme evidence that small-term latency variation does not alter performance and experience. Our findings, thus, are opposite to previous work investigating the effects of network jitter on PP and GX. For example, Amin et al. (2013) found that network jitter of 100 ms - which is comparable to our variation of $\pm 50 ms$, significantly increases task completion time in video games, compared to playing without it. Similarly, the authors found that network jitter also significantly decreases the overall GX. Since we were not able to replicate those findings using local latency jittering, our findings indicate that local and network-based jitter manifest their effect on performance and experience fundamentally different. This is in line with previous work by Liu et al. (2021d), in which the authors showed that local and network latency, generally, influence PP differently.

Lastly, investigating the interaction between base latency and its variation, we revealed inconclusive findings. On the one hand, inferential analysis suggests that all measures except one subscale of the PXI are unaffected of the interaction. On the other hand, the Bayesian analysis indicates that our data is in favor of a model that implies an effect by the interaction. Further investigating the interaction effect on the meaning subscale reveals equal evidence for H_0 and H_1. Hence, the inferential results showing a significant interaction between base latency and its variation are not supported by our Bayesian analysis. Overall, we found that NHST supports an interaction effect just on the meaning subscale, while our Bayesian analysis is in favor of a model that supports the interaction as a real effect on most measures. The diverging results can be explained by two reasons: First, it is possible that base latency and variation do generally interact, but that the effect is so small and subtle that we were not able to detect its influence in our study consistently. While it is theoretically possible that this is the case, it also raises the question on how pragmatically relevant the interaction effect may be. This line of reasoning, however, is in line with our findings regarding the interaction. NHST is, per design, more conservative than a Bayesian analysis since it is bound to p-values to detect significance. On the other hand, its Bayesian counterpart is not tied to fixed decision boundaries. Hence, a Bayesian analysis can indicate true effects of a variable even if classical NHST is not yet able to consistently detect a significant difference.

Second, it is also possible that base latency and latency variation do not interact. This implies that the significant results obtained in our study when investigating the interaction effect must be attributed to a statistical type I error. While we did use a family-wise error correction for post-hoc testing, a false-positive can never be completely ruled out. However, the results obtained by the Bayesian analysis indicate that our results are more likely affected by a type II error, which leads to an inflation of false-negative results.

Our results regarding the interaction between latency and its variation are inconclusive. Additional work is required to further entangle the interplay between both variables. In the context of this thesis, however, it is evident that small-term variation, as a standalone factor, does not alter PP and experience (RQ1).

3.1.4.2 Limitations and Future Work

While we found that latency variation does not negatively affect video game players, this study still has some limitations. First, our sample of participants in this study only partially represents the population of interest. Besides the strong gender imbalance among our participants, most of them were computer science students and, thus, not representing a high level of diversity. Future work, should aim to further generalize our findings by investigating short-term latency variation with a more diversified participant pool including players from different ages, educational levels and cultural backgrounds.

On the same note, as latency is especially interesting in the context of e-sports, future research should rigorously control for the participants' skill levels. While we asked participants in our study to self-rate their skill level in playing FPS games, we did not quantitatively assess their actual skill. Self-rated assessments are heavily biased and typically not highly reliable. The participants' skills are of particular interest since previous work indicates that more experienced players are more likely to perceive small differences in latency (Liu et al., 2021a). Hence, it is possible that latency variation, as investigated in this paper, does affect expert players but not players with a lower level of gaming skill. Furthermore, it is also possible that the player's individual skill level not only alters their performance but also their felt GX. Previous work discusses if players that perform better in a video have a higher level of enjoyment, and thus a higher level of GX, compared to players performing worse (Klimmt et al., 2009). Hence, future research should either control player skill more strictly or measure a reliable metric to use it in a statistical analysis.

In addition, it is important to recognize the limitations presented by the sample size in our study. With a sample of only 28 participants, there are limits to statistical power and precision. Although previous research suggests that latency effects can be detected even with a smaller sample size, the limited number of participants in our study may make it difficult to identify small effects. Therefore, future studies should investigate local latency variation using a larger sample to improve generalizability and reliability.

Moreover, we only tested the effects of varying latency for a FPS game. However, previous work, for example, the work by Claypool and Claypool (see Section 2.2.2.6 for more example), indicates that latency's effect strongly depend on various game characteristics. For example, in fighting games like Street Fighter, frame-perfect input is necessary to perform certain actions, such as blocking opponents' attacks, or successfully performing a combo. With a time window of only 16.67 milliseconds (assuming a 60 Hz game loop), small-term latency variation might affect such actions much more than fighting a bot in a shooting game. Therefore, our findings can not yet be generalized to the broad landscape of video game genres, as further studies are needed. Hence, our study could be replicated with other games to learn about the influence of latency jitter in different genres.

Lastly, our inferential analysis revealed an interaction effect between base latency and its variation on the meaning subscale of the PXI. While a subsequent Bayesian analysis generally favors a model in which the interaction effect does affect all experience measures, we found no evidence for either H_0 or H_1 about the effects of the interaction on the particular meaning subscale. Nevertheless, the identification of an interaction effect between base latency and its variation on the meaning subscale of the PXI raises important questions about the limitations of this study and the implications for future research. First, it is possible that the interaction between base latency, its variation and the meaning subscale may be more complex than previously thought. This highlights the need for more detailed investigations into the nature of this interaction and the potential role of other variables that may be confounding the results. Second, the study's reliance on a specific apparatus (Sauerbraten) and methods may have limited the generalizability of the results, as different experimental setups may yield different outcomes. This may have resulted in biased results on the meaning subscale of the PXI in particular, which may not be universally representative of real-world gaming situations. We hypothesize that playing Sauerbraten in general does not induce a great extent of meaning in the players, and thus, that the inferential differences are more likely caused by random variation instead of a real effect. Hence, to address these limitations, future research should explore the effects of the interaction between base and latency in a variety of settings, using different apparatûus and methods to ensure the validity of our findings. In conclusion, the identification of an interaction effect between base latency and its variation on the meaning subscale of the PXI highlights the need for further investigation and the potential limitations of the study. The inconclusiveness of our findings underscores the need for caution in interpreting these results and for future research to continue exploring the complex relationship between these variables.

3.1.5 Conclusion

In this section, we presented the results of a study ($n = 28$) investigating the effects of local latency jitter (small-term latency variation) on PP and GX in a FPS game. Participants played with two levels of mean base latency and two levels of latency variation.

Our work contributes to the extending body of work, showing that a high latency reduces performance and experience. Furthermore, we highlight that small-term latency variation as a standalone factor does not significantly influence video gaming sessions. On the contrary, a Bayesian analysis found evidence that an isolated latency variation does not alter performance and experience. Lastly, we contribute preliminary insight into the complex interaction between base latency and its variation. While NHST generally suggests that both variables do not interact, Bayesian analysis indicates that there may be a small and subtle effect. We argue that further work is required to consistently entangle the interplay between latency and its variation.

In light of this thesis's scope, this work shows that small-term latency variation does not affect PP and GX and, thus, answers this thesis's first research question (RQ1)(cf. Table 1.1).

3.2 Understanding Long-Term Variation of Latency (Study II)

Through the previous study, we learned about the effects of two types of latency: (1) constant latency and (2) fast-paced and small-term latency variation. Since latency is caused by multiple factors (Wimmer et al., 2019; Kaaresoja & Brewster, 2010; Noordally et al., 2016) and thus is prone to variation, there is a third type: (3) long-term variation latency (switching latency). Previous work does not account for switching latency and its' effects on players. Evidently, latency is neither constant nor are changes only induced by jitter, but it also undergoes medium-term switches between ever-changing levels. As a result, players may experience several different latency levels within a single gaming session. This situation is even more critical in light of the sharp increase in the number of mobile gamers (Statista, 2021b) and new gaming paradigms, such as cloud gaming (Shea et al., 2013). Mobile players are more strongly exposed to switching latency caused by handshakes between cell towers and network load balancing techniques on the carrier's end (Grigorik, 2013). Despite this, it currently is unclear how switching latency affects performance and experience in video games (RQ2, cf. Table 1.1). Particularly, in light of latency compensation techniques, it is essential to understand how a long-term change from one latency level to another level influences players and gaming sessions.

This section is based on the following article:

Halbhuber, D., Schwind, V., & Henze, N. (Oct. 2022e). "Don't Break My Flow: Effects of Switching Latency in Shooting Video Games." In: *Proc. ACM Hum.-Comput. Interact.* 6.CHI PLAY, pp. 1–20. DOI: 10.1145/3549492.

3.2.1 Background and Research Rationale

To answer if long-term latency variation (latency switches) affects PP and experience, we developed a video game exposing players to switching latency by alternating between three latency levels using three different frequencies. Emphasizing the ecological validity, we developed the game to conduct the study remotely and in the wild. To account for local latency, we determined the local latency of typical gaming setups playing our game. Using the game, we conducted a study with 264 players. Our analysis shows that switching latency significantly decreases the players' feelings of flow. Simultaneously, we found that switching latency significantly reduces the perceived challenge and the experienced tension. Besides these subjective measures, we found significant adverse effects on the players' performances, such as the accuracy and the achieved scores. We also found that increasing the number of switches increases the adverse effects of switching latency. Considering all gathered data, we conclude that games should prioritize constant latency. Furthermore, we found that the starting condition of a gaming session is crucial for PP and experience. Our work shows that the condition under which a gaming session starts fundamentally shapes the further course of the session. A bad start condition leads to a worse overall experience, while a good start condition improves it. In light of latency compensation, our work shows that compensation systems should always aim to provide a as stable as possible latency environment. While small-term variation seems to have no negative influence on PP and GX, a long-term variation, a latency switch, may negatively alters both.

3.2.2 Apparatus and Prestudy

To investigate how switching latency between different constant latency levels affects PX and PP, we developed a fast-paced 2D Shoot 'em up game. This type of game features elements found in the majority of game genres, such as quick target selection or target tracking, and thus is susceptible to latency (Beigbeder et al., 2004; Liu et al., 2021c). Developing a custom game has multiple advantages: (1) individual player skill does not bias the experiment's data, since our game was not openly available prior to this work, (2) low-level logging of performance metrics, such as accuracy, score, and click behavior, is realizable, and (3) directly manipulating latency is possible. We conducted a pre-study to test our game and stress-test our infrastructure. Then, we investigated the local latency of typical hardware setups playing our game, allowing us to account for local latency in the wild.

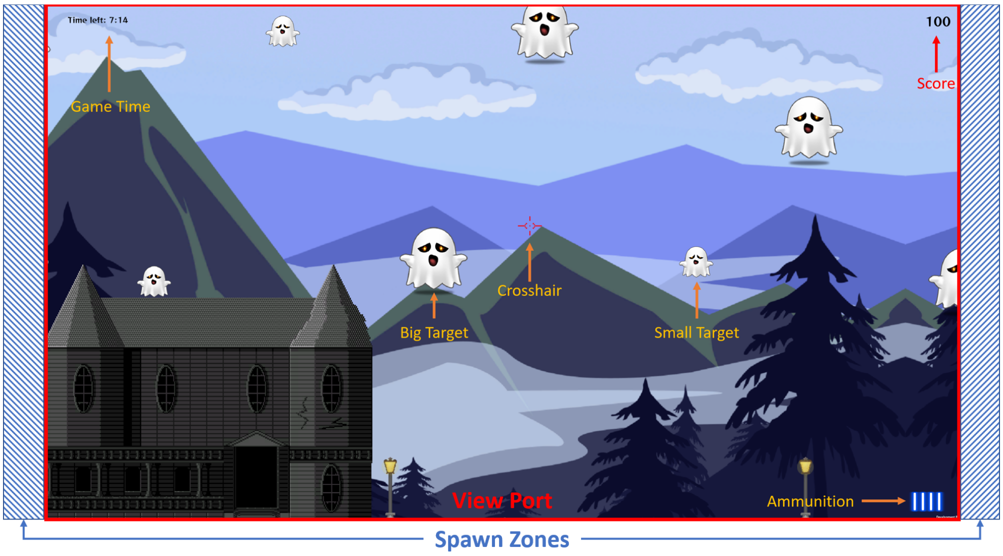


Figure 3.5: Screenshot of the developed 2D Shoot 'em up game. The red box shows the player's view - the viewport. To the left and right of the viewport are the blue-shaded spawn zones of the targets. Additionally, the screenshot shows the game time (top left), the current score (top right), the players' ammunition (bottom right), the crosshair (center), and large (center) and small targets (center).

3.2.2.1 Implementation

The players' objective in the developed game is to shoot as many targets as possible to increase their score before the game time elapses. Each target hit rewards the player with ten points. Figure 3.5 shows a screenshot of the developed game – the different game elements, such as targets, the player's ammunition, and the time left, are highlighted correspondingly. The targets spawn randomly on the left and right regions outside the player's screen. After spawning, they move with fixed horizontal speed to the opposite spawn zone. Targets disappear when leaving the viewport. The horizontal speed of the targets varies randomly within a fixed range. Targets leaving the players' viewports do not grant any points. Players do not lose points for missing targets. To keep players motivated while playing, we increase the difficulty by adding an ammunition game mechanic. Players have to reload their virtual weapon every five shots, which renders them unable to shoot for a brief moment and prevents them from spamming fire. To be successful, players need to manage their resources and plan ahead. After eight minutes of playtime, the game ends automatically and refers participants to a post-experience questionnaire. We developed the game using Unity3D (Version 2020.2.0f1). To ensure that each session is reproducible, we set a fixed seed for Unity3D's random number generator and locked the game's target frame rate at 60 frames per second (FPS). Finally, we added functions to create latency by buffering user inputs. We coupled these functions with a frequency generator alternating between different latency levels.

We used the browser as the game's target platform. Using the browser allows us to conduct all experiments remotely and in the wild. In-the-wild studies have an inherently higher ecological validity since participants use their own equipment in a familiar and comfortable setting. However, on the other hand, in-the-wild studies come with a reduced internal validity since we cannot fully control the experiment's environment and its execution. The advantages of investigating gamers in their own environment and using their own equipment outweigh the disadvantages of in-the-wild studies in our work. Additionally, playing the game in the browser entails multiple advantages. First, participants do not have to install any additional software to participate in the study. Second, content delivery is straightforward since we only need to provide a hyperlink to our web server. Finally, using the browser allows participants to complete the study within a single application. The game, the data collection, and the questionnaire were all realized using the browser - participants did not have to switch applications or even web pages. To prevent network latency, we used *WebGL* to deliver the game to the players. This enables hardware acceleration by the players' computers. Thus, all rendering and processing are performed locally.

3.2.2.2 Pre-study and Play Testing

We conducted an online pre-study to test the developed game. We gathered qualitative and quantitative data to identify and implement possible improvements.

Apparatus

We hosted our game on a publicly reachable web server for the study. Participants played the game on their own devices without installing or obtaining additional software. Our institution's web server managed content delivery and did not require further user input. Participants could take part using a WebGL-compatible browser of their choice, such as a current version of Google Chrome (for example, 116.0.X).

Procedure and Task

We first informed the participants via the crowd-sourcing platform prolific.co[1] about the study's purpose and provided a hyperlink to our web server. After giving informed consent to data collection, participants could follow the hyperlink to our website. Upon entering the website, they were assigned a random, unique identification and were presented with a start button. By clicking the button, participants started the game and the data collection. Participants were instructed to earn as many points as possible by shooting targets. They did not receive any further instruction. After 8 minutes, the game ended automatically and forwarded the participant to the post-experience questionnaire. All collected data is anonymous and not traceable to individuals. The user study and the data collection received ethical clearance as per the ethics policy of the University of Regensburg.

Participants

We used prolific.co to recruit 24 participants (5 female, 19 male). As we estimated a total duration of 12 minutes for our study, participants were compensated with £1.75 for their contribution. Their average age was 24.5 years (*SD* = 4.75 years), with ages ranging from

[1] `https://prolific.co/`

19 years to 36 years. All participants were screened for prior gaming experience using Prolifc's screening interface. This ensured that all participants had comparable gaming skills.

Descriptive Results

In total, 24 participants played our game for 3.2 hours. We recorded the participants' mouse movements and click behaviors throughout the gaming sessions. Additionally, we also logged frame rates and system specifications (CPU, GPU, OS, RAM, and screen resolution), as well as in-game metrics, such as score, hit rate, and opponent movements. Using our in-game logging, we recorded a total of 77 973 unique data samples. On average, each player performed 637.6 shots (*SD* = 169.4 shots) and managed to hit 418.6 targets (*SD* = 141.5 hits).

We used the 33-item Game Experience Questionnaire (GEQ) (IJsselsteijn et al., 2013) to quantitatively evaluate PX and coupled it with questions focused on the game's quality. We analyzed the GEQ with its subscales: Competence, Sensory, Flow, Tension, Challenge, Negative Affect, and Positive Affect. A maximum of 5 points could be assigned in each subscale. Participants on average gave 3.29 points (*SD* = 0.60 points) in the Competence subscale, 2.18 points (*SD* = 0.65 points) in the Sensory subscale, 3.11 points (*SD* = 0.70 points) in the Flow subscale, 2.55 points (*SD* = 0.94 points) in the Tension subscale, 2.72 points (*SD* = 0.39 points) in the Challenge subscale, 2.70 points (*SD* = 0.96 points) in the Negative Affect subscale and 3.00 points (*SD* = 0.99 points) in the Positive Affect subscale. The evaluation of the GEQ shows that none of the subscales were rated negatively, hence validating and establishing the game as the subject of our further research. Further, all participants stated in the qualitative feedback that they were able to complete the study without problems and did not encounter any bugs in the game or uncertainty about the study's procedure. A flawless procedure and bug-free apparatus are crucial for a large-scale in-the-wild study to be successful since we can not support participants during the study.

3.2.2.3 Estimating Local Latency

Local latency, the latency caused by one's hardware, such as the computer, the periphery, and the monitor, needs to be accounted for when investigating the effects of switching latency. However, since we aspire to conduct an in-the-wild study to maximize ecological validity, we can not control the setup used by our participants. To tackle this, we estimated

Overview of local latency measurements					
OS	**CPU**	**RAM**	**GPU**	**Mouse**	**Mean Local Latency**
Win 10 Pro 21H1	Ryzen 7 1800X	32 GB	GTX 1080Ti	Mamba Elite	11.40 ± 2.28 frames
Win 10 Home 21H1	Ryzen 7 4800U	16 GB	GTX 1660Ti	DeathAdder V2	14.65 ± 4.34 frames
Win 10 Pro 20H2	i5-10500	16 GB	RTX 2070	HP 280 Silent	8.49 ± 1.47 frames
Win 10 Home 20H2	Ryzen 3 1200	8 GB	GTX 980Ti	Logitech G203	16.60 ± 1.65 frames
Win 10 Pro 21H1	i5-1145G7	8 GB	Intel Iris Xe	Dell MS116-BK	17.30 ± 3.51 frames
Win 10 Home 21H1	AMD 3020E	4 GB	AMD Radeon	Logitech M90	11.45 ± 2.54 frames

Table 3.2: Shows all systems for which local latency was measured. Each system was measured 20 times. Measurements were done using a high-FPS Camera (GoPro 7, 240 FPS). Additional information for each system, such as OS, CPU, and GPU are provided in the table as well. Mean local latency is specified in frames. Local latency of all measured systems spans from 8.5 frames to 17.30 frames. The mean local latency for all systems is 13.32 frames (*SD* = 4.15 frames) which translate to an average local latency of 55.51 ms (*SD* = 17.29 ms.)

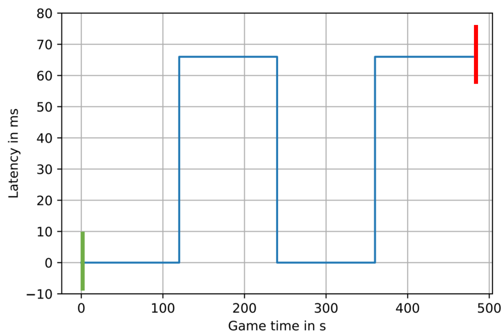
typical local latency while playing our game by testing multiple gaming setups and measuring the corresponding latency. The mean of these values will be considered as the average local latency for the remainder of this paper. Following the procedures of related work (Ivkovic et al., 2015; Long & Gutwin, 2018), local latency was measured using a 240 fps camera (4,167 ms/frame, GoPro Black 7), which captured both the system's mouse and the game screen. Manually comparing the physical mouse click with an update in the game allowed us to determine the local latency of six different systems. We measured each system 20 times. While playing our developed game, the mean local latency is 13.32 frames (*SD* = 4.15 frames, *N* = 120), which translates to a mean local latency of 55.51 ms (*SD* = 17.29 ms). This latency is considered to be the baseline latency of our game. All further latency values include this baseline without explicitly mentioning it. Table 3.2 shows further details to all measured systems as well as the mean measured local latency of each system.

3.2.3 Method

After developing and validating our apparatus in the pre-study, we used it to investigate how switching latency between different constant latency levels in a single gaming session affects GX and PP. We artificially added multiple latency levels via input buffering to our game to achieve long-term latency variation and used different frequencies to switch between the different latency levels. Subsequently, we tested all combinations of latencies and the number of switches during the gameplay.

3.2.3.1 Study Design

To control latency levels and switches between different levels, we utilized three independent variables (IVs): (1) START LATENCY - which corresponds to the latency participants started playing with, (2) TARGET LATENCY - which specifies to which value START LATENCY switches to, and (3) LATENCY SWITCHES - defines how often the latency in the gaming session switches (from START LATENCY to TARGET LATENCY and vice versa). START LATENCY and TARGET LATENCY both have three levels: (I) 0 ms, (II) 33 ms, and (III) 66 ms. LATENCY SWITCHES is likewise factorized in three levels: (I) 0 switches, (II) 3 switches, and (III) 12 switches. Combining all IVs results in eleven unique conditions (LATENCY SWITCHES x START LATENCY x TARGET LATENCY). Since we investigate the effects of switching to one level of artificially added latency, we excluded combinations switching between two levels of artificially added latency. Participants were randomly assigned to one of the eleven conditions. Figure 3.6 depicts, exemplary, the latency progression over one gaming session in one condition (left side). Additionally, the right table lists all tested conditions (right side).



Overview of study conditions			
Condition	**Start Latency**	**Target Latency**	**Switches**
1	0 ms	0 ms	0
2	33 ms	33 ms	0
3	66 ms	66 ms	0
4	0 ms	33 ms	3
5	0 ms	33 ms	12
6	0 ms	66 ms	3
7	0 ms	66 ms	12
8	33 ms	0 ms	3
9	33 ms	0 ms	12
10	66 ms	0 ms	3
11	66 ms	0 ms	12

Figure 3.6: The left side shows the latency progression for condition six in one gaming session. The green mark depicts the game‘s start. The red mark symbolizes the end of the gaming session. The latency switches between 0 ms and 66 ms on a unipolar rectangle wave with a frequency of 66.68 mHz which equals 3 latency switches. The table on the right lists all eleven unique combinations of START LATENCY, TARGET LATENCY and LATENCY SWITCHES. Participants were randomly and evenly assigned to the eleven conditions.

The latency in the game switches between START LATENCY and TARGET LATENCY using a unipolar rectangle oscillation defined by LATENCY SWITCHES. The upper bound (66 ms) of START LATENCY and TARGET LATENCY was defined in accordance to related work, which also investigated 66 ms latency (Henze et al., 2016). We also examined 33 ms and 0 ms latency to allow for detailed analysis. The first level of LATENCY SWITCHES (0 switches) relates to previous work investigating constant latency values (for example, Liu et al., 2021c; Liu et al., 2021d). The second level (3 switches) is the first value, which allows for balanced latency conditions, meaning participants are playing the same amount of time in START LATENCY and TARGET LATENCY and, thus, the least amount of switches testable. To also investigate a high level of switches, we set the last level of LATENCY SWITCHES to 12 switches, which allows for a direct comparison with the previous levels.

We recorded data about the participants' experiences and their performances. To measure participant performance, we used two dependent variables: (1) *Score*, which increased every time a player successfully hit a target. Players did not lose points for missing a target. And (2) *Target Offset*, which quantifies how accurately players hit the targets. *Target Offset* is defined by the Euclidean distance between the ideal hit point in the center of the target and the 2D coordinates of the actual impact. A lower value, thus, corresponds to higher accuracy, and vice versa, a higher value means the hit was further off of the ideal hit point.

To measure the perceived gaming experience, we again utilized the 33-item GEQ with its seven subscales (IJsselsteijn et al., 2013).

3.2.3.2 Apparatus

The study's apparatus was similar to the apparatus used in our pre-study. The game was again hosted on a web server accessible to the public. Participants started and played the game on their own computers with a WebGL-compatible browser of their choice, such as a current version of Google's Chrome (for example, 116.1.x). We modified the game to reflect the presented conditions - creating eleven different game versions, each incorporating one of eleven unique combinations of START LATENCY, TARGET LATENCY and SWITCHING FREQUENCY (cf. Figure 3.6). Each participant played a single version of the game.

3.2.3.3 Procedure and Task

Via Prolific, we informed the participants about the study's procedure and provided a hyperlink to our web server. After giving informed consent to the data collection, the participants could follow the hyperlink to the game. Following the hyperlink, participants were assigned a random and unique identification and were presented a start button, which, upon clicking it, started the data collection and the game. Participants were blind to the exact purpose of the study (to investigate switching latency), but were told to test a novel game. The participants' goals was to earn as many points as possible by shooting the targets. After eight minutes of game time, the game ended automatically and forwarded the participants to the post-experience questionnaire. All collected data is anonymous and not traceable to participants. The user study and the data collection received ethical clearance as per the ethics policy of the University of Regensburg.

3.2.3.4 Participants

We used the crowd-sourcing platform Prolific to recruit a total of 264 participants (55 female, 205 male, two non-binary, and two preferred not to say). We excluded one participant due to possible game manipulation attempts. Thus, one condition was tested with 23 participants. The other ten conditions were tested with 24 participants each. Participants who participated in our pre-study were excluded from attending the main study. Additionally, participation in more than one condition was not possible. Participants were compensated with £1.75 for an estimated study time of 12 minutes. The average participant age was 23.68 years (*SD* = 5.45 years), ranging from 18 years to 49 years. All participants were screened for prior gaming experience using Prolific's screening interface. Pre-screening for gaming experience ensured that all participants had comparable gaming skills.

3.2.4 Results

We evaluated the participants' mouse movements, click behaviors, frame rates, system specifications, and in-game metrics such as scores and accuracy. In total, participants played the game for 35,2 hours. On average, each participant fired their weapon 595.80 times (*SD* = 129.42 shots) and successfully hit 405.08 targets (*SD* = 113.62 hits).

We structure the further analysis in two parts: (1) Analysis of the post-experience questionnaire and (2) analysis of performance-related measures.

3.2.4.1 Game Experiences Questionnaire

Following the authors' guidelines (IJsselsteijn et al., 2013), we analyzed each subscale of the GEQ separately. We used a mixed model ANOVA with LATENCY SWITCHES

Mixed model ANOVAs of the Game Experience Questionnaire					
Measure	**Effector**	**DF, Residual**	**F-Value**	**p-Value**	η_p^2
Competence	Latency Switches	2, 252	0.529	0.590	<0.001
	Start Latency	2, 252	0.219	0.804	<0.001
	Latency Switch x Start Latency	4, 252	0.413	0.799	<0.001
	Latency Switch x Start Latency x Target Latency	2, 252	0.283	0.75	<0.001
Sensory	Latency Switches	2, 252	1.533	0.218	0.01
	Start Latency	2, 252	0.391	0.677	<0.001
	Latency Switch x Start Latency	4, 252	0.958	0.431	0.01
	Latency Switch x Start Latency x Target Latency	2, 252	0.252	0.592	<0.001
Flow	**Latency Switches**	**2, 252**	**49.079**	**<0.001**	**0.28**
	Start Latency	2, 252	0.141	0.869	<0.001
	Latency Switch x Start Latency	4, 252	1.526	0.195	0.02
	Latency Switch x Start Latency x Target Latency	2, 252	0.997	0.370	<0.001
Tension	**Latency Switches**	**2, 252**	**4.662**	**0.010**	**0.04**
	Start Latency	2, 252	0.734	0.481	<0.001
	Latency Switch x Start Latency	4, 252	0.410	0.801	<0.001
	Latency Switch x Start Latency x Target Latency	2, 252	0.040	0.961	<0.001
Challenge	**Latency Switches**	**2, 252**	**3.056**	**<0.001**	**0.14**
	Start Latency	2, 252	0.225	0.689	<0.001
	Latency Switch x Start Latency	4, 252	0.617	0.399	0.02
	Latency Switch x Start Latency x Target Latency	2, 252	1.021	0.188	0.01
Neg. Affect	Latency Switches	2, 252	0.274	0.760	<0.001
	Start Latency	2, 252	0.099	0.906	<0.001
	Latency Switch x Start Latency	4, 252	1.676	0.156	0.03
	Latency Switch x Start Latency x Target Latency	2, 252	0.068	0.934	<0.001
Pos. Affect	Latency Switches	2, 252	1.156	0.317	<0.001
	Start Latency	2, 252	0.095	0.909	<0.001
	Latency Switch x Start Latency	4, 252	0.593	0.668	<0.001
	Latency Switch x Start Latency x Target Latency	2, 252	0.012	0.988	<0.001

Table 3.3: Results of the GEQ mixed model ANOVA analysis. Each row represents one measurement testing for either main effects of Latency Switches and Start Latency or interaction effects for Latency Switches x Start Latency or Latency Switches x Start Latency x Target Latency. We found significant main effects of Latency Switches on the *Flow*, *Tension* and *Challenge* scores. Significant results are shown in bold.

nesting Start Latency and Target Latency, to analyze each subscale of the GEQ. Table 3.3 shows all statistical results of this analysis. To improve clarity, we only focus on significant ANOVA results and the according post hoc tests.

ANOVA showed a significant main effect of Latency Switches on *Flow* ($p < 0.001$, $\eta_p^2 = 0.28$), *Tension* ($p = 0.010$, $\eta_p^2 = 0.08$) and *Challenge* ($p = 0.007$, $\eta_p^2 = 0.04$). For post hoc analysis, we used a Tukey-test to reveal significant differences in *Flow*, *Tension* and *Challenge*. Starting with the *Flow* subscale, Tukey's test showed that participants' ratings did not significantly differ between playing with 0 and 3 Latency

SWITCHES (adjusted $p = 0.152$, $d = 0.257$). However, the test showed significant differences between playing with 0 and 12 LATENCY SWITCHES (adjusted $p < 0.001$, $d = 1.329$) as well as between playing with 3 and 12 LATENCY SWITCHES (adjusted $p < 0.001$, $d = 1.285$). Figure 3.7 (left) depicts the mean *Flow* scores grouped by LATENCY SWITCHES. Playing with 12 LATENCY SWITCHES lead to significant lower *Flow* scores compared to playing with 0 or 3 LATENCY SWITCHES.

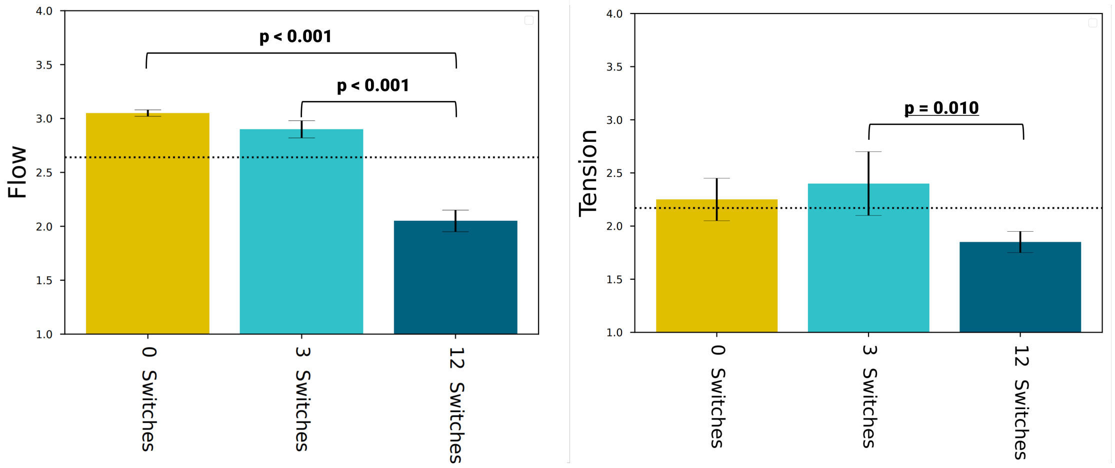


Figure 3.7: Depicts *Flow* (left) and *Tension* (right) data of the GEQ assigned by the players. Data is grouped by LATENCY SWITCHES. Error bars depict the standard error. Additionally, all data is color coded by LATENCY SWITCHES: Yellow groups 0, light-blue 3, and dark-blue 12 LATENCY SWITCHES. Significant differences between LATENCY SWITCHES are highlighted. The mean score over all subscales is provided via the dotted line. Participants playing with 12 LATENCY SWITCHES assigned the lowest *Flow* and *Tension* scores.

Next, we analysed *Tension* using Tukey's test for pairwise post hoc comparison. We found no significant differences between playing with 0 LATENCY SWITCHES and playing with 3 LATENCY SWITCHES (adjusted $p = 0.836$, $d = 0.085$) nor between playing with 0 LATENCY SWITCHES and playing with 12 LATENCY SWITCHES (adjusted $p = 0.082$, $d = 0.356$). However, we found significant differences between playing with 3 LATENCY SWITCHES and playing with 12 LATENCY SWITCHES (adjusted $p = 0.010$, $d = 0.424$). Figure 3.7 (right) shows the mean *Tension* scores grouped by LATENCY SWITCHES. Participants were significantly less tense when playing with 12 LATENCY SWITCHES compared to playing with 3 LATENCY SWITCHES.

Lastly, we further analysed the *Challenge* subscale using Tukey's test. Tukey's test found no significant difference comparing 0 LATENCY SWITCHES to 3 LATENCY SWICHES (adjusted $p = 0.846$, $d = 0.086$). However, we found significant differences between playing with 0 LATENCY SWITCHES and 12 LATENCY SWITCHES (adjusted $p = 0.012$, $d = 0.4887$), as well as between playing with 3 LATENCY SWITCHES and playing with 12 LATENCY SWITCHES (adjusted $p = 0.032$, $d = 0.346$). Figure 3.8 shows the mean *Challenge* scores grouped by LATENCY SWITCHES. Participation with 12 LATENCY SWITCHES induced the lowest amount of challenge in the participants. A small increase to 3 LATENCY SWITCHES, on the other hand, did not have a significant effect on the perceived challenge, compared to the *Challenge* rating playing with 0 LATENCY SWITCHES.

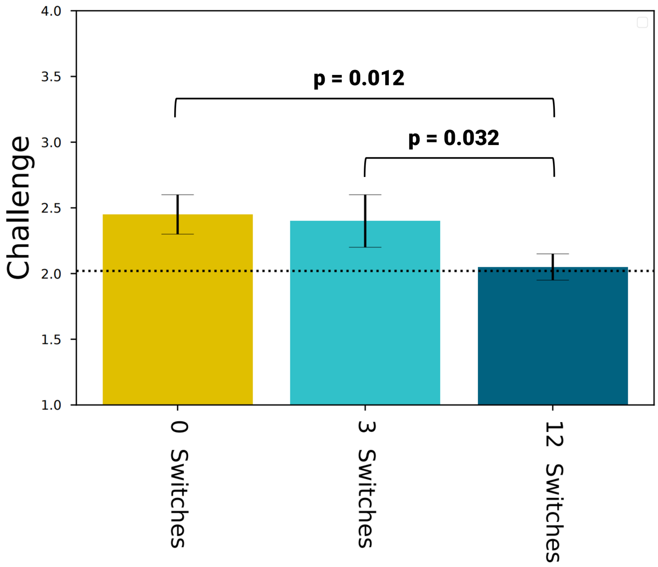


Figure 3.8: Depicts *Challenge* data of the GEQ assigned by the players. Data is grouped by LATENCY SWITCHES. Error bars depict the standard error. Additionally, all data is color coded by LATENCY SWITCHES: Yellow groups 0, light-blue 3, and dark-blue 12 LATENCY SWITCHES. Significant differences between LATENCY SWITCHES are highlighted. The mean score over all subscales is provided via the dotted line. Participants playing with 12 LATENCY SWITCHES assigned the lowest *Challenge* scores.

Mixed model ANOVAs of Score and Target Offset					
Measure	**Effector**	**DF, Residual**	**F-Value**	**p-Value**	η_p^2
Score	Latency Switches	2, 252	2.012	0.125	0.02
	Start Latency	2, 252	1.434	0.239	0.01
	Latency Switch x Start Latency	**4, 252**	**2.856**	**0.024**	**0.04**
	Latency Switch x Start Latency x Target Latency	2, 252	0.114	0.891	<0.001
Target Offset	**Latency Switches**	**2, 252**	**3.412**	**0.034**	**0.38**
	Start Latency	**2, 252**	**4.952**	**0.007**	**0.04**
	Latency Switch x Start Latency	**4, 252**	**2.672**	**0.032**	**0.04**
	Latency Switch x Start Latency x Target Latency	2, 252	0.575	0.563	<0.001

Table 3.4: Results of the *Score* and *Target Offset* mixed model ANOVA analysis. Each row represents one measurement testing for either main effects of LATENCY SWITCHES and START LATENCY or interaction effects for LATENCY SWITCHES x START LATENCY or LATENCY SWITCHES x START LATENCY x TARGET LATENCY. We found significant main effects of LATENCY SWITCHES and START LATENCY on *Target Offset* as well as significant interaction effects LATENCY SWITCH x START LATENCY on *Score* and *Target Offset*.

3.2.4.2 Performance Measures

Next we investigated whether LATENCY SWITCHES, START LATENCY and TARGET LATENCY had a significant effect on *Score* and *Target Offset*. We used a mixed model nested ANOVA with LATENCY SWITCHES nesting the factors START LATENCY and TARGET LATENCY - Table 3.4 shows all results of this analysis. In the following, to improve clarity, we only focus on significant ANOVA-results and the according post hoc tests.

Score

On average, participants scored 5 095.71 points (*SD* = 1 646.12 points). The ANOVA showed a significant interaction effect for LATENCY SWITCHES X START LATENCY (p = 0.024, η_p^2 = 0.04). We used a Tukey-test for post hoc testing of LATENCY SWITCHES X START LATENCY. Tukey's test showed a significant effect between 0 ms START LATENCY/12 LATENCY SWITCHES and 0 ms START LATENCY/0 LATENCY SWITCHES (adjusted p = 0.031, d = 0.571). Also the test revealed significant differences between playing in combination 0 ms START LATENCY/12 LATENCY SWITCHES and 0 ms START LATENCY/3 LATENCY SWITCHES (adjusted p = 0.034, d = 1.085). All other combinations did not show significant differences (all p > 0.05).

Participants playing with 0 ms START LATENCY and 0 LATENCY SWITCHES on average scored 6 035.6 points (*SD* = 2938.3 points). Playing with 0 ms START LATENCY and 3 LATENCY SWITCHES lead to an average score of 5922.5 points (*SD* = 975.4 points). Participating with 0 ms START LATENCY and 12 LATENCY SWITCHES resulted in a mean score of 4758.4 points (*SD* = 1169.5 points). Participant scored significantly fewer points when playing with 0 ms START LATENCY and 12 LATENCY SWITCHES compared to playing with 0 ms START LATENCY and 3 LATENCY SWITCHES and 0 ms START LATENCY and 0 LATENCY SWITCHES.

Figure 3.9 depicts mean *Score* values for conditions with significant differences.

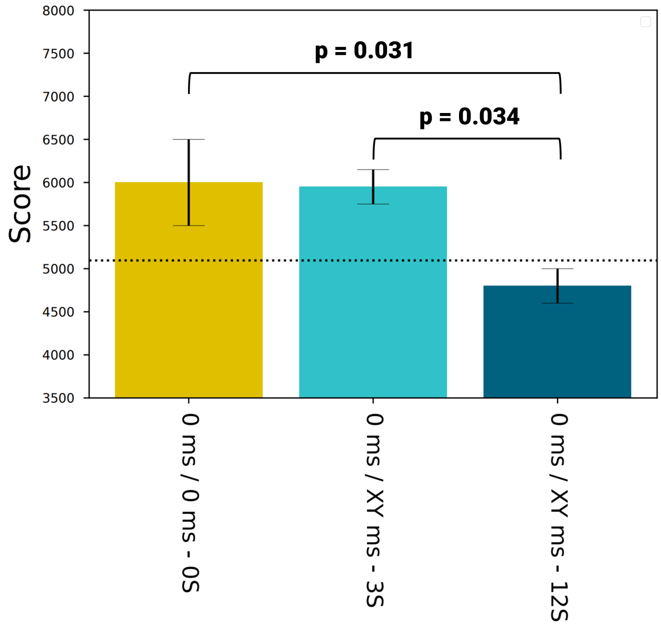


Figure 3.9: Shows mean score values players reached for conditions with signficant differences. X-ticks are coded with the independent variables START LATENCY / TARGET LATENCY - LATENCY SWITCHES. Additionally, all data is color coded by LATENCY SWITCHES: Yellow groups 0, light-blue 3, and dark-blue 12 LATENCY SWITCHES. The middle shows mean *Score* values grouped by LATENCY SWITCHES and the same color code. Error bars show the standard error.Additionally, significant p-values and standard error as error bars are provided. Participants playing with 0 ms START LATENCY reached a significant lower *Score* when playing with 12 LATENCY SWITCHES compared to playing with 0 or 3 LATENCY SWITCHES.

Target Offset

On average, participant were 0.263 UWC (*SD* = 0.151 UWC) off of an ideal hit. We, again, used a mixed model nested ANOVA with LATENCY SWITCHES nesting the factors START LATENCY and END LATENCY. ANOVA showed a significant main effect of LATENCY SWITCHES on *Target Offset* (p = 0.034, η_p^2 = 0.38). Figure 3.10 shows the mean *Target Offset* sorted by LATENCY SWITCHES. Participants playing with 0 LATENCY SWITCHES on average reached an *Target Offset* of 0.257 UWC (*SD* = 0.017 UWC), 3 LATENCY SWITCHES lead to a mean value of 0.262 UWC (*SD* = 0.006 UWC) and playing with 12 LATENCY SWITCHES caused the highest deviation from an ideal hit with a mean *Target Offset* of 0.272 UWC (*SD* = 0.005 UWC).

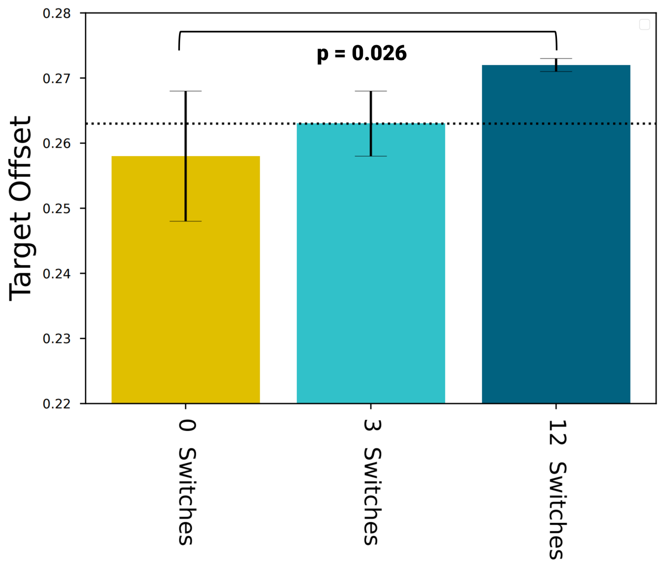


Figure 3.10: Shows mean *Target Offset* values players reached by number of latency switches. Error bars depict the standard error. X-ticks are coded with the independent variables START LATENCY / TARGET LATENCY - LATENCY SWITCHES. Significant differences are highlighted via p-bars. The mean *Target Offset* values is depicted via the dotted line. Participants deviated significantly stronger from an ideal hit when playing with 12 LATENCY SWITCHES compared to playing with 0 LATENCY SWITCHES.

We found a significant main effect of START LATENCY on *Target Offset* (p = 0.007, η_p^2 = 0.04). Participants playing with 0 ms START LATENCY averagely reached an *Target Offset* of 0.257 UWC (*SD* = 0.013 UWC), playing with 33 ms START LATENCY lead to a mean *Target Offset* of 0.265 UWC (*SD* = 0.002 UWC), and lastly participating in a 66 ms condition resulted in a mean *Target Offset* of 0.273 UWC (*SD* = 0.001 UWC). Figure 3.11 (left) depicts the mean *Target Offset* values grouped by START LATENCY. Further investigation, revealed an interaction effect for LATENCY SWITCHES X START

LATENCY ($p = 0.032$, $\eta_p^2 = 0.04$). Based on this results, we used a Tukey-test for post hoc investigation of LATENCY SWITCHES and START LATENCY. Tukey's test found significant differences in *Target Offset* between 0 and 3 LATENCY SWITCHES (adjusted $p = 0.026$, $d = 0.031$), but did not reveal significant difference for all other combinations (all adjusted $p > 0.05$). A Tukey-test showed significant differences between 0 ms and 66 ms START LATENCY (adjusted $p = 0.008$, $d = 0.114$), but no significant differences in *Target Offset* for all other combinations (all adjusted $p > 0.05$). Upon investigation of the interaction between LATENCY SWITCHES X START LATENCY, Tukey's test found significant differences between playing with 0 ms START LATENCY/0 LATENCY SWITCHES and 33 ms START LATENCY/0 LATENCY SWITCHES (adjusted $p = 0.007$, $d = 0.210$) as well as between 0 ms START LATENCY/0 LATENCY SWITCHES and 66 ms START LATENCY/0 LATENCY SWITCHES (adjusted p 0.011, $d = 0.265$). Additionally, the test revealed significant differences between playing with 33 ms START LATENCY/0 LATENCY SWITCHES and 0 ms START LATENCY/3 LATENCY SWITCHES (adjusted $p = 0.019$, $d = 0.054$) as well as between 66 ms START LATENCY/0 LATENCY SWITCHES and 0 ms START LATENCY/3 LATENCY SWITCHES (adjusted $p = 0.026$, $d = 0.107$). All other combination did not reveal significant differences (all adjusted $p > 0.05$). Figure 3.11 (right) displays the significant differences found in the pairwise comparison.

Overall, we found evidence that participants performed worse while playing with with 12 LATENCY SWITCHES compared to 0 LATENCY SWITCHES and also performed worse playing with 66 ms START LATENCY compared to playing with 0 ms START LATENCY. Participants playing with 0 ms START LATENCY and 0 LATENCY SWITCHES achieved significantly better *Target Offset* values compared to 33 ms START LATENCY and 0 LATENCY SWITCHES as well as compared to 66 ms START LATENCY and 0 LATENCY SWITCHES. Additionally, participants playing with 0 ms START LATENCY and 3 LATENCY SWITCHES also performed better compared to playing with 33 ms START LATENCY and 0 LATENCY SWITCHES as well as compared to playing with 66 ms START LATENCY and 0 LATENCY SWITCHES.

3.2.5 Discussion

Our analysis revealed that switching latency significantly affects experience and performance. To discuss and explain the found effects, we refer to prior research investigating the effects of latency on GX and PP. Additionally, we discuss novel effects not yet investigated in previous work. In the following, we systematically discuss those effects

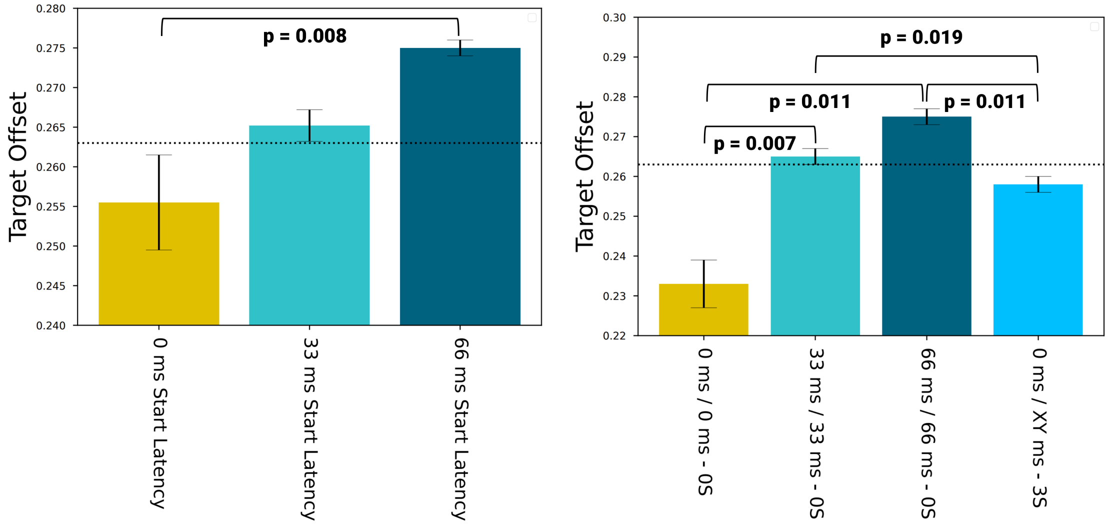


Figure 3.11: Left shows the *Target Offset* grouped by START LATENCY (x-ticks). Error bars depict the standard error. Participants deviated significantly stronger from an ideal hit when playing with 66 ms START LATENCY compared to playing with 0 ms START LATENCY. The right side visualizes the interaction effect of START LATENCY x LATENCY SWITCHES. Participants, when playing with 0 LATENCY SWITCHES, performed worse when playing with 33 ms or 66 ms START LATENCY compared to playing with 0 ms START LATENCY. Additionally, they performed better when playing with 3 LATENCY SWITCHES and 0 ms START LATENCY compared to playing with 0 LATENCY SWITCHES and 33 ms or 66 ms START LATENCY.

on the feeling of flow, tension, and challenge. Additionally, we discuss the effects on the players' scores and in-game accuracy. We conclude this section with a discussion of the implications for developers, gamers, and researchers alike.

3.2.5.1 Flow, Tension, and Challenge

We found an effect of latency switches on the perceived flow. Flow, firstly described in 1975 by Csikszentmihalyi (Csikszentmihalyi, 1997; Hendin & Csikszentmihalyi, 1975), corresponds to the mental state of *being in the zone*. In addition to its influence in areas such as reading, sports, and mental activities, flow also has a significant impact on the gaming experience in video games (Cowley et al., 2008). Video games are often designed to maximize the experienced flow, thus creating an activity that is enjoyable for the sheer sake of doing it, even at great cost, such as neglecting mundane everyday tasks, for the player (Csikszentmihalyi, 1990; Grimes & Feenberg, 2009). Our study showed that participants playing with twelve latency switches perceived significantly less flow than participants playing with none or three switches. However, we did not find significant differences between playing with zero and three switches. This suggests that the flow of participants is not disturbed by a few switches (three), and participants, thus, stay *in the zone*. On the other hand, more frequent switching lead to participants experiencing less flow. Our data shows that the experience was significantly disturbed by latency switches. Every time latency changed, the input-output paradigm the participant was dealing with changed as well. Each switch changed how the game had to be played, as it changed how the mouse behaved, how responsively shots were fired, and how the game itself reacted to the user input. Participants had to adjust to these changes every time, effectively preventing them from entering *the zone*. On the surface, these findings may seem obvious, as a change within the game session changes perceived flow compulsorily. However, we think that those findings are the most influential of this work, as they show that in a gaming context, it is not always the best approach to aim for low latency if this means sacrificing latency stability.

Additionally, we found that the number of latency switches had an effect on the feeling of tension and the perceived challenge offered by the game. Similarly to the flow rating, we found that participants rated both constructs lowest when playing with twelve latency switches. At first glance, this seems contradictory to our other findings, as it seems like players were less tense and challenged when playing with twelve latency switches. While this may be true on the surface, we believe this effect directly correlates to the reduced flow state. Flow, being described as situated between boredom and anxiety (Hendin & Csikszentmihalyi, 1975), is responsible for a certain feeling of tenseness. While not in a high-flow state, our participants were not as involved in the game as they would have been in a high-flow environment. The same applies to the perceived challenge – without proper involvement, participants did not see a real challenge in the game while still performing worse than participants in other conditions. Surprisingly, we did not find any indication in favor of this hypothesis in the post-experience qualitative feedback data. While not consciously aware of it, participants rated the game with more switches as a lower challenge while simultaneously performing worse than participants playing with no switches.

3.2.5.2 Score

We found an interaction effect between the number of latency switches and the start latency. Participants playing with 0 ms start latency and no or three switches obtained significantly more points than participants playing with 0 ms start latency and twelve switches. While this does not allow generalized conclusions about the effect of the independent variables on the achieved scores, it enables us to pose hypotheses that are in line with previous research. First, our analysis revealed the achieved score did not significantly differ between playing with 0 ms, 33 ms, and 66 ms start latency or between playing with zero, three, or twelve latency switches. We hypothesize that the score metric is robust to latency and thus robust to the variation of it, as it is a metric describing the overall performance of the participants. This is in line with previous work (Claypool et al., 2014), which found that PP is stable up to 100 ms latency. Second, participants' performances started to be negatively affected when introducing latency switches. However, these effects occurred only when participants started playing with 0 ms, and even then, only when the difference in the number of switches was high. Consequently, we found differences between playing with zero switches and twelve switches and between playing with three switches and twelve switches, but not between playing with zero and three switches. Surprisingly, this behavior was not observable when participants started playing with 33 ms or 66 ms start latency. We infer that participants playing with 0 ms start latency started their gaming experiences with optimal latency conditions and that , thus, switching latency impairs an otherwise ideal gaming session. On the other hand, starting with 33 ms or 66 ms start latency exposes the participant to a sub-optimal setting. Since participants in these conditions started their session with artificially added latency, the additional negative effects of switching latency did not influence gaming performance as much as in starting with 0 ms latency. Nevertheless, we found evidence that latency switching worsens the game performance in an otherwise optimal gaming session.

3.2.5.3 Target Offset

Participants playing with 12 latency switches aimed more poorly and deviated stronger from the optimal hit point than participants shooting with zero or three latency switches. In addition, the start latency has a significant effect on the participants' shooting behaviors. Participants who started the game with a latency of 66 ms were significantly less accurate than those who began with a latency of 0 ms. This is in line with prior research investigating the effects of latency on accuracy in video games (Beigbeder et al., 2004), which found that latency can reduce accuracy by up to 50%. Our analysis also showed that while in a latency-switching gaming session, the target latency, the latency to which the current latency alternates, does not affect the participants' accuracies. Nevertheless, we found that the combination of start latency and the number of latency switches creates a significant interaction effect on the player accuracy. An in-depth analysis of this interaction showed that switching latency might improve PP in some instances. While this finding on the surface seems to be not intuitive, further investigation revealed that a positive benefit for the players arises when compared to playing with constant high latency. Since this improvement happens in conditions with low switching values (three), we hypothesize that participants could utilize the 0 ms latency periods independently of the previous latency. While in these 0 ms periods, participants were able to substantially improve their accuracy in such a manner that it positively affected their overall accuracy rating. Contrary to this, when participating in a condition with a high number of switches, participants could not improve their rating significantly compared to constant latency.

Summarizing our findings, we showed that switching latency negatively impacts accuracy. We also found edge cases in which participants can benefit from latency switches. To the best of our knowledge, no previous research has investigated this behavior. Thus, we are only able to speculate about its origins. However, we conclude that two factors caused this improvement: (1) Participants only improved when playing with three latency switches. Thus, we assume that participants could utilize the 0 ms periods because they had enough time to familiarize themselves with the game behavior. This familiarization was not likely to happen in conditions with twelve switches - participants did not have time to internalize the new gaming environment. (2) Participants only improved when starting with 0 ms latency. We conclude this is caused by participants starting their gaming session with optimal latency conditions. Interestingly, starting with 33 ms or 66 ms latency did not lead to any improvement, regardless of the number of switches. Therefore, we assume that the start condition for any gaming session is crucial for the overall performance during the rest of the session.

3.2.5.4 Implications

Our findings have implications for future game design and thus for game developers. To optimize GX, developers should minimize fluctuations and aim for a stable latency, even though this could mean accepting a higher but stable latency over a lower but switching one. This can be achieved either by using a predictive system to reduce latency (Henze et al., 2016) to fixed values or by artificially adding latency to provide a consistent experience (Li et al., 2018; Diot & Gautier, 1999).

Our findings are relevant to gamers as well. While most gamers may not have the means to thoroughly control their latency, knowing that low latency is not obligatorily better than high latency is valuable. Particularly in highly competitive e-sports scenarios, it is important to factor in not only the average latency but also its stability when, for example, evaluating a past gaming session. Everyday gamers can minimize latency variability by using Quality of Service (QoS) protocols in their home equipment. Using such protocols allows the user to prioritize applications in the routing protocol. Although this may have only a small impact, it may improve GX and PP.

Researchers of latency in video games potentially benefit from our findings, too. We showed that constant latency is inherently different from switching latency. While this is known in the context of jitter, it has not been demonstrated for long-term variation in the way our work did. Additionally, jitter and the latency we investigated in this work differ strongly by the strength of fluctuation and frequency. We encourage other researchers to incorporate our findings in future work. Treating latency as a constant for elongated duration, whether discretized or approximated over many samples, potentially leads to the missing of effects on GX and PP. Lastly, our work also has implications for the design and development of latency compensation techniques. We showed that a stable latency is more beneficial to PP and GX compared to varying latency. This means that compensation systems should aim to provide an as stable as possible latency, even if this may entail deliberately not compensating latency further.

3.2.5.5 Limitations and Future Work

We investigated switching latency and compared its effects to constant latency. Our game either stayed at a constant latency or oscillated to 0 ms latency to isolate the effects of the artificially added latency and create a balanced dataset that enabled us to compare the different conditions. This approach has its limitations, as we cannot investigate effects occurring when switching between two levels of artificially added latency. Similarly, as we only estimated the local latency of the players playing our game, is possible that the actual local latency in our study differed. Furthermore, as we investigated three and twelve latency switches, there might exist a sweet spot between those two values we missed. The found edge cases regarding the effects of latency switches on the PP, namely the score, suggest this is a feasible consideration. Future work should continue to investigate the effects and implications of switching between two artificially added latency levels and whether there are latency-switching sweet spots.

Furthermore, we found that the starting condition of a gaming session is crucial for a player's performance and experience. Our work showed that the condition under which a gaming session starts significantly influences the further course of the session. A bad start condition leads to a worse overall gaming experience, while a good start condition improves the overall gaming experience. Future work should further investigate this aspect.

Despite the presented results, focusing only on one video game genre does not depict the whole gaming landscape. Future work should aim to replicate our findings in other game genres, such as FPS or racing games. Investigating different genres is valuable not only in the context of this work but also in the context of research on video games. Further deepening our knowledge about latency and its effects on players is crucial to developing games with a high gaming experience.

3.2.6 Conclusion

In this section, we presented our work in which we developed a 2D Shoot 'em up game that we inflicted with switching latency to investigate if long-term latency variation affects PP and experience. In a study with 264 participants, we found that long-term latency variation significantly affects GX and PP. We show that long-term latency instability significantly decreases the perceived flow, effectively preventing participants from *entering the zone*. In addition, we found effects on the experienced tension and the perceived challenge. Furthermore, we showed that participants were significantly less accurate when playing in a more volatile environment than participants playing in steady conditions.

In summary, we found that switching latency negatively influences GX and PP stronger than a constant high latency. However, while this is true for the overall picture, we also revealed edge cases where participants could benefit from switching latency. In total, however, the disadvantages of switching latency outweigh the advantages.

Summarizing the present work within the framework of this thesis, we show that long-term latency variation affects GX and PP. This section, thus, answers the second RQ

(RQ2, cf. Table 1.1). Overall, this means that latency compensation techniques should aim to minimize long-term latency variation. Reducing latency to a lower level is not necessarily beneficial for PP or GX if it entails sacrificing stability.

4

Understanding Latency Perception

In the previous chapter we learned that small-term latency variation does not affect PP and GX (RQ1). On the other hand, we showed that long-term variation does affect a player's experienced flow, the perceived challenge and tension as well as the objective in-game performance (RQ2). However, latency's magnitude and volatility are not the only factors that can vary in a real-world gaming session. In a real gaming session, latency can also vary by how it is perceived by the players. Previous work established that visual latency, a delay between player action and the moment the player sees the results of their action, reduces PP and GX (Liu et al., 2021d; Claypool & Claypool, 2006; Claypool & Claypool, 2006; Claypool et al., 2014). However, less is known about the influence of auditory latency, which refers to a delay between an action and hearing the effect of this action. In light of ever emerging wireless audio gaming equipment such as Bluethooth headsets, which can have a transmission latency of up to 200 ms (habr.com, 2019), this can have serious ramifications for players. Furthermore, since latency compensation produces high computational load it is necessary to establish if auditory latency in a standalone scenario affects PP and GX. This is even more relevant of online multiplayer games, which often calculate game events server-sided. In such a scenario it can happen that a player shoots their virtual weapon but hears the sound of successfully hitting a target with an increased offset. This offset is caused by a high latency to the server which has to confirm and verify each in-game actions before propagating it to all other players. In summary, it is crucial to understand how auditory latency affects PP and GX. Hence, the first half of this chapter describes two studies to answer how auditory latency influences

PP and GX (RQ3). To achieve this, we introduced different levels of artificial auditory latency to two FPS games – one self-developed and one commercially available – and let participants play with them in two independent studies. In the first study, we found that standalone auditory latency does not affect PP and GX when playing the self-developed FPS. Participants achieved the same in-game performance and had the same experience over all levels of auditory latency. To extend those findings, we conducted the second study using a commercially available and widely played FPS and higher auditory latency levels. In the second study, we found that auditory latency increases the feeling of tension and decreases the positive feelings associated with the gaming session. On the highest level tested, we even found that auditory latency increases the negative emotions towards the game. Additionally, we showed that the effects of auditory latency are particularly pronounce for experienced video game players.

In the second half of this chapter, we deepen our understanding of the effects of latency by investigating how its effects are modulated by how players visually perceive the gameplay. This is essential, since previous work comes to different conclusions on how the used in-game perspective alters gaming experience (Gorisse et al., 2017; Monteiro et al., 2018) and how it interacts with latency (Claypool & Claypool, 2006). While one body of work argues that the in-game perspective, with being a part of the game's genre definition, dictates the latency sensitive of the gaming session (Claypool & Claypool, 2006), another line of work indicates it does not (Schmidt et al., 2017). Thus, it is unclear how the in-game perspective alters the effects of latency. Subsequently, it is also unclear how latency compensation techniques should behave, for example, if the player manually changes in-game perspective in a game such as The Elder Scrolls Online (Bethesda, 2022) in which fluid change between in-game perspectives is possible. However, this is crucial information considering that latency compensation is a computational expensive task. Hence, to investigate how the in-game perspective alters the effects of latency (RQ4), we conducted a user study with a self-developed FPS game. In the game we incorporate the three most common in-game perspectives for video games: First-person view (FPV), third-person view (TPV), and bird's-eye view (BEV). For the study, we artificially increased the gaming session latency by buffering player input. Using this setup, we found that the in-game perspective – how the player visually perceives the game – does not alter the effects of latency.

In the last part of this chapter, we investigate how the current level of latency in a game should be communicated to the players. Previous work demonstrated that users' expectations about systems alter how they interact with them (Kosch et al., 2022), thereby influencing their experience. These expectations are formed by cues and prior experience with the system. An effect based solely by an expectation is called Placebo (if the effect is positive) or Nocebo (if the effect is negative) (Holmes et al., 2016; Montgomery & Kirsch, 1996). While previous work showed that using interactive system can trigger an Pla- or Nocebo effect (Michalco et al., 2015; Denisova & Cairns, 2015b; Kosch et al., 2022), it is unclear if this is also the case in regards to latency. However, this is essential, since knowledge about a expectation-based latency effect informs about how latency should be communicated to the players. In case the game communicates a high latency, for example by displaying it the game UI, and the mere fact that the player thinks that they are playing with a high latency leads to a expectancy-induced performance and experience degradation, it would be advisable to obfuscate it. On the other hand, it may also be possible that communicating a lower-as-real latency, for example 15 ms instead of 100 ms, to the player triggers a Placebo-induce performance and experience increase. However, currently it unclear how the expectation of latency alter PP and GX (RQ5). To answer this question, we conducted a study, in which participants played a video game with four levels of pretended latency, which is the latency that is communicated to player. Crucially, all gamerounds were played with 75 ms of actual latency, we merely changed its presentation in the game. We call the change of displayed latency value, without changing the actual latency, phantom latency. We show that a high phantom latency reduces PP and GX. Participants were least accurate and effective when playing with 120 ms of phantom latency while feeling the least competent and tensest. We concluded that the effects of latency are at least partly expectation-based. Hence, we also show that the mere expectation of latency can alter the course of the gaming session.

This chapter present the experimental setting, the procedure, and the results of all four studies. All findings are discussed in detail within the context of the conducted study.

4.1 Understanding Audio Latency in Custom Video Games (Study III)

Auditory latency, the temporal offset between performing an in-game action and hearing the corresponding sound to this action in video games, has not been widely investigated.

This is surprising since sound is an essential part of every immersive virtual environment. Especially in video games, a well-designed audio landscape is known to increase immersion and involvement of the players (Grimshaw et al., 2008; Gormanley, 2013). Through fittingly engineered sounds, video games evoke emotion in players, causing fear, bliss, or even anxiety (Toprac & Abdel-Meguid, 2011). In addition to being unambiguously useful in increasing game immersion and player engagement, sounds in video games are also used to convey game-relevant information and aid in increasing the players' performances. Whether a player-controlled avatar in the FPS Apex Legends (EA, 2022) groans because they were hit by an adversary's projectile, the car in Forza Horizon (Microsoft, 2021) beeps when the traction control is activated, or just to indicate that a skill is ready to use in League of Legends (Riot, 2021) - sounds in video games are omnipresent.

Despite its obvious relevance in video games, sound as a modulator of latency has not been widely investigated. While the works of Kaaresoja and Brewster (2010) and Ye et al. (2018) help in understanding how auditory latency arises and how it differs, the authors do not investigate the effects of auditory latency performance and experience. Other work in this line of research by Simon et al. (2013) investigated the effects of auditory latency in disc jockey (DJ) interfaces and found that novice users were not negatively influenced by auditory latency. Simultaneously, they found that experienced DJs performed worse, starting at an auditory latency of 130 ms. Simon et al.'s work suggest that auditory latency is negligible for the day-to-day user. While this work aids in understanding the effects of auditory latency on users, it focused on a highly specific user group (DJs), and, thus, is not generalizable to the vast spectrum of users of interactive systems, particularly not to video games. Hence, while there is some work investigating auditory latency in highly specified user scenarios, it is still unclear if auditory latency affects PP and GX in video games (RQ3). This and the next section present two studies designed and conducted to answer this question.

This section is partly based on the following article:

Halbhuber, D., Huber, M., Schwind, V., & Henze, N. (2022b). "Understanding Player Performance and Gaming Experience While Playing a First-Person Shooter with Auditory Latency." In: *Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '22. Bremen, Germany: Association for Computing Machinery, pp. 24–30. ISBN: 9781450392112. DOI: 10.1145/3505270.3558333.

4.1.1 Method

To investigate the effects of auditory latency on PP and GX, we customized a publicly available FPS game since the FPS genre has been shown to be particularly latency-sensitive (Claypool & Claypool, 2006; Beigbeder et al., 2004). The game we utilized was originally developed by us to investigate novel latency compensation techniques. In a study, we showed that the game is sensitive to latency. Players rated the game with a lower experience and performed worse when playing with high latency compared to playing with reduced latency. Considering that the game and its' players are negatively affected by latency, it is a viable apparatus to investigate auditory latency. Furthermore, since it is a custom game, our investigations are not as strongly influenced by prior experience, as it would be the case when investigating a commercial game. Additionally, using a custom-developed game allows us to directly log performance metrics of the players, such as their scores, accuracy, and time between shots. Lastly, the self-developed implementation empowers us to manipulate visual and auditory latency independently.

In the following, we describe how we modified the game and implemented artificial auditory latency. Next, we measured the baseline auditory latency of the game. Measuring the baseline auditory latency is crucial to establishing meaningful results in the concluding study. Lastly, we designed and conducted a study with 24 participants playing the modified game to investigated whether the added auditory latency influences PP and GX.

4.1.1.1 Game Development

The game's source code we used is free of charge and publicly available in this thesis's OSF repository. The game was developed using Unity3D (Version 2019.f16.2) for our modification, we used Unity3D (Version 2021.2.7f1). The game's intended target platform was the browser, since we did not plan to conduct a remote study we changed the game's target platform to Windows.

In the game, the player's goal is to shoot randomly appearing monsters before they reach the player. The player has to survive against an ever-increasing number of monsters. If a fixed number of monsters reached the players (before the player was able to shoot them), the players lost the game, and the game reset.

To introduce auditory latency, we reworked every sound-generating game object, such as the player's weapon, firing feedback, hit recognition, and sound generated by enemies spawning. Effectively, this means that if, for example, an enemy spawns, the actual event (the spawning of an enemy) was triggered but the corresponding sound (the sound generated by the spawning enemy) was delayed for a fixed amount of time. Players in the original version were only able to locate enemies visually. We added a specifically designed sound to allow players to localize opponents by sound using Unity‘s 3D-Sound engine. Unity uses a sophisticated approach to simulate sound as close to reality as possible. By utilizing dedicated sound-listening and sound-emitting game objects, the game engine can recreate complex auditory phenomena, such as the Doppler effect. To bring the auditory game elements even more into focus, we added walls to the game world. These walls block the view on the monsters. Hence, players had to rely strongly on auditory information to track enemies. Through these modifications, we emphasized the auditory components of the game. Players had to rely on and utilize auditory game information to perform well and to prevent the monster from reaching them. Additionally, we removed the tutorial of the original game. Since the game was previously used in a remote study, it had an in-depth tutorial to familiarize players with the game controls and mechanics, which is not needed in a study conducted in person. Figure 4.1 shows two screenshots from the modified game. The left side of Figure 4.1 shows the modified game world - walls are highlighted in white, possible enemy spawn points in red, and the players' starting point in blue. The right side of the figure shows the player's view during gameplay, with a hostile monster moving towards the player.

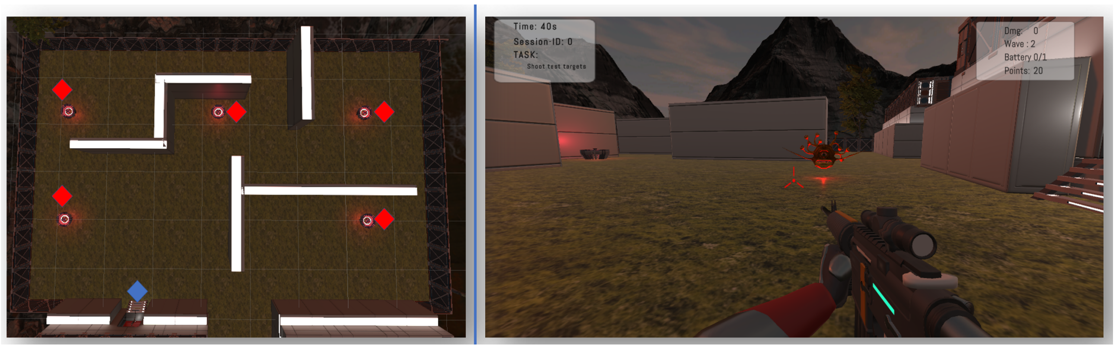


Figure 4.1: Shows two game screenshots. The left side shows an aerial overview of the game world. The red marker indicates enemies' spawn points, the blue marker the players' starting point, and the white areas the walls. The right side of the figure shows the players' views during gameplay. Players are provided several game-related information such as score and time left via the UI. Additionally, the figure shows a hostile monster moving towards the player.

4.1.1.2 Local Latency

To measure local auditory latency, we utilized a 240 fps camera (GoPro Black 7). However, contrary to related work investigating visual latency (Long & Gutwin, 2018; Ivkovic et al., 2015), we first had to determine the camera's offset between visual and auditory input. This is not required when measuring the latency between two visual events - for example, a user's mouse click and the subsequent firing of the weapon - but since we were interested in measuring the latency between a visual event (the clicking of the mouse button) and a subsequent auditory event (the firing sound), the camera's offset between visual and auditory input needs to be determined. To measure the audio-video offset of the GoPro we recorded a sound-generating event - the poking of a cup using a metal pin. By examining the recorded material frame-by-frame, we visually determined the exact moment the pin touched the cup - this moment serves as the starting point of our offset measurement. Further investigating the recorded material, again frame-by-frame, while observing the audio input channel allowed us to establish the moment audio was first recorded. The audio peak represents the end point of our offset measurement. The laboratory was completely silent while performing the experiment to ensure that no external noise distorts our measurements. Investigating the number of frames between the start (the poke) and endpoint (the audio peak) of the recorded event allowed us to estimate the audio-video offset of the used camera. We ran this experiment 20 times and found that the GoPro Black 7 records material with an audio-video offset of 29.85 frames (SD = 0.91 frames). Considering its frame rate of 240 frames/s this corresponds to an offset of 124.37 ms.

Utilizing the found values of the cameras's audio-video offset we measured the auditory latency baseline of our test system running the modified game. We again recorded a visual event - a user clicking the mouse button to fire their weapon - and compared it to the auditory effect - the sound generated by the weapon firing heard through wired headphones. To minimize deviation in the measurement, we ran the experiment 20 times. Comparing the recorded material frame-by-frame revealed a baseline auditory latency of 49.85 frames (SD = 2.17 frames). Subtracting the audio-video offset of the used camera, this measurement corresponds to a baseline auditory latency of 83.33 ms. All investigated auditory latencies are based on the measured baseline auditory latency without explicitly mentioning it.

4.1.1.3 Study Design

We conducted a study to test if auditory latency impacts PP and GX. We used AUDIO LATENCY as a within-subject variable. The levels of AUDIO LATENCY are based on the latency range of the commercially predominantly used Bluetooth protocol, which averages from 150 ms to 250 ms (McPhail, 2022). Additionally, visual latency is known to negatively affect PP and GX at 25 ms (Liu et al., 2021c) and respectively 125 ms (Liu et al., 2021d). To cover all ranges, we categorized AUDIO LATENCY in four levels: (I) 0 ms, (II) +50 ms, (III) +100 ms and (IV) +200 ms auditory latency. To measure GX, we used the GEQ (IJsselsteijn et al., 2013) with its seven sub-scales: Competence, Sensory, Flow, Tension, Challenge, Negative Affect, and Positive Affect. To determine PP, we recorded in-game data from the participants playing the game. PP is measured using three dependent variables: (1) *Score* - the amount of points players achieved by shooting monsters, each monster awarded 10 points, (2) *Accuracy* - quantifies how accurately the players shoot and is built as the quotient of the total amount of shots fired and the number of hits, and (3) *EnemyHit* - corresponds to the number of monsters that hit the players avatar.

4.1.1.4 Apparatus

As apparatus, we used the same setup as in our baseline auditory latency measurement. The game was executed on a stationary workstation in our laboratory in full-screen mode. The workstation (Intel i7, Nvidia GT970, 16 GB RAM) was attached to a monitor (24" FullHD @60Hz), a computer mouse (Logitech M10), and a wired headset. Participants played the game using the same wired headphones evaluated in the baseline measurements. Participants played with all artificially added auditory latency levels, resulting in four tested conditions: (1) *0 ms*, (2) *+50 ms*, (3) *+100 ms* and (4) *+200 ms* of artificially added auditory latency.

4.1.1.5 Procedure and Tasks

Participants were greeted at our institution's laboratory by the experimenter. After signing a consent form and thus agreeing to data collection, participants were seated in front of the computer running the game in full-screen mode. Participants were not informed about the exact purpose of the study (investigating the effects of auditory latency) but were told to test a novel game. Participants played each condition for ten minutes and were told to obtain as many points as possible by shooting monsters. No further instructions were given. After playing ten minutes, the current gaming session ended automatically, and the system referred the participants to the GEQ. In conclusion to completing the GEQ, the next round, with a different amount of artificially added auditory latency, started. Order of conditions was balanced using a Latin Square to prevent sequencing effects. After finishing all conditions, participants were referred to a final post-experience questionnaire. In the final questionnaire, participants were asked to state if they noticed auditory latency and, if they did, how they think it impaired their experience and performance. Upon completion of the final questionnaire, participants were debriefed and left the laboratory. The user study and the data collection received ethical clearance as per the ethics policy of the University of Regensburg.

4.1.1.6 Participants

We invited 24 participants (5 female, 19 male) using our institution's mailing list. Mean participant age was 23.16 years (SD = 2.34 years), ranging from 19 to 30 years. Participants were asked to rate their prior gaming experience and their prior experience in playing FPS games on a 5-point Likert item spanning from no experience at all to very experienced. Participants' mean self-rated experiences with video games in general was 4.41 points ($SD = 0.85$ points), and with FPS games 3.63 points ($SD = 1.29$ points). Thus, participants rated themselves as very experienced with video games in general but less experienced with FPS games. All participants were students at our institution and were compensated with credit for their study course.

4.1.2 Results

In the following, we first describe the gathered data, then we report results of a classical null-hypothesis analysis investigating if AUDIO LATENCY had an effect on game experience and PP. We conclude with a section presenting the results of a Bayesian inference analysis testing for effects induced by AUDIO LATENCY.

Shapiro-Wilk's test was used to determine if the data follows a normal distribution. Results show violation of normality for all GEQ subscales ($p < .05$) and shows that performance measures are normally distributed ($p > .05$). Hence, we use non-parametric Kruskal-Wallis tests (Kruskal & Wallis, 1952)(GEQ subscales) and ANOVAs (performance measures) for null-hypothesis testing. All pairwise cross-factor comparisons are

Game Experiences Scores							
Audio Latency	**Competence**	**Sensory**	**Flow**	**Tension**	**Challenge**	**Neg. Affect**	**Pos. Affect**
0 ms	3.10 ±1.07	1.60 ± 0.76	2.14 ± 0.94	1.11 ± 0.25	1.81 ± 0.94	1.54 ± 0.65	2.68 ± 1.01
+50 ms	2.6 ± 0.80	1.43 ± 0.95	2.06 ± 0.86	0.81 ± 0.81	1.51 ± 1.10	1.41 ± 1.04	2.45 ± 0.83
+100 ms	2.83 ± 0.73	1.31 ± 0.97	2.02 ± 0.87	0.89 ± 0.88	1.72 ± 1.02	1.14 ± 0.92	2.43 ± 0.91
+200 ms	2.85 ± 0.74	1.45 ± 0.99	2.21 ± 0.88	0.87 ± 0.99	2.02 ± 0.94	1.18 ± 1.03	2.56 ± 0.87

Performance Measures			
Audio Latency	**Score**	**Accuracy**	**EnemyHits**
0 ms	1705.41 ± 422.26	0.17 ± 0.04	4.01 ± 5.59
+50 ms	1852.01 ± 504.85	0.17 ± 0.05	6.21 ± 9.79
+100 ms	1767.08 ± 429.94	0.17 ± 0.04	3.92 ± 5.08
+200 ms	1706.66 ± 326.65	0.17 ± 0.04	5.04 ± 5.25

Table 4.1: Shows the mean scores and standard deviation of each subscale of the Game Experience Questionnaire (top) as well as the mean and standard deviation of the performance (bottom). Participants assigned playing with 0 ms the highest game experience score, expect for the Flow and Tension subscale. Both subscales were assigned the highest score when playing with 200 ms auditory latency. Participants reached the highest score when playing with 50 ms auditory latency, but simultaneously received the most amount of monster hits in this condition. Accuracy remained almost stable over all all conditions.

conducted using Wilcoxon tests and are Bonferroni α corrected. For Bayesian inference, we used JASP (Wagenmakers, 2022) and the default prior probability distribution recommended by Wagenmakers et al. (2018). Bayesian post-hoc tests were conducted using Bayesian t-tests. Posterior odds were corrected utilizing Westfall's approach to the correction for multiplicity (Westfall et al., 1997; de Jong, 2019). Bayes factors (Lavine & Schervish, 1999; Kass & Raftery, 1995) are formulated as BF_{01} which indicates how much more likely H_0 over H_1 is, and are interpreted using Lee and Wakenmakers' postulation (2014) to Bayes factor interpretation.

4.1.2.1 Descriptive Results

To rate GX, we utilized the GEQ, which was answered by the participants on a 5-points Likert-item. Table 4.1 (top) shows mean GEQ scores rated by the participants for each level of AUDIO LATENCY. Participants assigned the highest scores when playing in the 0 ms condition in all categories except of the Flow and Tension categories. In both, Flow and Tension, the 200 ms condition was assigned the highest scores by participants.

Performance measures are based on in-game data logs. In our study we measured three variables: (1) *Score*, (2) *Accuracy* and, (3) *EnemyHits*. Table 4.1 (bottom) shows mean values for each variable categorized by levels of AUDIO LATENCY. Participants reached the highest *Score* values when playing in the +50 ms condition (1852.01 points, SD = 504.85 points) and the lowest when playing in the 0 ms condition (1705.41 points, SD = 422.26 points). *Accuracy* values remained stable over all condition, only in the +50 ms condition standard deviation fluctuated stronger (0.17, SD = 0.05). Participants were most hit by hostile monsters when playing in the +50 ms condition (6.21 hits, SD = 9.79 hits), and were hit the least when playing with 100 ms artificially added auditory latency (3.92 hits, SD = 5.08 hits).

4.1.2.2 Null-Hypotheses Testing

Kruskal-Wallis' test using the within-subject factor AUDIO LATENCY test revealed no significant effects on the GEQ sub-scales Competence (χ^2 = 2.14, p = 0.54, df = 3), Flow (χ^2 = 0.29, p = 0.96, df = 3), Sensory (χ^2 = 1.68, p = 0.64, df = 3), Challenge (χ^2 = 3.08, p = 0.38, df = 3), Negative Affect (χ^2 = 4.35, p = 0.22, df = 3), and Positive Effect (χ^2 = 0.59, p = 0.89, df = 3). The test did reveal a significant effects of AUDIO LATENCY on Tension subsection (χ^2 = 8.08, p = 0.04, df = 3). Alpha-corrected Bonferroni post-hoc tests, however, did not reveal significant differences in pairwise-comparison for Tension (all $p > 0.05$)

An ANOVA, with the within-subject factor AUDIO LATENCY, did not reveal significant difference in the performance measures *Score* (F(1,23) = 0.68, p = 0.91), *Accuracy* (F(1,23) = 0.15, p = 0.93) and *EnemyHits* (F(1,23) = 0.58, p = 0.62).

4.1.2.3 Bayesian Inference

Previous Kruskal-Wallis tests and ANOVAs consistently did not reveal significant effects of AUDIO LATENCY on any measure. However, classical null-hypothesis testing can only reveal differences between distributions, thus it either accepts or rejects a null hypothesis. It can not, by design, reveal if the missing of a significant difference is an indication for equivalence between the investigated distribution. For example, although a classical null-hypothesis test did not find a difference between distribution A and distribution B, it does not mean that A and B are equivalent. Thus, to investigate equivalence in our data we conducted a Bayesian analysis (Cleophas & Zwinderman, 2018; Wagenmakers et al., 2018). Contrary to classical null-hypothesis testing Bayesian inference calculates probabilities for both: H_0 and H_1.

We found moderate evidence (Lee & Wagenmakers, 2014) for correct acceptance of H_0 in the distributions of Competence (BF_{01} = 5.45), Challenge (BF_{01} = 7.46), Tension

Post-Hoc Bayesian t-test BF01 values corrected for multiple testing - GEQ								
Level 1	**Level 2**	**Comptence**	**Sensory**	**Flow**	**Tension**	**Challenge**	**Neg. Affect**	**Pos.Affect**
0 ms	+50 ms	1.24	2.91	3.33	1.13	2.72	3.15	2.58
	+100 ms	2.27	2.02	3.17	2.11	3.35	1.08	2.51
	+200 ms	2.43	3.05	3.47	2.13	2.74	1.54	3.19
+50 ms	+100 ms	2.76	3.21	3.44	3.32	3.17	2.42	3.47
	+200 ms	2.61	3.47	3.39	3.40	1.44	2.73	3.24
+100 ms	+200 ms	3.46	3.12	3.24	3.47	2.27	3.45	3.16

Post-Hoc Bayesian t-test BF01 values corrected for multiple testing - Performance Measures				
Level 1	**Level 2**	**Score**	**Accuracy**	**EnemyHits**
0 ms	+50 ms	2.09	3.39	2.43
	+100 ms	3.13	3.37	3.47
	+200 ms	3.48	3.25	2.92
+50 ms	+100 ms	2.92	3.48	2.32
	+200 ms	1.89	3.46	3.13
+100 ms	+200 ms	3.06	3.45	2.78

Table 4.2: Shows BF_{01} values for post-hoc testing using Bayesian t-tests for data of the Game Experience Questionnaire(top) as well as all gathered performance measures (*Score*, *Accuracy*, *EnemyHits*) (bottom). All measures are categorized by the levels of artificially added auditory latency. Tests were corrected for multiplicity using Westfall's correction (Westfall et al., 1997). All post-hoc tests revealed anecdotal (BF_{01} > 1 and BF_{01} < 3) to moderate (BF_{01} > 3) evidence for H_0.

(BF_{01} = 9.03), Negative Affect (BF_{01} = 6.06), and strong evidence for H_0 in Flow (BF_{01} 15.77), Sensory (BF_{01} = 11.47) and Positive Affect (BF_{01} = 11.67). The data from the subscales of the GEQ have not been influenced by AUDIO LATENCY with at least moderate (BF_{01} > 3) and partly with strong evidence (BF_{01}>10). A Bayes factor BF_{01} of 15.77, as in the Flow subscale, indicates that the data is 15.77 times more likely under the null hypothesis postulating no effects induced by AUDIO LATENCY. Table 4.2 (top) shows Bayes factors BF_{01} for all pairwise post-hoc comparisons for the GEQ data categorized by levels of AUDIO LATENCY. In post-hoc testing we found anecdotal (BF_{01} > 1 and BF_{01} < 3) to moderate (BF_{01} > 3) evidence for H_0 regarding data from the GEQ.

For the performance measures we found moderate evidence for *Score* (BF_{01} = 8.53), and *EnemyHits* (BF_{01} = 9.31), and strong evidence for *Accuracy* (BF_{01} = 16.58) in favor of accepting H_0. Post-hoc tests are, again, depicted in table 4.2 (bottom) and show anecdotal to moderate evidence as well.

4.1.3 Discussion

Our results consistently show that artificially added auditory latency does not induce a significant effect on player experience and performance in a custom FPS game. A Bayesian analysis of the data revealed strong evidence that the data gathered while playing with four different amounts of added auditory latency is equivalent. With this work, we contribute first evidence that auditory latency does not affect players in the same way visual latency does. To explain this we refer to prior work investigating effects of visual latency – or the combination of visual and auditory latency – on players.

In this section, we first discuss the evidence for equivalence of GX and PP between tested auditory latency levels. We then conclude with exploring the implications of our findings for gamers, developers, and latency researchers.

4.1.3.1 Equivalence of Game Experience and Player Performance

We did not find any significant effects on the players' GXs. However, the lack of significant effects induced by auditory latency could be caused by several reasons.

First, it is possible that the game is not sensitive to auditory latency but only to visual, respectively combined latency. This could be because the used game is a fast-paced FPS game. Hence, players are put under constant stress. They have to fight a never-ending stream of monsters to achieve a score as high as possible. Since it is a stress-inducing situation, it is possible that players more heavily relied on their most dominant sense - visual perception. This is in line with work, showing through a large-scale study that humans prioritize their visual perception over all other senses (San Roque et al., 2015). This could also be applicable to of fast-paced games. Since players had to react quickly and accurately they only focused on the visual input and thus did not notice the auditory delay. While this, on the surface, may seem a disadvantage of the game used, we conclude that it is rather an indication that games vary in auditory latency sensitivity. This means that not all games, and thus the perceived GX of players, are affected by auditory latency in the same way. This has already been shown in regards to visual latency by Claypool and Claypool (2006) and Sabet et al. (Sabet et al., 2020a).

Second, individual player skill and experience with a certain game may change the perception of auditory information. Consider, for example, CS:GO, which is a highly competitive high-stakes team-based shooter. Starting at a certain skill level, perceiving and reacting to auditory cues is essential to perform well. Whether a player listens for enemy footsteps or tries to locate where a gunshot originates, every information obtained could potentially lead the own team to victory. Previous work showed that player skill alters the effects of latency. Liu et al. (2021a) for instance, showed that higher-skilled players are stronger affected by the negative effects induced by latency. The same could be true for auditory latency. We, however, used a custom game to prevent introducing a bias caused by previous experience in a game. Since our participants had no experience with it, we created a fair and comparable situation for our study. It is possible that auditory latency only starts affecting players at a certain skill level, which we potentially missed by using a custom game.

Lastly, another reason for the equivalence found between the tested auditory latency levels may be that the chosen levels, in combination with the measured baseline auditory latency of the test system, are too small. It is possible that there is an auditory latency threshold we did not cross. Similarly to the different latency thresholds for different game genres proposed by Claypool and Claypool (2006), it is reasonable to assume that too-low auditory latency, in general, is not perceivable by humans. We designed our tested latency levels on the latency range of the commercially used aptX codec. We did this because there is no reason to assume a higher auditory latency in the wild. Thus, the investigated latency values are those everyday gamers have to deal with. If auditory latency is not perceivable under a certain threshold, and if we did not cross that threshold in our study, it means that auditory latency may have no relevance in real-world gaming sessions.

Our findings are valid in regard to the tested custom game. However, one needs to be careful to generalize our findings to the vast gaming landscape with its countless types of games, genres, and players. It is possible that other games lead to different results. Nevertheless, our work is the first step to shine light on the effects of auditory latency on players and to provide means to create a fair and fun gaming environment for every gamer alike.

4.1.3.2 Implications of our Findings

Our findings have implications for players, game developers, and researchers.

Day-to-day players directly benefit from our findings since we showed that an increase in auditory latency is neglectable in regards to GX and PP. Thus, spending the extra money to buy specialized low-latency audio equipment such as headsets and headphones is unjustified. However, it is possible that auditory latency does affect players with a high skill level. Nevertheless, since most players do not compete at such a high skill level, for example, in high-stakes CS:GO tournaments, our findings are applicable to most parts of the gaming community.

Game developers potentially profit from our findings as well. Since we showed that auditory latency does not affect players with the same magnitude as visual latency does, our conclusions allows developers to prioritize resources accordingly. This means that reducing visual latency should be prioritized in the development pipeline. Thus, developers should center on providing visually responsive games to the players. Audio latency, while it should not be neglected totally, can be categorized with a lower severity which allows developers to implement, test, and fix more urgent game elements.

Finally, our findings are most impactful for latency researchers and the research community. We found first evidence that auditory and visual latency do not impact players in the same way. Previous work demonstrated that latency in video games, generally speaking, does decrease PP and GX (Liu et al., 2021a; Beigbeder et al., 2004; Claypool & Finkel, 2014). However, previous work does not distinguish between visual and auditory latency. Thus, it is possible that the effects of auditory latency have not been accounted for. We encourage researchers to start to differentiate between visual and auditory latency. Both types of latency could have highly different effects on players. Ideally, researchers should measure visual and auditory latency independently, design experiments considering both types of latency, and report accordingly. This would enable us to deepen our knowledge about latency and its' effects on players further.

4.1.4 Conclusion

In this work, we investigated the effects of auditory latency in a custom FPS. Twenty-four participants played a custom game with four levels of added auditory latency. We found that auditory latency does not alter PP and GX. To further investigate, we performed a Bayesian analysis. We found strong evidence that the data of all tested auditory latency levels does not differ. Hence, we provide first evidence that auditory latency does not affect gamers with the same magnitude that visual latency does. However, one needs to be careful to generalize our findings to every type of game and gamer. Our findings are valid in regard to the tested game, but this does not necessarily mean that they are applicable in regards to all types of games and gamers. Future work should focus on further researching auditory latency as a concept on its own rather than investigating it combined with visual latency. Ultimately, further deepening our understanding of latency, its types, and the individual effects of auditory and visual latency allows us to better understand gamers and games alike.

4.2 Understanding Audio Latency in Commercial Video Games (Study IV)

In the previous section we gathered first evidence indicating that auditory latency does not seem to affect PP and GX. However, given the work's limitations, such as the use of a custom developed video game, a conclusive generalization about the effects of auditory latency in video games is unattainable. Hence, to strengthen our findings we conducted a second study using a commercially available and widely played FPS game and higher auditory latency levels. Furthermore, we investigate how the players' skill levels alter the effects of auditory latency, since previous work showed that latency has stronger effect on experienced video game players (Liu et al., 2021a). This section reports the experiment, its results and its implications.

This section is partly based on the following article:

Halbhuber, D., Köhler, A., Schmidbauer, M., Wiese, J., & Henze, N. (2022c). "The Effects of Auditory Latency on Experienced First-Person Shooter Players." In: *Proceedings of Mensch Und Computer 2022*. MuC '22. Darmstadt, Germany: Association for Computing Machinery, pp. 286–296. ISBN: 9781450396905. DOI: 10.1145/3543758.3543760.

4.2.1 Method

To investigate the effects of auditory latency on PP and GX in commercial video games, we modified a computer's audio output pipeline to induce four levels of controlled auditory latency and installed the FPS game CS:GO on it. Then, we conducted a study with 24 participants to investigate if auditory latency influences PP and GX using this setup.

4.2.1.1 Apparatus

For this study's apparatus, we used a stationary workstation in our laboratory. CS:GO was installed and executed in full-screen mode. CS:GO is a team-based tactical shooter and frequently used in research investigating the effects of latency in video games (Liu et al., 2021c; Liu et al., 2021a). Two opposing teams compete for an objective or aim to eliminate the enemy team in the game. However, since playing in a team involves playing with other players (either humans or AI-controlled), this is not suitable for a controlled study. In team-based gameplay, it is impossible to control for all variables such as the skill of the other players, communication in the team, or synergy between different players. Hence, to guarantee replicability, we used, in line with related work (Liu et al., 2021c; Liu et al., 2021a), CS:GO's Deathmatch mode. In Deathmatch there are no teams. Each player fights on their own. The goal in this mode is to eliminate as many opposing players as possible in a given amount of time. To further increase replicability, we disabled the ability to buy weapons - all players played with the automatic rifle M4A1 and were not able to change the weapon during gameplay. Additionally, we controlled the game map and prevented players from changing it. All gaming sessions were played on the map Mirage, which is one of the most played maps in CS:GO (Liu et al., 2021d; Liu et al., 2021c). We set the duration of each Deathmatch round to 5 min of gameplay. While playing, participants faced AI-controlled bots with medium difficulty.

We used Voicemeeter Banana[1] (VBAN) to manipulate the workstation's auditory latency. VBAN is a free-to-use audio mixer application designed to mix and manage multiple audio streams. By installing a virtual sound device, the application can delay a local audio stream; thus, the application can introduce auditory latency out-of-the-box. VBAN supports continuously delaying the audio stream for up to 500 ms. The workstation (Intel i7, Nvidia GTX980TI, 16 GB RAM) was attached to a 1440p@60Hz monitor (HP E272q), a wired mouse (Sharkoon Shark Force), and a wired keyboard (Dell L100). For audio output, we used wired headphones (Superlux HD-681 Evo).

4.2.1.2 Study Design

We conducted a study to investigate if auditory latency impacts PP and GX in CS:GO. Furthermore, we investigate if player skill alters the effect of auditory latency. We used AUDIO LATENCY and REPETITION as within-subject variables and participants' EXPERIENCE as between-subject variable. The levels of AUDIO LATENCY are based on the range of the latency of the commercially predominantly used BT protocols, which ranges from 40 ms (*aptX low latency*) (McClintock, 2014) to 270 ms (sub-band-coding (SBC)) (habr.com, 2019). Furthermore, visual latency is known to influence GX within the chosen range negatively (Liu et al., 2021c). To fully explore the effects of auditory latency, we also investigated a considerably high level of 500 ms. While the last level is higher than typical latency found in modern gaming setups, there is some indication that

[1] https://vb-audio.com/Voicemeeter/banana.htm

audio latency can reach up to this level using smartphones and bluetooth headphones (R., 2021). Thus, we categorized AUDIO LATENCY in four levels: (I) *0 ms*, (II) *40 ms*, (III) *270 ms*, and (IV) *500 ms* of added auditory latency. These levels also cover a wide enough range to potentially reveal trends in the effects of latency as seen in similar studies on visual latency (MacKenzie & Ware, 1993; Liu et al., 2021c). We also coded the number of times a participant played with one auditory latency configuration in REPETITION. REPETITION is categorized in two levels: (I) *1st* - first time playing with a particular auditory latency level, and (II) *2nd* - second time playing with a particular auditory latency. We recorded REPETITION to investigate potential habituation effects of auditory latency. Furthermore, we used EXPERIENCE as an independent between-subject variable. In line with related work, which also investigated the effects of latency in dependency of player skill in CS:GO (Liu et al., 2021a), we categorized EXPERIENCE on two levels: (1) *Expert* - participants with more than 100 hours of playtime in CS:GO, and (2) *Novice* - participants with less than 100 hours of playtime in CS:GO.

To measure GX, we again used the GEQ (IJsselsteijn et al., 2013). In our study we utilized the subscales Competence – indicating how skillful players felt, Flow – stating how immersed the players were, Tension – revealing how tense or annoyed the players were, Challenge - describing the level of challenge experienced, Positive Affect – indicating a positive gaming experience, and Negative Affect – corresponding to a negative gaming experience. As other previous work (Liu et al., 2021a) did, PP was measured by recording the players' scores in the game. *Score* increased every time a player successfully eliminated an opposing bot. Players did not lose points if they were eliminated.

In summary, we tested four different conditions based on the levels of AUDIO LATENCY. All participants played twice with each level of auditory latency. Latency levels were presented to the participants following a Latin square design to prevent a sequence effect. Participants played two consecutive rounds on each level of latency, reflected in the REPETITION variable.

4.2.1.3 Procedure and Task

Participants were greeted at our institution's laboratory by the experimenter. Then participants were informed about the procedure and the general purpose of the study. However, they were blind to the exact purpose (investigating the effect of auditory latency). After signing the consent form and thus agreeing to data collection, participants were seated in front of the computer running the game in full-screen mode. Next, each participant played one warm-up round of CS:GO Deathmatch without headphones and audio. In the warm-up round participant could ask questions about the game and procedure. After the warm-up, participants had to fill out a questionnaire about their demographic, prior gaming experience with games, and particularly their gaming experience with CS:GO. Subsequently, each participant played two 5-min rounds of Deathmatch with each level of AUDIO LATENCY. Thus, overall, participants played eighth rounds of CS:GO. After each round, participants filled out the GEQ on a different computer, and we recorded the round's score. After playing all eight rounds, participants were debriefed. In debriefing, participants were informed about the manipulated auditory latency in the game and had the opportunity to give qualitative feedback about their gaming sessions. The study took about 75 minutes per participant. Figure 4.2 shows one participant playing CS:GO in our study. The study and the data collection received ethical clearance as per the ethics policy of the University of Regensburg.

4.2.1.4 Participants

In line with the number of participants tested by previous work (Long & Gutwin, 2018; MacKenzie & Ware, 1993; Liu et al., 2021c), we invited 24 participants (9 female, 15 male) using our institution's mailing list. Mean participant age was 22.21 years (*SD* = 6.08 years), ranging from 20 to 51 years. Participants could obtain credit for their study course as compensation for participating.

4.2.2 Results

Twenty-four participants played eight rounds of CS:GO. Thus, we recorded 192 score measurements and 192 responses to the post-experience questionnaire. Furthermore, we recorded 28 relevant notes made by our participants in qualitative feedback. We first report the statistical analysis of gathered GX metrics in the following. We then continue to report the analysis of the logged and analyzed performance metrics. We conclude by outlining the qualitative feedback received from our participants.

4.2.2.1 Game Experience

Descriptive data showing the mean score and standard deviation for each subscale of the GEQ is shown in Table 4.9. The data in the table is grouped by the tested levels of AUDIO LATENCY and separated by EXPERIENCE (top and bottom).

Figure 4.2: Depicts a participant playing Counter-Strike: Global Offensive in our study. Participants played with wired headphones and four levels of controlled auditory latency (0 ms, 40 ms, 270 ms and 500 ms).

For statistical analysis we used a three-way mixed model ANOVA (AUDIO LATENCY: *0 ms*, *40 ms*, *270 ms*, *500 ms* x REPETITION: *1st*, *2nd* ∼ EXPERIENCE: *Novice*, *Expert*) as the prerequisites for ANOVA were met (Shapiro-Wilk test for all measures p >0.05). Three-way ANOVA showed no significant main effect of AUDIO LATENCY on the subscales Challenge (F(3,66) = 2.55, $p = 0.065$, $\rho_T = 0.91$) and Flow (F(3,66) = 1.80, $p = 0.145$, $\rho_T = 0.92$). ANOVA, however, revealed a significant effect of AUDIO LATENCY on Competence (F(3,66) = 3.85, $p = 0.029$, $\rho_T = 0.85$), Positive Affect (F(3,66) = 4.09, $p = 0.027$, $\rho_T = 0.93$), Negative Affect (F(3,66) = 3.77, $p = 0.026$, $\rho_T = 0.87$) and Tension (F(3,66) = 6.84, $p = 0.002$, $\rho_T = 0.86$). Subsequently, we investigated the subscales showing significant differences using a Wilcoxon test with Bonferroni Alpha-correction. Wilcoxon's test found significant differences between *0 ms* and *500 ms* ($p = 0.041$), and *270 ms* and *500 ms* ($p = 0.017$) for the Positive Affect subscale. All other pairwise comparisons for Positive Affect were not significant (all $p > 0.185$). Further post-hoc tests revealed a significant difference between *0 ms* and *500 ms* for the Negative Affect subscale ($p = 0.048$) while all other comparisons did not reach significance (all $p > 0.066$). We also found significant differences for Tension between *0 ms* and *270 ms* ($p = 0.003$) as well as between *270 ms* and *500 ms* ($p = 0.002$), all other pairs were not significant (all $p > 0.169$). We found no significant differences in the pairwise comparison of the subscale Competence (all $p > 0.095$). Generally, playing with a higher level of AUDIO LATENCY led to a decreased Positive Affect and an increased level of Tension and Negative Affect. Figure 4.3 shows the statistical analysis of all significant subscales.

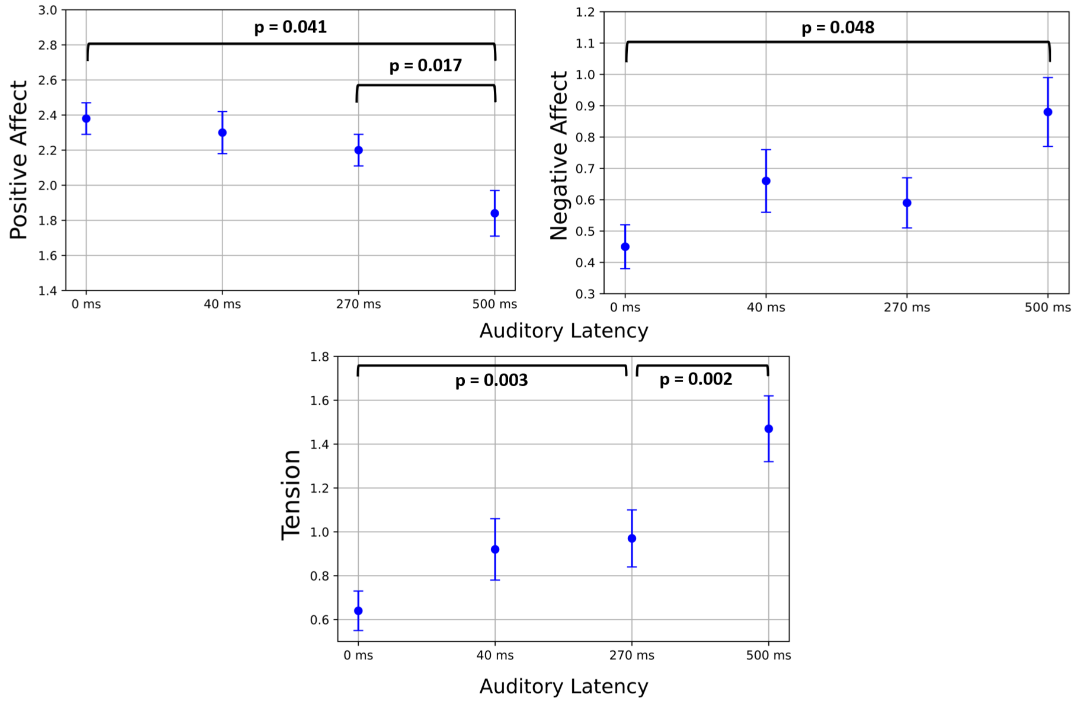


Figure 4.3: Shows the scores provided by all participants, averaged over both repetitions, for the Positive Affect (top left), Negative Affect (top right), and Tension (bottom) subscale of the Game Experience Questionnaire (IJsselsteijn et al., 2013). Significant differences are marked. Error bars show the standard error. Participants rated the Positive Affect subscale significantly worse when playing with 500 ms of auditory latency compared to playing with 0 ms and 270 ms of auditory latency. Similarly, the Negative Affect was significantly higher when playing with 500 ms compared to playing with 0 ms. Lastly, the experienced Tension was rated the highest when playing with 500 ms of auditory latency. Post-hoc testing showed significantly higher Tension rating when playing with 270 ms compared to playing with 0 ms, and when playing with 500 ms compared to playing with 270 ms.

An ANOVA revealed no main effect for REPETITION ($p = 0.108$). However, we found a significant main effect of EXPERIENCE on the Competence subscale ($F(1,22) = 7.26$, $p = 0.013$). *Experts* provided significantly higher Competence rating compared to inexperienced *Novice* players ($p = 0.014$). However, EXPERIENCE did not influence the players Positive Affect ($F(1,22) = 0.42$, $p = 0.523$), Negative Affect ($F(1,22) = 1.16$, $p = 0.292$), Challenge (F(1,22) = 0.00, p = 0.994), Tension (F(1,22) = 0.03, p = 0.870), and Flow (F(1,22) = 1.98, p = 0.173). Further investigation showed neither an interaction between AUDIO LATENCY X REPETITION on the Positive Affect (F(1,22) = 0.44, p = 0.677), Negative Affect (F(1,22) = 0.35, p = 0.765), Competence (F(1,22) = 0.27, p = 0.850), Challenge (F(1,22) = 1.62, p = 0.201), Tension (F(1,22) = 2.11, p = 0.110), and Flow (F(1,22) = 0.56, p = 0.647), nor between AUDIO LATENCY X EXPERIENCE on the Positive Affect (F(1,22) = 1.14, p = 0.333), Negative Affect (F(1,22) = 1.31, p = 0.279), Competence (F(1,22) = 0.21, p = 0.812), Challenge (F(1,22) = 1.21, p = 0.948), Tension (F(1,22) = 0.97, p = 0.391), and Flow (F(1,22) = 0.39, p = 0.757). Moreover, there was no significant interaction between REPETITION X EXPERIENCE for Positive Affect (F(1,22) = 0.16, p = 0.697), Negative Affect (F(1,22) = 0.46, p = 0.503), Competence (F(1,22) = 0.76, p = 0.392), Challenge (F(1,22) = 0.96, p = 0.338), Tension (F(1,22) = 0.02, p = 0.902), and Flow (F(1,22) = 0.80, p = 0.382). And no interaction effects between AUDIO LATENCY X REPETITION X EXPERIENCE for Positive Affect (F(1,22) = 1.43, p = 0.249), Negative Affect (F(1,22) = 0.61, p = 0.594), Competence (F(1,22) = 0.46, p = 0.712), Challenge (F(1,22) = 0.13, p = 0.922), Tension (F(1,22) = 1.79, p = 0.162), and Flow (F(1,22) = 2.42, p = 0.074).

Additionally, we investigated the effects of AUDIO LATENCY and REPETITION using EXPERIENCE as a between-subject variable and split all players in two even (N = 12) groups (*Novice* vs. *Experts*). We found a significant effect of AUDIO LATENCY on the subscale Tension (F(1,33) = 4.45, p = 0.028) for the *Novice* group. However, alpha-corrected post-tests did not reveal significant differences (all $p > 0.092$). The other subscales Positive Affect (F(1,33) = 1.07, p = 0.362), Negative Affect (F(1,33) = 0.93, p = 0.435), Challenge (F(1,33) = 1.22, p = 0.318), Flow (F(1,33) = 0.82, p = 0.489), and Competence (F(1,33) = 1.70, p = 0.210) were not significantly affected by AUDIO LATENCY. REPETITION also showed no significant effects on Positive Affect (F(1,33) = 0.05, p = 0.830), Negative Affect (F(1,33) = 0.38, p = 0.548), Challenge (F(1,33) = 3.13, p = 0.104), Tension (F(1,33) = 0.17, p = 0.690), Flow (F(1,33) = 0.40, p = 0.541), and Competence (F(1,33) = 4.26, p = 0.063). There were no significant effects of LATENCY X REPETITION in the NOVICE group on Positive Affect (F(3,33) = 0.82, p = 0.384), Negative Affect (F(3,33) = 0.39, p = 0.665), Competence (F(3,33) = 0.67, p = 0.565), Challenge (F(3,33) = 0.88, p = 0.459), Tension (F(3,33) = 2.20, p = 0.106), and Flow (F(3,33) = 2.75, p = 0.066).

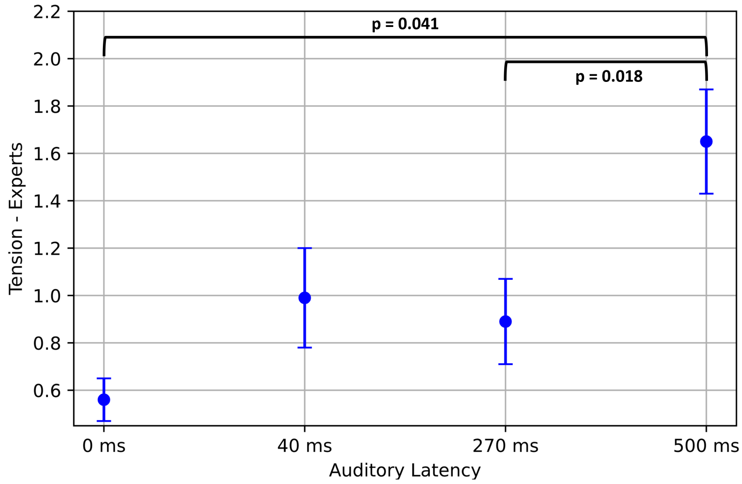


Figure 4.4: Shows the scores given by the *Expert* group in the Tension subscale of the Game Experience Questionnaire (IJsselsteijn et al., 2013). Significant differences are highlighted. Error bars show the standard error. *Experts* rated the game with 500 ms auditory latency with the highest Tension rating. Post-hoc testing showed that playing with 500 ms of auditory latency led to a significantly higher Tension rating compared to playing with 0 ms and 270 ms of auditory latency.

Investigating the *Expert* group revealed a significant effect of AUDIO LATENCY on Positive Affect (F(1,33) = 4.09, $p = 0.027$) and Tension (F(1,33) = 3.78, $p = 0.035$). However, the subscales Negative Affect (F(1,33) = 2.9, $p = 0.079$), Challenge (F(1,33) = 1.42, $p = 0.253$), Flow (F(1,33) = 1.40, $p = 0.259$), and Competence (F(1,33) = 2.68, $p = 0.072$) did not show significant effects. Post-hoc tests revealed no significant differences for Positive Affect(all p =>0.062). However, we found significant differences in Tension between *0 ms* and *500 ms* ($p = 0.041$), as well as between *270 ms* and *500 ms* ($p = 0.018$) of audio latency in the *Expert* group, while there were no significant differences between all other comparisons (all $p > 0.827$). *Experts* playing with a 500 ms of AUDIO LATENCY experienced the significantly highest level of Tension. Figure 4.4 shows the scores given by *Experts* in the Tension subscale - significant differences are highlighted. There were no significant effects of LATENCY X REPETITION in the EXPERT group on Positive Affect(F(3,33) = 0.97, $p = 0.407$), Negative Affect (F(3,33) = 0.66, $p = 0.585$), Competence (F(3,33) = 0.04, $p = 0.990$), Challenge (F(3,33) = 0.86, $p = 0.447$), Tension (F(3,33) = 1.72, $p = 0.197$), and Flow (F(3,33) = 0.86, $p = 0.472$).

Game Experiences Scores - Novice Players						
Audio Latency	**Tension**	**Competence**	**Flow**	**Challenge**	**Pos. Affect**	**Neg. Affect**
0 ms	0.72 ± 0.83	1.56 ± 1.26	2.67 ± 0.81	1.62 ± 0.62	2.13 ± 1.04	0.48 ± 0.60
40 ms	0.85 ± 0.88	1.53 ± 1.27	2.74 ± 0.59	1.72 ± 1.10	2.18 ± 1.01	0.51 ± 0.63
270 ms	1.04 ± 0.96	1.48 ± 0.94	2.54 ± 0.66	1.64 ± 0.64	2.12 ± 0.73	0.5 ± 0.56
500 ms	1.28 ± 1.03	1.10 ± 0.83	2.53 ± 0.68	1.83 ± 0.46	1.88 ± 0.65	0.66 ± 0.61

Game Experiences Scores - Expert Players						
Audio Latency	**Tension**	**Competence**	**Flow**	**Challenge**	**Pos. Affect**	**Neg. Affect**
0 ms	0.56 ± 0.49	2.55 ± 0.75	2.47 ± 0.76	1.63 ± 0.92	2.62 ± 0.72	0.43 ± 0.49
40 ms	0.99 ± 1.05	2.43 ± 0.80	2.32 ± 0.90	1.65 ± 0.88	2.43 ± 0.95	0.80 ± 0.74
270 ms	0.89 ± 0.89	2.29 ± 0.70	2.13 ± 0.63	1.65 ± 0.74	2.28 ± 0.90	0.68 ± 0.54
500 ms	1.65 ± 1.08	2.11 ± 0.95	2.21 ± 0.61	1.88 ± 0.68	1.80 ± 1.06	1.10 ± 0.90

Table 4.3: Shows the mean scores and standard deviation of each subscale of the Game Experience Questionnaire (IJsselsteijn et al., 2013) for each level of tested AUDIO LATENCY. Additionally, the data is separated by the two levels of EXPERIENCE. Top shows the data of *Novice* players, bottom shows the *GEQ* scores rated by *Expert* players.

4.2.2.2 Player Performance

Participants, on average, achieved an in-game score of 240.4 points $\pm$ 118.8 points. The average score was highest when playing with 0 ms of AUDIO LATENCY (252.5 points $\pm$ 125.4 points). All performance measures are normal distributed (Shapiro-Wilk test for all measures $p > 0.05$). Hence, we used a three-way ANOVA (AUDIO LATENCY: *0 ms*, *40 ms*, *270 ms*, *500 ms* X REPETITION: *1st*, *2nd* $\sim$ EXPERIENCE: *Novice*, *Expert*), however, ANOVA showed no significant main effect of AUDIO LATENCY ($F(2,66) = 1.84$, $p = 0.154$) on *Score*. A three-way ANOVA also revealed no significant effect of *Repetition* ($F(1,66) = 1.72$, $p = 0.229$) on *Score*. Further investigation showed no interaction between AUDIO LATENCY X REPETITION ($F(3,66) = 2.14$, $p = 0.108$), AUDIO LATENCY X EXPERIENCE ($F(3,66) = 0.63$, $p = 0.581$), REPETITION X EXPERIENCE ($F(3,66) = 1.71$, $p = 0.205$), and no interaction between AUDIO LATENCY X REPETITION X EXPERIENCE ($F(3,66) = 0.32$, $p = 0.798$).

To investigate the effects of AUDIO LATENCY and REPETITION in dependence of EXPERIENCE, we, again, split the data in two even groups (N = 12, *Novice* vs *Experts*). *Novice* players achieved a mean score of 149.92 points $\pm$ 51.46 points, while *Experts* achieved an average score of 330.95 points $\pm$ 83.79 points. ANOVA revealed no significant effect of AUDIO LATENCY ($F(1,33) = 1.38$, $p = 0.274$) nor of REPETITION ($F(1,33) = 0.01$, $p = 0.979$) on *Score* in the *Experts* groups. Similarly AUDIO LATENCY

In-game Scores by Experience			
Audio Latency	**Novice**	**Expert**	**Novice + Expert**
0 ms	151.1 ± 50.4	348.7 ± 94.8	253.2 ± 122.0
40 ms	145.0 ± 52.8	333.2 ± 82.9	239.1 ± 117.0
270 ms	148.7 ± 46.4	317.8 ± 80.6	233.3 ± 107.1
500 ms	148.4 ± 55.0	324.1 ± 72.1	236.2 ± 108.8

Table 4.4: Shows mean in-game scores achieved by each groups and combined. Experts achieved the highest in-game score when playing with *0 ms* AUDIO LATENCY.

had no effect (F(1,33) = 0.78, p = 0.51) on the *Score* of *Novice* players. However, ANOVA revealed a significant effect of REPETITION (F(1,33) = 5.31, p = 0.004) on the *Score* of *Novice* Players. Inexperienced players achieved a significantly higher *Score* in the *2nd* round. Furthermore, we found no interaction between AUDIO LATENCY and REPETITION neither for EXPERTS (F(3,33) = 1.04, p = 0.379) nor for NOVICES (F(3,33) = 1.69, p = 0.189). Table 4.4 shows mean *Score* values and standard deviation for each EXPERIENCE group separately and both groups combined for each level of tested auditory latency.

4.2.2.3 Qualitative Feedback

In general, all participants enjoyed participating in the study. However, some participants stated that they felt like the game was not responding to their actions. All Experts were able to tell that we manipulated the auditory latency of the game. Only one Novice came to the same conclusion. Novices mainly thought we manipulated in-game mechanics such as the frame rate of the game or the bot difficulty. Manipulation check showed that no participants, regardless of the level of experience, could correctly tell in which round the auditory latency was the highest.

4.2.3 Discussion

In this section, we first discuss found effects of auditory latency on players' gaming experiences. Next, we continue by examining the influence of auditory latency on PP. Subsequently, we summarize with a general discussion on the our findings and our work. Lastly, we conclude this section by exploring the implication of our findings for gamers, developers, and game researchers.

4.2.3.1 Game Experience

Our results consistently show that 40 ms of auditory latency did not influence GX while playing CS:GO. However, starting at 270 ms auditory latency negatively affects the perceived tension while playing as well as worsen the positive associations with the game which results in a heighten negative and a reduced positive affect towards the game. At 500 ms players significantly stronger associated negative feelings with the game. Furthermore, we found that the adverse effects of auditory latency on the perceived tension while playing are more pronounced for experienced players. Our findings align with related work investigating the effects of visual latency in video games. First, previous work showed that increasing visual latency modulates the perceived GX such as the negative (Liu et al., 2021c) and positive affect. Second, related research showed that visual latency affects experienced players stronger than inexperienced players (Liu et al., 2021a). Our work shows that the same is true for auditory latency. Considering that we did not find an effect of auditory latency on the GX of novice players but did find an effect when investigating the data of experienced players, we can conclude that the tolerance for auditory latency decreases with prior knowledge and experience with the game. We argue that there are two possible explanations for the differences found for the in-game experience: Previous work showed that visual latency tolerance increases with task complexity (Battle et al., 2019). Since inexperienced players have little knowledge about the game and its mechanics, the perceived complexity is higher than the perception of high-skilled players. Thus, their latency threshold is higher. Our second perspective on the found effects is that more experienced players, which have spent a considerable amount of time (>100 hours) in the game, are disturbed by the shifted perceptual input-output schema. Those players are used to the game responding in a certain way. They, additionally, are used to playing on their own setup; by disturbing the mental model of how the game behaves typically, we also disturb their gaming experience. Every time they played with an artificially increased auditory latency level, their mental representation of the game was violated. The higher the auditory latency was, the more pronounced this violation manifested. A larger discrepancy between what the players are used to and how the game actually responded in the study led to a more severe negative influence on the perceived gaming experience.

Our findings regarding the effects of auditory latency on the gaming experience for small amounts ($\leq$ 40 ms) are negligible. On the other hand, our work shows that higher auditory latency ($\geq$ 270 ms) negatively affects the gaming experience. However, auditory latency is not as impacting compared to visual latency, which starts negatively influencing GX at 25 ms (Liu et al., 2021c).

4.2.3.2 Player Performance

While a downwards trend is noticeable in the achieved scores, our analysis did not reveal a significant effect of auditory latency on players' performances, neither for novice nor expert players. We did find a significant effect of the number of times one particular auditory latency level was played on the score of the inexperienced players. However, we assume that the increase of score in the second round by novice players is not due to habituation to the auditory latency but rather due to novice players learning the game mechanics. Thus, novice players started getting better in the game instead of adapting to the auditory latency. This is also supported by the fact that we did not see the same significant increase of points in the expert group. Achieved points stayed the same in both rounds for the experienced players. Hence, we cannot conclude, as related work does for visual latency (Sabet et al., 2019b) that players adapt to auditory latency.

The lack of significant effects of auditory latency on PP might be due to multiple reasons. For example, as games predominately use the visual channel to convey game-relevant information, players might rely stronger on visual information than on auditory information. As we reported in the previous study, this may also affect players in this study. Since CS:GO, and especially the mode we tested in our study, puts players in a position where they have to defend themselves against a never-ending stream of enemies, it may be possible that stress was induced. Thus, it is more likely that players rely solely on their most dominant sense - visual perception.

Overall, we found that auditory latency does not negatively influence PP. Furthermore, comparing auditory latency to visual latency, it is evident that the adverse effects of auditory latency on PP are not as pronounced as those of visual latency.

4.2.3.3 Implications of our Findings

Our findings have implications for gamers, developers, and researchers alike. Gamers should be aware of the effects that auditory latency may have on them. Starting at 270 ms of auditory latency, these negative effects can be induced by commercially available sound equipment using the SBC codec. Therefore, gamers should avoid buying equipment using the SBC codec and instead obtain headphones using the aptX low latency standard, which has an auditory latency of less than 40 ms. However, while this recommendation is generally applicable, it may not be necessary to exchange existing equipment. As we showed, the effects of auditory latency are more pronounced for experienced gamers. Casual gamers, spending only little time gaming, may not profit from the reduced auditory latency of pricier aptX low latency hardware.

Game developers can also benefit from our findings. Since we showed that auditory latency does not affect players to the same extent as visual latency, developers can prioritize their resources accordingly. This means that reducing visual latency should be a priority in the development pipeline. As we highlighted in the previous study in Section 4.1, developers should focus on providing players with visually appealing and responsive games. While audio latency should not be neglected entirely, it can be ranked at lower severity, allowing developers to implement, test, and fix more critical game elements first. This implication is particularly relevant in light of novel gaming paradigms such as cloud-based game streaming (CGS). In CGS, games are streamed via the Internet; A high network latency thus leads to higher overall latency. Since auditory latency does not affect players with the same magnitude as visual latency, CGS providers should prioritize delivering visual information if the players' connections are not stable or fast enough to receive audio and visual information synchronously.

Finally, the most important implication of our results concerns researchers and the research community. We found further indication that auditory and visual latency do not affect players in the same way. Previous work showed that latency in video games generally reduces player experience and performance. However, previous work has not distinguished between visual and auditory latency. Therefore, it is possible that the effects of auditory latency have not been considered. As we showed that auditory latency does affect high-skilled players, we encourage researchers to distinguish between visual and auditory latency. Both types of latency have different effects on players. Ideally, researchers should measure visual and auditory latency independently, design experiments considering both types of latency, and report accordingly.

4.2.3.4 Limitations and Future Work

With our work, we provide first empirical evidence that auditory latency affects the GX of expert gamers in CS:GO. However, since we tested only CS:GO, our findings may not generalize to the entire gaming landscape. Future work, thus, should investigate the effects of auditory latency in other video games and game genres, such as rhythm games or RTS games. Furthermore, considering that we found negative effects of auditory latency on expert players despite the relatively small number of participants in this group (n = 12), it is possible that we missed smaller effects. Future work should build on our work and further investigate the negative effects of auditory latency on expert players. Moreover, since auditory latency is seldom a stand-alone issue in real gaming scenarios, and players are constantly confronted with a mixture of visual, auditory, input, and network latency, future work should also address the interaction between different types of latency. It might be the case that the negative effects of auditory latency are influenced by other types of latency, such as visual latency.

Moreover, we conducted this work amidst the ongoing Covid-19 pandemic. Although all regulations of our institution were constantly controlled and thoroughly documented, we nevertheless minimized the amount of in-person exchange required. Thus, we also aimed to minimize the number of participants needed to run our study. We did this by following related work investigating visual latency. Previous work, showed that a small number of participants is sufficient to investigate the effects of latency in video games generally (Long and Gutwin: n = 18 and n = 20 (2018), MacKenzie and Ware: n = 8 (1993)) and in CS:GO (Liu et al.: n = 25 (2021c)) in particular. However, while previous and our work shows that this approach reveals significant effects when investigating latency, it also may results in missing small effects of auditory latency. Investigating a larger number of participants may allow for a more detailed analysis of the effects of auditory latency on GX and PP.

Generally, it needs to be noted that our test system had an inherent auditory latency that we could not manipulate directly. Computers have an auditory latency of up to 70 ms. This baseline auditory latency is influenced by the applications running, the used operating system, the installed sound drivers, and the utilized hardware. Our system's baseline latency might be too high to reveal the effects of adding small amounts (40 ms) of auditory latency to CS:GO. There may be an auditory latency threshold that our system's baseline crossed. However, since end users' systems as well have a base latency our work replicates a natural gaming setup. Nevertheless, with our work, we successfully showcased the effects of large amounts of added auditory latency.

Furthermore, while the distinction between expert and novice players based on the hours previously played in CS:GO was used in prior work and seems to be reasonable based on the differences we found between the groups, this distinction might not always reflect the skill of an individual player. In addition, the threshold of 100 hours was chosen based on results of prior studies and is not necessarily ideal. Some people might learn the game faster due to their talents and strengths. Others perform better with less hours because they played similar games before. Moreover, if participants are used to different hardware than provided in our lab, they could perform worse than someone with less experience but similar peripherals. Consequently, the classification based on hours played is simple and in our case effective but could be improved in future works to achieve a more nuanced and realistic graduation of skill levels.

Lastly, another limitation of this work can be found in the game we used. We tested a FPS game for our work because this type of game is extremely fast-paced and requires split-second decision-making to perform well. Additionally, previous work showed that the adverse effects of visual latency are particularly pronounced in FPS games (Claypool & Claypool, 2006; Liu et al., 2021d; Liu et al., 2021c). While our findings are valid for CS:GO, one needs to be careful to generalize it to the vast landscape of gaming with its numerous type of genres, games, and gamers. It is possible that investigating auditory latency in other games or genres, such as arcade games, leads to the unraveling of effects unnoticed by our work.

4.2.4 Conclusion

In this work, we investigated the effects of auditory latency in CS:GO. Twenty-four participants with two different skill levels (novices and experts) played eight rounds of Deathmatch in which they had to fight an endless stream of AI-controlled enemies. Each round was inflicted with one of four levels of controlled auditory latency (0 ms, 40 ms, 270 ms and 500 ms). We found that expert players experience a significantly higher level of tension and a significantly decreased positive affect starting at 270 ms of auditory latency. At 500 ms auditory latency, those negative effects are amplified, and additionally, experts started to significantly stronger associate the game with negative feelings. We found no effect of auditory latency on the GX of novice players. Furthermore, we did not reveal significant effects of auditory latency on players' performances - regardless of their skill level.

With our work, we deepen our knowledge about latency and its effects on players, which aids researchers in better understanding players and their interaction with games. Furthermore, game developers can build upon this knowledge to develop games with optimized gaming experience.

4.3 Understanding In-Game Perspective and Latency (Study V)

In the previous two sections (Section 4.2 and 4.1), we learned about auditory latency and its effects on video game players to answer this thesis's third RQ (cf. Table 1.1). While the first study indicated that auditory latency does neither affect PP nor GX, the second study revealed that its effects are depended of the players' skills. We found that experienced players are potentially affected by auditory latency. However, latency not only influences the auditory components of a game but can also affect the game's visual presentation. Previous work repeatedly showed that visual latency (and/or a combination of visual and auditory latency) has significant negative effects on PP and GX (Liu et al., 2021c; Claypool & Claypool, 2006; Claypool & Finkel, 2014). However, it is unclear how the visual characteristics or game mechanics of the game alter the game's latency sensitivity.

Hence, previous work takes different approaches and, for example, investigates how game characteristics, such as the game's feedback frequency, the importance of player actions, or the required spatial accuracy when carrying out in-game actions, alter the game's latency sensibility (Sabet et al., 2020a). That work zeros into game mechanics to establish which factors manipulate the effects of latency on a micro level. Similarly, previous work also aims to categorize latency sensitivity on a macro level using, for example, the game's pacing (Schmidt et al., 2017) or its genre (Claypool & Claypool, 2006). On this note, Claypool and Claypool (2006) argue that a game's genre and how the player visually perceives the game world are critical factors influencing the game's latency sensitivity. In their work, the authors highlight that different exemplary game genres, such as FPS, RPG, and RTS games, incorporate different in-game perspectives. Thus, the authors conclude that different video game genres have different thresholds before latency negatively affects performance and experience. However, while the in-game perspective certainly is part of how video game genres can be defined, video game genres are fluent. The perspective alone does not define the game's genre, as evidenced by FPS/RTS hybrids such as Executive Assault 2 (Hesketh, 2022). Genres differ in countless other aspects, such as pacing, narrative methods, and interaction techniques. Nevertheless, the in-game perspective plays a unique role in how the players visually perceive and experience the game. Whether it is a First-Person View (FPV), Third-Person View (TPV), or Bird's-Eye View (BEV), the perspective offers distinct visual and spatial information to the player. It affects how players perceive the game environment, navigate through it, and make strategic decisions. Moreover, the in-game perspective can influence the player's immersion, spatial awareness, and emotional engagement with the game (Denisova & Cairns, 2015a; Monteiro et al., 2018). While the in-game perspective is a crucial element of how the game is perceived and played, it is unclear how the effects of latency are altered by it. Thus, it is unclear how the visual in-game perspective alters the effects of latency (RQ4).

This section is partly based on the following article:

Halbhuber, D., Schauhuber, P., Schwind, V., & Henze, N. (2023b). "The Effects of Latency and In-Game Perspective on Player Performance and Game Experience." In: *Proceedings of the ACM on Human-Computer Interaction* 7.CHI PLAY, pp. 1308–1329. DOI: 10.1145/3611070.

4.3.1 Background and Research Rationale

One reason for the small number of works investigating the in-game perspective as a modulator of latency is that most games only integrate one or two in-game perspectives. Therefore, using commercial video games, it is hard to consider the perspective's influence on latency perception in isolation.

Our work starts solving this problem by providing insight into how the in-game perspective alters the effects of latency on PP and GX. To achieve this, we developed a shooting video game that features the same core gameplay with three different in-game perspectives. Using the game, we conducted a within-subjects study with 36 participants playing with three different perspectives (FPV, TPV, and BEV) and two levels of latency (low and high). Inferential analysis shows that players performed significantly worse when playing with a high latency independently of the used in-game perspective. The adverse effects of latency manifested in players hitting fewer targets and needing more time to shoot targets successfully. We also found that high latency significantly reduced perceived ease of control, progress feedback, and challenge appropriateness. Furthermore, high latency led to a reduced feeling of mastery, immersion, autonomy, and overall enjoyment. However, using Bayesian analysis, we found that our data support a model in which latency and perspective do not interact. We found evidence for a model demonstrating no true effect of the in-game perspective on PP or GX. Our findings are crucial to video game and latency researchers, as we demonstrate that the in-game perspective does not necessarily dictate the latency sensitivity of video games. This allows future research to investigate the interaction with other aspects of video games, such as their pacing or narrative methods and latency.

4.3.2 Apparatus

To investigate how the used in-game perspective alters the effects of latency on PP and GX, we developed a custom video game. We developed a shooting game since this type of game allows translating the core gameplay (shooting and tracking targets) to different perspectives without fundamentally altering the game mechanics. Furthermore, previous work showed that shooting games are particularly negatively affected by latency since these games require timely target tracking and selection to perform well (Claypool & Claypool, 2010). After developing the game, we used it to measure the local latency of our test setup. Since we aspire to investigate the effects of latency, we need to account for local latency, which is inherently part of the computer setup due to polling and update rates of the used technical equipment such as the monitor, computer mouse, keyboard, and the workstation (Wimmer et al., 2019).

4.3.2.1 Implementation

In the first step, we designed and developed the game world and the player avatar. Figure 4.5 shows the game world from a top-down tilted angle. The game world comprises one large main room (Figure 4.5, B) enclosed by two smaller rooms. The small left room (Figure 4.5, A) serves as the starting point for the player, in which the controllable avatar spawns. The small room on the right (4.5, C) acts as the endpoint for the gaming session.

Next, we implemented three avatar controllers, one for each in-game perspective. Each controller allows the player to move the avatar and the avatar's viewport. Additionally, the controllers enable players to move a virtual weapon using the mouse and to shoot the weapon using a left mouse click. Avatar movement is controlled using the WASD keys (W = forward, A = left, S = backward, D = right). Rotational movement is controlled by using the mouse in FPV and TPV. In BEV, the rotational movement of the avatar follows the virtual weapon's cross-hair. Figure 4.6 depicts the player's view from FPV (left), TPV (middle), and BEV (right). Translating between FPV and TPV is quickly done. The viewport only needs to be slightly offset. However, translating avatar movement and behavior to BEV from FPV or TPV without fundamentally changing the presented amount of information or the game's interaction technique is non-trivial.

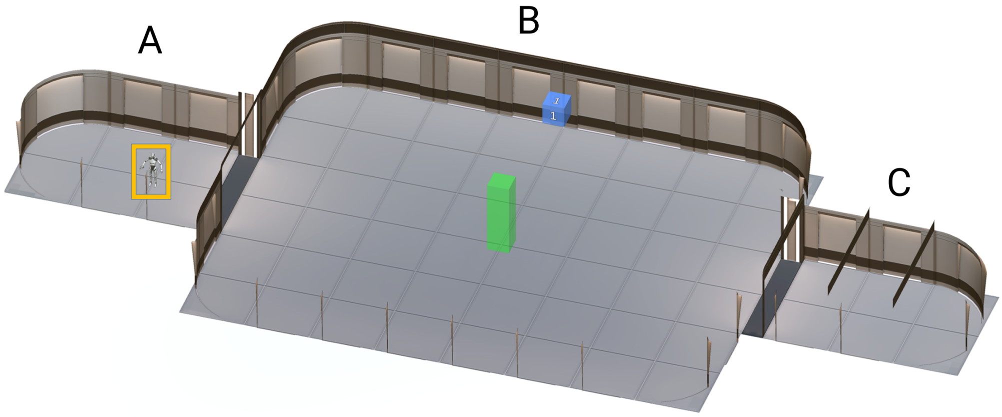


Figure 4.5: Depicts the game world of our custom shooting game. In the room on the left (A), players start playing with their avatar (yellow box). In the middle room (B), the actual gameplay takes place. Players move their avatars to the starting point (green bar) to start the study. Players moved their avatars to the end room (C) when finished.

BEV offers, per design, more information to the players. Players can see targets not only in front of them but in the whole game world. To prevent BEV from biasing our work by granting an advantage over other perspectives, we developed a custom occlusion map. This map is applied to the player's viewport when playing with BEV and limits the player's view to a cone in front of the avatar (see Figure 4.6). This approach mimics the viewport of FPV and TPV and is a common technique in tactical games such as X:COM (Firaxis, 2022) or Divinity 2: Original Sins (Larian Studios, 2022). Furthermore, changing the in-game perspective from FPV or TPV to BEV transforms the target selection task from a three-dimensional to a two-dimensional problem, effectively removing one dimension. This shift has notable implications for the player's interaction with the game environment and the challenges they face in selecting targets. In FPV, players typically navigate and interact with objects in a three-dimensional space, requiring them to account for depth, distance, and spatial relationships when selecting targets. On the other hand, with BEV, the game presents a flattened representation of the environment, where distance and depth are often conveyed through visual cues such as size, shading, or perspective. As a result, the target selection process becomes predominantly focused on the 2D plane. This reduction in complexity potentially makes target selection easier.

Nevertheless, since players are stationary while shooting stationary targets in our game and thus are not deeply interacting with the three-dimensional space, we argue that the actual differences between in-game perspectives are negligible.

Despite that, using BEV not only entails advantages. Since the player is situated above the game world and thus farther away from the actual gameplay, some tasks get more difficult. This is particularly the case in shooting games, in which the player is positioned further off from the targets, thus they are smaller and fundamentally harder to hit. However, since the whole game world's size is decreased, players require less real-world mouse movement to cover the same amount of in-game distance as in FPV or TPV. We anticipate that the increase in difficulty induced by the decreased target size and the decreased path length balance each other. This anticipation aligns with Fitts' law (Fitts, 1954). The law states that reducing the size of a target increases the time required to reach it in a 2D selection task, while reducing the distance to the target decreases the movement time. Hence, the overall challenge in our game should be constant over all perspectives.

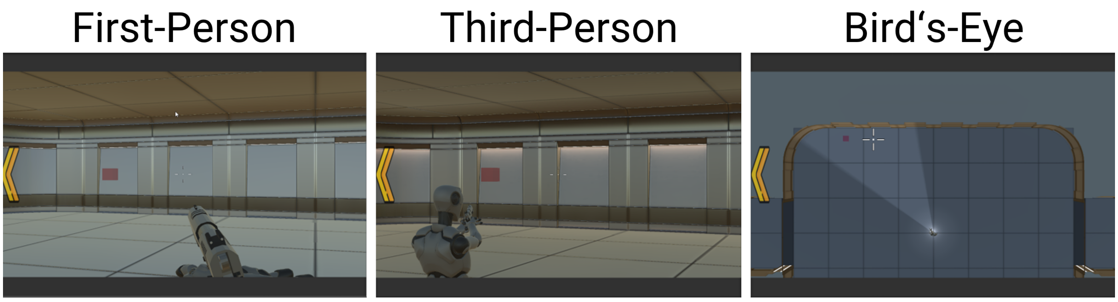


Figure 4.6: Depicts three screenshots from our game. Each subfigure depicts the players' view from a different in-game perspective. Left shows the viewport in First-Person-View. The middle subfigure shows the Thrid-Person-View and the right depicts the view in Bird's-Eye-View. Additionally, each subfigure depicts a red target, which players had to shoot in the study. Furthermore, the right also depicts how we reduced the Birds-Eyes-View inherent advantage by using a cone-shaped occlusion map. Players using the Birds-Eye View could only see targets within the highlighted cone.

The player's objective in the game is to shoot a fixed number of targets. To start the main task of the game, the player has to move to the starting point. The green bar in Figure 4.5 highlights the starting area. Once the player reaches the starting area, the avatar can no longer be moved using the keyboard. However, players can still aim using the computer mouse. Stationary targets appear in a controlled but randomized order in a designated target area in front of the player. Our game features two types of targets: (1) Red targets, which the player needs to shoot as fast as possible, and (2) blue targets, which the player has to track by placing the weapon's cross-hair on it. The blue targets are always positioned right before the player's starting point (see green bar in Figure 4.5) and act as inter-trial fixation points. Players must actively move their mouse to find and shoot the appearing red targets. Players obtain points for successfully tracking and shooting targets. The faster players shoot or track the target after its appearance, the more points are awarded. However, players do not lose points for missing targets. While playing, we log all game events to a local database, such as successful hits, tracks, misses, and reaction times. We used *Unity 3D* (Version 2020.2.16f) to develop the game.

4.3.2.2 Measuring Local Latency

We measured the game's local latency, comprised of the used hardware and the game's run-time, to determine precisely what latency players will face. For the measurement, we installed our game on a stationary workstation in our laboratory. The workstation (Intel i7, Nvidia GT970, 16 GB RAM) was attached to a monitor (24" FullHD @60Hz), a computer mouse (Razor Viper 8K), and headphones. The game ran in full-screen mode. Using a 240 fps camera (4,167 ms/frame, GoPro Black 7) and the procedure used by previous work (Ivkovic et al., 2015; Long & Gutwin, 2018), we captured both the system's mouse and the game screen. By manually comparing the physical mouse click with an in-game event (firing of the virtual weapon), we determined the local latency of our setup. We repeated this measurement 20 times. We found that our game has an average local latency of 11.46 frames (SD = 3.15 frames, n = 20), which translates to 47.75 ms (SD = 13.12 ms). This latency serves as the baseline latency of our game, and all subsequent latency values include this baseline without explicitly stating it.

4.3.3 Method

We conducted a study using our game to investigate how different in-game perspectives and different levels of latency affect the GX and PP. To simulate different latency levels we used input buffering techniques in the game. Input buffering involves delaying the processing of user inputs for a specified duration, thereby mimicking the effect of local latency. In our study, we implemented two latency levels by introducing fixed delays in the processing of user inputs.

Specifically, we modified the game's code to incorporate a predetermined delay for each input (mouse and keyboard) received from the players. This delay was set to correspond to the desired latency level. By introducing delays in this way, we created a controlled environment that allowed us to investigate the effects of latency on gameplay performance and experience without introducing confounding factors.

4.3.3.1 Study Design

To control for perspective and latency, we utilized two independent variables (IVs) in a 2 x 3 within-design: (1) LATENCY, which corresponds to the latency participants are playing with. LATENCY has two levels: (I) *low* - which refers to 0 ms of artificially added latency and *high* - which represents playing with 150 ms of artificial latency. We chose the *high* latency in line with previous work, which shows varying thresholds of up to 150 - 180 ms (Armitage, 2003; Long & Gutwin, 2018) for latency tolerance for video games in general and shooting games in particular. The second IV is (2) PERSPECTIVE - which refers to the in-game perspective participants are playing with. PERSPECTIVE has three levels: (I) First-Person-View (*FPV*), (II) Third-Person-View (*TPV*), and (III) Bird's-Eye-View (*BEV*).

To measure the players' performance and GX, we used a set of dependent variables (DVs). In line with related work (Claypool & Claypool, 2006), we utilized *Reaction Time* and *Accuracy* to quantify the objective PP. *Reaction Time* is the time required to hit a target. A lower *Reaction Time*, thus, corresponds to less time required for shooting the appearing target. *Accuracy* quantifies the participants' accuracy and is built by the ratio of total shots to successful shots. Therefore, a higher *Accuracy* refers to a higher level of precision.

We used the latest version of the 30-item PXI (Abeele et al., 2020) with its subscales: Ease of Control, Progress Feedback, Audiovisual Appeal, Clarity of Goals, Challenge, Mastery, Curiosity, Immersion, Autonomy, and Meaning to quantitatively evaluate the GX. Furthermore, we added the recommended questions to asses Enjoyment to the questionnaire. The instrument was administered according to original work (Abeele et al., 2020).

4.3.3.2 Apparatus and Procedure

For the study, we used the same system for which we determined the local latency of our game. The game was executed in full-screen mode. The study was conducted in our laboratory, which was quiet and free of external disturbances.

Upon arrival, participants were greeted at our laboratory. Participants were not informed about the exact purpose of the study (investigating the interaction between latency and perspective) to prevent a bias induced by the anticipation of latency (Kosch et al., 2022) but were told to test a novel game. They were informed about the study's procedure after giving informed consent and agreement to data collection. Each participant started the study with a short warm-up phase in our game. Following the warm-up, the actual study started, in which participants aimed to maximize their score by shooting targets. Participants were told they would receive a gift card if they beat the game's high score to further incentivize them. However, all participants received the gift card, independently of if they beat the game's high score. Participants played six rounds, each with a unique combination of LATENCY and PERSPECTIVE. Conditions were balanced using a Latin Square to avoid sequence effects. After each round, an in-game performance overview showcasing the game's high score and the participant's points obtained in the previous round was presented. Next, participants filled out the PXI and Enjoyment questionnaire before the following condition started. Finally, the study was concluded with a debriefing, in which the participants were informed about the detailed purpose of the study. We estimated a total duration of one hour for the study, which was designed following the research ethics policy of the University of Regensburg.

4.3.3.3 Task

In each of the six rounds, participants started playing with one combination of LATENCY and PERSPECTIVE. At the start of each round, the participants' avatar was situated in the starting area (see Figure 4.5, A). Participants could move freely for as long as they wished in the game world to accommodate themselves to the game controls and the new perspective. Then, participants started the actual task by moving the avatar to the center of the main room of the game world. After entering the starting position (green bar in Figure 4.5, B), the game restricted the avatar's movement. From this point forward, participants were not able to move the avatar but only to move the virtual weapon. After the avatar was locked in place, the first inter-trial target (blue) appeared. Participants had to position the cross-hair of their virtual weapon for 2 seconds on the blue target. We utilized the inter-trial target to ensure that each trial (shooting of a red target) started with the same preconditions independent of the in-game perspective. After 2 seconds, the blue target disappeared, and a stationary red target spawned with a randomized size at a random position within the designated spawn area. Spawn position and sizes for red targets for all rounds were randomized once and held constant over all participants. We manipulated red targets' sizes using *Unity*'s scaling component. The targets' sizes varied between 0.4 and 2.0 in reference to *Unity*'s standard *Cube* primitive (Unity 3D, 2023). Figure 4.7 depicts a blue inter-trial target and the spawn area for the targets in red. After shooting the red target, it disappeared, and the blue target in front of the avatar spawned again. This course of action was repeated 30 times (30 blue and 30 red targets). Upon completion, the participants were notified, and the avatar was free to move again. To end the round participants moved the avatar to the end zone (see Figure 4.5 C). This procedure was repeated for all six combinations of LATENCY and PERSPECTIVE.

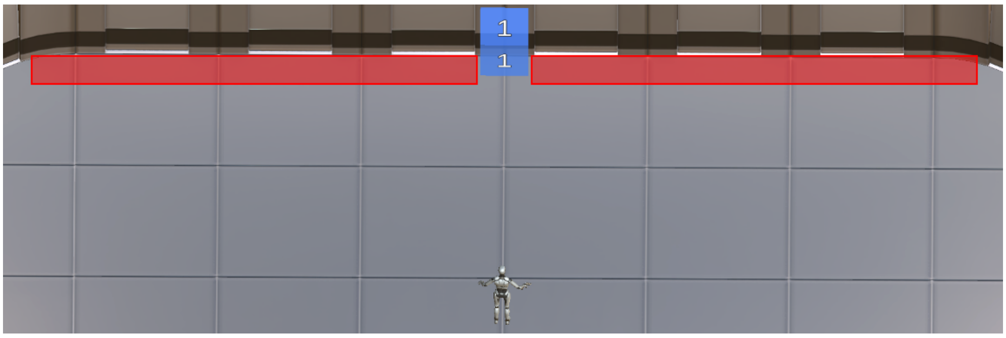


Figure 4.7: Depicts an aerial view of the middle room of our game world. The avatar is situated at the bottom of the figure. In front of the avatar is a blue target, which the players had to track to spawn the next red target. The red targets spawned within the red area left and right of the blue target. The spawn area was not depicted in actual gameplay but is shown in this figure for clarification purposes only.

4.3.3.4 Participants

We recruited 36 participants (25 male, ten female, and one non-binary) using our institution's mailing list and social media. The participants' ages ranged from 19 to 35 years, with a mean age of 23.42 years (*SD* = 3.77 years). Thirty-two participants were right-handed, and four participants were left-handed. We allowed participants to choose which hand to use to control the computer mouse (right or left). All participants decided to use the right hand. Twelve participants said they play video games daily, eleven participants play video games weekly, four participants play a few times per month, and the remaining nine participants play less than a few times monthly. Thirty-one participants were students at our institution and received credit for their course of study. The other five participants had technical backgrounds (Computer Science, Engineering, and Software Development). All participants received an additional 5 € gift card as compensation.

4.3.4 Results

In the following, we report the analysis of all measures. Descriptive results can be found in Table 4.5 (performance measures), Table 4.6 (enjoyment measures), and Table 4.7 (PXI). All gathered data were tested for normality using Shapiro-Wilk tests (normal distribution assumed if $p > .05$). All measures violated Shapiro-Wilk's test for normality. Thus, we used a rank-aligned two-way RM-ART-ANOVA (non-parametric data) (Wobbrock et al., 2011) for the inferential assessment of LATENCY and PERSPECTIVE (LATENCY: *low* vs. *high*) x (PERSPECTIVE: *FPV* vs. *TPV* vs. *BEV*) with repeated measures on both factors. We used an alpha level of .05 for all statistical tests (significance assumed if $p <$ .05).

Since classical null hypothesis testing only reveals differences between distributions - ANOVA either accepts or rejects a hypothesis - it can not reveal if an insignificant difference indicates a similarity between the investigated distribution. Thus, we conducted a Bayesian analysis to investigate the similarity in the data of our conditions. Rather than just rejecting a hypothesis, an Bayesian ANOVA calculates the probability that the acceptance of the null hypothesis (no differences in distribution) is correct (Cleophas & Zwinderman, 2018; Wagenmakers et al., 2018). Contrary to classical null hypothesis testing, thus, Bayesian inference calculates probabilities for both: H_0 and H_1. For Bayesian analysis, we used the R-package (R Core Team, 2023) BayesFactor (Morey & Rouder, 2022). Bayes factors (Lavine & Schervish, 1999; Kass & Raftery, 1995) are formulated as BF_{10}, which indicates how much more likely a model that supports H_1 over H_0 is. Bayes factors are interpreted in line with Andraszewicz et al.s' postulation (Andraszewicz et al., 2015), which categorizes continuous Bayes factors into discrete levels of evidence. Table 4.8 depicts the used Bayes factor evidence categories. To increase readability, we structure the following sections by independent instead of dependent variables.

Mean and Standard Deviation of Performance Measures - M(SD)					
Measure	Overall	Latency		Perspective	
Reaction Time	20.17(19.64)	*low*	15.81(12.15)	*FPV*	16.32(12.44)
				TPV	15.69(12.99)
				BEW	15.43(10.93)
		high	24.54(24.20)	*FPV*	25.23(27.20)
				TPV	24.50(25.17)
				BEV	23.88(19.60)
Accuracy	0.64(0.15)	*low*	0.72(0.12)	*FPV*	0.71(0.15)
				TPV	0.73(0.11)
				BEW	0.73(0.11)
		high	0.56(0.12)	*FPV*	0.54(0.12)
				TPV	0.55(0.12)
				BEV	0.57(0.12)

Table 4.5: Shows mean and standard deviation of *Reaction Time* and *Accuracy* overall and for each level and combination of LATENCY and *Perspective*. Overall, participants needed 20.16 s (*SD* = 19.64 s) to find, track, and shoot appearing targets. Furthermore, participants had an average precision of 0.64 (*SD* = 0.15), which translates to 64 % successful shots.

Mean and Standard Deviation of Enjoyment Measures - M(SD)					
Measure	Overall	Latency		Perspective	
Enjoyment	1.41(1.32)	*low*	1.82(0.90)	*FPV*	1.92(0.84)
				TPV	1.76(0.93)
				BEW	1.78(0.94)
		high	0.99(1.53)	*FPV*	1.03(1.46)
				TPV	0.92(1.55)
				BEV	1.01(1.60)

Table 4.6: Shows mean and standard deviation of the *Enjoyment* data overall, and for each level and combination of LATENCY and *Perspective*. Overall, participants rated their enjoyment with 1.41 points (*SD* = 1.32 s).

4.3.4.1 Inferential Analysis

Latency

ART-ANOVA revealed a significant main effect of LATENCY on *Reaction Time* ($F(1,35)$ = 643.775, $p < 0.001$, $\eta_p^2 = 0.94$) and on *Accuracy* ($F(1,35) = 144.660$, $p < 0.001$, η_p^2 = 0.81). Figure 4.8 depicts mean *Reaction Time* (left) and *Accuracy* (right) values for both

Mean and Standard Deviation of Player Experience Inventory - M(SD)											
	Overall	Lat.	M(SD)	Pers.	M(SD)		Overall	Lat.	M(SD)	Pers.	M(SD)
EoC	2.20(0.74)	*low*	2.42(0.61)	*FPV*	2.41(0.64)	**PrF**	-0.46(1.37)	*low*	-0.29(1.41)	*FPV*	-0.28(1.37)
				TPV	2.46(0.61)					*TPV*	-0.25(1.41)
				BEV	2.41(0.59)					*BEV*	-0.33(1.48)
		high	1.98(0.79)	*FPV*	2.0(0.76)			*high*	-0.61(1.32)	*FPV*	-0.59(1.34)
				TPV	2.0(0.90)					*TPV*	-0.68(1.40)
				BEW	1.96(0.73)					*BEW*	-0.62(1.27)
AvA	0.80(1.60)	*low*	0.92(1.50)	*FPV*	0.95(1.50)	**CoG**	2.33(0.84)	*low*	2.40(0.76)	*FPV*	2.39(0.85)
				TPV	0.96(1.35)					*TPV*	2.35(0.80)
				BEV	0.87(1.68)					*BEV*	2.46(0.64)
		high	0.67(1.70)	*FPV*	0.71(1.51)			*high*	2.25(0.91)	*FPV*	2.25(0.94)
				TPV	0.77(1.76)					*TPV*	2.28(0.92)
				BEW	0.54(1.85)					*BEW*	2.24(0.90)
Cha	1.58(1.07)	*low*	1.91(0.81)	*FPV*	1.98(0.82)	**Mas**	0.92(1.18)	*low*	1.41(0.96)	*FPV*	1.51(0.85)
				TPV	1.89(0.75)					*TPV*	1.44(1.02)
				BEV	1.87(0.86)					*BEV*	1.28(1.02)
		high	1.25(1.20)	*FPV*	1.37(1.07)			*high*	0.41(1.17)	*FPV*	0.42(1.14)
				TPV	1.22(1.18)					*TPV*	0.32(1.14)
				BEW	1.15(1.35)					*BEW*	0.50(1.24)
Cur	0.28(1.52)	*low*	0.44(1.47)	*FPV*	0.36(1.44)	**Imm**	1.35(1.19)	*low*	1.54(1.09)	*FPV*	1.80(0.90)
				TPV	0.49(1.52)					*TPV*	1.5(1.16)
				BEV	0.49(1.49)					*BEV*	1.34(1.16)
		high	0.11(1.56)	*FPV*	0.01(1.45)			*high*	1.16(1.25)	*FPV*	1.22(1.04)
				TPV	0.15(1.55)					*TPV*	1.18(1.31)
				BEW	0.16(1.71)					*BEW*	1.07(1.41)
Aut	-0.53(1.51)	*low*	-0.35(1.54)	*FPV*	-0.24(1.63)	**Mea**	0.25(1.31)	*low*	0.30(1.28)	*FPV*	0.31(1.27)
				TPV	-0.48(1.47)					*TPV*	0.29(1.33)
				BEV	-0.33(1.55)					*BEV*	0.29(1.28)
		high	-0.70(1.46)	*FPV*	-0.57(1.52)			*high*	0.20(1.34)	*FPV*	0.33(1.24)
				TPV	-0.75(1.47)					*TPV*	0.01(1.41)
				BEW	-0.79(1.41)					*BEW*	0.25(1.37)

Table 4.7: Shows the scores given by participant for each *Player Experience Inventory* Abeele et al., 2020 subscale. Answers were given on a 7-point Likert scale from -3 (strongly disagree) to +3 (strongly agree). The table provides further details by breaking down the given scores for each level of LATENCY and PERSPECTIVE. Overall participants rated the gaming session with 2.20 points (*SD* = 0.74 points in *Ease of Control* (EoC), -0.46 points (*SD* = 1.37 points) in *Progress Feedback* (PrF), 0.80 points (*SD* = 1.60 points) in *Audiovisual Appeal* (AvA), 2.33 points (*SD* = 0.84 points) in *Clarity of Goals* (CoG), 1.58 points (*SD* = 1.07 points) in *Challenge* (Cha), 0.92 points (*SD* = 1.18 points) in *Mastery* (Mas), 0.28 points (*SD* = 1.52 points) in *Curiosity* (Cur), 1.35 points (*SD* = 1.19 points) in *Immersion* (Imm), -0.53 points (*SD* = 1.51 points) in *Autonomy* (Aut), and 0.25 points (*SD* = 1.31 points) in *Meaning* (Mea).

levels of LATENCY. When playing with *low* LATENCY compared to playing with *high*

Bayes Factor BF_{10}			Interpretation
	$>$	100	Extreme evidence for H_1
30	-	100	Very strong evidence for H_1
10	-	30	Strong evidence for H_1
3	-	10	Moderate evidence for H_1
1	-	3	Anecdotal evidence for H_1
	1		No evidence
1/3	-	1	Anecdotal evidence for H_0
1/10	-	1/3	Moderate evidence for H_0
1/30	-	1/10	Strong evidence for H_0
1/100	-	1/30	Very strong evidence for H_0
	$<$	1/100	Extreme evidence for H_0

Table 4.8: Evidence categories for Bayes factors. Adjusted for BF_{10}, H_0, and H_1 from Andraszewicz et al. (2015).

LATENCY participants were significantly faster (*low*: 15.817 s (*SD* = 12.157 s), *high*: 24.541 s (*SD* = 24.205 s)) and more accurate (*low*: 0.729 (*SD* = 0.128), *high*: 0.561 (*SD* = 0.125)).

ART-ANOVA also found significant effects of LATENCY on Ease of Control (F(1,35) = 25.555, $p < 0.001$, $\eta_p^2 = 0.42$), Progress Feedback (F(1,35) = 11.534, $p = 0.0017$, η_p^2 =

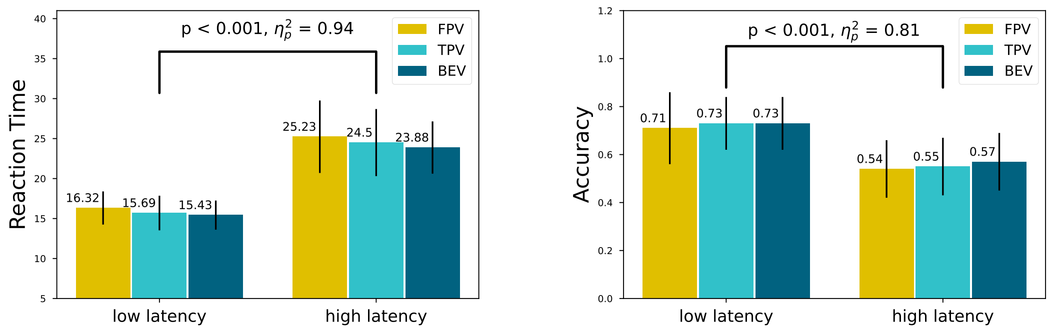


Figure 4.8: Depicts mean *Reaction Time* (left) and *Accuracy* (right) values grouped by LATENCY. Both subfigures provide p-values and effect size for the within LATENCY comparison. Error bars depict the standard error. Participants shoot targets significantly faster (*low*: 15.817 s (*SD* = 12.157 s), *high*: 24.541 s (*SD* = 24.205 s)) and with a significantly higher precision (*low*: 0.729 (*SD* = 0.128), *high*: 0.561 (*SD* = 0.125)) when playing with the *low* level of LATENCY.

0.24), Challenge (F(1,35) = 24.328, $p < 0.001$, $\eta_p^2 = 0.41$), Mastery (F(1,35) = 49.639, $p < 0.001$, $\eta_p^2 = 0.58$), Curiosity (F(1,35) = 5.904, $p = 0.0203$, $\eta_p^2 = 0.14$), Immersion (F(1,35) = 6.574, $p = 0.0147$, $\eta_p^2 = 0.15$), Autonomy (F(1,35) = 5.973, $p = 0.0197$, $\eta_p^2 = 0.14$), Enjoyment (F(1,35) = 30.143, $p < 0.001$, $\eta_p^2 = 0.46$), and no significant effect on Audiovisual Appeal (F(1,35) = 2.667, $p = 0.1113$, $\eta_p^2 = 0.07$), Clarity of Goals (F(1,35) = 0.950, $p = 0.3362$, $\eta_p^2 = 0.02$) or Meaning (F(1,35) = 2.071, $p = 0.1589$, $\eta_p^2 = 0.05$). Participants had a higher feeling of control and progress feedback, rated the challenge of the game as more appropriate, were stronger immersed, derived a stronger feeling of mastery, curiosity, and autonomy, and overall had more fun when playing with *low* LATENCY than playing with *high* LATENCY. Figure 4.9 and Figure 4.10 depict all significant differences grouped by level of LATENCY.

Perspective

ART-ANOVA found no significant effect of PERSPECTIVE on *Reaction Time* (F(2,70) = 1.881, $p = 0.1600$, $\eta_p^2 = 0.05$) or *Accuracy* (F(2,70) = 0.629, $p = 0.5359$, $\eta_p^2 = 0.01$). Furthermore, ART-ANOVA also found no significant effect on Ease of Control (F(2,70) = 0.228, $p = 0.7959$, $\eta_p^2 < 0.01$), Progress Feedback (F(2,70) = 0.084, $p = 0.9194$, $\eta_p^2 < 0.01$), Audiovisual Appeal (F(2,70) = 0.095, $p = 0.9089$, $\eta_p^2 < 0.01$), Clarity of Goals (F(2,70) = 0.037, $p = 0.9633$, $\eta_p^2 < 0.01$), Challenge (F(2,70) = 0.718, $p = 0.4908$, $\eta_p^2 = 0.02$), Mastery (F(2,70) = 0.172, $p = 0.8419$, $\eta_p^2 < 0.01$), Curiosity (F(2,70) = 1.230, $p = 0.2982$, $\eta_p^2 = 0.03$), Immersion (F(2,70) = 1.836, $p = 0.1670$, $\eta_p^2 = 0.04$), Autonomy (F(2,70) = 1.101, $p = 0.3381$, $\eta_p^2 = 0.03$), Meaning (F(2,70) = 1.300, $p = 0.2788$, $\eta_p^2 = 0.03$), and Enjoyment (F(2,70) = 0.463, $p = 0.6310$, $\eta_p^2 = 0.01$).

Latency x Perspective

ART-ANOVA found no significant interaction effect of LATENCY X PERSPECTIVE on *Reaction Time* (F(2,70) = 0.105, $p = 0.9003$, $\eta_p^2 < 0.01$) or *Accuracy* (F(2,70) = 0.103, $p = 0.9017$, $\eta_p^2 < 0.01$). Furthermore, ART-ANOVA also found no significant effect on Ease of Control (F(2,70) = 0.162, $p = 0.8503$, $\eta_p^2 < 0.01$), Progress Feedback (F(2,70) = 0.558, $p = 0.5748$, $\eta_p^2 = 0.01$), Audiovisual Appeal (F(2,70) = 0.589, $p = 0.5573$, $\eta_p^2 = 0.01$), Clarity of Goals (F(2,70) = 1.582, $p = 0.2128$, $\eta_p^2 = 0.04$), Challenge (F(2,70) = 0.020, $p = 0.980$, $\eta_p^2 < 0.01$), Mastery (F(2,70) = 1.538, $p = 0.2218$, $\eta_p^2 = 0.04$), Curiosity

(F(2,70) = 0.078, p = 0.9245, $\eta_p^2 < 0.01$), Immersion (F(2,70) = 1.570, p = 0.2151, η_p^2 = 0.04), Autonomy (F(2,70) = 0.3365, p = 0.7153, $\eta_p^2 < 0.01$), Meaning (F(2,70) = 0.6705, p = 0.5146, η_p^2 = 0.01), and Enjoyment (F(2,70) = 0.648, p = 0.5259, η_p^2 = 0.01).

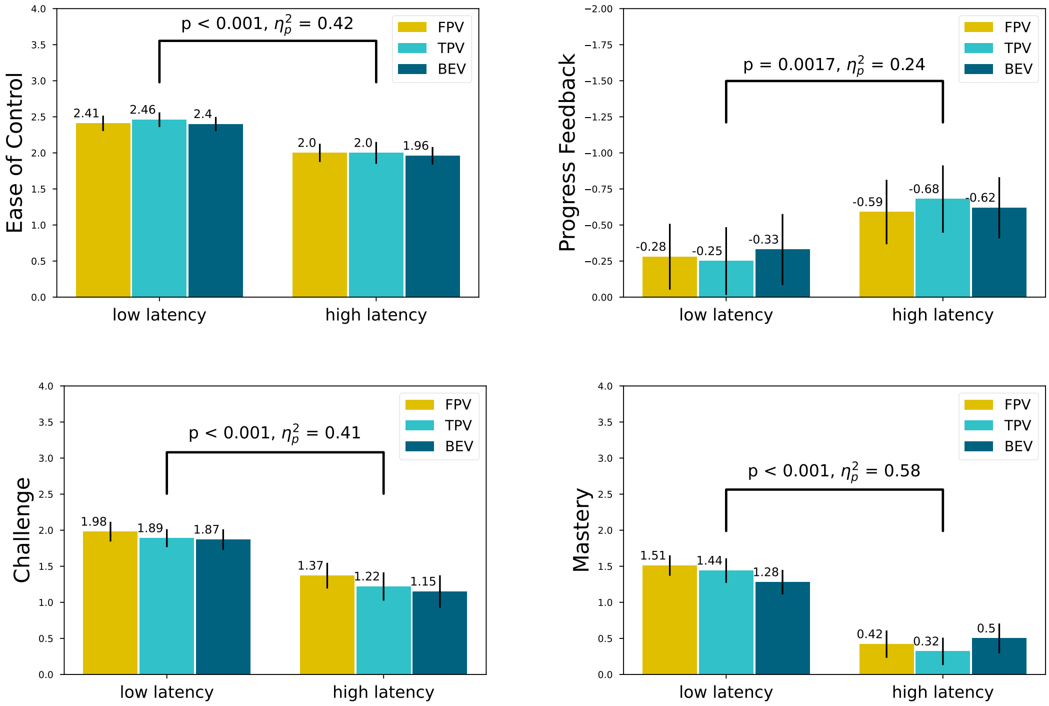


Figure 4.9: Depicts mean values of the subscale Ease of Control, Progress Feedback (inverted Y-scale), Challenge, and Mastery of the Player Experience Inventory (Abeele et al., 2020) grouped by LATENCY. The subfigures also provide p-values and effect sizes for the within LATENCY comparison. Error bars show the standard error. On average, participants had a higher feeling of control and a stronger sense of how they were doing in the game. Additionally, they derived a greater sense of mastery when playing the game with *low* LATENCY. Furthermore, participants felt that the game was significantly less of an appropriate challenge when playing with *high* LATENCY.

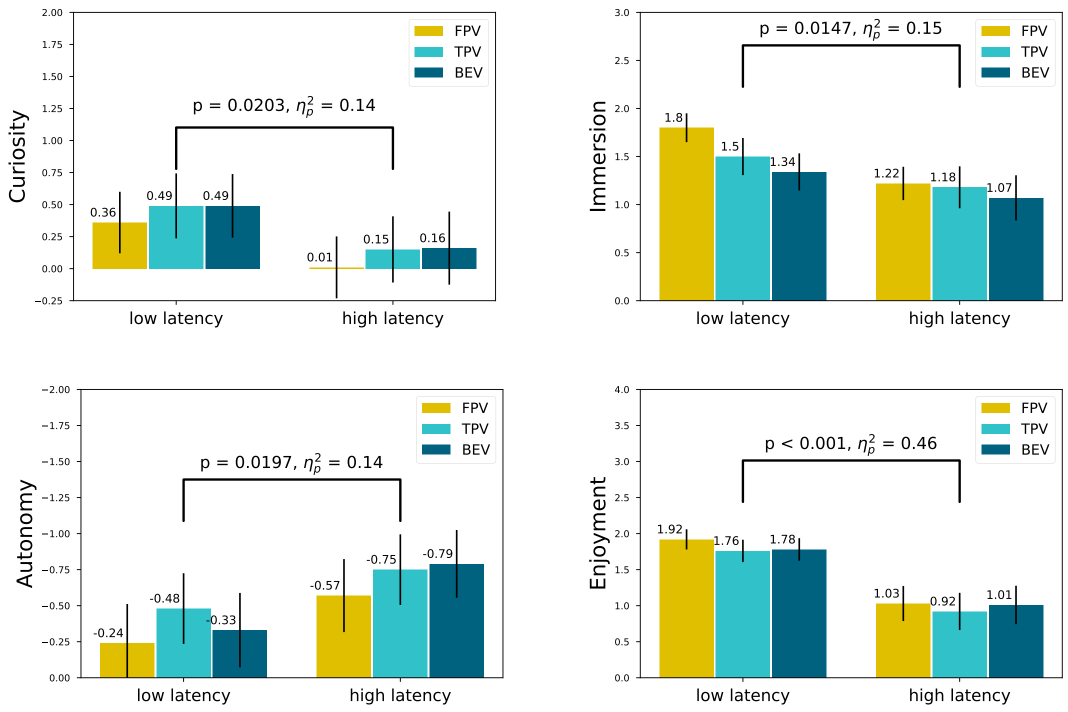


Figure 4.10: Depicts mean values of the subscale Curiosity, Immersion, and Autonomy (negative Y-scale) of the Player Experience Inventory (Abeele et al., 2020) and the mean values of the Enjoyment dimension grouped by LATENCY. The subfigures also provide p-values and effect sizes for the within LATENCY comparison. Error bars show the standard error. On average, participants were more curious about the game, were stronger immersed in the game, had a higher feeling of autonomy, and had a more joyful experience when playing with *low* LATENCY.

4.3.4.2 Bayesian Analysis

Previous inferential tests consistently did not reveal any significant effects of PERSPECTIVE or the interaction between LATENCY and PERSPECTIVE on either the performance or experience measures. Thus, we used a Bayesian two-way RM-ANOVA to further asses the influence of PERSPECTIVE.

Latency

We found extreme evidence in support of a model that postulates that LATENCY has a true effect (acception of H_1) on *Reaction Time* ($BF_{10} > 100$, *error* $\pm$ 0.55 %, $R^2 = 0.699$) and a true effect on *Accuracy* ($BF_{10} > 100$, *error* $\pm$ 0.55 %, $R^2 = 0.575$).

Investigating the effects of LATENCY on player experience, we found evidence for a model that accepts H_1 and postulates a true effect of LATENCY on Ease of Control ($BF_{10} > 100$, *error* $\pm$ 0.54 %, $R^2 = 0.419$), Challenge ($BF_{10} > 100$, *error* $\pm$ 1.09 %, $R^2 = 0.622$), Mastery ($BF_{10} > 100$, *error* $\pm$ 0.48 %, $R^2 = 0.534$), Immersion ($BF_{10} = 96.889$, *error* $\pm$ 0.46 %, $R^2 = 0.595$), Enjoyment ($BF_{10} > 100$, *error* $\pm$ 0.52 %, $R^2 = 0.680$), and Progress Feedback ($BF_{10} > 100$, *error* $\pm$ 0.43 %, $R^2 = 0.798$). Furthermore, we found strong evidence that LATENCY has a true effect on Autonomy ($BF_{10} = 20.935$, *error* $\pm$ 0.4 %, $R^2 = 0.721$), on Clarity of Goals ($BF_{10} = 19.462$, *error* $\pm$ 0.68 %, $R^2 = 0.852$), and on Curiosity ($BF_{10} = 19.086$, *error* $\pm$0.39 %, $R^2 = 0.757$). Lastly, we found anecdotal evidence in support of no effect of LATENCY (acception of H_0) on Audiovisual Appeal ($BF_{10} = 0.520$, *error* $\pm$ 1.04 %, $R^2 = 0.514$) and Meaning ($BF_{10} = 0.224$, *error* $\pm$ 0.59 %, $R^2 = 0.672$).

Perspective

We found evidence in support of a model that postulates no true effect (accepting H_0) of PERSPECTIVE on *Reaction Time* ($BF_{10} = 0.087$, *error* $\pm$ 0.58 %, $R^2 = 0.699$) and no true effect on *Accuracy* ($BF_{10} = 0.081$, *error* $\pm$ 0.28 %, $R^2 = 0.575$).

Investigate the effects of PERSPECTIVE on player experience, we found up to very strong evidence for a model that accepts H_0 and postulates no effect of PERSPECTIVE on Ease of Control ($BF_{10} = 0.052$, *error* $\pm$ 0.98 %, $R^2 = 0.419$), Challenge ($BF_{10} = 0.100$, *error* $\pm$ 0.46 %, $R^2 = 0.622$), Mastery ($BF_{10} = 0.055$, *error* $\pm$ 0.38 %, $R^2 = 0.534$), Enjoyment ($BF_{10} = 0.069$, *error* $\pm$ 0.34 %, $R^2 = 0.680$), Progress Feedback ($BF_{10} = 0.051$, *error* $\pm$ 0.46 %, $R^2 = 0.798$), Audiovisual Appeal ($BF_{10} = 0.068$, *error* $\pm$ 0.38 %, $R^2 = 0.514$), *Clarify of Goals* ($BF_{10} = 0.056$, *error* $\pm$ 0.29 %, $R^2 = 0.852$), and Curiosity ($BF_{10} = 0.093$, *error* $\pm$ 0.68 %, $R^2 = 0.757$). Furthermore, we found moderate evidence in favor of H_0, for Autonomy ($BF_{10} = 0.151$, *error* $\pm$ 0.32 %, $R^2 = 0.721$) and Meaning ($BF_{10} = 0.109$, *error* $\pm$ 0.42 %, $R^2 = 0.672$), and anecdotal evidence for no effect on Immersion ($BF_{10} = 0.557$, *error* $\pm$ 0.36 %, $R^2 = 0.595$).

Latency x Perspective

Investigating the interaction between LATENCY and PERSPECTIVE, we found strong evidence in support of a model that postulates that PERSPECTIVE X LATENCY has no true effect (accepting H_0) on *Reaction Time* (BF_{01} = 0.089, *error* ± 1.22 %, R^2 = 0.699) and no true effect on *Accuracy* (BF_{10} = 0.089, *error* ± 2.37 %, R^2 = 0.575).

Investigating the effects of LATENCY X PLAYER on player experience, we found strong evidence for a model that accepts H_0 and postulates no interaction effect of LATENCY X PERSPECTIVE on Ease of Control (BF_{10} = 0.085, *error* ± 4.2 %, R^2 = 0.419), Audiovisual Appeal (BF_{10} = 0.092, *error* ± 2.29 %, R^2 = 0.514), Enjoyment (BF_{10} = 0.093, *error* ± 1.17 %, R^2 = 0.680), Curiosity (BF_{10} = 0.085, *error* ± 1.62 %, R^2 = 0.757), and Challenge (BF_{10} = 0.100, *error* ± 2.79 %, R^2 = 0.622). Furthermore, we found moderate evidence of no true interaction effect on Progress Feedback (BF_{10} = 0.104, *error* ± 1.28 %, R^2 = 0.798), Autonomy (BF_{10} = 0.106, *error* ± 1.69 %, R^2 = 0.721), Meaning (BF_{10} = 0.167, *error* ± 1.93 %, R^2 = 0.672), Immersion (BF_{10} = 0.187, *error* ± 1.88 %, R^2 = 0.595), Mastery (BF_{10} = 0.196, *error* ± 1.59 %, R^2 = 0.534) and *Clarity of Rules* (BF_{10} = 0.197, *error* ± 1.78 %, R^2 = 0.852).

4.3.5 Discussion

Inferential analysis shows that latency in our game significantly reduces PP and GX. Using a Bayesian analysis, we strengthen our findings. We found up-to extreme evidence for a model that implies a true effect of latency on PP (all BF_{10} > 100) and a true effect of latency on most subscales of the PXI and the enjoyment of the game (all BF_{10} > 19.086). However, we also found anecdotal evidence that latency does not affect the audiovisual appeal of the game (BF_{10} = 0.520) and the level of meaning derived from playing it (BF_{10} = 0.224). Furthermore, via inferential analysis, we found that the in-game perspective does not significantly alter performance or experience. Using a Bayesian approach, we found that the in-game perspective may not manipulate gaming experience (0.051 < BF_{10} < 0.557). We found a similar effect of the interaction between latency and the in-game perspective. Our inferential analysis showed no significant interaction. Additionally, we found up-to strong evidence (0.085 < BF_{10} < 0.197) that supports a model with no true effect of the interaction on gaming experience.

In the following, we first discuss our results regarding PP. We then shed light on how latency and in-game perspective alter the functional components (ease of control, progress feedback, audiovisual appeal, clarity of goals, and challenge) of the PXI. Subsequently, we explore the implications of our findings regarding the psycho-social consequences (mastery, curiosity, immersion, autonomy, and meaning) of the PXI and the enjoyment of the game itself (Abeele et al., 2020). We continue by showcasing the implications of our findings for latency and video game research. Lastly, we conclude this section with a discussion about our work's limitations and possible future work.

4.3.5.1 Player Performance

Our work shows that players were less accurate and required more time to shoot the in-game targets when playing with high latency. The temporal asynchronicity between player input and the latency-induced game reaction led to a performance decay. These findings are in line with previous work, which showed that latency negatively affects game performance (Claypool & Claypool, 2006; Beigbeder et al., 2004; Liu et al., 2021d) and accuracy in particular (Liu et al., 2021b; Liu et al., 2021c). However, while a large body of work investigates the effects of latency in different games, the in-game perspective has been largely neglected. Previous work typically investigated the effects of latency in a single game (Beigbeder et al., 2004), and thus a single perspective, or only related the effects of latency in different video games to each other (Claypool & Claypool, 2006). One notable exception is the work by Schmidt et al. (2017). Similarly to our findings, the authors found that the in-game perspective in one game did not change the effects of latency. However, given that the authors also used three games from different genres, their work cannot produce a generalized conclusion on the interaction of latency and in-game perspective. Our work, in contrast, extends upon previous work and demonstrates that the in-game perspective in a single game does not alter the players' performance potential or the actual in-game performance.

4.3.5.2 Functional Consequences

We found that latency significantly and negatively impacts three of the five constructs targeted at the functional components of the PXI (ease of control, progress feedback, and challenge). These constructs generally describe how the immediate gaming experience is altered as a direct result of game elements and design choices. Hence, our work shows that introducing latency to the game directly influences how players experience the game on a functional level. This is generally in line with previous work, which shows that latency leads to a decreased gaming experience. However, previous work often treats gaming experience as a single dimension (Liu et al., 2021b; Liu et al., 2021c) or used instruments that are not well-suited for a comprehensive comparison with the PXI (Sabet et al., 2019a). Hence, our work builds on previous findings and extends it by an assessment of how latency manipulates GX using the validated PXI-scale.

Our results show that players rated the game with significantly higher ease of control and a higher level of progress feedback when playing with low latency. Conversely, this means players had a significantly harder time performing targeted in-game action and judging if the performed action was good or bad when playing with high latency (Abeele et al., 2020). Thus, playing with high latency leads to a less intuitive input-out paradigm. Latency increases the loop-throughput time in Card's (Card et al., 1983) MHP and leads to increased asynchronicity between in- and output. Ultimately, this increased asynchronicity in the interaction decreases the ease of control. Furthermore, since the game's response, for example, the visual feedback when a target was successfully hit, was delayed, players did not know how well they were performing in the game, decreasing the level of perceived progress feedback.

Our analysis also reveals that players rated the game's challenge significantly less appropriate when playing with a higher level of latency. This is in line with our previous work, which also shows that latency influences the perception of a game's challenge bidirectionally. In section 3.2, for example, we showed that switching between latency levels leads to a reduced feeling of challenge compared to a constant level of latency. Previous work, however, does not provide conclusive evidence of how latency alters the perception of challenge. While most works suggest that latency increases the challenge by decreasing the interaction's responsiveness (Long & Gutwin, 2018), we indicated in section 3.2 that it reduces challenge as a byproduct of reducing the game's flow. We argue that in this investigation it is a combination of both. While it is evident that the game becomes harder by delaying action feedback (such as target hit registration), latency also alters how the players perceive the game's challenge. When playing with high latency, the immediate response is missing. Players would not receive confirmation if their actions were successful right after the action. This possibly leads to a feeling of unfairness. Subsequently, this feeling of unfairness and the increased difficulty caused by input-output asynchronicity leads to a reduced level of challenge appropriateness. Previous work arguing that playing with a higher latency puts players at unfair disadvantages over other players playing with lower latency or the game itself indicates that this is a reasonable conclusion (Ishibashi & Kaneoka, 2005; Brun et al., 2006; Zander et al., 2005).

Lastly, using an inferential analysis, we found no significant influence of the in-game perspective on the functional components of the PXI. Additionally, a Bayesian ANOVA, strengthens the inferential results and suggests no true effect of the in-game perspective on functional components. Changing the in-game perspective did not immediately change the gaming experience on a functional level. This is in strong contrast to previous work, which showed that how the player is situated in the game world, i.e., the players' in-game perspective, dictates what level of latency is acceptable for the gaming session (Claypool & Claypool, 2006). Based on our data, we propose that latency thresholds are not definable by categorizing games using their in-game perspective. Different aspects, such as the game's pacing (Schmidt et al., 2017), the importance of player actions, or the required spatial accuracy when carrying out in-game actions (Sabet et al., 2020a) alter how latency affects the functional components of the gaming session. Our results suggest that the in-game perspective as a standalone variable does not affect latency nor interacts with it. Thus, previous findings that showed an effect of in-game perspective more likely originated from testing different games and game mechanics.

4.3.5.3 Psycho-Social Consequences and Enjoyment

Our work also reveals a significant impact of latency on the psycho-social consequences of the gaming experience as well as on the overall enjoyment of the gaming session. The psycho-social consequences of the PXI describe second-order emotions experienced by the players as a result of playing the game. We found that latency decreased four of the five constructs in the psycho-social consequences dimensions (mastery, curiosity, immersion, and autonomy) and the game's overall enjoyment.

These results highlight the importance of reducing latency in video games as it directly affects the players' emotional experience. Latency significantly impacts the players' feelings of control, excitement, and engagement while playing the game. A lower latency results in a more seamless and enjoyable gaming session, allowing players to fully immerse themselves in the virtual world and feel a sense of mastery and autonomy. On the other hand, a high latency negatively impacts the players' psycho-social consequences and overall enjoyment of the game. This is in line with previous work, which showed that increasing latency decreases players' perceived effectiveness, their experienced fun (Liu et al., 2022), and increases their frustration (Long & Gutwin, 2018). Furthermore, we found no significant effects on the psycho-social consequences of the GX induced by the in-game perspective or by the interaction of latency and in-game perspective. Again, a Bayesian approach revealed evidence for a model that supports no true effect of the in-game perspective. Thus, our Bayesian analysis supports that altering the in-game perspective of the game does not alter the players' psycho-social consequences induced by playing the game.

4.3.5.4 Implications of our Findings

This work's findings have implications for researchers and game developers. The implications of our work are twofold: First, we show that the used perspective in our game does not alter the player's performance or gaming experience. Game developers can leverage this insight to design games that prioritize gaming experience across different in-game perspectives. Since the perspective does not significantly affect latency's impact, developers can allocate resources and potentially implement strategies to reduce latency and enhance overall gameplay, regardless of the chosen perspective. This knowledge allows for more flexible and inclusive game design decisions, accommodating a wider range of player preferences without compromising performance. Furthermore, the knowledge that in-game perspective does not significantly impact latency effects simplifies the development process. Developers can allocate their resources more efficiently without the need for perspective-specific latency optimizations. This streamlined approach can lead to reduced development time and costs, enabling teams to focus on other critical aspects of game design and performance improvement. Second, our work also replicates and builds upon previous work investigating the effects of latency on PP and gaming experience (Long & Gutwin, 2018; Liu et al., 2021d; Claypool & Finkel, 2014). However, our work goes one step further and shows that latency always has negative consequences and is not imperatively dependent on the player's perspective of the game world. Furthermore, our work suggests that findings regarding latency from one in-game perspective are transferable to other in-game perspectives. This reduces the need to validate the effects of latency found in one study using one particular in-game perspective in additional studies with different perspectives. These results highlight the potential for researchers to shift their focus from the in-game perspective as a primary factor affecting latency's impact. Instead, they can explore other aspects, such as network conditions, game mechanics, player characteristics, or environmental factors that may have a more fundamental influence. By broadening the scope of investigation, researchers can gain a deeper understanding of the complex dynamics of latency and identify additional mitigating factors.

4.3.5.5 Limitations and Future Work

While our work demonstrates that the in-game perspective does not alter PP and GX in our game and that latency and in-game perspectives do not interact, our work still has limitations. In the following, we discuss these limitations and present new avenues to investigate the interaction between latency and in-game perspective.

One limitation of our work is the used game. We developed our own game to translate fundamental game mechanics to different in-game perspectives without changing the core gameplay. Our results, predominantly the Bayesian analysis, supports the assumption that we achieved this without changing the GX. However, our game neither has an elaborate story nor requires a comprehensive input strategy. Hence, one needs to be cautious about generalizing to the ever-increasing landscape of video games with its countless types of games and genres. Nevertheless, our work provides the next step toward a better understanding of latency and the factors influencing its impact. Future work should build on our work and try to further parameterize different components that influence the effects of latency in video games.

Furthermore, it is crucial to acknowledge the limitation imposed by our study's participant sample and sample size. First, our sample primarily consisted of university students. While focusing on university students allowed for an accessible and homogenous recruitment and data collection, it also limits our work's generalizability. Second, the sample size of 36 participants presents inherent limitations in terms of statistical power and precision. Despite previous work showing that a relatively small number of participants is sufficient to detect the effects of latency (MacKenzie and Ware: n = 8 (1993), Long and Gutwin: n = 18 and n = 20 (2018), Liu et al.: n = 25 (2021c)), the limited number of tested participants in our study may hinder the ability to detect subtle and small effects. Hence, future work should investigate the interaction between latency and the in-game perspective with a larger sample. Additionally, future research endeavors should aim to diversify the participants pool to include individuals from different age groups, educational backgrounds, and cultural contexts to strengthen the generalizability and robustness of our findings.

Another limitation of our work relates to our findings regarding the psycho-social consequences induced by the gaming session. While our data show that in-game perspective does not alter the psycho-social consequences of the players induced by playing our game, this finding does not generalize to the broader field of digital media, such as story-driven games or videos. Cinematography, for example, shows that different camera angles, light setups, and scenery evoke different emotions in viewers or players. To further investigate the psycho-social consequences manipulated by the in-game perspective, future work should craft a video game that focuses on a compelling narrative targeted at creating highly engaging gaming sessions. Similarly to our work, this custom game could be translated to different in-game perspectives and used in a study to inform about the effects of the perspective.

4.3.6 Conclusion

This section presented a study with 36 participants playing a custom shooting game with two levels of latency and three different in-game perspectives. We found that latency significantly decreases PP and gaming experience. Further, we did not find a significant effect of the in-game perspective or the interaction between latency and the in-game perspective on our measures. We continue by analyzing the gathered data using a Bayesian approach, which supports the assumption that neither the in-game perspective nor the interaction with latency influences GX and performance. We discuss and conclude, thus, that the in-game perspective does not necessarily dictate the effects of latency in video games. Our work is a crucial step towards a better understanding of latency, how video game players perceive it, and how other factors manipulate its effects in video games. Finally, we conclude that the in-game perspective, as a standalone characteristic, is not a fruitful metric to define a game's latency sensitivity. Previous approaches to categorizing latency sensitivity by video game genre or in-game perspective may neglect in-game factors that fundamentally alter the effects of latency. Ultimately, our findings allow future latency research to exclude the in-game perspective and shift its focus to exploring other game characteristics that may alter the effects of latency in video games. By identifying other factors that influence latency sensitivity in video games, researchers can develop a more nuanced understanding of how game design affects player experience, which can help game developers design games that are more enjoyable and engaging for players.

4.4 Understanding Latency Expectancy (Study VI)

In the previous part of this chapter, we investigated the effects of auditory latency and how the visual in-game perspective alters the effects of latency. We found that auditory latency affects experienced video game players (cf. Section 4.1 and 4.2), but also found that the in-game perspective does not modulate its effects. Latency, in our study (cf. Section 4.3) always had negative effects, independent of the used in-game perspective. So far, we have clarified two crucial aspects of latency in video games. However, there are still open questions we need to answer to be able to use the knowledge we have gained to develop novel latency compensation techniques. The most pressing question is, how the current level of latency should be communicated to the player, for example, via an in-game overlay. Current games, such as Fortnite (Epic Games, 2020), use small numerical indicators in the game UI to let players know their latency. However, the mere display of latency can potentially trigger an expectation in the player (Kosch et al., 2022). It can change the player's mental model of the player-game interaction loop. Thus, it is possible that the mere display of latency leads to performance and experience degradation. Currently, it is unclear if the expectation of latency alters PP and GX (RQ5). To start unraveling this question, we can refer to previous work from cognitive science, psychology, and medicine investigating the so-called place effect.

In medicine, a placebo is a sham treatment or substance with no known therapeutic value or active ingredients respectively (Beecher, 1955; Arnstein et al., 2011; Kosch et al., 2022), which has a positive effect on a patient. For example, placebos can relieve pain (Colloca, 2019; Montgomery & Kirsch, 1996), support the treatment of physical conditions (Kaptchuk, 1998), and even aid the therapy of psychiatric disorders such as anxiety disorder (Holmes et al., 2016). The primary factor underlying the placebo effect is one's expectation of improvement. These expectations are formed by previous experience, contextual cues, and biological characteristics that ultimately determine the response to a placebo (Holmes et al., 2016; Montgomery & Kirsch, 1996). On the other hand, the complement to a placebo is called nocebo, in which a sham treatment or procedure leads to a harmful response (e.g., blocking therapeutic effects of clinical drugs (Colloca & Barsky, 2020)).

Users' interactions with digital environments are also fundamentally shaped by their expectations about it. Thus, the interplay between humans and computer-based systems is potentially subject to the pla- and nocebo effect. Previous work showed, for example, that telling users that they interact with an intelligently adapting user interface decreases their task completion rate compared to users testing the same system without said suggestion. Crucially, both user groups tested the same system without an intelligently adapting user interface (Kosch et al., 2022). In a subsequent study, the authors demonstrated that the placebo effect, induced by suggesting an adaptive interface, sustained even after the initial interaction (Kosch et al., 2022). Other work investigated how the prior expectation about a video game's rating influences the actual users' ratings of the game (Michalco et al., 2015). Users with a higher primed expectancy about the game's rating rated the game significantly better compared to users with no or a lower primed expectancy. Similar research in this line of work showed that players of video games rated a tested game with higher GX and achieved higher gaming performance if they believed they played in cooperation with an intelligent agent (Denisova & Cairns, 2015b). A considerable amount of evidence shows that expectancy-based effects fundamentally form the experience and performance in digital environments. Nevertheless, one example of a concept crucially relevant and potentially susceptible to expectancy-based effects but yet not researched in that manner, is latency.

While a large body of work investigated the effects of latency in video games, less is known about the potential effects of latency perception and expectation. Investigating expectancy-based effects induced by suggested latency (phantom latency) is crucial for a range of reasons. First, researchers investigating latency may need to account for the effect of phantom latency when designing studies and briefing participants. Second, developers must know how to communicate latency to users without interfering with their performance and experience. Third, players must be aware of potential effects induced by the mere perception and expectation of latency. It is currently unknown how phantom latency and potentially induced expectancy-based effects manipulate GX and performance in video games.

Previous work shows that sham treatments or procedures known as a placebo, which are based on the recipient's expectation (Beecher, 1955; Arnstein et al., 2011; Kaptchuk, 1998), aid in treating physical (Colloca, 2019; Montgomery & Kirsch, 1996; Kaptchuk, 1998) and psychological conditions (Holmes et al., 2016). On the other hand, expectations may also hinder a successful treatment, known as the nocebo effect (Colloca & Barsky, 2020). Previous work also shows that users of digital environments are susceptible to effects induced by the users' expectations of it (Kosch et al., 2022; Michalco et al., 2015; Denisova & Cairns, 2015b). One factor fundamentally influencing the interaction in said environments is latency (MacKenzie & Ware, 1993; Annett et al., 2014; Jota et al., 2013). Latency particularly affects players of video games (Eg et al., 2018; Beigbeder et al., 2004; Claypool & Claypool, 2006; Liu et al., 2021d; Liu et al., 2021c). However, while a large body of work investigated the effects of latency in video games, how the expectation of latency alters PP and GX (RQ5) is unclear.

This section is based on the following article:

Halbhuber, D., Schlenczek, M., Bogon, J., & Henze, N. (2022d). "Better Be Quiet about It! The Effects of Phantom Latency on Experienced First-Person Shooter Players." In: *Proceedings of the 21st International Conference on Mobile and Ubiquitous Multimedia*. MUM '22. Lisbon, Portugal: Association for Computing Machinery, pp. 172–181. ISBN: 9781450398206. DOI: 10.1145/3568444.3568448.

4.4.1 Background and Research Rationale

Our work closes this gap by providing in-depth insights into how latency expectation affects GX and performance in a fast-paced FPS video game. To achieve this, we developed a custom latency overlay for CS:GO. Using our overlay, we conducted a study in which we presented experienced CS:GO players four pretended levels of latency (30 ms, 60 ms, 90 ms, and 120 ms) while playing the game. We call this pretended latency phantom latency. Each participant played with all levels of phantom latency. Crucially, all rounds were played with an actual latency of 75 ms. Our analysis shows that phantom latency significantly alters GX and performance. Players were significantly less accurate, dealt less damage per hit, and had a significantly lower feeling of competence when playing with 120 ms of phantom latency compared to playing with 30 ms. Holistically considering all gathered data, we demonstrate that researchers investigating latency need to account for expectancy-based effects induced by a study or the researchers themselves. Furthermore, we show that the mere perception of latency influences players' performances and experiences. Video game players and developers should be aware of the bidirectional effects induced by phantom latency and its perception.

4.4.2 Method

To investigate how phantom latency affects PP and GX in video games, we conducted a study with high-skilled participants playing CS:GO. We used CS:GO since fast-paced video games have already been shown to be notably affected by the adverse effects of latency (Claypool & Claypool, 2006; Liu et al., 2021d; Liu et al., 2021c). Therefore, we modified the game with a self-developed overlay that can indicate any latency value.

4.4.2.1 Apparatus

We used CS:GO's Deathmatch game mode, in which all players compete against each other without forming teams. In line with previous work, all gaming rounds were played on Mirage (the most played map in CS:GO) using the AK-47 (the most used weapon in CS:GO) (Liu et al., 2021d; Liu et al., 2021c). We prevented players from switching or obtaining other weapons via the in-game console. Furthermore, to prevent confounding variables from playing against other human players, we conducted the study using CS:GO's built-in bots (hard difficulty).

We developed a custom overlay for CS:GO using Java to display phantom latency to the participants. Since latency in the wild is composed of numerous factors, it never is perfectly constant as we discussed in previous sections of this thesis (cf. Section 3.1 and 3.2). Thus, we added random variation to the displayed phantom latency. When using the overlay, the displayed phantom latency randomly varies within a range of 3 ms; i.e., if set to a latency of 30 ms, the overlay displays a phantom latency between 27 ms and 33 ms. The displayed value is updated every 400 ms within that range to establish a more natural latency behavior. Figure 4.11 depicts a screenshot of an unmodified CS:GO version (left) and a screenshot of CS:GO while using our developed latency overlay (right).

The game was executed at a fixed 144 frames per second on a stationary high-end workstation with an Intel i9-11900K, an Nvidia GeForce GTX 3080, 32 GB RAM, and an M.2 SSD. In addition, the workstation was connected to a HyperX Cloud II headset, a Corsair K100 keyboard, a Logitech G502 HERO gaming mouse, and an MSI Optix 27" monitor. We measured the latency of our system running CS:GO using the game's internal server architecture, which allows us to manipulate and control the latency of gaming sessions.

4.4.2.2 Study Design

We designed a within-subjects study and utilized the independent variable (IV) PHANTOM LATENCY which is factorized on four levels: (I) *30 ms*, (II) *60 ms*, (III) *90 ms* and, (IV) *120 ms*. The levels of PHANTOM LATENCY were designed in accordance with related work, which showed different latency thresholds before impacting GX and

Figure 4.11: Shows two screenshots from the first-person shooter game *Counter-Strike: Global Offensive*. The left side of the figure shows an unaltered screenshot. The right side shows a screenshot with our latency overlay enabled in comparison. In this screenshot 30 ms of latency are depicted in the upper right corner of the game using numerical values (green). The red box, the red arrow, and the "Phantom Latency" description were not shown in an actual gaming session.

performance (Claypool & Claypool, 2006; Claypool & Finkel, 2014; Beigbeder et al., 2004). Liu et al. (2021d), for example, investigated negative effects of latency in a range from 25 ms latency up to 125 ms latency. Further, we excluded a *0 ms* level for PHANTOM LATENCY, because latency in the wild is never zero. Each participant played with each level of PHANTOM LATENCY. Crucially, we controlled the true latency in all conditions to remain at 75 ms. The rationale to set the true latency to 75 ms was two-fold: (1) It was crucial to conceal the fact that latency over conditions did not change. Hence, we required a baseline that induced negative latency effects but was neither notably too low nor too high since all participants were highly familiar with CS:GO and how latency affects the game. Secondly, (2) we did not use one of the tested levels of phantom latency (30 ms, 60 ms, 90 ms, and 120 ms) as a baseline to prevent a match and possibly interaction between the displayed level of phantom latency and the true latency. All conditions were balanced using a Latin Square to prevent sequence effects.

We recorded a range of dependent variables to measure the participants' GX and performance. We used the 33-item GEQ (IJsselsteijn et al., 2013) with its seven subscales Sensory, Flow, Competence, Positive Affect, Negative Affect, Tension, and Challenge to quantitatively evaluate GX and coupled it with qualitative questions focused on the participants' experiences with latency in the past gaming round and the participants' perceptions of latency.

We operationalized PP in different variables: (1) *Score* - the overall amount of points achieved for hitting and eliminating adversary bots, (2) *Kills* - numbers of enemies eliminated, (3) *KD-Ratio* - the ratio of enemy kills and in-game deaths. A higher *KD-Ratio* indicates a more efficient and effective gaming session, and vice versa, a lower *KD-Ratio* correlates to worse performance. (4) *Number of Headshots* (*NoH*) - number of headshots dealt. A headshot in CS:GO deals maximum damage and instantly eliminates the enemy. However, they are hard to accomplish and get even harder the further the target is away and the more the target moves. Reliably hitting headshots is a crucial skill for every CS:GO, player. Lastly, we used (5) *Damage Per Hit* (*DpH*) as additional DV. Besides headshots, CS:GO also implements different hit zones on the avatar. For example, shooting an enemy in the foot does less damage than shooting an enemy in the chest. Performing effectively in CS:GO requires players to deal the highest amount of damage with the lowest possible hits.

4.4.2.3 Procedure and Task

Participants were greeted at the laboratory by the experimenter. After briefing them about the general procedure of the study, they gave informed consent. Participants were told that they would test a novel latency representation in CS:GO (our latency overlay). They were, however, blind to the study's exact purpose (testing the effects of phantom latency on GX and performance). After the introductory briefing, participants were asked to fill out a demographic questionnaire, answering questions about their age, gender, handedness, experience in FPS games, hours played in CS:GO, their CS:GO rank, what they consider to be high latency, and how they usually deal with high latency while gaming. Next, participants were led into a separate room where the gaming setup was situated. CS:GO was already executed in full-screen mode on the stationary workstation. The participants were seated in front of the workstation and started the gaming sessions with a 10-minute warm-up round in CS:GO's Deathmatch. A warm-up was conducted to allow the participants to familiarize themselves with the gaming setup and the Deathmatch mode. After the warm-up, participants had a two-minute break before the first gaming round with PHANTOM LATENCY started. Each round lasted 10 minutes. Participants were asked to answer the GEQ after each round. After answering the GEQ, participants were asked if they felt latency in the last round and, if so, how they felt it manifested in the game. Next, participants had another 2-minute break before starting the subsequent round.

4.4.2.4 Participants

Previous work showed that the effects of latency on GX and performance could reliably be detected with a comparatively small number of participants (Liu et al., 2021d; Liu et al., 2021c; MacKenzie & Ware, 1993; Long & Gutwin, 2018). In line with this work, we also recruited 24 participants (21 male, 3 female) through our institution's mailing list. The participants' age ranged from 18 years to 42 years, with an average age of 25.71 years (*SD* = 5.25 years). Twenty-two participants were right-handed, and the other participants were left-handed. Nevertheless, all participants operated the computer mouse using their right hand and the keyboard using their left hand. Since we investigate how one's expectation of latency influences the course of the gaming session, one must be at least familiar with the concept of latency in video games. Therefore, to take part in our study, participants had to have at least 100 hours of experience in CS:GO. Participants experience in CS:GO ranged from 100 hours to 2895 hours, with an average of 535.16 hours (*SD* = 556.68 hours). Besides mere playtime, participants were also screened for their rank in CS:GO. CS:GO has an ELO-based ranking system with 19 internal ranks - 19 being the lowest and 1 being the highest possible. Participants' ranks ranged between rank 1 and rank 12. On average, participants were ranked on rank 9.59 (*SD* = 4.52), which corresponds to a medium rank overall.

4.4.3 Results

Twenty-four participants played four rounds of CS:GO with different levels of phantom latency. Thus, we collected 96 responses to the post-experience questionnaire and 96 recorded performance measurements.

In the following, we present the descriptive and statistical analysis of the GX measures. Then, we continue by reporting the analysis of performance measures. Finally, we conclude by outlining the qualitative feedback received in the study.

4.4.3.1 Game Experience Questionnaire

Descriptive data showing the mean score and standard deviation for each subscale of the GEQ and each level of PHANTOM LATENCY is shown in Table 4.9. Answers were given on a 5-point Likert-item (0 minimum, 5 maximum).

For statistical analysis we used a one-way ANOVA (PHANTOM LATENCY: *30 ms*, *60 ms*, *90 ms*, and *120 ms*) as no prerequisite for ANOVA was violated (Shapiro-Wilk

	Game Experience Questionnaire Scores						
Phantom Latency	**TEN**	**COM**	**FLO**	**CHA**	**POS**	**NEG**	**SEN**
30 ms	0.54 ± 0.53	2.64 ± 0.59	3.48 ± 0.52	2.11 ± 0.81	2.54 ± 0.69	0.39 ± 0.49	1.25 ± 0.59
60 ms	0.66 ± 0.65	2.29 ± 0.46	3.51 ± 0.53	2.22 ± 0.82	2.33 ± 0.49	0.43 ± 0.54	1.02 ± 0.62
90 ms	1.04 ± 1.12	2.04 ± 0.46	3.41 ± 0.78	2.04 ± 0.75	2.14 ± 0.58	0.68 ± 0.87	1.00 ± 0.55
120 ms	1.39 ± 1.07	1.51 ± 0.69	3.25 ± 0.59	2.51 ± 0.87	1.73 ± 0.75	0.42 ± 0.67	0.83 ± 0.58

Table 4.9: Shows the mean scores and standard deviation of the subscales Tension (TEN), Competence (COM), Flow (FLO), Challenge (CHA), Positive Affect (POS), Negative Affect (NEG) and Sensory (SEN) of the Game Experience Questionnaire for each level of tested PHANTOM LATENCY.

test for all GEQ measures $p > 0.05$). ANOVA showed a significant effect of PHANTOM LATENCY on Tension (F(1,23) = 4.599, p = 0.004, η_p^2 = 0.131), on Competence (F(1,23) = 13.985, p < 0.001, η_p^2 = 0.313), and on Positive Affect (F(1,23) = 7.225, p < 0.001, η_p^2 = 0.191). We found no statistically significant effect of PHANTOM LATENCY on Flow (F(1,23) = 0.819, p = 0.487, η_p^2 = 0.026), on Challenge (F(1,23) = 1.633, p = 0.187, η_p^2 = 0.051), on Negative Affect (F(1,23) = 1.031, p = 0.383, η_p^2 = 0.033), and on Sensory (F(1,23) = 1.015, p = 0.390, η_p^2 = 0.032).

Next, we further investigated all significant results via Tukey tests. P-values and confidence intervals are corrected for multiple comparisons. Tukey's test showed significant differences between 30 ms and 120 ms (p_{Tukey} = 0.006, d_{Cohen} = -0.967, CI95=[-1.521, -0.186]) of PHANTOM LATENCY and between 60 ms and 120 ms (p_{Tukey} = 0.026, d_{Cohen} = -0.826, CI95=[-1.396, -0.729]) for the Tension subscale. For the Competence subscale, we found significant difference between 30 ms and 90 ms (p_{Tukey} = 0.007, d_{Cohen} = 0.957, CI95=[0.127, 1.081]), between 30 ms and 120 ms (p_{Tukey} = 0.001, d_{Cohen} = 0.815, CI95=[0.669, 1.623]), between 60 ms and 120 ms (p_{Tukey} < 0.001, d_{Cohen} = 1.254, CI95=[0.315, 1.268]), and, between 90 ms and 120 ms (p_{Tukey} = 0.019, d_{Cohen}=0.858, CI95=[0.065,1.018]) of PHANTOM LATENCY. For the Positive Affect subscale we found significant difference between 30 ms and 120 ms (p_{Tukey} < 0.001, d_{Cohen} = 1.254, CI95=[0.337, 1.288]) as well as between 60 ms and 120 ms (p_{Tukey} = 0.012, d_{Cohen} = 0.959, CI95=[0.128, 1.080]). Figure 4.12 shows the results of the post-hoc comparison for Tension, Competence, and Positive Affect. Participants were significantly more tense in 120 ms PHANTOM LATENCY condition than in the 30 ms and 60 ms condition. Furthermore, participants had the significantly lowest feeling of competence when playing with the highest level of PHANTOM LATENCY. Lastly, we

	Performance Measures				
Phantom Latency	**Score**	**Kills**	**KD-Ratio**	**NoH**	**DpH**
30 ms	552.04 ± 117.85	47.67 ± 10.41	2.72 ± 0.59	22.42 ± 7.48	21.84 ± 5.78
60 ms	521.83 ± 106.39	44.33 ± 10.34	2.17 ± 0.53	18.92 ± 6.01	20.81 ± 4.51
90 ms	497.79 ± 108.71	41.71 ± 10.09	1.91 ± 0.71	17.46 ± 6.81	19.31 ± 3.31
120 ms	469.71 ± 110.94	39.99 ± 10.04	1.74 ± 0.59	17.04 ± 5.78	18.98 ± 3.68

Table 4.10: Shows the mean and standard deviation of *Score*, *Kills*, *KD-Ratio*, *Number of Headshots* (NoH), and *Damage per Hit* (DpH) for each level of tested PHANTOM LATENCY.

also found that playing with 120 ms PHANTOM LATENCY led to a significantly lower Positive Affect associated with the gaming session compared to playing with 30 ms and 60 ms.

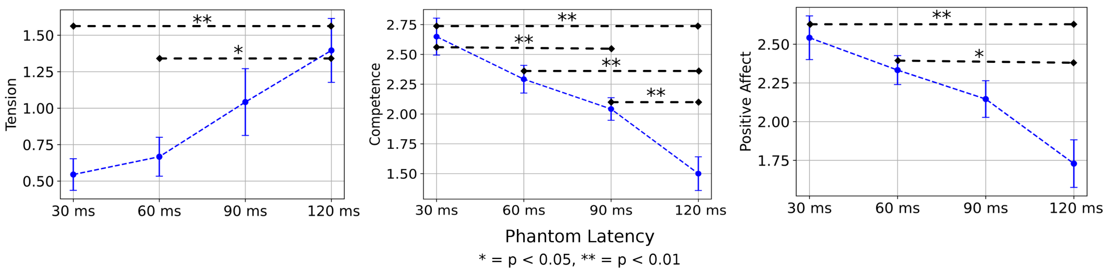


Figure 4.12: Scores given in the Tension, Competence, and Positive Affect subscale of the Game Experience Questionnaire (IJsselsteijn et al., 2013). Significant differences are highlighted via asterisks. Error bars show the standard error. Participants were significantly more tense in 120 ms PHANTOM LATENCY condition than in the 30 ms and 60 ms condition and had the lowest feeling of competence when playing with the highest level of PHANTOM LATENCY. We also found that playing with 120 ms PHANTOM LATENCY led to a significantly lower Positive Affect associated with gaming session compared to playing with 30 ms and 60 ms.

4.4.3.2 Player Performance

Mean *Score*, mean *Kills*, mean *KD-Ratio*, mean *NoH* and, mean *DpH* for each level of PHANTOM LATENCY are shown in Table 4.10.

We again used a one-way ANOVA (PHANTOM LATENCY: *30 ms*, *60 ms*, *90 ms*, and *120 ms*) to investigate for statistical differences as ANOVA prerequisites were not violated (Shapiro-Wilk test $p > 0.05$). ANOVA revealed no significant effect of PHANTOM LATENCY on *Score* ($F(1,23) = 2.387$, $p = 0.074$, $\eta_p^2 = 0.072$) and *Kills* ($F(1,23) = 2.694$, $p = 0.051$, $\eta_p^2 = 0.081$). However, ANOVA detected significant difference in the data of

KD-Ratio (F(1,23) = 3.873, $p = 0.012$, $\eta_p^2 = 0.112$), *NoH* (F(1,23) = 3.330, $p = 0.098$, $\eta_p^2 = 0.023$), and *DpH* (F(1,23) = 3.136, $p = 0.029$, $\eta_p^2 = 0.093$). Next, we used Tukey's test to conduct a post-hoc comparison for all significant ANOVA results. P-values and confidence intervals, again, are corrected for multiple comparisons. Tukey's test revealed significant differences in *KD-Ratio* ($p_{Tukey} = 0.016$, $d_{Cohen} = 0.876$, CI95=[0.072, 0.994]), in *NoH* ($p_{Tukey} = 0.028$, $d_{Cohen} = 0.820$, CI95=[0.424, 10.326]), and in *DpH* ($p_{Tukey} = 0.041$, $d_{Cohen} = 0.775$, CI95=[0.072, 5.646]) between 30 ms and 120 ms of PHANTOM LATENCY. Participants playing with 120 ms of PHANTOM LATENCY had a significantly worse kill-to-death ratio (meaning they died more often while eliminating fewer enemies), hit significantly fewer headshots, and dealt significantly less damage per shot in general compared to playing with 30 ms of PHANTOM LATENCY Figure 4.13 and Figure 4.14 depict PP and significant differences between conditions.

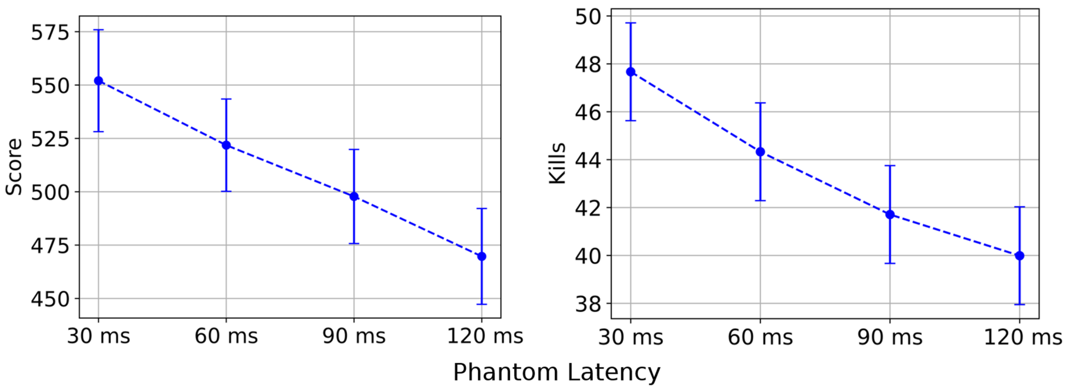


Figure 4.13: *Score* and *Kills*. Error bars show standard errors. We found no signficant differences in both measures.

4.4.3.3 Qualitative Feedback

After each 10-minute round of playing CS:GO with one of the four levels of PHANTOM LATENCY, participants were asked if they felt latency in the last round and, if so, how they felt that the latency in the game manifested itself, i.e., what effect it had. Out of all 96 individual rounds played, participants felt latency in 43 rounds (44,79 %). Nineteen participants (79.167 %) felt latency while playing with 120 ms of PHANTOM LATENCY and 15 participants (62,5 %) were sure there was latency when playing with 90 ms of PHANTOM LATENCY. Still, seven rounds (29.167 %) were associated with latency, while a suggested PHANTOM LATENCY of 60 ms was presented to the participant. In the 30 ms

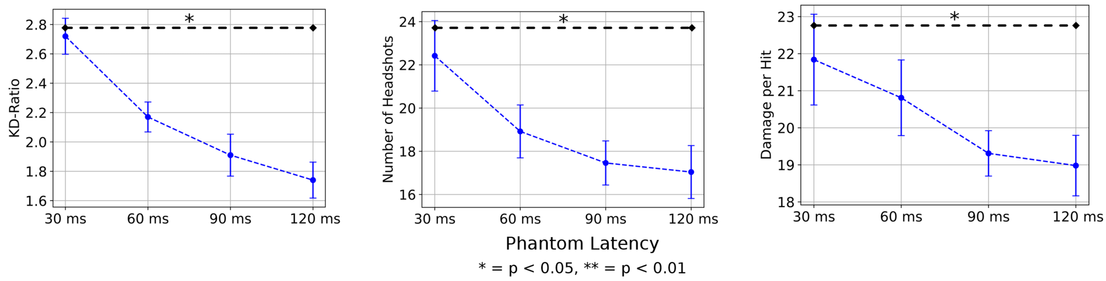


Figure 4.14: Results for *KD-Ratio*, *Number of Headshots*, and *Damage per Hit*. Error bars show standard errors. Significant pairwise comparisons are highlighted via asterisk. Participants playing with 120 ms of PHANTOM LATENCY had a significantly worse kill-to-death ratio, hit significantly fewer headshots and dealt significantly less damage per shot in general compared to playing with 30 ms of PHANTOM LATENCY.

PHANTOM LATENCY condition only two participant (8.333 %) felt latency. Crucially, all participants played with the same actual latency of 75 ms in all rounds. Increasing the amount of displayed latency also increased the likelihood of participants feeling the effects of latency. The higher the displayed latency, the higher the ratio of participants reporting to have felt it in the last gaming round.

4.4.4 Discussion

Our results consistently show that phantom latency - a latency merely suggested and displayed to the player - has significant effects on the PP and GX (RQ5). In this section, we first discuss and contextualize phantom latency's effects on the feeling of tension and competence. We continue to shed light on how phantom latency alters the positive feelings associated with the game. Next, we discuss the effects of phantom latency on the objectively measured PP. We showcase the implication of our findings for researchers, game developers, and gamers alike. We conclude this section by discussing our study's limitations and possible future works.

4.4.4.1 Effects on Tension, Competence, and Positive Affect

Our work shows that players were tenser when playing with 120 ms than when playing with 30 ms or 60 ms of phantom latency. We also found the same systematic regarding the positive affect associated with the game - players achieved a lower positive affect score when playing with 120 ms of phantom latency compared to playing with 30 ms and 60 ms of phantom latency. Lastly, we also found that phantom latency altered how competent players felt while playing. Playing with the highest level of phantom latency (120 ms) lead to a significantly reduced feeling of competence compared to playing with 30 ms, 60 ms, or even 90 ms of phantom latency.

Our findings regarding the experienced tension and the associated positive affect align with our previous work, which showed that true latency increases tension and reduces all positive feelings and emotions towards digital video games (cf. Section 3.2). A reduced positive affect indicates that the players experienced less joy and pleasure and generally reduced perceived fun of playing the game (IJsselsteijn et al., 2013). The enjoyment of an activity has a systematic influence on the performance during this activity (Ashby & Isen, 1999) - this is known as the performance-enjoyment link in video games (Schmierbach et al., 2014; Rogers et al., 2015). Our work extends prior work and shows that the effect of latency on the GX is not entirely technical but at least partly expectation-based. The mere suggestion or presentation of latency, as done in our work via our custom latency overlay, is enough to induce an expectancy-based effect, significantly altering the experiences obtained in the gaming session. Evidently, we also found that phantom latency reduced the subjective feeling of competence in our study's players. This is particularly interesting because all tested players were highly skilled and played at least 100 hours of Counter-Strike: Global Offensive (mean = 535.16 hours, *SD* = 556.68 hours). One possible origin of the effects of phantom latency on the feeling of competence is that players may tried to adapt to the phantom latency based on prior experience with true latency. One could assume that all players knew how true latency manifests in the game and which techniques usually work to compensate for it (such as over- or undershooting). Since our study always had the same true latency of 75 ms, previous experience with latency did not provide any advantage and may even result in a reduced feeling of competence. Players may tried to adapt to a varying latency, which ultimately was constant.

We did not find a significant effect of phantom latency on the other subscales of the GEQ. This indicates that the subscales Sensory, Flow, Negative Affect, and Challenge are either not susceptible to phantom latency or an expectation-based effect in general.

4.4.4.2 Effects on Player Performance

Our findings consistently show an effect of phantom latency on gaming performance. In line with other work researching the effects of true latency in video games, we show that players were less accurate and less effective when playing with the highest phantom latency (120 ms) compared to playing with its lowest level (30 ms). In addition, players had a significantly worse kill-to-death ratio, hit significantly fewer headshots, and subsequently dealt significantly less damage per hit.

However, contrary to previous work, which showed that true latency directly influences game performance (Claypool & Claypool, 2006; Beigbeder et al., 2004), our work shows that the players' performances seem to be less affected by phantom latency. This behavior was somewhat expected since the performance is a hard quantitative measure and less prone to subjective manipulation (Kosch et al., 2022). Nevertheless, phantom latency significantly and negatively influenced PP at its highest level. The effects of phantom latency on performance can be explained using the classical placebo, respectively nocebo, paradigm. Players in our study performed worse because they anticipated performing worse based on their experience with the game. The players' expectations of performing in a specific condition ultimately manipulated how they performed. The number of given headshots demonstrates the successful suggestion of latency by our latency overlay. A headshot in CS:GO is the purest form of skill and competence in the game since it is exceedingly hard to hit a target as small as the head when enemies are moving around. While it is almost impossible to hit such a small target (depending on how far away the enemy is) when playing with 120 ms true latency, our induced phantom latency would not have a technical effect. Nevertheless, players in the 120 ms phantom latency condition hit significantly fewer headshots than in the 30 ms phantom latency condition. This again demonstrates the performance-enjoyment link discussed in the GX section. Players not only felt less competent after playing a round with high phantom latency, but they also actually behaved less competently, as showcased by the significantly reduced number of hit headshots.

4.4.4.3 Implications for Researchers and Game Developers

Our findings have implications for researchers and game developers. First, video game researchers profit from our findings since we show that the expectation of a gaming session may fundamentally change the course of a session. This may be relevant when investigating novel interfaces, new game mechanics, or game elements. Latency researchers, in particular, benefit from our work as we show that latency has an inherent expectancy-based component. This is especially the case when testing experienced players who are used to the game or system behaving in a certain way. We conclude that it thus may be best to keep the participants blind to the actual study's goal when investigating latency, as the mere expectancy of latency may already alter the study's outcome. Finally, our findings are also relevant to previous work investigating the effects of latency in interactive systems. Since previous work did not directly account for the nocebo effect of latency, some of the results may overstate the effect of latency. While it is undoubtedly true that latency has a real effect, revisiting previous approaches to latency and its compensation may be worthwhile with the expectancy-based effects in mind.

Second, game developers should also be aware of the effect of displaying latency. Based on our findings, displaying latency may not always be advisable when optimizing for GX. A poor GX has consequences - not only for players but also for game developers and publishers. In the severest scenario, an unsatisfactory GX will result in the game being canceled (Forbes, 2019; Video Games Chronicle, 2020; Kotaku, 2019). A range of current video games, such as PlayerUnknown's Battleground (Krafton Inc., 2020), already aim to disguise latency using color-switching icons instead of textual values. However, it is unclear if concealing latency using an icon is enough to prevent players from forming an expectation about latency in the game.

4.4.4.4 Limitations and Future Work

Our work demonstrated an effect induced by the expectancy of latency in video games. Nevertheless, our work has limitations, presenting new avenues to investigate expectancy-based effects in video games.

In medicine, placebo (placebo vs. baseline) or placebo-controlled (placebo vs. active treatment) studies require at least two controlled groups of conditions (Kosch et al., 2022; Kaptchuk, 1998). Such a study design allows researchers to compare a potential placebo effect against an actual and no treatment. Our study only investigated the effects of suggested latency (phantom latency) in a within-subject study design without a dedicated control group. This approach is valid because the actual effects of true latency in video games, especially in CS:GO, have been demonstrated numerous times by previous work (Liu et al., 2021d; Liu et al., 2021c). Nevertheless, conducting a placebo-controlled study investigating latency in video games may allow a comparison of the true and the expectancy-based effect of latency. Thus, we encourage future work to investigate expectancy-based latency effects in a placebo-controlled study that operationalizes the treatment (true latency vs. phantom latency) in its design.

Furthermore, since the placebo and nocebo effects are entirely based on the participants' expectations, we can not conclusively be certain about their origin. Our work showcased an expectancy-based effect of phantom latency. However, we can not be sure what triggered the effect. As we postulate and discuss, it is possible that the expectancy of latency decreased gaming experience and performance. On the other hand, it is also possible that the lack of latency variation negatively influenced our participants. Participants expected and anticipated varying levels of latency in the study; as soon as the gaming session started, those expectations were violated, possibly leading to a decrease in experience and performance. While this may be another possible explanation for our findings, we concluded that it is highly unlikely. Considering that 79.168 % of the participants were sure that they felt latency while playing with 120 ms of phantom latency, it is implausible to assume that the participants' expectations were explicitly violated. Nevertheless, it may be beneficial to investigate further what circumstances allow and support the formation of an expectancy-based latency effect.

Lastly, in our work, we controlled for the players' individual skills in CS:GO to increase the work's reliability and validity. Nevertheless, it is possible that skill influences the formation of expectancy-based latency effects beyond our control mechanisms. In our study, we investigated the effects of phantom latency on players ranked 1 to 12 in CS:GO's internal ranking system. However, while all of our participants can be considered high-skilled players, there still is a difference in experience, technique, and play style between a rank 1 player and a rank 12 player. Thus, future work should investigate how phantom latency affects GX and performance if all tested players are on the exact same skill level.

4.4.5 Conclusion

This section presented a study with 24 participants playing CS:GO with phantom latency. In our study, we primed participants with four levels of phantom latency (30 ms, 60 ms, 90 ms, and 120 ms) using a self-developed latency overlay, while the true latency in the gaming session was not manipulated and stayed consistent in all conditions. We found that the mere suggestion of latency significantly and bidirectionally influences PP and GX. Participants were significantly tenser, felt less competent, and associated the game with significantly less positive feelings the higher the amount of phantom latency. Besides the subjective experience, we also found that participants' objective performances were lowered by phantom latency. They achieved a significantly reduced kill-to-death ratio, hit fewer headshots, and dealt less damage per hit when playing with the highest level of phantom latency. We discuss our findings and conclude implications for researchers and game developers. We discuss that it may not always be the best approach to display latency to players if one wants to optimize PP and GX. Furthermore, we discuss that previous work investigating latency may overstate latency's technical effects by not accounting for its expectancy-based component. Researchers must be aware that a participant's expectancy of the system or part of the system (such as latency) can alter the outcome of an investigation. In light of this thesis and the development of novel latency compensation techniques, we show that informing players of the current latency can change their GX and PP.

5

Compensating Latency via Artificial Neural Networks

The previous two chapters allowed us to learn about different aspects of latency and its effects in video games. We learned that small-term latency variation does not affect PP and GX (RQ1). However, we also gained the understanding that a long-term switch between latency levels does, on the other hand, have negative consequences on the gaming session and its players (RQ2). Furthermore, we also learned that auditory latency is only relevant for highly skilled video game players (RQ3) and that the visual in-game perspective does not modulate the effects of latency (RQ4). Lastly, the last section of the previous chapter also demonstrated that how latency is communicated to the players in a gaming session is highly relevant. We showed in a study that the mere suggestion of latency, for example via an in-game overlay, has ramifications on PP and GX (RQ5). The knowledge gained in previous studies allows us to derive five concrete recommendation for future latency compensation approaches:

First, small-term latency variation is negligible, long-term switches should be avoided. Hence, latency compensation should aim for long-term stability. Second, auditory latency does not have to be accounted for individually. Thus, if a differentiation between visual and auditory latency is necessary, a compensation system should focus on providing visual responsivity and a reduction of visual latency. Third, the in-game perspective does not alter the effects of latency. Hence, latency compensation always has to aim for latency reduction, independent of the current in-game perspective. Fourthly, the expectation of latency alters PP and GX. In conclusion, latency compensation should obfuscate the current latency. Lastly, participants in latency compensation studies should not be informed about the compensation technique, since this can bias the results.

In this chapter, we utilize these recommendations and present novel latency compensation techniques based on ANNs. As we learned in Section 2.3 previous work investigated different latency compensation techniques and methods, such as geometrical manipulation (Lee et al., 2019), time warp (K. Lee & C. Chang, 2017; Bernier, 2001), user input extra- and interpolation (Liu et al., 2022), or game world alteration (Gutwin et al., 2004). However all of these methods have certain limitations or disadvantages (such as always favoring the shooter in time warp (Bernier, 2001)), a more general and flexible solution is missing. Deep learning-based ANNs, on the other hand, showed tremendous capability to predict user behaviour and inputs. In previous work, ANNs have successfully been used to compensate for latency in AR/VR settings (Buker et al., 2012) and in touch screen interaction (Henze et al., 2017; Henze et al., 2016).

To develop and evaluate our technique we conducted three user studies. We simulated network latency in a cloud gaming (CG) scenario for all three studies, since users of CG services are most affected by latency. As already briefly discussed in previous parts of this thesis, in a CG scenario, games are streamed via video to the players. While streaming a game, the user's input is sent to the streaming server via the Internet. The cloud gaming server receives the input, calculates reactions to the input, renders the game, and sends the results back to the player as a video stream. This extensive communication via the Internet increases the overall latency for video game players using CG services. CG has the potential to trigger a paradigm shift in the gaming industry – the ability to play cutting-edge AAA productions from anywhere is promising. However to be usable and accepted by video game players CG needs a mean to reduce latency. Hence, we situated our research within a CG scenario.

In the first study, we used a self-developed FPS game, which we also used to evaluated auditory latency in Section 4.1. In the game, players have to fight endless waves of enemies to survive. We used a self-developed game, since this allows us access game-internal states and information. Commercial video games, on the other hand, often are black boxes and restrict access to the game's internal logic, state and code. The additional data gained by using a self-developed game may serve as a source for ANN training.

We added artificial latency to the game via input buffering and coupled the game to a ANN trained on data of players playing the game. We used the ANN to compensate for four levels of latency and found that the system's concept works. Players using the ANN prediction performed better and had a better gaming experience. Hence, we conclude that ANNs can compensate for the negative effects of latency in custom video games (RQ6). In the second study we deployed the same approach for a commercial video game – Age of Empires 2 (AoE2). AoE2 is a relatively slow-paced and tactical RTS game, which makes it a suitable object for the next line of investigation of using ANNs to compensate for latency. Contrary to the first study we did not use game-internal information to train the ANN. Thus, ANN training was exclusively based on data that is available to the player of the game instead of relying on game-internal information. In the study we found comparable results as previously. Our ANN was able to compensate for some of the negative effects of latency (RQ7), players had a higher level of experience albeit we did not found an increase in performance. The last study of this chapters combines the two previous approaches. We deployed an multi-model ANN-based latency compensation system in the fast-paced, high-action, extremely competitive video game CS:GO. However, contrary to previous investigation we did not train the ANN on fixed prediction. Previous work, similar to the previous two studies, trained ANNs on constant latency values (Henze et al., 2016; Henze et al., 2017). Thus, the resulting ANNs could only compensate for a constant latency value. Latency, however, is never perfectly constant as we discussed in Chapter 2 and showed in Chapter 3. Previous work introduces a certain degree of fuzziness by compensating for a constant latency. Hence, in the last study we parameterized latency as input to the ANN itself. This means that the ANN is highly flexible and can adapt itself to varying latency environments. In a study, we found that our ANN is able to compensate for varying latency (RQ8). Players playing with our compensation system had an increased performance and a better gaming experience.

This chapter presents the background, the research rational, the data collection, the experimental setting, the procedure, and the results of all three studies. All findings are discussed in detail within the context of the conducted study. Background and research rationale for individual studies are provided if suitable.

5.1 Artificial Neural Networks in Custom Video Games (Study VII)

Cloud game streaming services, such as Google's Stadia (Google, 2020) or Blade's Shadow (Shadow, 2020), offer gamers a variety of advantages compared to conventional gaming platforms. The entire computing load is borne by the provider's server. This server provides remote play and renders the game via video stream to the players. The local computer only has to display the received video stream, which significantly reduces hardware requirements for playing graphically demanding video games. Players do not have to constantly worry about upgrading their gaming PC to meet the latest game requirements (Sun & Claypool, 2019). Additionally, gamers do not have to install games on their own devices – games are pre-installed on the server and playable almost instantly.

While cloud streaming has advantages over conventional gaming systems it simultaneously entails some drawbacks. Due to their architecture and mechanism, streaming services for games are inflicted with a higher latency than conventional gaming platforms, such as a gaming PC or a gaming console. Popular game magazines such as EuroGamer (Eurogamer, 2020) and PCGamer (PCGamer, 2019) report an average latency of 183 ms for Google's game streaming service Stadia. Despite efforts to compensate for latency, either by game-internal or game-external approaches, current commercial game streaming services still are afflicted by latency, which negatively influences PP and GX. Poor performance and dissatisfied users ultimately lead to users abandoning the service. This has economic ramifications for the providers, up to financial bankruptcy (PCMag UK, 2015).

Previous research emphasizes the capabilities of predictions based on machine learning models (Le et al., 2017; Henze et al., 2016) and shows that video games benefit from even small reductions in latency (Liu et al., 2021c). Currently, however, it is unclear if ANN-based latency compensation can be used in video games to reduce the negative effects of latency. To answer this question, we first establish if ANNs can be used for latency compensation in custom video games (RQ6), in which we have full access to the game's internal state.

5.1.1 Background and Research Rationale

Previous research, not only in the field of HCI, emphasizes the capabilities of predictions based on machine learning models (Le et al., 2017; Henze et al., 2016) and shows that video games benefit from even small reductions in latency (Liu et al., 2021c). This work highlights how to compensate for latency by using ANN models that predict the player's avatar movement. In this work, we developed a real-time 3D video game to record player input and derive ANNs that are able to predict a player's movement in the game. We simulate latency corresponding to the delays of current streaming services and test these ANN in the same real-time gaming environment to predict the recent avatar position and orientation within the current game state. In a subsequent evaluation, we found that prediction-based latency compensation using ANNs significantly enhances PP and gaming experience in high latency setups. Our study shows that latency-based delays in real-time games can be reduced and gaming experience and performance improved using only a few parameters obtained from the player's avatar.

This section is based on the following article:

Halbhuber, D., Henze, N., & Schwind, V. (Oct. 2021). "Increasing Player Performance and Game Experience in High Latency Systems." In: *Proc. ACM Hum.-Comput. Interact.* 5.CHI PLAY, pp. 1–20. DOI: 10.1145/3474710.

5.1.2 Data Collection and Apparatus

Similar to previous work (Le et al., 2017), we followed a data-driven approach to compensate latency using ANNs in a high latency gaming session. This approach entails formulating a goal, such as compensating for latency, gathering real-world user data, developing appropriate ANNs, and testing them in a conclusive evaluation. Hence, to test if ANNs are able to compensate for latency, we first designed and developed an FPS game, which we can use to gather game data, such as avatar position, avatar orientation, and a number of in-game metrics. Second, we gathered training data for the ANN in a data acquisition study in which participants were instructed to play the game. Next, we trained deep learning-based algorithms using the data set and tested the capability of in-game avatar prediction for latency compensation. We optimized the model's parameter to minimize the loss on a separate test set. Finally, we evaluated our prediction model in a user study with 96 participants playing the game while we recorded metrics to obtain PP and GX.

5.1.2.1 Game Development

One requirement for the game and the user study was barrier-free access for potential participants during the COVID-19 pandemic. To meet this requirement, we conducted the data acquisition and main study in this work exclusively online instead of in a laboratory setting. Although this decreases the internal validity and controllability of local phenomena, such as input latency, it emphasizes the ecological validity using an in-the-wild approach. To conduct our studies online, we used the browser as the game's target platform. As the game is fully hardware-accelerated, all processing and rendering is performed locally on the players' computers.

Since players of FPS games react particularly sensitive to latency (Claypool & Claypool, 2006) we selected them as the target group of our research. Next, we designed and developed the game world and the avatar in Unity3D (Version 2019.f16.2). Additionally, we implemented a first-person controller allowing the players to control the avatar within the game world. The game world is divided into three parts: The first two parts comprise an introductory tutorial (basic controls and shooting), which allows the player to familiarize themselves with all avatar controls and the UI. The third part of the game world is the actual game, in which the player is situated in a jungle environment with enemies spawning at five different portals. Figure 5.1 shows an aerial view of the implemented game world – the different game parts are highlighted. Additionally, Figure 5.2 shows the player's view while playing in the game arena.

The player's objective during the game is to shoot a fixed number of hostile monsters before they reach the player's avatar. If a monster reaches the player, the player loses points from a pool of limited life points. In case the player has been hit four times by the enemies, the game was over and restarted in the third game section. If the player manages to shoot all enemies, the player reaches the next game level while earning points equivalent to the number of enemies shot. To keep the player motivated while playing, we gradually increase the game's difficulty with each level by increasing the number and speed of enemies.

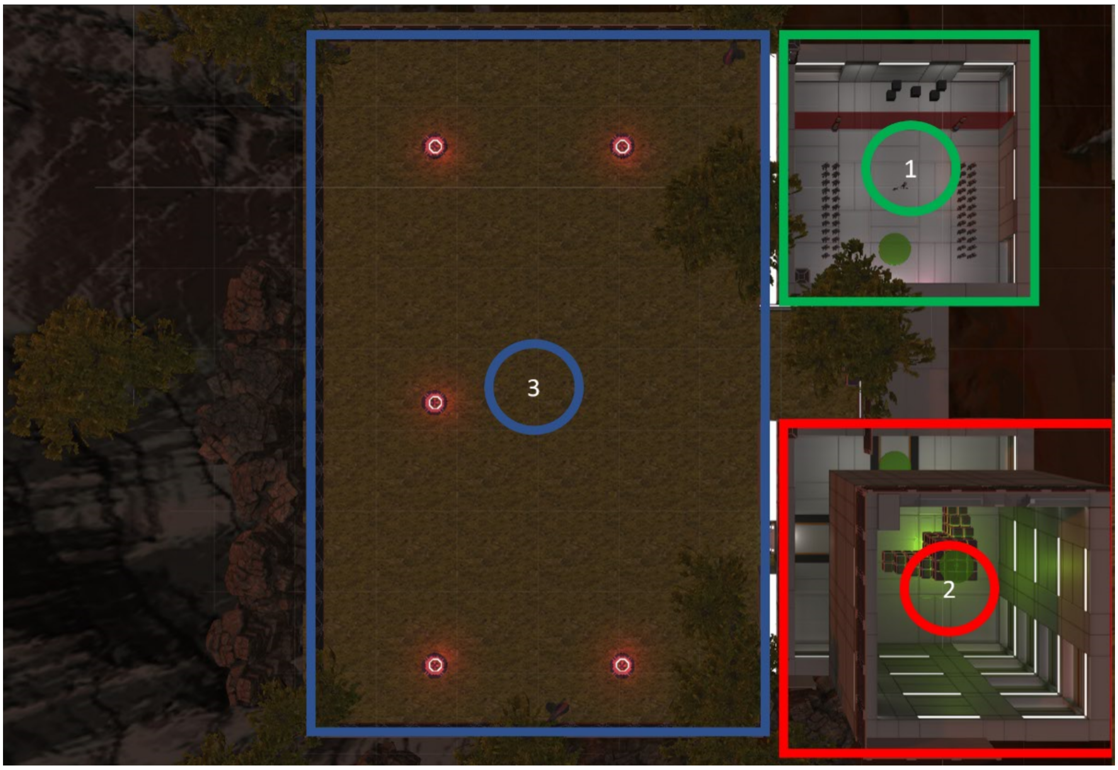


Figure 5.1: Shows a top view of the designed game world. The different game areas are color-coded. The green and red game areas were parts of the tutorial. In the blue game area, players have to compete against the AI opponents.

5.1.2.2 Game Data Collection

We conducted a data acquisition study to collect the in-game data necessary for training an ANN capable of predicting avatar movement in real time.

Apparatus.
For the study, we hosted the game on a publicly reachable web server. Participants played the game on their own devices in a browser of their choice. It was not necessary to download game files or manually install the game on the player's devices. Content delivery is fully handled by our web server and does not require any additional user input.

Procedure and Task.
After entering the website, participants were presented with a consent form including the purpose of the study. After giving informed consent to data collection, participants

Figure 5.2: Shows the players' in-game view while playing in the third game area.

could move on to the game start screen. Participants were aware of the data collection but not of the precise purpose of the collection or the exact type of data collected. The study received ethical clearance via the ethics policy of the University of Regensburg. By clicking on the start button, participants started playing the game. All further instructions for the study were handled directly in the game via a game interface containing a dialog box. The players were guided by the game through tutorial A (basic controls) and tutorial B (shooting). Upon reaching the third game area, the participant's goal was to fight and survive the enemies waves while simultaneously obtaining as many points as possible. After 800 seconds of playtime, the game automatically closed and ended the session.

Participants.

We recruited 24 participants (9 female, 14 male, 1 preferred not to say) evenly from two sources: a mailing list of our institute and crowd-sourcing via *prolific.co*. Both groups were compensated either with credit for their study course or £3 as monetary compensation. A total duration of 20 minutes has been estimated for participation. Their average age was 25.8 years (SD = 6.0 years), with the age ranging from 18 years to 41 years.

5.1.3 Model Development

The gathered data has been processed in three steps to gain a model able to predict player input and movement: (1) pre-processing of the data, (2) using the data for training deep learning models, and (3) integrating the developed model into the game.

5.1.3.1 Data Pre-Processing

We logged a total of 602 943 unique samples from the participants in the data collection study. The data is divided into three categories: (1) continuous frame data, (2) event-based data, and (3) system specifications. The majority of the data consists of continuous frame data. One frame consists of the following data points: (1) frames passed since game start, (2) seconds passed since game start, (3) X-, (4) Y-coordinates of the current raw mouse position in pixel, (5) X-, (6) Y-, (7) Z-coordinates of the current avatar position as well as (8) X-, (9) Y- and (10) Z-coordinates of the current avatar orientation. We recorded a total of 522 165 frame samples. On average, participants generated 21 756 frames ($SD = 8.625$) in each gaming session. 80 788 samples (80 764 events, 24 system specifications) were generated from event-based data and system specification logs. Event-based data contains information about in-game events, such as the player firing, the player dying, the player losing all health, or the player hitting an enemy with a shot. Participants fired their virtual weapons a total of 20 060 times ($M = 835.6$, $SD = 800.5$) while fighting against 3 228 enemies. In total, participants played for 19 200 seconds (5.3 hours) with an average frame rate of 30.2 FPS ($SD = 4.9$).

For this study, we trained solely on continuous frame data of each player. In detail, each row of the final training data set maps one frame consisting of the following eight data points: (1-2) raw mouse coordinates in X, Y, (3-5) avatar coordinates in X, Y, Z, (6-8) avatar orientation in X, Y, Z. To increase prediction accuracy we enlarged the prediction baseline. Instead of inferring based on one frame our ANN uses five successive frames to predict future avatar positions and orientations. Finally, we performed the training of the ANN based on 40 unique input values using this approach.

5.1.3.2 Latency Determination

To determine how far the model should predict, we determined typical latency values in game streaming applications. We based our prediction values on current latency measurements of popular game magazines such as EuroGamers (2020) and PcGamer (2019).

The elaborated latency of Google's Stadia based on six different reports averages at 183 ms (SD = *103* ms). Although the standard deviation of the reported measures is high, we choose to use the mean as the upper bound of the artificially added latency, based on a trifold reasoning: (1) Technical advancements in cloud gaming are leading to an ever-reducing latency. An optimistic choice with a low latency corresponds better to a real-world scenario in ongoing improvement rather than the choice of conservative high latency. (2) Ng et al. (2014) showed that users perceive latency down to 1 ms, following the authors findings, it is clear that lower latency values play a crucial role in the interaction between humans and computers. Choosing a higher upper artificial latency limit, would have lead to neglecting and blurring of the lower values. And, (3) since previous work already demonstrated that high latency values highly influence GX and PP, we choose to investigate lower latency values, which rendered us able to perform a finer graded and more detailed compensation via the ANN prediction. This, in turn, allows for more detailed analysis of the added latency, the ANN prediction and the effects on participants playing without compensation technique. Consecutively, we defined the maximum prediction value of our system to be at 180 ms (rounded down). Thus the trained model is able to compensate for 180 ms latency in a high latency environment such as a game streaming service.

5.1.3.3 ANN Training and Model Development

The ANN's goal is to compute the avatar position and orientation in 180 ms based on the given training data set. Crucial for choosing the ANN architecture was the time needed to predict the next output. Since the output had to be generated and merged with the game in real-time, the duration for inference had to be minimized. The prediction must not interfere with or slow down the execution of the gameloop. Considering typical game frame rates from 30 to 60 frames per second (FPS), inference and merge had to be done within 33 ms to 16 ms. Thus, we choose a lightweight ANN implementation based on a Multistep-Dense-Network architecture (Tensor Flow, 2015).

The network's first layer (L_1 = 40 neurons) is used to transform multi-dimensional input data into a one-dimensional array of scalars. This layer is followed by four fully connected hidden layers (L_2=1024 neurons, L_3 = 128 neurons, L_4 = 128 neurons, L_5 = 1024 neurons). The number of hidden layers and the number of neurons implemented in each layer were determined using grid search (Feurer & Hutter, 2019). The last hidden layer passes the processed data to the last layer – the output layer (L_6 = 8 neurons). To account for overfitting, we added a drop-out (rate = 0.1) function between the last hidden layer and the output layer (Srivastava et al., 2014). The output layer, in turn, outputs the avatars orientation and position in the chosen prediction value, e.g. 180 ms. To account for non-linearity, we solely used Rectified Linear Unit (ReLU) (Glorot et al., 2011). We optimized the data using adaptive movement estimation (ADAM) (Kingma & Ba, 2014) with a batch size of 64 samples, a learning rate of 0.1 and a loss implementation based on the mean squared error (MSE). Through Keras callback function, we dynamically changed the learning rate in the training to enable the underlying back-propagation method in our ANN to deal with local optimization minima. Figure 5.3 shows the structure of our ANN.

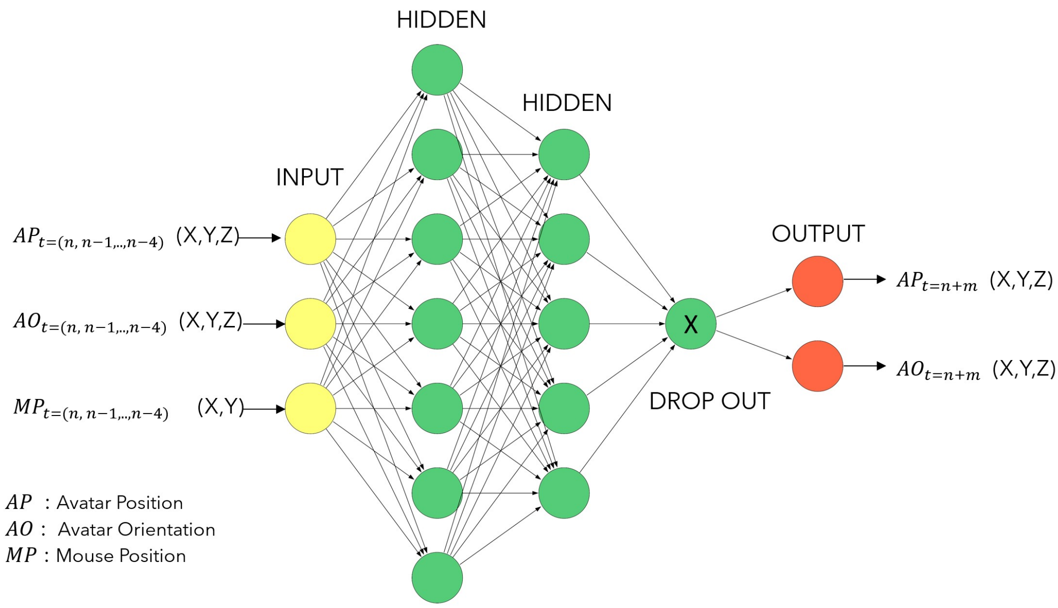


Figure 5.3: Depicts the conceptual structure of our artificial neural network used for latency compensation. Due to size limitations, not all layers and neurons are depicted. We input the current and the four most recent($t = (n, n-1, .., n-4)$) avatar positions, orientations as well as mouse positions. The input (yellow) reduces the multi-dimensional input to a one-dimensional tensor. Consecutively, the networks' hidden layers (green) follow, which are connected to a drop-out before feeding to the output layer (red), which outputs avatar position and orientation for the desired moment of time ($t = n + m$).

We trained additional prediction values aside from the 180 ms forecast. The additional values are factors of 180 ms making their effect comparable to the original 180 ms prediction. Thus, we trained the model to predict 60 ms, 120 ms, and 180 ms in the "future". We trained each prediction mode until no further improvement in loss optimization was observable, which was achieved after 45 epochs of training. Loss was calculated using Unity's Worldspace Coordinates (UWC) and the MSE. The final losses were 19.24 UWC for the *+60 ms* model, 42.31 UWC for the *+120 ms* model and 48.32 UWC for the *+180 ms* model. The training was performed on a PC with Windows 10, AMD Ryzen 7 1800X, 64 GB RAM, and two NVIDIA GeForce GTX 1080 TIs with a total of 22 GB VRAM. Training the ANN took about 4.4 hours for each prediction mode, totaling an overall training time of 13.2 hours.

5.1.3.4 Model Integration

To predict the avatar's position and orientation, we integrated the model in the environment of the game. We used Tensorflow.JS (Abadi et al., 2016; Smilkov et al., 2019) to serve the model online. Inferences from the model can be requested directly from the game via JavaScript. For inference, the model needs the previously defined 40 input values. We implemented various game functions to cache avatar position, avatar orientation, and mouse position. Once the caching function saved five frames, the game sends a request for inference to the model, waits for the result, and applies the received values directly in the game by updating the avatar position and orientation. Consecutively, the game repeats this process for every frame, discarding the last cached frame and adding the current frame to the rolling cache. All these processing steps are performed within one game loop, i.e. within a single frame. Thus, the model inference does not increase the game's execution time. Figure 5.4 shows the prediction pipeline.

5.1.4 Method

We conducted an online study to test the hypotheses that different PREDICTION TIMES impact GX and PP. We used PREDICTION TIME as a between-subject variable. In addition to a baseline model with (1) *0 ms* prediction, which is achieved by completely disabling the ANN prediction, we tested three different ANN models: (2) *+60 ms*, (3) *+120 ms* and (4) *+180 ms*. The four levels of PREDICTION TIME were evenly distributed and randomly assigned to the participants.

We collected data about PP and GX for each participant. Performance is measured in three dependent variables: (1) *MaxScore* - maximum number of points achieved, (2) *HitCoef* – hit-shot-ratio and (3) *EnemyHit* – number of enemy hits. We used the

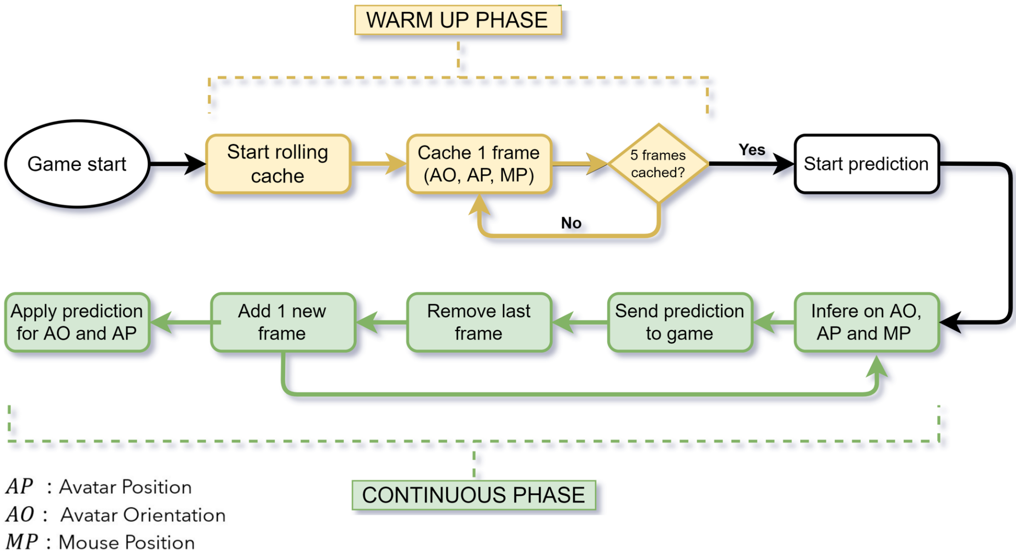


Figure 5.4: Depicts a flow diagram of the prediction pipeline. With the start of the game, the warm-up phase starts (orange) as well. In warm up the pipeline catches five consecutive frames. After five frames, the warm-up phase ends, and the continuous phase (green) starts. In this phase, the pipeline infers on Avatar Position (AP), Avatar Orientation (AO) and Mouse Position (MP). Then, the prediction is sent back to the game. Before applying it in the game, the oldest frame in the cache is replaced with the current frame, which triggers a new prediction iteration.

maximum number of points achieved instead of the average score because we wanted to measure peak performance instead of average performance. To measure GX, we used the GEQ (IJsselsteijn et al., 2013), with its subscales: Competence, Sensory, Flow, Tension, Challenge, Negative Affect, and Positive Affect.

5.1.4.1 Apparatus

The apparatus was similar to the one used for data collection. We hosted the game on a publicly reachable web server. Participants played the game on their own devices in a browser of their choice. In dependence on their assigned conditions, players either played the *0 ms* baseline game or were supported by one of the three ANN models (*+60 ms*, *+120 ms*, or *+180ms*). Resulting in the following four conditions: (I) *0 ms*, (II) *+60 ms*, (III) *+120 ms* and (IV) *+180 ms*. In all conditions, we added 180 ms of artificial latency via input buffering.

5.1.4.2 Procedure and Task

On entering the web page, participants were presented with a consent form. After giving informed consent, participants were able to start the game. Participants were aware of and agreed to data collection but were not aware that they tested different ANN models. Consequently, they did not know that they were being assisted in their gaming session by a latency compensation technique based on ANNs (*+60 ms, +120 ms, +180 ms)*. The study received ethical clearance as per the ethics policy of our institute. All further study instructions were handled directly in the game via a UI. The game guided the participants through the tutorial. Upon reaching the third game area, the participants' goal was to survive against the enemies while simultaneously obtaining as many points as possible. After 800 seconds of playtime, the game automatically closed the gaming session and opened the final questionnaire within the browser window.

5.1.4.3 Participants

We recruited 96 participants (23 female, 70 male, 3 preferred not to say) through the crowd-sourcing platform Prolific.co. Thus, each condition was tested using a total of 24 participants. Participants who participated in our data collection study could not attend. Participants were compensated with £3, and we estimated a total duration of 20 minutes for their participation. The average age of the participants was 24.6 years (*SD* = 5.7), with the age ranging from 18 years to 51 years.

5.1.5 Results

In total we recorded 2 652 949 unique samples from the participants. The major part is the continuous frame data, which sums up to a total of 2 297 526 unique frames. On average, participants generated 23 933 frames per session (*SD* = 9 487 frames). Event-based data, as well as the unique system specification data, add up to 355 423 samples (35 5327 events and 96 system information). During the conducted user study, participants fired their virtual guns a total of 51 533 times (M = 536.80, *SD* = 635) and fought against a total of 13 732 enemies. The game was played by the participants for 86 373 seconds (23.99 hours/25.9 min per participant) in all conditions.

5.1.5.1 Maximum Score (MaxScore).

Before further analysis we checked *MaxScore* for normal distribution using the Shapiro-Wilk test. Neither data from *0 ms* (W = 0.76, p = <.001), *+60 ms* (W = 0.81, p = <.001), *+120 ms* (W = 0.77, p = <.001) nor data from *+180 ms* (W = 0.86, p = <.001) is normally distributed. Subsequently, we conducted a Kruskal-Wallis test which showed a significant effect of PREDICTION TIME ($\chi^2(3)$ = *17.59*, p = <.001) on the maximum game score of the players. Pairwise comparison using Wilcoxon signed rank test showed, in combination with a Bonferroni α correction, a significant difference between *0 ms* and *+60 ms* (W = 114, p = .005), between *0 ms* and *+120 ms* (W = 112, p = .005) as well as between *0 ms* and *+180 ms* (W = 107.5, p = <.001). Mean *MaxScore* values, as well as p values are depicted in Figure 5.5. Increasing the prediction increased PP. Players performed best in the *+180 ms* condition.

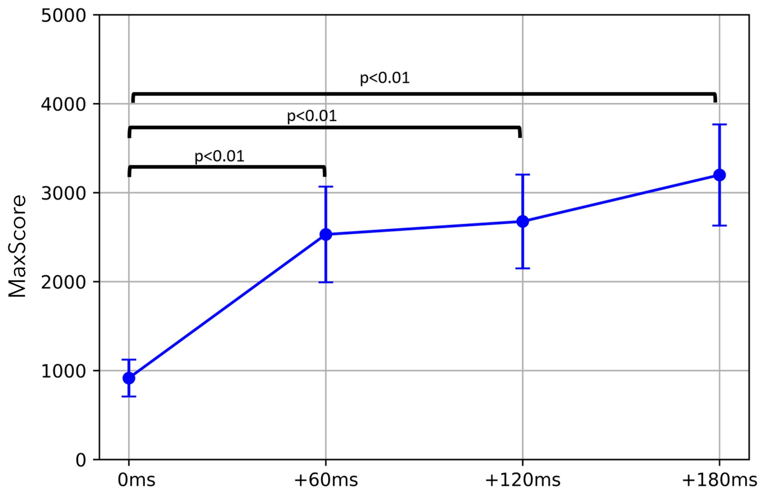


Figure 5.5: Shows the evaluation of MaxScore. Illustrates the average maximum score achieved by all players over all conditions. Significant differences are marked at the corresponding places via p-bars. We found significant differences between *0 ms* and *+60 ms*, between *0 ms* and *+120 ms*, and between *0 ms* and *+180 ms* after Bonferroni correction. The error bars show the standard error.

5.1.5.2 Shot-to-Hit Ratio (HitCoef).

Shapiro-Wilk test showed that the gathered data from *0 ms* (W = 0.94, p = 0.23), *+60 ms* (W = 0.96, p = 0.39) and *+120 ms* (W = 0.98, p = 0.39) follows a normal distribution.

Data from the *+180 ms* condition does not fit a Gaussian distribution (W = 0.89, p = 0.01). Since not all *HitCoef* data is parametric, we choose to use Kruskal-Wallis test again. The test showed a significant effect of PREDICTION TIME ($\chi^2(3)$ = *34.39*, p = <.001) on the shot-to-hit ratio of the players. Pairwise comparison using Wilcoxon signed rank test showed, in combination with a Bonferroni α correction, a significant difference between *+120 ms* and *0 ms* (W = 50, $p < .001$), between *+120 ms* and *+60 ms* (W = 53, $p < .001$), between *0 ms* and *+180 ms* (W = 52.4, $p < .001$), and between *+120 ms* and *+180 ms* (W = 76.0, $p < .001$). Mean *HitCoef* values, as well as p values are depicted in Figure 5.6 (left). Increasing the prediction up to 120 ms increased players shot-to-hit ratio. Predicting 180 ms did not increase the *HitCoef* further. Player performed best in the *+120 ms* condition.

5.1.5.3 Hit by Enemies (EnemyHit).

Using Shaprio-Wilk we determined that data from condition *0 ms* (W = 0.92, p = 0.08) and condition *+120 ms* (W = 0.94, p = 0.24) is normally distributed. Data from condition *+60 ms* (W = 0.82, p = <.001) and *+180 ms* (W = 0.91, p = 0.03) is not normally distributed. We did not find significant differences in the times a player got hit by enemies ($\chi^2(3)$ = *3.73*, p = 0.29) using the Kruskal-Wallis test. However, looking at Figure 5.6 (right), a general trend is recognizable, but yet to be evidently proven. Figure 5.6 (right) shows the mean values of *EnemyHit*. Increasing PREDICTION TIME did not significantly decrease the time players got hit by enemies.

5.1.5.4 Game Experience Questionnaire.

Based on the authors' recommendations, we evaluated each category of the GEQ individually (IJsselsteijn et al., 2013), starting with an analysis for normal distribution. Table 5.1 shows the evaluation of the Shapiro-Wilk test on the different questionnaire categories.

Oneway ANOVA for parametric, independent data showed no significant differences in the categories Competence ($F(95)$ = 1.14, p = .32), Flow ($F(95)$ = 0.74,p = .52) and Challenge ($F(95)$ = 0.25, p = .85). Kruskal-Wallis statistical test for non-parametric data showed no significant differences in the categories Sensory ($\chi^2(3)$ = *1.99*, p = 0.57), Tension ($\chi^2(3)$ = *5.02*, p = 0.17) and Negative Affect ($\chi^2(3)$ = *2.79*, p = 0.42). The test revealed significant difference in the category Positive Affect ($\chi^2(3)$ = *12.16*, p = <.001).

Further analysis via Wilcoxon test showed significant differences between the baseline *0 ms* condition and the *+120 ms* condition (W = 118.5, $p < 0.01$). Increasing PREDICTION

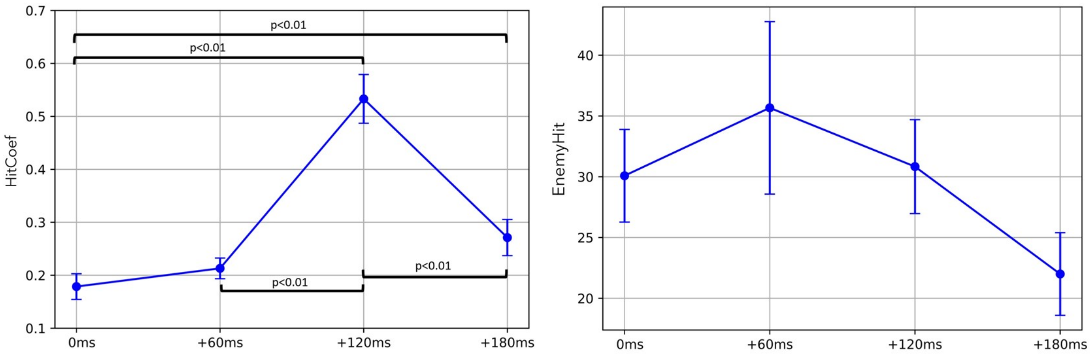


Figure 5.6: Shows the evaluation of the shot-to-hit coefficient (HitCoef) (left) and enemy hits (EnemyHit) (right). The left diagram illustrates the average hit quotient achieved by all players across all conditions. Significant differences are shown at the corresponding places via p-bars. Significant differences could thus be determined after Bonferroni correction between *0 ms* and *+120 ms*, *0 ms* and *180 ms*, *+120 ms* and *+60 ms*, and between *+120 ms* and *+180 ms*. The error bars show the standard error. The right diagram illustrates the average number of hits that opponents have scored at the end of the game. Despite a noticeable downward trend, no significant differences in the different distributions could be found. The error bars show the standard error.

TIME to 120 ms, significantly increased the experienced positive feelings and emotions that player perceived or experienced while playing. Participants assigned the *+120 ms* condition the highest Positive Affect score.

5.1.6 Discussion

Our results indicate that our latency compensation technique has the highest impact on the players' scores with maximum prediction (+180 ms). Generally, we showed that the participants achieve a higher maximum score in the game with all predictions. The highest game scores were achieved in the +180 ms condition. These results are in accordance with the latency/performance limits of (Claypool et al., 2014). According to the authors, FPS games are most affected by latency. Claypool et al. state that games of this genre experience negative effects starting at a latency value of 100 ms. Furthermore, the authors explain that a further reduction of latency does not lead to a significant

Shapiro-Wilk Test GEQ Categories				
	0 ms	+60 ms	+120 ms	+180 ms
COM	W = 0.96, p = 0.44	W = 0.95, p = 0.44	W = 0.96, p = 0.56	W = 0.96, p = 0.66
SEN	W = 0.92, p = 0.06	W = 0.87, **p = <0.01**	W = 0.92, p = 0.06	W = 0.92, p = 0.07
FLO	W = 0.96, p = 0.57	W = 0.93, p = 0.15	W = 0.92, p = 0.09	W = 0.98, p = 0.92
TEN	W = 0.92, p = 0.07	W = 0.91, **p = 0.05**	W = 0.88, **p = <0.01**	W = 0.95, p = 0.24
CHA	W = 0.95, p = 0.25	W = 0.94, p = 0.17	W = 0.96, p = 0.46	W = 0.94, p = 0.15
NEG	W = 0.96, p = 0.58	W = 0.93, p = 0.09	W = 0.91, **p = 0.05**	W = 0.90, **p = 0.02**
POS	W = 0.89, **p = 0.01**	W = 0.95, p = 0.26	W = 0.97, p = 0.68	W = 0.93, p = 0.15

Table 5.1: Shows the evaluation of the Shapiro-Wilk test for the different questionnaire categories (Competence (COM), Sensory (SEN), Flow (FLO), Tension (TEN), Challenge (CHA), Negative Affect (NEG) and Positive Affect (POS)) and conditions (*0 ms*, *+60 ms*, *+120 ms* and *+180 ms*).

improvement in performance. We also did not find a significant increase in performance by further decreasing latency. We found no significant difference in the maximum score achived between +60 ms, +120 ms and +180 ms.

Considering the average values for HitCoef, we showed that participants using the +120 ms condition achieved the highest shot-to-hit-ratio. Consequently, latency compensation of more than 120 ms did not lead to a further improvement of the shot-to-hit ratio. The HitCoef ratio can be equated with the accuracy of the participants. Participants are more accurate when a larger number of opponents are hit with fewer shots. Despite the decreasing accuracy in the +180 ms condition, we show that the accuracy of the participants is significantly improved by our predictive system. These results are also consistent with the work of Beigbeder et al. (2004). The authors show that latency-induced effects reduce the accuracy of players by up to 50%. Our work shows that these effects can be compensated by predicting avatar position and orientation. The compensation of latency significantly increases the accuracy of participants in systems with latency. This is generally caused by the reduced input/display offset. By predicting, the participant gets a more direct reaction to the input action than in a system without prediction. We conclude that due to the real-time representation in the game, participants have more time to place the crosshairs more consciously and consequently more accurately, which ultimately increases the shot-to-hit-ratio.

The third analyzed variable is the number of hits the players received by enemies (EnemyHit). We found no significant difference for the number of enemy hits, although

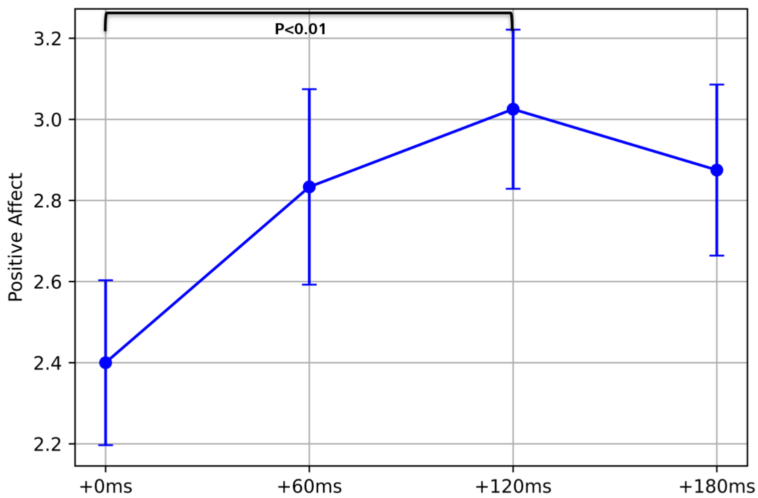


Figure 5.7: Shows the evaluation of the positive affect score of the game experience questionnaire. Illustrates the average rating submitted by all players overall conditions. Significant differences are marked at the corresponding places via p-bars. Significant differences between *0 ms* and *+120 ms* were determined after Bonferroni correction. The error bars show the standard error. Players significantly stronger associated the game with a Positive Affect in the *+120 ms* condition compared to the *0 ms* condition.

a descending trend can be recognized when looking at the mean values of the measure. Thus, the variable first rises slightly in the comparison of condition 0 ms and condition +60 ms, but then falls continuously over condition +120 ms to its minimum average value in condition +180 ms. The missing significant differences can be attributed to the circumstance that the number of enemy hits depends strongly on the ingame position of the avatar. Tactical and anticipatory behavior of the participants was paramount. Since the opponents attack from any side, the participants benefit more strongly from anticipatory gameplay than from a latency-free environment. Our ANN cannot support the tactical positioning of the avatar.

We found significant differences in the positive affect category of the GEQ. All other categories showed no significant differences. Positive affect describes the feelings and emotions experienced by the players during the game. These feelings and emotions include, for example, joy, pleasure, and the subjectively perceived fun of playing (IJsselsteijn et al., 2013). Participants in the study rated the condition with 120 ms prediction with the highest positive affect score. Participants in this condition most strongly associated their experience with the game with fun. Positive affect, in particular the enjoyment of an activity, has a systematic and positive influence on the performance during this activity (Ashby & Isen, 1999). This is also shown by the in-game metrics discussed above. Participants in the +120 ms condition achieved better performance scores and experienced more fun while playing than participants in all other conditions. Therefore, we argue that using a system to compensate for latency significantly increases the GX by increasing the perceived positive effect.

GX is crucial for the success of any game. The failure of countless high production-value examples, such as Anthem (Forbes, 2019), Fallout 76 (Kotaku, 2019), and Warcraft 3: Reforged (Video Games Chronicle, 2020), illustrated the effects of neglecting GX in the game design process. In this regard, our work showed, as prior work did (Sabet et al., 2020b; Beigbeder et al., 2004; Eg et al., 2018), that perceived latency negatively influences GX and performance. Additionally, our work presents a solution to this quandary – prediction-based latency compensation. Our findings and the resulting implication are relevant to game streaming providers, game developers, and researchers as well.

Game streaming services providers must continue their endeavors to improve existing infrastructures. Server availability and density must be increased to further reduce the resulting network latency. Optimizing network latency will result in lower end-to-end latency for gamers, which in turn will improve GX. In addition, carriers should work closer with game development studios to enable them to account for the expected latency ranges in the design phase. Ideally, game streaming services should allow developers to incorporate the method presented in this section. In doing so, providers could offer an interface through which a prediction for latency compensation is trained and integrated - individually for each deployed game.

Game developers, on the other hand, need to be aware of the difference between traditional gaming platforms and game streaming services. Developing a game for a traditional system should not be equally approached as developing a game for a streaming service. Although developers use different tools to adapt the game to different platforms, latency is not taken into consideration. Adaption is carried out exclusively on a technical level. Future game developments should consider possible latency already in the design phase. Thus, latency-sensitive game sections should be adapted in such a manner that they become more robust against latency. Besides adjusted development, game studios should also consider the possibility of a prediction-based latency compensation in their game. In this case, game developers could incorporate a prediction directly into the game, similar to the method presented in this section. The prediction should react adaptive to the current latency level and thus ultimately enable a latency-independent GX.

Finally, the findings of our work and the presented method are relevant to other researchers as well. Researchers in the field of video games and researchers in HCI may profit from the presented method. Although we used game-specific parameters to train our model, it is likely that the presented approach is suitable for any kind of software operated by a mouse and keyboard. For example, by integrating mouse prediction, a software could achieve higher responsiveness and thus enhance the perceived UX.

5.1.7 Limitation and Future Work

Our ANN is trained by data from our game. This allows us to directly implement model inference into the game but simultaneously limits the usage of the ANN for said game. A more general approach to train and implement an ANN-based prediction could be used to evaluate our method in commercially available video games. This approach, although, would need to be implemented on the operating system level since most commercial video games do not allow interference from a third-party application.

Another limitation can be found in the trained prediction values - 60 ms, 120 ms, and 180 ms. We solely trained discrete fixed timings, but latency in real-world applications is not represented by a single discrete value but a dynamic range of values. In the context of our work, assuming discrete latency is valid to fully control the environment but future work should focus on developing and investigating adaptive systems that adjust the prediction to latency measured in real time.

The ANN architecture in this section is based on commonly used deep learning architectures. A more sophisticated and complex model may be able to generate more precise predictions and further improve our results. We optimized the ANN until we were satisfied with the loss achieved; Our results show that this approach can already improve PP and experience. The architecture and the model parameters can, however, be further optimized – for example, the number of hidden layers could be increased when more training data is available. We encourage further research and provide all data and the model to the public. This enables future work beyond latency research. For example, a deeper analysis of the collected data sets can provide further insights into the behavior of gamers.

Additionally, future work could provide insight into whether our presented method is valid in different gaming contexts, such as networked multiplayer games. In such games, different latency levels create an unjust situation. Players with high latency are more disadvantaged by latency than players with lower latency. Especially in competitive esports, players often travel far to compete with other players under identical and controlled conditions. High latency can make the difference between virtual survival and death and consequently can cost the players millions in prize money (Statista, 2021a; EarlyGame, 2020). An adaptive system, similar to the system presented in this work, could bring the latency of all gamers to a common denominator thus creating a fair gaming environment for competitive players.

5.1.8 Conclusion

In this section, we presented a novel approach to compensate for high latency in custom video games using ANN. We predict the position and orientation of the in-game avatar in a self-developed 3D video game. This prediction effectively reduces the necessary time to process a feedback loop between player and game, since our system anticipated the player's next move and implements it in the game before the actual user input is received.

The prediction of our system is based on data of 24 participants playing a self-developed first-person video game. This data was collected in a data collection study and includes information about avatar position and orientation, the position of enemies as well as some event-based information, such as data about firing behavior and the current score and maximum score of the players. After data collection, we trained an ANN to predict changes in avatar position and orientation based on past frames. We trained three different levels of predictions: +60 ms, +120 ms, and +180 ms – the highest value is based on the current latency of commercial game streaming services. After training, we conducted a second study with 96 participants to validate the approach. In the study, the participants played with 180 ms of artificial latency and one of the prediction modes.

In our second study, we showed that players with +120 ms and +180 ms prediction achieved significantly higher game scores than in the other conditions. Additionally, we showed that by using the +120 ms model players achieved higher accuracy values compared to all other conditions (0 ms, +60 ms, +180 ms). Furthermore, participants rated the +120 ms condition with significantly higher scores in the positive affect category of the GEQ. In summary, we were able to show that negative latency-based effects, such as performance degradation, can be compensated by our system. Ultimately, this enables players to achieve low-latency performance and GX.

5.2 Artificial Neural Networks in Slow-Paced Games (Study VIII)

In the previous section, we showed that ANNs are capable of compensating for high latency in custom video games by using game-internal information, such as the avatar's orientation and position (RQ6). Players playing with the ANN-based compensation achieved a better performance and had a higher level of GX. However, typically for-profit game developers and publishers do not grant access to game-internal data to their players. Accessing and altering the game's internal state is often prohibited to prevent cheaters and hackers from levering this data to obtain an unfair advantage over other players or the game. This, of course, makes third-party latency compensation techniques that rely on game-internal data nearly impossible. Hence, in this section we investigate a ANN-based latency compensation technique that only uses data and information that is available without accessing the game's internal logic or mechanic. For this investigation, we utilize the RTS game Age of Empire 2. AoE2 is a comparatively slow game, which makes it more robust for latency induced effects. However, given that this is a novel method, yet not investigated in previous work, it is reasonable to use a more robust and slower video game for the first step of ANN-based latency compensation in commercial video games. Ultimately, the overarching RQ this section answers is whether ANNs can be used to compensate for the negative effects of latency in commercial video games (RQ7).

This section is based on the following article:

Halbhuber, D., Seewald, M., Schiller, F., Götz, M., Fehle, J., & Henze, N. (2022f). "Using Artificial Neural Networks to Compensate Negative Effects of Latency in Commercial Real-Time Strategy Games." In: *Proceedings of Mensch Und Computer 2022*. MuC '22. Darmstadt, Germany: Association for Computing Machinery, pp. 182–191. ISBN: 9781450396905. DOI: 10.1145/3543758.3543767.

5.2.1 Background and Research Rationale

In this section, we show how to compensate for the latency of commercial video games without modifying the game or accessing the internal game state. We use ANNs to compensate for the adverse effects of latency by predicting future movement of the computer mouse in the RTS game AoE2. To achieve this, we conducted a data collection study with 21 participants playing AoE2 and Empire Apart (DESTINYbit, 2022). We used the gathered data to develop, test, and benchmark two different ANNs. The first ANN predicts the future mouse position in 50 ms based on past mouse movements. The second ANN predicts future mouse position in 50 ms based on past mouse movement and in-game images. We evaluated both ANNs and found that visual data does not lead to a better prediction than just using the mouse movement data. Thus, our second study with twelve participants investigated the effects of an ANN-based latency compensation using past mouse movement data while playing AoE2 with latency. Our evaluation shows that the developed ANN significantly enhanced the GX compared to playing with high latency and no prediction. We conclude that ANNs trained only on externally available data, can be used in commercial RTS games to reduce the adverse effects of latency on the GX.

5.2.2 Data Collection and Apparatus

In line with the previous section, we again used a data-driven approach to develop and evaluate the presented ANNs. First, we developed custom tools to remotely record mouse movement and video output while playing any video game. In a data collection study, we then used the developed tools to collect data from 21 gamers playing the RTS games AoE2 and Empire Apart. We used two different games to increase the ANN's potential to generalize over different games.

5.2.2.1 Development of the Data Gathering Tool

We gathered all data to train our ANNs in-the-wild to maximize ecological validity. While this approach introduces some randomness, for example, when considering local latency, it simultaneously potentially increases the ANN's generalization capability. Since the data is not collected in a laboratory setting, it contains more variation. Gathering data in the real world allows the ANNs to potentially learn a more nuanced version of gamers playing video games, which is not constrained by the setting of a laboratory study.

We exclusively used Python 3 to develop our data-gathering tool. The tool consists of different methods for starting games, capturing and converting gameplay images, recording mouse inputs, multi-threaded data processing, and archiving and uploading the gathered data. For automatically installing and starting games we used PyWinHook (Tungsteno, 2022), PyNput (Palmér, 2022) and Steam (Valve, 2022). For image capturing, resizing, and converting, we used OpenCV2 (OpenCV Team, 2022). For mouse movement recording, we utilized PyWinHook, and for uploading the gathered data to a remote server, we made use of PySFTP. We created an executable from our Python scripts using PyToExe (Vollebreg, 2022).

After starting the program, it automatically logs in to the game library application Steam with hard-coded login credentials. If our application can not find a current Steam installation, it tries to download and install it. After logging into Steam, the tool waits for user input to start the data collection. Data is only collected if an appropriate game is running and maximized to protect participants' right to privacy. Upon ending the game session, i.e., if the game gets closed, the tool automatically finishes uploading the gathered data before closing itself.

Our tool records the current mouse position in the game every 5 ms. Mouse positions, however, are only recorded if the current mouse position changed compared to the mouse position 5 ms ago to minimize the amount of logged data and to prevent identical mouse position entries. Furthermore, the software captures, resizes (to 848 pixel by 480 pixel), and starts to upload one gameplay image every 41.67 ms (recording in 24 fps). We limited the resolution of the gameplay images to reduce the time needed to upload the data to our server while obtaining as much visual information as possible. Recording 1 hour of gameplay still produced about 10 GB of data. Figure 5.8 shows two game screenshots our data gathering tool recorded.

5.2.2.2 Data Collection Study

Using the developed tool, we conducted a data collection study to acquire the necessary gameplay data to develop ANNs capable of predicting user inputs to reduce latency without accessing game-internal states or data.

Figure 5.8: Shows screenshots of the Real-Time Strategy games Empires Apart (DESTINYbit, 2022) (left) and Age of Empire 2 (Microsoft Games, 2022) (right).Both screenshots show the game interface, some player-controlled units and buildings from a vertical bird's eye view. We used the games in a remote data collection study to obtain data suitable for training our latency compensation system. The screenshots were recorded using our data-gathering tool.

Apparatus

For the study, we sent our data-gathering tool to our participants. Participants played the games on their devices and thus did not have to install any software manually. The study ran automatically and did not require any additional input from the experimenter.

Procedure and Task

After giving informed consent to the data collection, participants received an e-mail with our tool and detailed instructions. Participants started the tool by double-clicking the received executable. After confirmation by the participants and maximization of the game, the data logging (mouse movement + gameplay images) started. Next, participants were asked to play one hour of free play, which is a sandbox mode in both games, allowing them to start playing right away. After playing for one hour, participants closed the game. Our tool automatically compressed and uploaded the gathered data to our remote server. The study, and thus the data collection, received ethical clearance by the University of Regensburg's ethics policy.

Participants

Twenty-one participants (4 f, 17 m) were recruited via our institute's mailing list. Participants were selected independent of age, gender, and experience playing RTS games. All participants were students and compensated with credit for their course of study. The study took about 70 minutes per participant. Participants' average age was 25.3 years (SD = 3.6 years), ranging from 20 years to 36 years.

5.2.2.3 Development of the Artificial Neural Networks

Overall, we collected 1 412 802 gameplay images (due to rare transmission errors, the number of collected images does not reflect the theoretical maximum) and 3 479 488 mouse positions in our data collection study. We used the gathered data to develop two deep learning-based ANNs: (1) a Deep Neural Network (DNN) (Goodfellow et al., 2016, pp. 164-223) and, (2) a mixed-model Convolutional Neural Network (CNN) (Goodfellow et al., 2016, pp. 326-366) combined with a DNN. We optimized the ANNs' parameter to minimize the loss on a train-test set combination. The accuracy of both ANNs was evaluated using a separate validation set (80/10/10 split). Both ANNs' goal was to predict the mouse position in 50 ms. This prediction could be sent to the streaming server in a CGS scenario. The server receives this prediction after a certain time, for example, after 50 ms. However, since the server received a predicted position in the future, the subsequent rendering will not be affected by latency. This predictive method effectively reduces the gamer's experienced latency. We defined the prediction value for both ANNs per previous work, which shows that latency negatively affects gamers starting at 25 ms. Furthermore, we showed in Section 5.1 that a prediction of 60 ms already significantly increased gamer performance and GX. Thus, to increase the likelihood of the amount of latency and latency prediction influencing gamers, we set the prediction value to 50 ms.

General Description

Both ANNs were trained using a supervised deep learning approach. Hence, we defined the amount of data used for inference (the input) and the corresponding frame or mouse position as output in training. We used TensorFlow (Abadi et al., 2016) to train all models and *Optuna* (Akiba et al., 2019) for hyperparameter optimization. For training, we used a local server with an Intel i9-990k CPU, 16 GB Ram and two GPUs, one Nvidia RTX 2060 with 6 GB VRAM, and one Nvidia 1080Ti with 8 GB VRAM. As in the previous section, it was crucial for both ANNs that the time needed to predict the next output is minimized. Since the output of the ANNs has to be applied to the game in real-time, the duration for inference had to be as short as possible. The prediction must not slow down the interaction with the game. Since typical frame rates in games range from 30 to 60 frames per second (FPS), generating the next output and merging it back into the game had to be finished within 16 ms to 33 ms. We optimized the Mean Absolute Error (MAE) between the actual delta value of the mouse position in 50 ms and the predicted value for all models.

Training the Dense Neural Network

The first ANN uses the last ten mouse positions to predict a delta value for the mouse position in X and Y coordinates in 50 ms. Since we recorded one mouse position every 5 ms, the prediction is based on the data of the last 50 ms. Hence, the ANN's input consists of 10 pairs of X and Y coordinates. The network consists of five fully connected layers. The first layer L_1 - the input layer, has 20 neurons and passes the input to the first of three fully connected hidden layers (L_2 = 64 neurons, L_3 = 32 neurons, L_4 = 16 neurons). The last layer L_5 - the output layer - has two neurons and serves the predicted value of the mouse position in 50 ms in X and Y coordinates. We used ADAM (Kingma & Ba, 2014) with a batch size of 128 samples and a learning rate of 0.001 for back-propagation. As activation function we used TensorFlow's built-in ReLU implementation (Glorot et al., 2011). We used early stopping (Prechelt, 1998) to prevent overfitting and custom callback functions to deal with local optimization minima. The model was trained for 100 epochs, with one epoch taking approximately 36 minutes. In evaluation, the model achieves an MAE of 12.4 pixels per coordinate on the unknown validation set. Considering a full HD monitor with a resolution of 1920 pixels by 1080 pixel the ANN creates a prediction with an error of less than 1 % on the X-axis and about 1.2 % on the Y-axis. Generating one inference takes about 1.6 ms, which is fast enough considering our requirement of a prediction time of less than 16 ms.

Training the mixed-model Convolutional Neural Network

The second ANN uses five past game images and the corresponding mouse positions to predict the mouse position in 50 ms. Thus, the prediction is based on gameplay from the past 208 ms. The dense part of the network is identical to the first presented ANN. The convolutional part of the model consists of seven layers. L_1, a fully connected dense layer, receives the screenshot in the recorded resolution (848 px X 480 px) and scales it down by factor 10 to 84 px X 48 px. Additionally, L_1 grey-scales the input images. Layer L_2 ,the first convolution layer (neurons = 64, stride = $(3,3)$, padding = 3), is followed by a max pooling layer (pool = $(2,2)$). L_4 (convolution, neurons = 32, stride

= $(3,3)$, padding = 3) and L_5 (max pooling, pool = $(2,2)$) follow the same structure as L_2 and L_3. The next layer, L_6, is a flatten layer, which receives the multi-dimensional output from the previous layer and transforms it into a one-dimensional array. In the next step, this array is passed to a dense layer (L_7, neurons = 16), which generates X, Y coordinates predicting the mouse courser's position. The output of the convolutional and the network's dense parts are concatenated in a final dense layer. We, again, used ADAM with a learning rate of 0.1 and a batch size of 4 samples as well as ReLU to optimize. We utilize drop-out (drop-out rate = 0.2) (Srivastava et al., 2014) to prevent overfitting of the model. The second ANN was trained for ten epochs, with one epoch taking 36 hours to train. After training, the ANN achieved an MAE of 18.8 px in predicting the mouse position in 50 ms. Again, considering a full HD monitor the MAE corresponds to an error of less than 1 % on the X-axis and about 1,8 % error on the Y-axis. The model finishes one inference in 5.4 ms. Figure 5.9 shows an overview of the developed model (left) and an architecture plot of the convolution branch of the model (right).

Overall, we did not find that using gameplay images increases the prediction accuracy. Thus, we used the non-convolutional DNN for further investigation.

Integration of ANN to the Game

To predict the mouse position in 50 ms using the developed model, the model needs the means to communicate with the game. Although we trained the ANN on the data of two different games, we used only one game in the evaluation to test the generalizability of our training data. Since AoE2 is a commercial game, it is prohibited to manipulate the game or the current game state directly. Thus, we used the Python library PyNPut to override the mouse position on the operating system (OS) level. Using this approach, the game never receives the actual mouse hardware value but only the predicted values of our model. Upon starting the inference pipeline, the ANN waits for ten mouse positions to be received before starting to pipe the inference to the game. Via rolling cache procedure (similar to the procedure presented in Figure 5.4 in Section 5.1), the ANN removes the oldest mouse positions upon receiving a new position. Using this method allows the ANN to predict a future mouse position continuously.

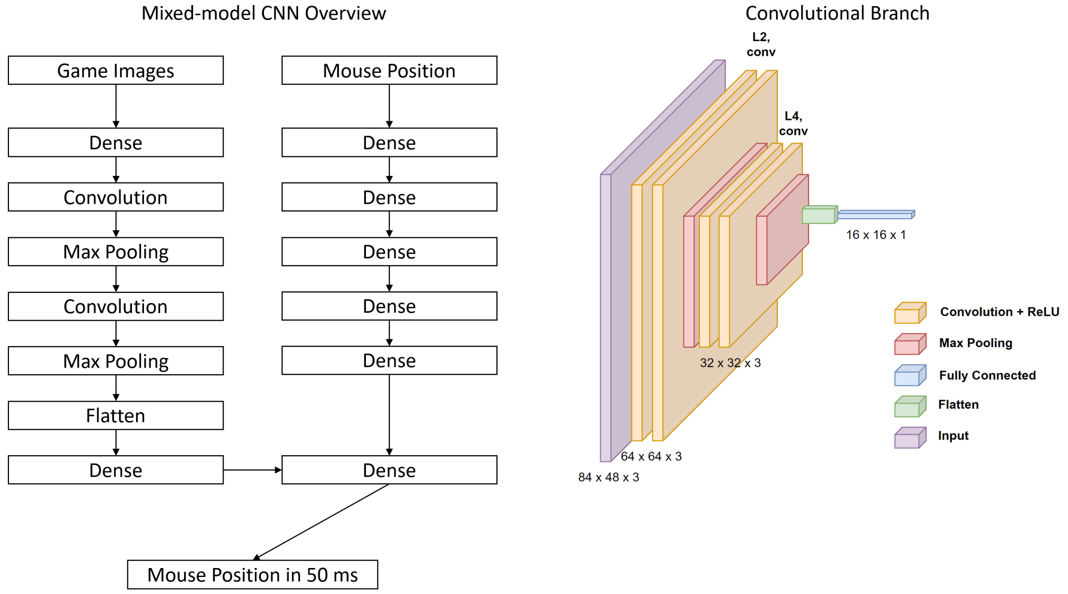


Figure 5.9: Shows an overview of the mixed-model *Convolutional Neural Network* (CNN) (left), and the architectural structure of its convolutional branch used for in-game image processing (right). In the model, mouse positions are processed by a series of dense layers, while the in-game images are processed by the convolutional branch. To reduce computational load, the original images are downsized to 84 pixels by 48 pixels. This downsized image is passed through two convolutional layer coupled to max pooling. The multi-dimensional image is then transformed to a one-dimensional array using a flatten layer. The output of the convolutional branch is fed to a dense layer, which also receives the processed mouse positions. Finally, the last dense layer outputs X,Y coordinates for the mouse position in 50 ms.

5.2.3 Method

We conducted a second study to determine if ANN-based prediction systems can compensate for the negative effects induced by latency. In the study, we created 50 ms artificial, controlled latency by buffering the mouse input on the OS level using a custom Python script and the Python library PyNPut.

5.2.3.1 Study Design

We investigated the effects of our prediction using a within-subject design. Hence, we used DELAY as an independent within-subject variable. The variable had three levels: (I) *No Latency* - which is used as a control condition and corresponds to playing with no latency and no applied prediction -, (II) *Latency* - which corresponds to playing with 50 ms of controlled latency and no prediction being applied in the gaming session -, and lastly, (III) *Prediction* - categorizing gaming sessions with a controlled latency of 50 ms and an enabled prediction compensating 50 ms of latency. To measure the GX, we used the in-game modules of the GEQ (IJsselsteijn et al., 2013). We used the six subscale Competence, Flow, Tension, Challenge, Positive Affect, and Negative Affect of the in-game module and expanded it by the four subscale Positive Experience, Negative Experience, Tiredness and Return to Reality by the post-game module to quantify the subjective effects of our system on the gamers.

We measured the effect of DELAY on the players' performances using the dependant variable *Score*. *Score* is generated using AoE2's built-in scoring system. All participants played with all levels of DELAY, resulting in three different conditions for our study. The condition order in the study was randomized to prevent a bias induced by sequence effects.

5.2.3.2 Apparatus

We installed AoE2 on a stationary workstation (Intel i9-9900k, 16 GB RAM, Nvidia RTX 2060 6 GB VRAM) in our laboratory. The workstation was attached to a monitor (24" FullHD @60Hz), a computer mouse (Logitech M10), and a wired headset. The laboratory was quiet and free of external disturbance.

5.2.3.3 Procedure and Task

Participants were greeted at our institution's laboratory by the experimenter. After giving informed consent and agreeing to the data collection, participants were seated at the workstation running AoE2. Participants were not informed about the exact purpose of the study (investigating the effects of the developed prediction system). However, they were briefed about the general procedure of the study. Next, all participants played three rounds of AoE2's free-play mode for 15 minutes. After each round, they filled out the GEQ. The order of game rounds, and hence prediction modes, was randomized. Upon finishing the third round and the third time filling out the GEQ, we collected demographics and information about past gaming experiences in RTS games and AoE2. Lastly, participants were debriefed. Participation in the study took about 1 hour. The study received ethical clearance as per the University or Regensburg's research ethics policy.

Score Game Experience Questionnaire										
Latency	Com.	Flo.	Ten.	Cha.	Pos.	Neg.	Pos. E.	Neg. E.	Tired.	Real
No Delay	2.75/0.83	1.75/0.81	0.54/0.23	1.95/1.07	2.58/0.88	0.66/0.57	2.09/0.89	0.68/0.39	0.63/0.57	0.94/0.69
Delay	1.95/1.07	1.54/0.87	1.75/1.21	2.21/1.03	1.38/0.99	1.71/1.16	1.39/0.99	1.26/0.86	1.66/0.91	1.02/0.78
Prediction	2.25/1.37	1.71/1.01	0.95/1.08	1.52/1.02	1.84/1.05	0.71/0.54	1.85/1.05	0.81/0.58	0.71/0.72	1.05/1.01

Table 5.2: Shows the mean scores with standard deviation (mean/SD) for all subscale of the Game Experience Questionnaire (IJsselsteijn et al., 2013). The data is grouped by the three levels of LATENCY.

5.2.3.4 Participants

Using our institution's mailing list, we invited 12 participants (2 f, 10 m). The participant mean age was 23.16 years (SD = 2.31 years). Participants were chosen blind to age and gender. To prevent a bias induced by individual gaming skills, participants had little to no experience playing AoE2. Participant pre-screening was crucial since experienced AoE2 players deploy sophisticated gameplay strategies such as using shortcuts, queuing of unit construction, and simultaneously controlling independent units, which would have distorted the data of our study. Furthermore, all participants were students at our institution and were compensated for participation with credit for their study course.

5.2.4 Results

We independently evaluated each sub-scale of the GEQ. Table 5.2 shows the mean score and standard deviation for all subscale. All measures showed no violation of normality using Shapiro-Wilk's test (all $p > 0.05$). Hence, we used an analysis of variance (ANOVA) for further statistical testing. For post-hoc testing we used Tukey tests.

A one-way ANOVA (DELAY: *Latency* vs. *No Latency* vs. *Prediction*) showed no significant effect of DELAY on the in-game subscale Competence, Flow, Challenge, Positive Affect and no significant effect on the post-game subscale Positive Experience, Negative Experience, and Returning to Reality (all $p > 0.2$, all $\eta_p^2 < 0.1$). However, ANOVA revealed a significant main effect of DELAY on Tension (F(1, 11) = 4.632, p = 0.017, $\eta_p^2 = 0.219$). A Tukey post-hoc test showed significant differences between *No Latency* and *Latency* ($p_{tukey} = 0.014$, $d_{cohen} = 0.223$). All other differences were not significant (all $p_{tukey} > 0.137$, all $d_{cohen} < 0.091$). One-way ANOVA also revealed a significant main effect of DELAY on Negative Affect (F(1, 11) = 4.632, p = 0.005, η_p^2 = 0.278). Post-hoc testing showed significant differences between *Latency* and *No Latency* ($p_{tukey} = 0.009$, $d_{cohen} = 0.287$) and between *Latency* and *Prediction* ($p_{tukey} = 0.013$,

ANOVAs of Game Experience Questionaire										
	Com.	Flo.	**Ten.**	Chal.	Pos.	**Neg.**	Pos.E	Neg.E.	**Tired.**	Real.
DF	1	1	**1**	1	1	**1**	1	1	**1**	1
Residual	11	11	**11**	11	11	**11**	11	11	**11**	11
F-value	1.540	0.263	**4.632**	1.441	1.180	**6.369**	1.569	2.708	**7.188**	0.031
p-value	0.229	0.770	**0.017**	0.251	0.320	**0.005**	0.223	0.081	**0.003**	0.970
η_p^2	0.085	0.016	**0.219**	0.080	0.067	**0.279**	0.087	0.141	**0.303**	0.002

Table 5.3: Results of the one-way ANOVAs investigating the effects of DELAY on the different subscale of the Game Experience Questionnaire (IJsselsteijn et al., 2013). Significant results are displayed in bold. We found significant differences between Tension (Ten.), Negative Affect (Neg.) and Tiredness (Tired.). Participants playing with our presented latency compensation method associated the game with significantly less negative feelings and were significantly less tired after the gaming session.

d_{cohen} = 0.235). However, no significant difference between *No Latency* and *Prediction* (p_{tukey} = 0.991, d_{cohen} = 0.051). Furthermore, ANOVA revealed a significant main effect of DELAY on Tiredness. Post-hoc tests showed significant differences between *Latency* and *No Latency* (p_{tukey} = 0.005, d_{cohen} = 0.393) and between *Latency* and *Prediction* (p_{tukey} = 0.013, d_{cohen} = 0.281). However, no significant difference between *No Latency* and *Prediction* (p_{tukey} = 0.991, d_{cohen} = 0.096).

Overall, when using our predictive system participants had a lower Negative Affect and Tiredness compared to playing with 50 ms of latency and no support by the ANN. Table 5.3 shows all ANOVAs performed on the GEQ data - significant results are highlighted. Figure 5.10 shows the scores for the subscale Negative Affect (left) and Tiredness (right).

One-way ANOVA found no significant main effect of DELAY on *Score* (F(1, 11) = 0.914, p = 0.411, η_p^2 = 0.053). Participants on average achieved 10,042.42 points ± 1204,5 points.

5.2.5 Discussion

Our results show that using a deep learning-based predictive system to compensate for latency in commercial RTS games significantly reduces the negative effects of latency on the players' GX. Gamers playing Age of Empire 2 with a latency of 50 ms supported by our system associated the game with a significantly lower negative affect and experienced significantly lower tiredness than playing without the ANN. The objective

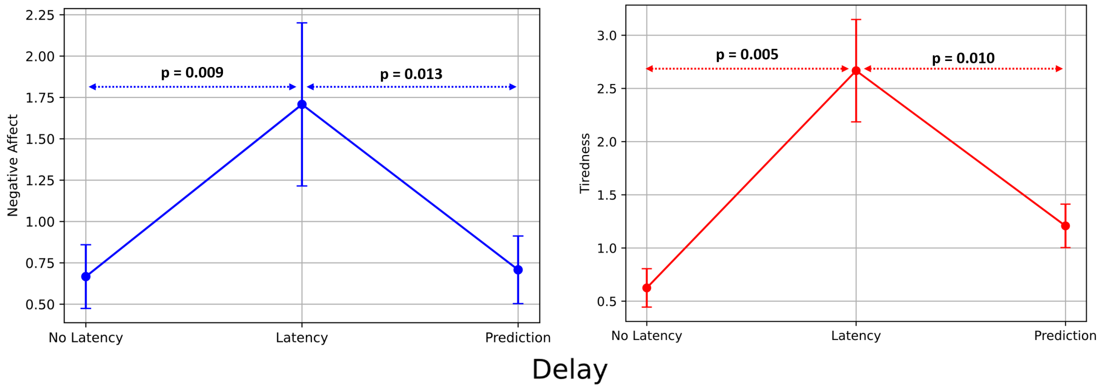


Figure 5.10: Shows the subscale Negative Affect (left) and Tiredness (right) of the Game Experience Questionnaire IJsselsteijn et al., 2013. Significant differences are highlighted via p-bars. Error bars depict the standard error. Participants rated the negative affect associated with the game significantly highest when playing with latency and no prediction. In the same manner, participants rated the gaming session with latency and no prediction with most tiring. We found no significant difference between playing with no latency and our prediction, hence our system did not negatively influence the game experience.

game performance remained constant in all tested conditions. Thus, our analysis shows that our system removes the negative effects of latency without introducing negative secondary effects.

In the following, we first explain and discuss the found effects based on previous work investigating the negative influence of latency on GX. We then discuss the implications of our findings and our work for game developers, researchers, and cloud-based game streaming providers.

5.2.5.1 Tension, Negative Affect, and Tiredness

Generally, previous work showed that latency negatively affects the GX of video games (Dabrowski et al., 2014). Liu et al. (2021), for example, showed that latency starting at 25 ms leads to an experience degradation. The authors showed that this decrease in QoE linearly correlates with the amount of latency - the higher the latency, the more pronounced the effects on the GX. In our work, we successfully compensated for the negative effects of controlled latency without negatively influencing the positive aspects of the gaming experience. We found a significant decrease in negative affect when playing with our ANN compensating latency compared to playing with latency and no prediction. A gamer's negative affect is a manifestation of their experienced negative emotions, such as anger, fear, and disgust while playing the game. We report similar effects of latency compensation based on deep learning on the GX in Section 5.1. Interestingly, in the previous section, contrary to the findings in this section, we found an increase in the positive affect experienced while playing with the compensation method instead of a decrease in the negative affect. This difference could be due to the different amounts of latency induced. While we used a baseline latency of 180 ms artificially added latency in Section 5.1, we evaluated our system during gameplay with 50 ms of latency in this section. Comparing both works indicates the evolution of latency-based effects. While compensating for high latency (180 ms) increased positive feelings, compensating for lower latency (50 ms) decreases negative feelings associated with and triggered by the gaming session. Both approaches, reducing low latency and high latency, are beneficial to optimizing the overall GX. A direct comparison suggests that compensating for different latency levels might improve different aspects of the GX.

Our analysis also showed that the tiredness induced by the gaming session was significantly lower when playing with our latency compensation ANN compared to playing without it. A gamer's tiredness after a gaming session indicates how exhaustive the session was. The exhaustion in gaming may be due to multiple reasons. One possible reason for a high level of exhaustion is a cognitively demanding gaming session, for example, if gamers have to simultaneously observe, control, and manage various game resources such as units in an RTS game (Chen et al., 2015). However, since we did not find a significant difference between playing without latency and playing with our system compensating latency, it is unlikely that the game itself induced a high level of tiredness. On the contrary, as we only found a significant increase in gamers playing with latency and no compensation, we conclude that latency in video games directly induces a higher level of exhaustion. The resulting de-synchronization induced by the latency between input and visual confirmation increased the cognitive demand of all in-game tasks. The absence of a responsive confirmation of a performed action led to that performed actions had to be observed and controlled for a more extended period. Thus, ultimately, latency induced higher tiredness.

We found that the tension experienced while playing with latency was significantly higher compared to playing without latency. This shows that we successfully induced latency in the game - hence, the increased tension. We did, however, not find a decrease in experienced tension when playing with our ANN's prediction. However, we also did not find a significant difference between playing with no latency and playing with the support of our ANN. Our ANN could not fully compensate for the negative effects of latency on the perceived tension. It, however, was able to lower the effects to the point that they are no longer statistically distinguishable from playing with no latency.

In summary, as prior work did (Claypool & Claypool, 2006; Claypool & Finkel, 2014; Liu et al., 2021c), our work shows that latency negatively influences the GX. Additionally, our work presents a solution to the latency problem in cloud-based game streaming - deep learning-based latency compensation. Furthermore, a comparison with the work of Section 5.1 indicates that compensating for different levels of latency might improve different aspects of gamers' experiences.

5.2.5.2 Implication of our Findings

Game developers should be aware of the different effects of latency and latency compensation techniques on gamers. Especially potential dissimilarities when compensating for high latency compared to a lower latency may be relevant in the game optimization process. Reducing a high latency increases the positive affect. Thus, it increases the fun and the number of positive emotions associated with the game. Reducing a low latency leads to the decrease of negative affect. Game developers can utilize this knowledge to improve games designed for cloud-based game streaming in the early development stages. Developers can focus on decreasing high latency early on, knowing that any negative association in play testing may be attributed to a remaining latency. Decreasing only high latency early on in the development allows for saving resources that can be utilized in other development areas. In later optimization stages, the remaining low latency may then be reduced to optimize and finalize the GX. Furthermore, game developers should be aware of the different latency conditions when simultaneously developing a game for conventional local gaming systems and cloud-based game streaming platforms.

Researchers may also benefit from our work. We showed that deep learning-based latency compensation techniques are well equipped to reduce negative effects induced by latency in commercial slow-paced video games. Furthermore, we showed that the different latency levels, and thus their compensation, affect gamers differently. Researchers can build on our work to further investigate the effects of latency compensation on GX in greater detail. We showed that deep learning-based latency compensation could reduce the negative effects of latency in AoE2. However, since we used the same game to train and evaluate our ANN, it remains unknown if ANNs can generalize across different games. We assume that by using a large enough data set of gameplay data of multiple games and genres, deep learning-based models could generalize over all games. Consequently, researchers should investigate the generalizability of deep learning-based latency compensation techniques in video games. The findings of our work and the presented method are also relevant to researchers outside of video games. Although we used game-specific parameters to train our system, the presented approach is potentially suitable for any software operated by a mouse and keyboard. For example, by integrating mouse prediction, any software could achieve higher responsiveness, thus enhancing the overall UX.

Finally, our findings and the method we used to compensate for latency are also relevant to cloud-based game streaming providers. Streaming providers need to continue to improve and upgrade their server infrastructure and their method of compressing and delivering their content. Further optimizing latency conditions in cloud-based game streaming is essential to alleviate it to the latency level of conventional gaming systems. To do so, providers can use the latency compensation technique presented in this section. Our method may be used for any type of game. In reducing the inherent latency of game streaming, providers can offer gamers a gaming platform with the same GX and performance potential as local gaming setups.

5.2.5.3 Limitations and Future Work

In our work, we did not find an increased accuracy when using visual material for the mouse prediction. However, the lower accuracy might be due to the small number of epochs trained. One can assume that an increased time spent on training increases the accuracy of the CNN and the mixed model. Both models have not yet converge to an optimization minimum in our evaluation. Future work, thus, should build on our work and further investigate the use of CNN and visual material for latency compensation techniques. Furthermore, the lack of increased prediction accuracy using images may be due to the image resolution used for training. In our work, we scaled the training images down to a resolution of 84 pixels by 48 pixels. Down-scaling was necessary to train the CNN in a reasonable amount of time. Nevertheless, even with down-scaled images, training one epoch took approximately 36 hours. Future work, thus, should investigate if it is a feasible approach to increase computational power for training.

5.2.6 Conclusion

This work presents a novel approach to compensate for latency in commercial video games by predicting the mouse position in 50 ms in Age of Empire 2 using ANNs. In contrast to previous work, we do not require the internal game state or to modify the game. Consequently, our approach can compensate for the latency of unmodified commercial games to reduce the negative effects on the GX induced by latency.

5.3 Artificial Neural Networks in Fast-Paced Games (Study IX)

In the previous two sections, we investigated if ANN-based latency compensation can be used in custom and slow-paced commercial video games. In the first study, we used game-internal information to predict a future game state. In the second study, we did not rely on game-internal information and used the user's raw mouse input to predict a future mouse position. Both types of systems allow the game to calculate a reaction before the actual player input happens, which effectively reduces the perceived latency by the players. We showed that both systems are able to reduce the negative effects induced by latency (RQ7 and RQ8).

However, while the approaches presented so far reduce latency's influence on PP and GX, both presented studies have a shared limitation: They predict fixed latency values. In other words, their adaptability to latency is limited. But, as we learned in Section 2.1, 3.1, and 3.2, latency in the wild is never constant. It varies depending on a multitude of factors, such as the equipment used, the Internet connection, load-balancing techniques by game providers, and much more. Thus, a latency compensation technique with high practical applicability needs to be able to account for a changing latency environment. In the previous two sections, we showed that ANN-based compensation techniques are well-suited to compensate for constant latency. Currently, however, it is unclear if an ANN-based compensation system can adaptively reduce the negative effects of latency on PP and GX in commercial, fast-paced video games (RQ9).

This section will be part of the following manuscript in preparation for submission:

Halbhuber, D., Wolff, C., Seewald, M., Schwind, V., & Henze, N. (n.d.[b]). "Leveling the Playing Field: Adaptive Latency Compensation in First-Person Shooters using Artificial Neural Networks." In: *Currently in preparation*.

5.3.1 Background and Research Rationale

This section highlights how to compensate for varying latency in a commercial, fast-paced FPS video game. We refined the approach presented in Section 5.2 and trained an ANN to predict a future player input by using only the information provided to the player at run-time. Contrary to the system presented in Section 5.1, the ANN in this section does not rely on game-internal data to predict future player inputs. To achieve this, we collected data from 15 participants playing CS:GO. While playing, we recorded computer mouse movements, keyboard inputs, and videos from the gameplay. We used the gathered data to train different ANNs. The first ANN predicts the mouse position and keyboard input in the future (similar to the system presented in Section 5.2). The second ANN is based on the object-detection framework Yolo (Redmon et al., 2016; Jocher et al., 2021) and trained on detecting enemies in the gameplay of CS:GO players (Roboflow, 2022). This information is used to better contextualize the player's current movement and behavior. Potentially, enriching the mouse movement and input prediction model with context increases its prediction accuracy and power. Previous work, reported in Section 5.1 and 5.2, showed contextual information is not necessarily needed for an ANN to be able to effectively compensate for latency in video games. However, we hypothesize that providing contextual information, such as information about currenlty visible enemies, potentially improves the prediction accuracy of an ANN-based latency compensation system even further.

We tested the final system in a study with 25 players. Contrary to previous studies, we simulated real-world fluctuating latency not by software-sided input buffering but by using an external Arduino microcontroller. This was necessary since running inference for two ANNs and running a graphically demanding video game simultaneously with a input buffering script generated too much computational load. This high computational load led to frame stuttering in the game and would have biased our study. Our results show that using adaptive latency compensation systems based on ANNs significantly increased PP and GX (RQ9). Players reached higher scores, were more accurate, and had an overall better experience when playing with our compensation system compared to playing with no compensation.

5.3.2 Data Collection and Apparatus

Several steps were necessary to develop the apparatus for our investigation. First, we developed a pipeline for detecting enemy movement in live gameplay. The enemy's position is parameterized and exported to be used in further training of the prediction model. Second, we conducted a data collection study in which we gathered mouse movement, input events, and screen recordings from players playing CS:GO. Then, we used the pipeline developed in the first step to identify enemies in the recorded gameplay videos. All this data, mouse movement, keyboard events, and the gameplay videos with marked enemies are then used to test and evaluate different ANN architecture to identify the ANN type best suited for latency compensation.

5.3.2.1 Enemy Detection Pipeline

We developed an enemy detection pipeline using the Yolo framework (Redmon et al., 2016; Jocher et al., 2021). The goal of the pipeline is to detect enemy movement and feed the enemy's position to the predictive system. This additional information about enemies provides a context and thus potentially increases the predictive model's accuracy. In other words, the predictive model should learn what movement and behavior are likely to happen in dependence on the information received from the detection pipeline.

Yolo is a single-shot, real-time object detection framework based on neural networks. Generally speaking, its goal is to localize a region of interest within an image and classify this region in accordance with the user's need. After inference, Yolo provides the user with bounding boxes and coordinates (X and Y) for all objects detected. Figure 5.11 shows an exemplary output from a Yolo prediction. The figure shows different images augmented with bounding boxes highlighting the detected objects.

Out of the box, Yolo can reliably detect and classify the 20 standardized Pascal object classes (Everingham et al., 2006; Everingham et al., 2010), such as persons, birds, cats, airplanes, bottles, chairs, and more (for a full list see Everingham et al., (2006)). However, Yolo's real power comes from its ability to be individualized via fine-tuning (Quinn et al., 2019). Fine-tuning is an approach in which the weights (the trained parameters) of a pre-trained model (in our case, baseline Yolo v5) are transferred to a new model, which is

Figure 5.11: Shows an exemplary output from the Yolo pipeline. The left image shows a street with different vehicles and people in the air. The right image shows a snippet from a popular movie showing a person pushing a boat into the ocean. Yolo correctly provided labels and colored bounding boxes for the right images but failed to identify the person in the left image correctly. Figure adapted from Everingham et al., (2006)

then trained on new data. The new model utilizes the pre-trained weights and thus is able to kick-start its own training process since it does not have to learn all the semantics of images. Furthermore, fine-tuning drastically reduces the required amount of image data needed for training. Instead of needing several hundred thousand images of a class (for example, of dogs), a few thousand images are sufficient to reliably detect and classify a class in an image. In our study, we utilized Yolo's ability to be fine-tuned to detect enemies in CS:GO. We fine-tuned the YoloV5 (Jocher et al., 2021) architecture on a public dataset of CS:GO. The dataset contains 6 370 images of the playable characters in CS:GO. In CS:GO, teams fight each other. One team is called Terrorists, and the other opposing team is called Counter-Terrorists. Our Yolo implementation was trained on 5939 images from the dataset (431 images validation set) and detects the probability that a character on screen is part of the Terrorists or Counter-Terrorist team, draws a bounding box around the character, and outputs the coordinates of the bounding box to separate text file. Figure 5.12 shows an example output from our Yolo implementation. The figure shows the player's view, a Terrorist teammate, and the information provided by Yolo (bounding box, probability of team affiliation).

Figure 5.12: Shows a screenshot from CS:GO gameplay with an example output from our Yolo implementation. The figure shows the player's view, a terrorist teammate, and the information provided by Yolo (bounding box, probability of team affiliation).

5.3.2.2 Data Collection Study

After developing and testing the enemy detection pipeline, we conducted a data collection study to gather natural data from players playing CS:GO. This data is necessary for training the final latency compensation ANN.

Apparatus

For the data collection study, we installed CS:GO on a stationary workstation (Intel i9-9900k, 16 GB RAM, Nvidia RTX 2060 6 GB VRAM) in our laboratory. The workstation was attached to a monitor (24" FullHD @60Hz), a computer mouse (Razer DeathAdder V2), and a wired headset. The laboratory was quiet and free of external disturbance. The game was executed in fullscreen mode. To gather the required data, we used a combination of microcontrollers and Python scripts. An Arduino intercepted all keyboard and mouse inputs, recorded them, and passed them on to the workstation. To capture the screen while playing, we used Python's D3Dshot (Brochu, n.d.) and Open-CV (OpenCV Team, 2022) library.

Procedure and Task

Participants were greeted at our institution's laboratory by the experimenter. After giving informed consent and agreeing to the data collection, participants were seated at the workstation running CS:GO. Participants were informed about the exact purpose of the study (gathering data for ANN training). Next, all participants played one round of CS:GO's team deathmatch mode for 10 minutes. Upon finishing the round, we collected demographics and information about past gaming experiences in video games generally and CS:GO specifically. Lastly, participants were debriefed. Participation in the study took about 25 minutes. The study and the data collection received ethical clearance as per the research policy of the University of Regensburg.

Participants

Using our institution's mailing list, we invited 15 participants (5 f, 10 m). The participants' mean age was 24.13 years (*SD* = 3.52 years) and ranged from 18 to 31 years. Participants were chosen blind to age and gender. Participants reported that they play 12 hours (*SD* = 4.88 years) weekly of video games on average, while they only play 6.2 hours (*SD* = 5.17 hours) of FPS games. All participants were familiar with CS:GO and played between 2 and 2 500 hours with an average playtime of 587.47 hours (*SD* = 7.42 hours). All participants were students at our institution.

5.3.2.3 Development of the Artificial Neural Networks

Overall, we collected 1 321 975 images and 592 500 mouse and keyboard actions in our data collection study. We used this data to develop, train, and test three deep learning-based ANNs: (1) a feed-forward network, (2) a long short-term memory (LSTM) network (Yu et al., 2019) and (3) an ANN based on the encoder-decoder architecture (Vaswani et al., 2017; Badrinarayanan et al., 2017). The goal of the ANNs was to predict player movements and actions. However, contrary to the ANNs presented in Section 5.1 and 5.2, we also parameterized latency. To introduce latency to the training data, we randomly offset the timestamp of consecutive data frames. We offset the timestamp randomly between 50 ms to 150 ms to simulate a latency variation between 50 ms and 150 ms. To mimic a naturalistic latency behaviour, we limited the difference of latency between two consecutive frames to 5 ms.

Since modern games typically are executed with 60 frames per second, timely inference by the ANNs was paramount. Considering 60 frames per second, the latency compensation model had a time frame of 16 ms to generate the next prediction. However, since detecting enemies using Yolo takes about 6 ms per prediction, the available time frame reduces down to no longer than 9 ms.

Data Pre-Processing

All 1321975 images were preprocessed using our enemy detection pipeline and merged with the recorded mouse and keyboard actions. The whole dataset was interpolated to get one data point every 5 ms. Since we used a supervised deep learning approach, we defined the input and the output of the latency compensation system. The final training input consists of the following data columns: (1) timestamp, (2)-(4) X- and Y-coordinates of the bounding box of the enemy (if applicable, 0/0 if not) generated by Yolo, (5) X-coordinate of current mouse position, (6) Y-coordinate of current mouse position, (7) status of left mouse button (up or down), (8) status of middle mouse button (up or down), (9) status of right mouse button (up or down), (10) current keys pressed and (11) current latency. We used the mean absolute error (MAE) between the actual mouse position and the predicted mouse position to optimize our models in training. We used TensorFlow (Abadi et al., 2016) to train all models and Optuna (Akiba et al., 2019) for hyperparameter optimization and architecture selection. Using Optuna, we systematically acquired hyperparameters, such as learning rate, batch size, and prediction baseline (how much data the ANN should use for one inference), for each model individually. All models were trained on a local machine with an Intel i7-13700K, 32 GB RAM, and one NVIDIA RTX 3070 TI with 8GB VRAM. We used a train, validation, and test split of 80/10/10. We report the final performance for each developed model in the following.

Feed-Forward Network

The classical feed-forward network uses the last nine mouse positions, button presses, and boundary boxes of the enemy AI to predict the player's next action. Since we interpolated the whole dataset to 5 ms steps, nine entries represent the last 45 ms of actions. Therefore, the input layer consists of 91 neurons. Overall, the network consists of six fully connected layers and two dropout layers (dropout rate = 0.2). Figure 5.13 depicts the architectural structure of the network. The output layer consists of 2 neurons, which serve to predict the future change of the X and Y movement of the mouse. We used different activation functions for different layers: L_2 uses ReLU, L_3 uses Sigmoid, L_5 uses ReLU, and L_6 uses linear regression. For backpropagation, we used ADAM

(Kingma & Ba, 2014) with a batch size of 128 and a learning rate of 0.001. We also used a custom callback function to reduce the learning rate if necessary. The model was trained for 50 epochs and achieved an MAE of 4.4 pixels per mouse movement. Generating one prediction takes about 3.7 ms, which is less than 9 ms and, thus, fulfills our requirement for timely inference.

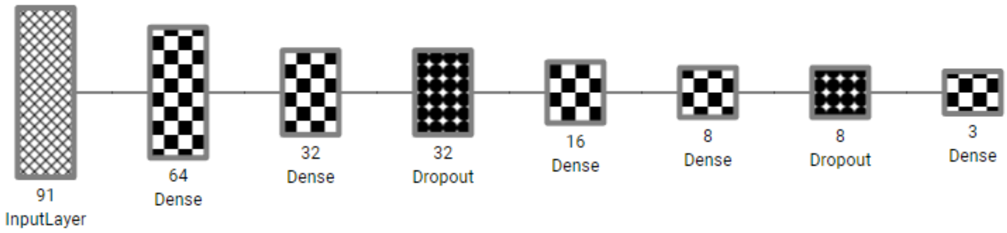


Figure 5.13: Shows the architectural structure of the feed-forward network. The figure also provides the names and number of neurons for all layers.

Long Short-Term Memory Network

Our second approach focused on the use of LSTM models since previous work showed that it is possible to predict the mouse position with the usage of LSTM models (Wei et al., 2021). We used the enemy AI boundary boxes, mouse position, button presses, and the current random latency as input for this network as a prediction baseline. Optuna suggested using the last 20 entries as time series input for the LSTM. Thus, the LSTM's input shape was given as (20,10). The input layer (L_1) is connected to an LSTM layer (L_2) with 32 neurons, which is connected to three fully connected stacked layers (L_3 - L_5). L_3 and L_4 use ReLu and linear regression as activation functions, and the last fully-connected layer (L_5) serves as the output layer with 2 neurons, which provides the same output as the feed-forward network. Figure 5.14 shows the architectural structure of this model. For backpropagation, we used ADAM (Kingma & Ba, 2014) with a batch size of 128 and a learning rate of 0.01. Again, custom callback functions were used to handle local optimization minima. The model achieved an MAE of 5.2 pixels per mouse movement. However, loss never full stabilized, and, thus, can not be considered optimized. The LSTM model achieved an inference time of 6.7 ms.

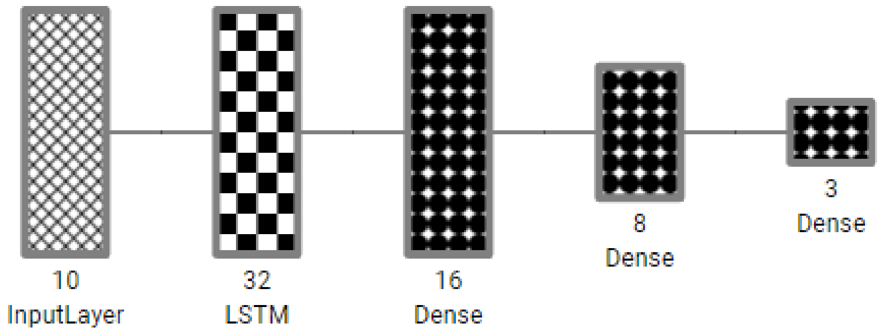


Figure 5.14: Shows the architectural structure of the LSTM Network. The figure also provides the names and number of neurons for all layers.

Encoder-Decoder Network

Based on previous work (Du et al., 2020; Yuan et al., 2020), we also investigated the use of an attention-based LSTM encoder-decoder network for latency compensation. Again, we used the enemy boundary boxes, mouse position, button presses, and the current random delay as input. Optuna suggests using the last ten data points as a prediction baseline. Thus, the input layer had a shape of (10,10). Figure 5.15 shows the network's architectural structure. The input layer (L_1) is followed by a one-dimensional convolutional layer L_2 (filters = 64, kernel size 2) and another one-dimensional convolutional layer L_3 (filters = 64, kernel size 6). Both (L_2 and L_3) use ReLU as an activation function. The convolution layers are followed by a one-dimensional max pooling layer L_4 (pool size = 2), which is connected to a flattened layer L_5. We used a repeat vector (10 times repetition) to get the output in shape for the following LSTM layer L_6 (neurons = 200) with ReLU as the activation function. The last two layers are fully connected layers, L_7 with 100 neurons and ReLU as activation function and L_8 as output layer with 2 output neurons. For backpropagation, we used ADAM with a batch size of 128 and a learning rate of 0.1. Again, we applied custom callback functions to deal with local optimization minima. The model achieved an MAE of 5.3 pixels per mouse movement and it took about 8.1 seconds to generated one inference, which is to slow to be used in a a high fps game loop.

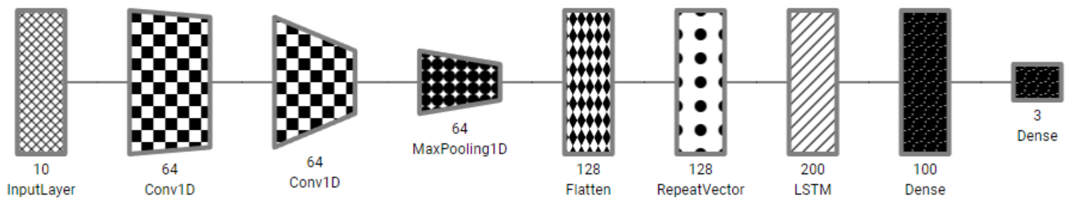


Figure 5.15: Shows the architectural structure of the decoder-encoder architecture. The figure also provides the names and number of neurons for all layers.

Final Implementation

Since the feed-forward network achieved the lowest MAE and got the fastest inference time, we chose to use it as the final prediction architecture. We used Python to integrate our model into the OS's input-output pipeline to manipulate the mouse and keyboard positions. We interpolated over the last 100 ms of user actions with 5 ms steps and fed the last 45 ms as input into the network. Upon receiving new user actions, we removed the last player action from the array and restarted the whole process to continuously predict mouse movements and button presses. This final implementation is similar to the flow diagram presented in Figure 5.4. Furthermore, since our model's prediction overrides raw hardware values on the OS level, an application running on the workstation cannot differentiate between raw mouse input or input generated by our predictive system. In the potential game, our prediction is presented as if the user moved the mouse as usual.

5.3.3 Method

To investigate if a ANN-based latency compensation system can adaptively compensate for the negative effects of latency in commercial video games, we conducted a study. In the study we predict the players' mouse movements to counteract latency. We created a naturalistic latency environment by using an Arduino, which randomly offsets player inputs via input buffering.

5.3.3.1 Apparatus

We used the same workstation as in our data collection study for this study. Similarly, CS:GO was executed in full-screen mode. The laboratory was quiet and free of external disturbances. To introduce latency to our apparatus, we used an Arduino Leonard (Arduino, n.d.), which uses an ATmega32u4 microcontroller. Coupled with a

USB host-shield (Nerpa Tech, n.d.), this microcontroller allows the Arduino to act as an input device. The Arduino is interjected between the input devices (mouse and keyboard) and the workstation. It receives the input events, delays them by a predefined latency, and forwards the final delayed input to the workstation. To mimic naturalistic behaviour, the Arduino does not create a constant but varying latency for each input event (mouse and keyboard). Latency values are randomly created between 0 ms and 150 ms. However, to prevent a too steep latency variation between changes, the Arduino changes latency step-wise and is limited to an increase or decrease of 5 ms between consecutive values. To allow our latency compensation system to parameterize the current random delay in its prediction, the Arduino propagates the currently applied level of latency via a public interface. Via this interface, the ANN is informed about the current level of latency. Figure 5.16 shows an exemplary latency progression generated by our Arduino.

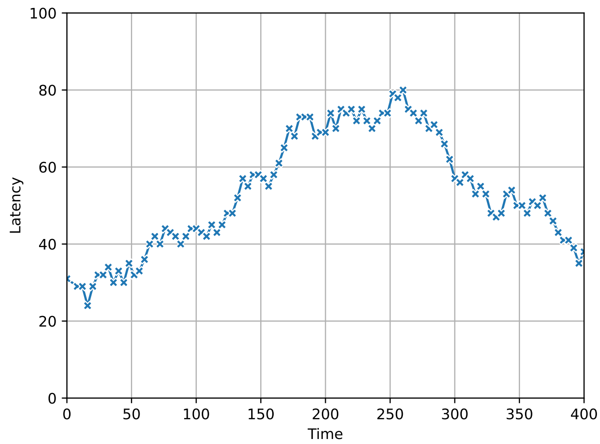


Figure 5.16: Shows an exemplary latency progression generated by the Arduino used in our study. In this figure, the Arduino generated a random latency between 0 ms and 100 ms. The difference of latency between two changes is limited to 5 ms, to prevent the Arduino from generating to steep changes. Please note that values are smoothed for presentation purposes.

5.3.3.2 Study Design

To control for the effect of latency and our prediction system, we used a single IV Latency Condition. Latency Condition has five levels: (I) *No Delay*, (II) *Delay 50 ms - 100 ms*, (III) *Prediction 50 ms - 100 ms*, (IV) *Delay 100 ms - 150 ms*, and (V) *Prediction 100 ms - 150 ms*. In the *No Delay* condition, no delay and no prediction are applied, and participants play with the unaltered game. In the delay conditions (2 and 3), the respective amount of delay is randomly added each second. In the prediction conditions (3 and 5), the respective amount of delay is added, and additionally, our latency compensation system aims to compensate for the added amount of delay. Each participant played with all five conditions. The order of conditions was balanced using a Latin square to prevent sequence effects.

To measure the players' performances we used four DVs: (1) *Kills* - quantifies how many enemies were eliminated by the player (more is better), (2) *Assists* - quantifies how many enemies the player dealt a substantial amount of damage, but did not get the kill (more is better), (3) *Headshots* - quantifies how many enemies the player eliminated via a headshot (more is better), and (4) *Deaths* - quantifies how many times the player died (less is better).

We quantify the subjective gaming experience using the miniPXI questionnaire (Haider et al., 2022), with its ten subscales: Audiovisual Appeal, Challenge, Ease of Control, Clarity of Goals, Progress Feedback, Autonomy, Curiosity, Immersion, Mastery, and Meaning. Additionally, we asked two further questions about the participants' opinions regarding their enjoyment and the current effects of latency in the game. Participants were asked how much they enjoyed the last round and how much influence they believed latency had on their performance and experience. All questions were answered on a 7-point Likkert item ranging from 0 to 7.

5.3.3.3 Procedure and Task

Upon arriving at the laboratory, participants were informed about the general framework of the study and were presented with a consent form. After giving informed consent and filling out a demographic questionnaire, they started playing their first round of CS:GO. Participants were not aware of the exact purpose of the study (testing ANN-based latency compensation methods) but were only told they would play five rounds of CS:GO with varying delay conditions. Participants played 7 minutes of Deathmatch using only a single weapon. The participants' goal in each round was to obtain as high a score as possible (by obtaining *Kills*, *Assists*, and *Headshots*). After each round, the browser automatically opened and presented the players with our questionnaire. After all five rounds, participants were debriefed and thanked for their attendance. We estimated a total of 45 minutes for the whole study. The study and its procedure received ethical clearance as per the ethics policy of the University of Regensburg.

5.3.3.4 Participants

We recruited a total of 25 participants (5 female, 20 male) for the study. Participants who took part in our data collection study could not participate in this study. The participants' mean age was 23.98 years (*SD* = 3.53 years), ranging from 18 to 33 years. They played on average 14.92 hours (*SD* = 7.86 hours) of video games each week, while on average 8.08 hours (*SD* = 7,43 hours) were spent playing FPS games. On average every attendee played 705.2 hours (*SD* = 912,13 hours) of CS:GO, ranging from 0 to 3500 hours.

5.3.4 Results

Twenty-five participants toke part in the study. Each of them played five rounds of CS:GO, with each round lasting for seven minutes. Thus, we collected 125 questionnaires results and 125 measurements of *Kills*, *Assists*, *Headshots* and *Deaths*. In the next section, we first report the statistical analysis of the experience data, before reporting the analysis of the performance measures. Since all data follows a normal distribution (Shaprio-Wilk all $p > 0.05$), we used a oneway RM-ANOVA with Bonferroni-Holm correction (Holm, 1979) to analyze the data. To increase readability, we summarize post-hoc tests and only report significant results.

5.3.4.1 Mini Player Experience Inventory, Enjoyment and Subjective Latency

Table 5.4 shows the descriptive results of the miniPXI and the questions regarding the participants' enjoyment, as well as the results to the subjective effects of latency.

ANOVA showed significant differences effect of LATENCY CONDITION on Challenge ($F(1,24) = 27.99$, $p < 0.001$, $\eta^2 = 0.54$) , Control ($F(1,24) = 4.92$, $p < 0.001$, $\eta^2 = 0.17$), Autonomy ($F(1,24) = 7.75$, $p < 0.001$, $\eta^2 = 0.24$), Curiosity ($F(1,24) = 25.19$, $p < 0.001$, $\eta^2 = 0.51$), Immersion ($F(1,24) = 43.58$, $p < 0.001$, $\eta^2 = 0.65$), Mastery ($F(1,24) = 37.59$, $p < 0.001$, $\eta^2 = 0.61$), Meaning ($F(1,24) = 35.47$, $p < 0.001$, $\eta^2 = 0.59$), Enjoyment ($F(1,24) = 64.98$ $p < 0.001$, $\eta^2 = 0.73$) and latency ($F(1,24) = 101.51$, $p < 0.001$, $\eta^2 = 0.81$).

Challenge

Next, we conducted post-hoc tests for each significant dimension. Starting with the Challenge dimension, post-hoc tests showed significant differences between the conditions *No Delay* and *Delay 100 ms-150 ms* (adjusted $p < 0.001$), *Delay 50 ms-100 ms* and *Delay*

Functional Constructs					
Condition	**Audiovisual Appeal**	**Challenge**	**Control**	**Goals**	**Feedback**
No Delay	5.72 ± 1.06	4.88 ± 1.76	5.68 ± 0.85	6.20 ± 0.82	5.80 ± 1.04
Delay 50 ms-100 ms	5.68 ± 1.18	4.28 ± 1.28	5.84 ± 0.90	6.36 ± 0.81	5.92 ± 0.86
Prediction 50 ms-100 ms	5.76 ± 0.93	4.60 ± 1.28	6.08 ± 0.64	6.32 ± 0.63	6.12 ± 0.88
Delay 100 ms-150 ms	5.44 ± 1.53	2.36 ± 1.47	5.76 ± 0.72	6.32 ± 0.69	6.08 ± 0.76
Prediction 100 ms-150 ms	5.72 ± 1.02	3.96 ± 1.06	6.12 ± 0.67	6.34 ± 0.66	6.04 ± 0.84

Psychosocial Constructs					
Condition	**Autonomy**	**Curiosity**	**Immersion**	**Mastery**	**Meaning**
No Delay	4.56 ± 1.12	4.64 ± 1.41	5.96 ± 1.06	5.48 ± 1.23	5.32 ± 0.99
Delay 50 ms-100 ms	3.52 ± 1.16	3.00 ± 1.04	3.64 ± 1.25	2.96 ± 1.10	2.88 ± 0.93
Prediction 50 ms-100 ms	4.08 ± 1.08	4.04 ± 1.24	5.16 ± 1.07	5.04 ± 1.06	4.88 ± 1.13
Delay 100 ms-150 ms	2.84 ± 1.07	2.08 ± 1.08	2.52 ± 1.16	2.24 ± 1.23	2.32 ± 1.14
Prediction 100 ms-150 ms	3.76 ± 1.20	3.76 ± 1.20	4.20 ± 0.96	4.20 ± 1.12	4.32 ± 1.31

Other Constructs		
Condition	**Enjoyment**	**Latency**
No Delay	5.96 ± 0.93	1.04 ± 0.20
Delay 50 ms-100 ms	2.72 ± 0.98	4.40 ± 0.96
Prediction 50 ms-100 ms	5.12 ± 0.93	2.64 ± 0.91
Delay 100 ms-150 ms	2.16 ± 1.11	5.84 ± 1.14
Prediction 100 ms-150 ms	4.64 ± 1.08	3.32 ± 1.07

Table 5.4: Shows the mean and standard deviation of each category of the mini PXI (Haider et al., 2022) combined with the question regarding enjoyment and latency for the different conditions of the study.

100 ms-150 ms (adjusted $p < 0.001$), *Prediction 50 ms-100 ms* and *Delay 100 ms-150 ms* (adjusted $p < 0.001$), *Prediction 50 ms-100 ms* and *Prediction 100 ms-150 ms* (adjusted $p = 0.011$) and between *Delay 100 ms-150 ms* and *Prediction 100 ms-150 ms* (adjusted $p < 0.001$). All other pairwise comparison yielded no significant results (all adjusted $p > 0.058$). Overall, our results show that only in high latency scenarios (100 ms-150 ms) the Challenge score is significantly affected, however this is counteracted in the *Prediction 100 ms-150 ms* condition which overall produced a higher score than the *Delay 100 ms-150 ms* condition and only differs compared to its lower counterpart. Figure 5.17 shows the Challenge dimension for all conditions.

Control

In the Control category pairwise comparison showed significant differences between *No Delay* and *Prediction 50 ms-100 ms* (adjusted $p = 0.02$) and between *No Delay* and *Prediction 100 ms-150 ms* (adjusted $p = 0.022$). All other combinations showed no significant difference (all adjusted $p > 0.05$). Our prediction significantly increased the

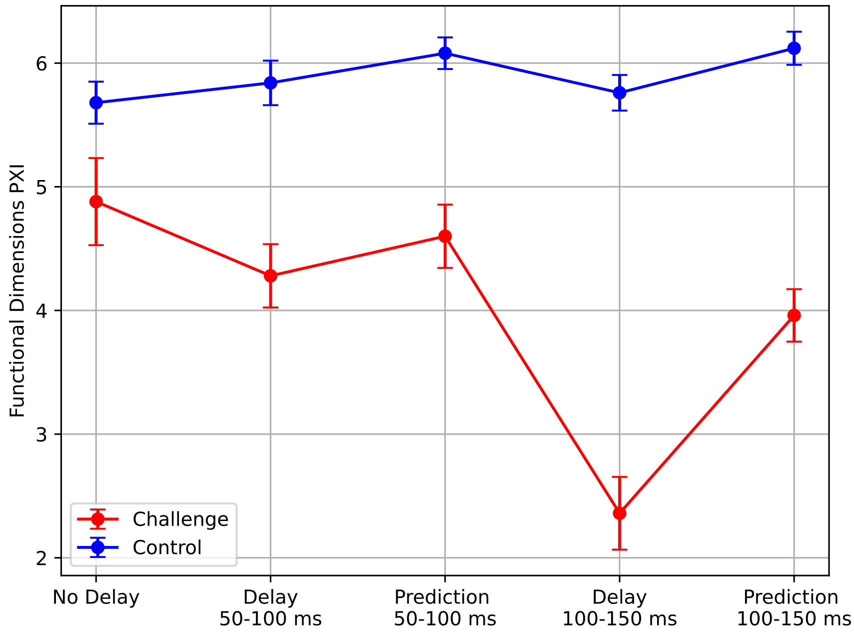


Figure 5.17: Shows the Challenge (red) and Control(blue) dimension of the Player Experience Inventory (Vanden Abeele et al., 2016). Overall, the data shows that players perceived the game's challenge to be significantly more appropriate when playing with no latency or with our ANN compensating for latency system. Furthermore, in the control section our data demonstrates that our ANN even increases the ease of controlling the game compared to the baseline condition with no delay. Errorbars show the standard error.

participants' feeling of control, while latency had no significant effect. Participants reported the highest feeling of control while playing with the highest level of prediction (*Prediction 50 ms-100 ms*). Figure 5.17 shows the data for the Control dimension for all conditions.

Autonomy

Next, we analyzed the effect of our latency compensation system on the Autonomy subscale. Post-hoc test revealed significant differences between *No Delay* and *Delay 50 ms-100 ms* (adjusted $p = 0.015$), *No Delay* and *Delay 100 ms-150 ms* (adjusted $p < 0.001$), *No Delay* and *Prediction 100 ms-150 ms* (adjusted $p = 0.047$) and *Prediction 50 ms-100 ms* and *Delay 100 ms-150 ms* (adjusted $p = 0.006$). All other pairwise comparison yielded no significance (all adjusted $p > 0.11$). Participants rated playing with no latency and no prediction with the highest feeling of autonomy. However, playing with any level of prediction resulted in higher scores than playing with latency and no prediction. Figure 5.18 (left) shows the data for Autonomy for all tested conditions.

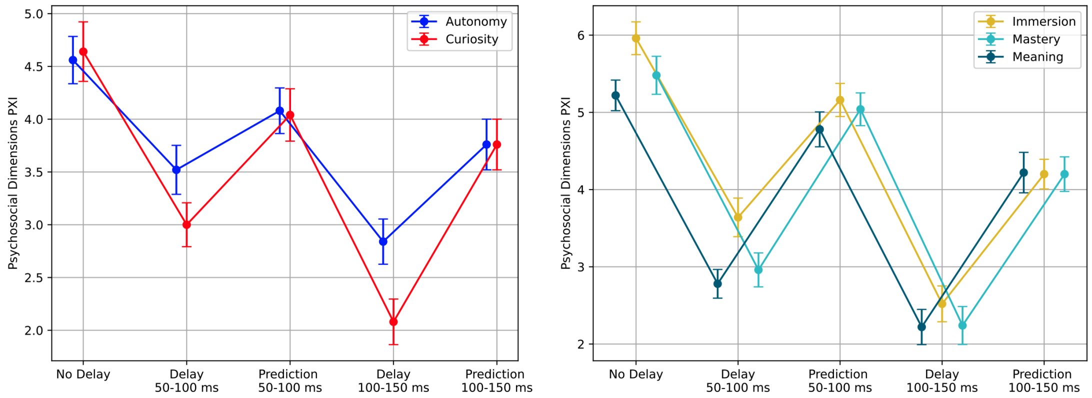


Figure 5.18: Shows the data for significant subscales of Player Experience Inventory's psychosocial dimensions. Overall, players achieved the best scores when playing with the control conditions with no delay and no prediction. However, the data consistently shows that our ANN can compensation for the negative effects induced by latency. Overall, players had a better gaming experience on a psychosocial level when playing with our predictive model compared to playing with no compensation and any level of latency. Errorbars show the standard error.

Curiosity

Post-hoc tests investigating Curiosity, found no significant differences between *No Delay* and *Prediction 50 ms-100 ms* (adjusted $p = 0.192$) and between *Prediction 50 ms-100 ms* and *Prediction 100 ms-150 ms* (adjusted $p = 0.347$). All other tests revealed significant differences (all adjusted $p < 0.011$). Overall, participants rated the prediction conditions higher than the delay conditions. Furthermore, the *No Delay* condition was rated with the significantly highest score. Figure 5.18 (left) shows the Curiosity data for all conditions.

Immersion

Next, we investigated the Immersion subscale via post-hoc testing. The pairwise comparison showed no difference between *Delay 50 ms-100 ms* and *Prediction 100 ms-150 ms* (adjusted $p = 0.053$). Every other comparison showed significant results (all adjusted $p < 0.01$). Participants rated the *No Delay* condition with the highest Immersion score. The prediction conditions generally scored higher than the delay conditions, except when comparing *Delay 50 ms-100 ms* (M = 3,64, SD = 1,25) and *Prediction 100 ms-150 ms* (M = 4,20. SD = 1,20), where we revealed no significant difference. Figure 5.18 (right) shows the Immersion data for all conditions.

Mastery

Post-hoc testing data from the Mastery subscale revealed no significant difference between *No Delay* and *Prediction 50 ms-100 ms* (adjusted $p = 0.17$) and *Delay 50 ms-100 ms* and *Delay 100 ms-150 ms* (adjusted $p = 0.062$). All other comparisons revealed significant differences (all adjusted $p < 0.01$). Generally, participants rated the conditions with no delay and with applied prediction with a significantly higher Mastery score compared to the latency conditions. Figure 5.18 (right) shows the Mastery data for all conditions.

Meaning

Investigating the pairwise difference of the Meaning subscale, we found no significant differences between *No Delay* and *Prediction 50 ms-100 ms* (adjusted $p = 0.179$), *Delay 50 ms-100 ms* and *Delay 100 ms-150 ms* (adjusted $p = 0.163$) and between *Prediction 50 ms-100 ms* and *Prediction 100 ms-150 ms* (adjusted $p = 0.179$). All other pairwise comparisons resulted in significant differences. Overall, participants rated the gaming session with the highest level of meaning when there was no delay. Figure 5.18 (right) shows the Meaning data for all conditions.

Enjoyment

Next, we analyzed the Enjoyment data. We found no significant differences between *Delay 50 ms-100 ms* and *Delay 100 ms-150 ms* (adjusted $p = 0.102$) and between *Prediction 50 ms-100 ms* and *Prediction 100 ms-150 ms* (adjusted $p = 0.102$). However, all other tests revealed significant differences (all adjusted $p < 0.01$). Overall, participants enjoyed playing the game the most, when not affected by latency. Figure 5.19 shows the Enjoyment data for all conditions.

Subjective Latency

Lastly, we analyzed the subjectively felt latency. We found significant differences between all tested condition (all adjusted $p < 0.05$). Generally, participants rated playing with no delay with the least amount of subjectively felt latency. Playing with delay led to a higher subjectively felt latency. However, introducing a latency compensation via prediction reduced the amount of subjectively felt latency again. While our prediction was not able to reduce the subjective impact of latency to a no delay level, we nevertheless significantly reduced latency's subjective influence on players. Figure 5.19 shows how players rated the subjectively experienced latency for all conditions.

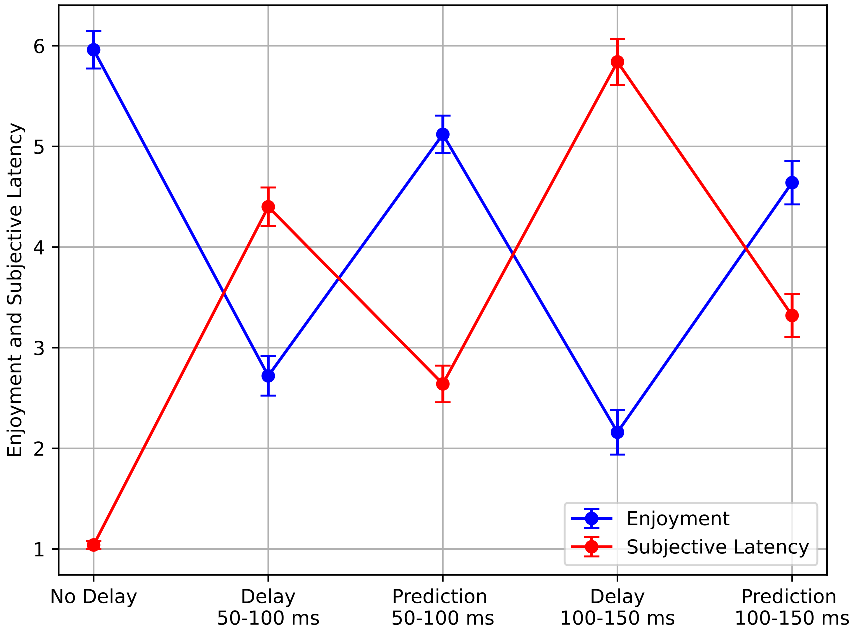


Figure 5.19: Shows the data of the additional questions asked regarding the player's enjoyment and subjective feeling of latency's impact on their gaming session. Overall players enjoyed the game most, when playing without latency and compensation system. However, when comparing to playing with latency, our compensation system alleviated scores almost to baseline level. A similar systematic is observable regrading the subjectively experienced influence of latency. With no delay added, players rated latency's impact – unsurprisingly – the lowest, when introducing delay player rated latency's influence significantly higher. On the other hand, coupling this with our latency compensation system, we effectively were able to reduce the negative effect of latency and decrease its subjective impact. Errorbars show the standard error.

Player Performance				
Condition	**Kills**	**Assists**	**Deaths**	**Headshots**
No Delay	23.52 ± 5.52	4.60 ± 2.53	10.08 ± 2.61	9.16 ± 5.32
Delay 50 ms-100 ms	11.60 ± 3.92	7.04 ± 2.99	10.64 ± 2.74	2.04 ± 1.24
Prediction 50 ms-100 ms	19.32 ± 5.71	4.68 ± 2.58	9.80 ± 2.27	6.56 ± 3.64
Delay 100 ms-150 ms	10.12 ± 3.19	7.00 ± 3.44	12.12 ± 2.70	1.68 ± 1.25
Prediction 100 ms-150 ms	18.00 ± 3.67	4.48 ± 2.26	11.04 ± 2.21	5.80 ± 2.72

Table 5.5: Shows the mean and standard deviation of the measured PP for the different conditions of the study.

5.3.4.2 Player Performance

Table 5.5 shows mean and standard deviation for *Kills*, *Assists*, *Deaths*, and *Headshots* for each condition tested. Again, we used a RM-ANOVA with different *Latency Conditions* factoring from *No Delay* to *Prediction 100 ms-150 ms*, to analyze in-game PP.

ANOVA showed significant differences for performance, for *Kills* ($F(1,24)$ =111.84, $p < 0.001$, $\eta^2 = 0.82$), *Assists* ($F(1,24) = 7.06$, $p < 0.001$, $\eta^2 = 0.227$), *Deaths* ($F(1,24) =$ 4,658, $p = 0.013$, $\eta^2 = 0.163$) and *Headshots* ($F(1,24) = 46{,}20$, $p < 0.001$, $\eta^2 = 0.66$).

Kills

Post-hoc testing revealed no differences between *Delay 50 ms-100 ms* and *Delay 100 ms-150 ms* (adjusted $p = 0.073$) and between *Prediction 50 ms-100 ms* and *Prediction 100 ms-150 ms* (adjusted $p = 0{,}101$). All other pairwise comparison revealed significant differences (all adjusted $p < 0.05$). Overall, participants performed best when playing with no delay. However, when introducing latency and prediction, our data shows that players performed significantly better when supported by our latency compensation system. Figure 5.20 (left) shows the *Kills* data.

Assists

Next, we analyzed the *Assists* data. Our tests showed significant differences between *Delay 50 ms-100 ms* and *Prediction 50 ms-100 ms* (adjusted $p = 0.014$), *Delay 50 ms-100 ms* and *Prediction 100 ms-150 ms* (adjusted $p = 0.007$) and between *Delay 100 ms-150 ms* and *Prediction 100 ms-150 ms* (adjusted $p = 0.002$). All other comparisons did not reveal significant differences (all adjusted $p > 0.05$). Generally, participants performed

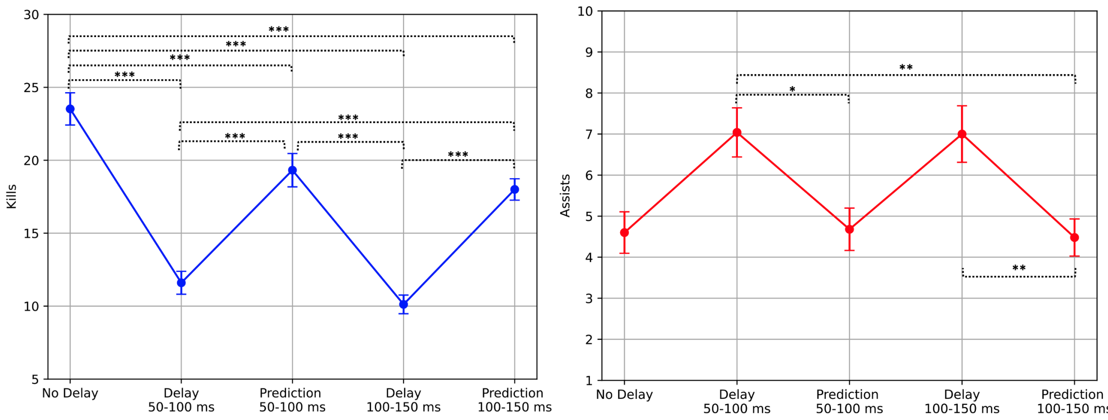


Figure 5.20: Depicts the amount of *Kills* (left) and *Asissts* (right) as a measure of player performance. Players achieved the most *Kills* when playing with the control condition. However, the data also shows that our compensation system was able to significantly reduce the negative effects of latency. Comparing playing with delay and no prediction and playing and prediction enabled evidently shows that. Similarly, players had a significantly lower number of *Assist* when playing with our prediction system, which indicates that players were able to finish eliminating the enemy. Significant difference are marked using Asterisks (* = $p < 0.05$, ** = $p < 0.01$, *** = $p < 0.001$). Errorbars show the standard error.

best when playing with no delay our one of our prediction modes. The high number of *Assists* in both latency conditions indicates that participants were not able to deal enough damage to enemies to achieve a kill. Figure 5.20 (right) shows the *Assists* data.

Deaths

Pairwise comparing *Deaths* data revealed significant differences for *No Delay* and *Delay 100 ms-150 ms* (adjusted $p = 0.012$) and for *Prediction 50 ms-100 ms* and *Delay 100 ms-150 ms* (adjusted $p = 0.003$). All other comparisons did not reach significance (all adjusted $p > 0.05$). Summarizing, our results shows that playing with a high delay leads to a significant higher *Death* count compared to playing with no delay and no prediction. Subsequently, our results also show that this increase in deaths is counteracted with an appropriate prediction effectively reducing the adverse effects of latency. Figure 5.21 (left) shows *Deaths* data for all tested conditions.

Headshots

Lastly, we analyzed the *Headshots* data. We found no significant differences between

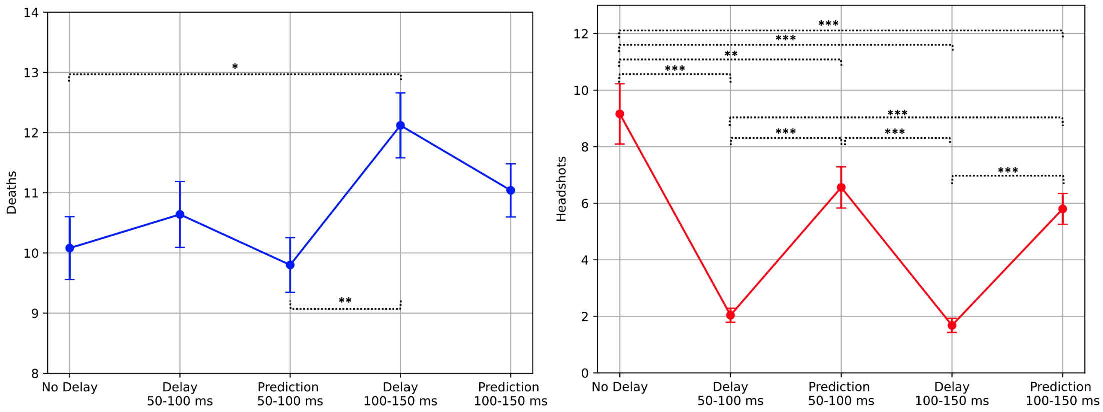


Figure 5.21: Depicts the amount of Deaths (left) and *Headshots* (right) as a measure of PP. Players had the most *Deaths* when playing with the high delay condition. However, the data also shows that our compensation system was able to significantly reduce the negative effects of latency. Comparing playing with high delay and no prediction and playing with high latency and prediction enabled evidently shows that players had a significantly lower number of *Deaths*. Similarly, latency and our prediction system significantly altered the amount of headshots performed by players. Using our prediction system players performed consistently and significantly better than playing with no latency compensation system. Significant difference are marked using Asterisks (* = $p < 0.05$, ** = $p < 0.01$, *** = $p < 0.001$). Errorbars show the standard error.

Delay 50 ms-100 ms and *Delay 100 ms-150 ms* (adjusted $p = 0{,}19$) and between *Prediction 50 ms-100 ms* and *Prediction 100 ms-150 ms* (adjusted $p = 0{,}19$). However, all other comparison showed significantly different amount of achieved *Headshots* (all adjusted $p < 0.05$). Overall, participants achieved the highest level of *Headshots* when playing with no delay and no prediction. However, when introducing latency the number of *Headshots* drops significantly. This drop is effectively and significantly counteracted by the prediction of our ANN. Participants playing with delay and the corresponding prediction performed significantly better than participants playing without the prediction. Figure 5.20 (right) shows the *Headshots* data.

5.3.5 Discussion

Our results reveal that our adaptive latency compensation system helped to improve some aspects of the players' experiences and almost every aspect of the players' performances when compared to playing with no compensation system. In the following, we first discuss how the prediction method improves player experience by contextualizing our findings regarding the PXI. Next, we shed light on the influence of our compensation system on the players' performances. To conclude this section, we provide potential avenues for future work and point out the limitations of this work.

5.3.5.1 Effects on Player Experience

Claypool et al. (2014), for example, as well as the work presented in this thesis, showed that as soon as latency exceeds 100 ms experience deteriorates significantly, which is in line with our results. The condition with the highest delay led to the worst results in almost every category of the questionnaire, while a prediction between 100 ms and 150 ms minimized the effect of latency for the subscales Challenge, Curiosity, Immersion, Mastery, Meaning, Enjoyment, and the subjectively felt level of latency. However, the control condition – with no delay added – scored higher than any other conditions, which confirms that scenarios without latency are the most pleasant ones and confirms the validity of our study. Furthermore, for the Control subscale of the PXI, our compensation system even led to better results than in the control condition (with no delay and no prediction), which indicates that our compensation method made aiming for targets, and thus playing the game, easier for the players. This is not necessarily a positive effect. Methods such as the one presented in this section could be used for cheating. Previous work discussed a proof-of-concept in which generative adversarial networks trained on player data can be used to cheat in an FPS game (Kanervisto et al., 2022). Overall, our work shows that adaptive latency compensation based on deep-learning ANNs can reduce or even eliminate the negative effects of latency on GX.

5.3.5.2 Effects on Player Performance

To measure PP, we analyzed four different variables. First, the number of headshots represents the accuracy of players. Beigbeder et al. (2004) demonstrated in previous work that latency decreases players accuracy up to 50 percent. Our findings also indicate that latency drastically decreases the count of headshots. Players in both prediction conditions achieved higher numbers of headshots than in both delay conditions, which highlights that ANNs can increase the accuracy of players in settings with low and high latency. Analyzing the second performance variable, which is comprised of the number of successful in-game kills, shows that players perform best in the control condition. However, we also show that, overall, when latency is introduced, the number of kills decreases significantly. This negative effect of latency, however, is counter-acted when playing with our latency compensation system. In our work, the number of kills significantly rises to almost no-latency levels when players are supported by our ANN's prediction. We found a similar systematic for the number of assists and number of player deaths. Overall, our findings are in line with previous work investigating the effects of latency on PP. Claypool et al. (2014), for example, showed that a high latency leads to a reduced PP. Our work comes to the same conclusion. However, we go one step further and show that the compensation of latency via adaptive ANNs significantly reduces latency's effect on PP.

In summary, our work shows that the developed ANN-based latency compensation system is well-suited to tackle the negative consequence of latency on PP and GX. When playing with latency, our prediction system enabled players to achieve performance levels and experience almost on par with playing with no latency.

5.3.5.3 Limitations and Future Work

In this work, we effectively compensated for the negative effects of varying latency using a set of adaptive deep-learning based ANNs. Although, we successfully reduced or even completely eliminated the negative influence of latency on PP and experience, this work has, nevertheless, some limitations.

First, we tried to keep the skill level of the participants as diverse as possible. The difference in skill is necessary to get broader results and get closer to a real-world scenario, where skill and previous experience also vary drastically. However, it could be useful to train different ANNs for different skill groups. Tailoring the predictive system to a player's individual skill level could improve our results even further. This is particular the case, since our previous work in Section 4.1 showed that some aspects of latency, in case of Section 4.1 the perceptual channel it is perceived by, is dependent of a player's skill level. The same could be true for latency compensation systems. Hence, future work should investigate the interplay between latency compensation and player skill.

Second, another limitation of our work is its heavy computational requirements. Running multiple inferences from different ANNs while buffering player input, rendering a 3D game, and capturing live gameplay footage is extremely computationally expensive. Future work, thus, should investigate how our approach can be optimized. For example, future work could research if the ANNs' inferences can be outsourced to a second machine, potentially a dedicated server hosted on the Internet. This would drastically reduce the computational load of the machine running the game, which would allow for a smoother gaming experience. We proposed a similar approach in Section 5.1.

5.3.6 Conclusion

In this work, we investigated if ANN-based latency compensation can be used to adaptively counteract the negative effects of latency on PP and GX. To achieve this, we trained two different ANNs. The first ANN is trained to detect enemies in the FPS game CS:GO. This information is provided to the second ANN, which is trained on data of 15 participants playing CS:GO. The second ANN's goal is to predict players' mouse movements and keyboard presses to compensate for latency. The information about enemies provided by the first ANN allows the compensation system to contextualize player behaviour and hence increases the prediction accuracy.

We used our latency compensation system in a study, in which 25 participants played with changing latency. To simulate a naturalistic latency behaviour, we used an Arduino which was interconnected between the workstation running CS:GO and the player's input devices (mouse and keyboard). In the study, participants played with five different latency conditions: (I) No latency, (II) varying latency between 50 ms and 100 ms, (III) varying latency between 100 ms and 150 ms, (IV) varying latency between 50 ms and 100 ms and our ANN-based latency compensation enabled, and (V) varying latency between 100 ms and 150 ms and our ANN-based latency compensation enabled.

Our results consistently show that our ANN-based compensation system improved PP and GX. Players using our system achieved significantly more in-game kills, had more assists, and more headshots. Simultaneously, they had a better GX while playing with our compensation compared to playing with latency and no compensation. Contrary to our previous work in Section 5.1 and Section 5.2, we did not train the ANNs in this section on fixed latency values for their prediction instead, we parameterized latency as input to the ANN and, thus, allowed it to adaptively change the level of compensation and prediction.

Summing up, this section shows that ANN are well suited to compensate for naturalistic ever-changing latency in commercial, fast-paced video games.

5.4 Synthesis Compensating Latency

In this chapter, we presented three studies that explored using ANNs to compensate for latency in a gaming context. We gradually increased the prediction difficulty in these experiments – from the first ANN using a data-rich environment with game-internal information to the last ANN in a fast-paced FPS game only using externally available data – to obtain maximal ecological results.

In the first study, we used an ANN to mitigate latency in a custom video game. Specifically, we trained an ANN using internal game data, focusing on the avatar's rotation and orientation. This approach enabled the ANN to predict and adapt in-game elements based on these players' inputs, effectively compensating for latency and enhancing the real-time gaming experience (RQ6). The second study followed a similar methodology, with multiple ANNs trained to forecast the mouse's future position in Age of Empires 2. These ANNs projected the movements of the player's mouse cursor, effectively compensating for latency in this slow-paced, commercial RTS game (RQ7). However, contrary to the first study using ANNs in custom video games, the ANN in this study did not have access to game-internal information and thus had to rely only on data that is available externally, such as player inputs and the video stream.

In the third study, we introduced an innovative multi-ANN system designed to adaptively compensate for latency in the competitive FPS game CS:GO. In this study, two interconnected ANNs collaborated seamlessly. The first ANN detects in-game enemies and provides this information to the second ANN, which predicts future mouse positions and uses the additional data from the first ANN to contextualize its prediction. This sophisticated approach reduced perceived latency, providing players with a fundamentally better performance and gaming experience in latency-inflicted gaming sessions (RQ8).

Our work demonstrates that ANN-based latency compensation for video games is a promising approach. ANNs offer versatile solutions across different gaming scenarios. From a custom game over a slow-paced commercial strategy game to a fast-paced shooting game, our ANNs consistently improve PP and experience in high latency settings. Our findings suggest that this improvement via ANNs is not limited to the games tested but can be extended to other gaming genres and scenarios.

6

Conclusion

This thesis elucidated different aspects of latency and its compensation in video games through nine studies. In the previous three chapters (Chapter 3, 4, and 5), we answered eight research questions (RQs) (cf. overview RQs in Table 1.1), in doing so we first showed how latency variation affects player performance (PP) and game experience (GX) (RQ1 and RQ2). Then, we investigated perceptual aspects of latency, such as auditory latency (RQ3), in-game perspectives (RQ4), and player expectations (RQ5), and how those alter the effects of latency. Lastly, this thesis culminates in the previous chapter (Chapter 5), in which we used the gained understanding of latency to develop novel latency compensation techniques based on artificial neural networks (ANNs)(RQ6, RQ7, and RQ8).

This chapter summarizes the contributions of this thesis and presents design guidelines to support researchers, game designers, and developers in developing future latency compensation techniques. Additionally, we look at future possibilities for latency compensation beyond a classical gaming scenario after discussing this work's limitations.

6.1 Summary of Contributions

In Chapter 3, we explored how latency variation affects PP and GX. In the first study, we found that small-term variation – latency that changes fast and volatile – does not affect PP and GX (RQ1). In the second study, however, we demonstrated that a long-term switch between latency levels, i.e., switching from one level of latency to another level for a longer duration, significantly influences the gaming session (RQ2). Ultimately, a long-term switch persistently changes the experienced input-out paradigm and affects both PP and GX. Both studies provide empirical evidence on how latency's variation alters its effects. Most importantly, it shows that assuming latency as a constant in an experimental setting does not necessarily bias the experiment. This is crucial since previous work and the work presented in this thesis often assume a constant latency as a proxy for real-world latency.

In Chapter 4, we investigated how the player's perception changes the effects of latency in four studies. We found that standalone auditory latency does not necessarily affect gaming sessions. However, a certain level of prior experience of players can lead to auditory latency being negatively influential on PP and GX (RQ3). Furthermore, we showed that the visual in-game perspective of a video game does not alter the effects of latency. Prior work came to different conclusions regarding the effects of the in-game perspective on a game's latency sensitivity (Sabet et al., 2020a; Schmidt et al., 2017). Crucially, our work demonstrates that latency is always negative independently of the used in-game perspective (RQ4). Lastly, we investigate how the mere perception of latency, for example, via an in-game overlay, alters PP and GX. We show that the mere expectation of latency – believing that one plays with latency – leads to reduced performance and experience (RQ5). This is essential because of two reasons: First, it shows that a player's expectation about a gaming session fundamentally changes how it plays out. And second, our work shows that it is not always beneficial to communicate the current level of latency to the players.

Based on the findings of Chapters 3 and 4 we implemented novel latency compensation systems based on ANNs in three independent studies. Chapter 5 focuses on the implementation and evaluation of these systems. All developed ANNs aim to reduce players' experienced latency by predicting what the player will do next. This prediction allows the system – the game – to calculate the effects of an input event before the actual input has happened. In turn, this prediction reduces the time required for an input-output loop and decreases perceived latency. In our work, we gradually increased the prediction difficulty from ANN to ANN and study to study. In the first study, we used game-internal information such as the player's avatar position and orientation to predict game-internal events. Using this data-rich environment, the ANN could effectively compensate for the negative effects of latency by predicting the player's next action (RQ6). In the next step, we only used game-external information to predict the computer mouse movement in a slow-paced real-time strategy (RTS) game in the second study since commercial video games typically do not allow access to their internal state or data. Nevertheless, we show that ANN-based latency compensation is a feasible approach in slow-paced commercial video games. Using our system, players rated the session with a higher level of GX than those playing with latency and no compensation system (RQ7). Lastly, our efforts accumulate in the multi-ANN system presented in the third study. In that work, we used two ANNs: The first ANN detects enemies in the fast-paced first-person shooter (FPS) game Counter-Strike: Global Offensive (CS:GO). This information is provided to a second ANN, which uses this information to enrich and contextualize its prediction of players' computer mouse movements. Overall, we found that ANNs that predict player input are well-suited to reduce the adverse effects of latency on PP and GX. Specifically, in the last study, we demonstrated that ANNs can compensate for unstable latency conditions and provide players with a consistent input-output paradigm despite varying latency, even in fast-paced commercial video games (RQ8).

6.2 Design Guidelines and Implications

In this section, we present the implications of this thesis. We formulate these implications as tangible design guidelines targeted for future research investigating latency and latency compensation, as well as game publishers and developers aiming to maximize their players' in-game performances and experiences. All guidelines are derived from the experimental studies conducted and the empirical findings of this thesis. We enriched them with previous work investigating the effects of latency in video games to provide an even broader context.

Stability is Key

Latency compensation techniques allow for a change in the current level of latency in an instant. However, without further adjustments, this should be avoided. This work, notably the experiment in Study 3.2, shows that a long-term switch between two latency levels can negatively affect the players' experiences and performances because players may partially habituate to one level of latency. While our work does not directly show the habituation effects of latency (cf. Section 4.2), fundamental work from psychology shows that humans generally perform better when they can expect specific outcomes from their actions (Wagener & Hoffmann, 2010). In psychology, this phenomenon is called time-based event expectancy (Volberg & Thomaschke, 2017; Thomaschke et al., 2011; Aufschnaiter et al., 2021). This paradigm states that humans internalize regularities and benefit from consistency, thus allowing them to mentally prepare for events, which increases their performance. However, we argue that a change in latency destroys this consistency and, hence, has adverse effects.

Consider the following example to better visualize why stability is vital and how this guideline should be applied. An experienced video game player living with their family plays their favorite fast-paced online multiplayer FPS game, such as CS:GO. While playing, the other family members watch Netflix, browse Social Media, or download their favorite photos from Flickr. All this heavy use of the Internet increases the latency of our CS:GO player since the family's router has to divide the available resources between all Internet users. Fortunately, the player has a prediction-based latency compensation system enabled, which reduces the otherwise high latency of 150 ms to an acceptable value of 75 ms. While this still is not ideal, it is better than playing with 150 ms. After some time, all other family members suddenly stopped using the Internet, allowing the family's router to dedicate all resources to the video game player. In doing so, the newly gained resources would reduce latency down to 20 ms, which is considered an ideal latency in previous work (Jota et al., 2013). However, this would entail a sudden change in latency, potentially significantly reducing experience and performance. In this scenario, the latency compensation system has two options: (1) stop compensating for latency and risk a latency change-induced performance and experience degradation, or (2) artificially increase the latency perceived by the user to 75 ms to provide a consistent gaming session and input-output paradigm. Given the empirical findings of this thesis, we argue that the latter is the better option. Always provide as stable an environment as possible for players.

This guideline, of course, does not entail that the compensation should let the player play with 75 ms of artificial latency all the time. On the contrary, the compensation system should make intelligent and context-aware decisions. Disabling the compensation, for example, between rounds in CS:GO, and, thus, providing the player with as low as possible latency is still advisable.

Always Consider Latency

Previous work comes to contradicting conclusions about how different video games are more or less sensitive to latency (Claypool & Claypool, 2010; Sabet et al., 2020a;

Schmidt et al., 2017). The general rule of thumb is that faster games, for example, FPS games, are more susceptible to latency since they require more timely and accurate input from the player. The work in this thesis fundamentally builds on that statement, i.e., we investigated fast-paced video games in all studies but one (cf. Study 5.2). Still, this does not mean that other types of games are not affected by latency. Video games differ by manifold factors, such as pacing, the narrative, or the used in-game perspective. Study 4.3 shows that latency always has negative consequences independent of the used in-game perspectives. This indicates that latency always affects PP and GX.

Pragmatically, game developers should always aim to implement latency compensation systems in their game. Just because a game is not an extremely fast-paced FPS game does not mean that latency will not affect the gaming session of slower-paced games. Hence, latency should always be considered, be it in game development or research using or investigating games.

Respect Individual Player Skill and Experience

Our work in Section 4.1, and previous work (Liu et al., 2021a), indicates that the effects of latency are modulated by individual player skill. More experienced players are more strongly affected by latency than lesser-skilled players. This is relevant not only for latency research but also for latency compensation.

Ideally, a latency compensation system should be aware of individual player skills and be able to adapt the level of compensation for each player individually. This becomes even more relevant in light of multiplayer games in which players with differing skill levels play with each other on the same server. A centralized latency compensation instance, for example, a dedicated server responsible for compensation using our presented ANN-based approach, could perform a skill-based latency compensation prioritization. The system could prioritize reducing latency for more skilled players since they are more sensitive to it. This could involve allocating more resources or using advanced techniques to minimize latency for highly skilled players, even at the expense of less experienced players who might not notice or be affected by slight delays. Similarly, the system could adapt its prediction and compensation algorithms based on the player's skill level. For highly skilled players, it might use more aggressive prediction and compensation techniques to ensure smoother gameplay. In contrast, for less experienced players, it might apply less aggressive compensation to prioritize overall system performance.

Considering player skills when designing and operating latency compensation systems can lead to more tailored and practical solutions. By recognizing that highly skilled players are more sensitive to latency, these systems can prioritize reducing latency for those players while still providing a reasonable experience for less skilled players. Ultimately, the goal is to provide a fair and enjoyable gaming experience for all players, taking their individual skill levels into account.

Consider Expectancy-based Latency Effects

Our work shows that the mere expectation of latency can lead to an expectation-based performance and experience drop. This has consequences for latency researchers, latency compensation, and game developers. First, researchers should avoid informing participants about the exact purpose of their studies if they investigate the effects of latency (as we did in all compensation experiments; see, for example, study design Study 9, 5.3.3.3). If the participants expect to be testing a setting inflicted with latency, this can bias the investigation's results.

Second, this guideline also considers that most video games have some sort of user interface that shows the current level of latency. This is done to inform the players of their current connection quality, for example, in a massively multiplayer online game. However, this entails that the players expect the displayed latency. Since latency's effects are not entirely technical but at least partly expectation-based, communicating the current latency to players can have negative consequences.

Pragmatically, game and latency compensation developers need to be aware of the expectancy-based effects of latency. This may even entail not communicating a high latency to players because this may worsen the gaming session further.

Provide Context for Latency Compensation Systems

The latency compensation system should always be provided with contextual information about the current game state. While this is hard to achieve when investigating commercial video games, we showed in Study 5.3 that it is worth the effort. Our work provided the compensation system with information about enemy positions in live gameplay. The prediction system then utilized this information to contextualize the players' behaviors. Ultimately, this allowed the compensation system to provide robust latency compensation, which could even reliably compensate for unstable latency.

The same approach could be used in other games and latency compensation methods. In open-world games, for example, with expansive environments, providing contextual information about the player's current location, mission objectives, and nearby interactive elements (e.g., non-player characters, items, and obstacles) can help the latency compensation system predict the player's actions more accurately. Another example is the use of latency compensation in multiplayer online battle arena (MOBA) games such as League of Legends or Dota 2. In these games, contextual information about the hero compositions, map objectives, and the actions of other players is vital. Latency compensation systems that consider this information could improve the responsiveness of hero abilities and tactics, making gameplay more enjoyable and competitive.

Overall, it is evident that providing an ANN-based latency compensation system with contextual information is beneficial. Input-output paradigms change with the game played. Each game requires a unique interaction – taking these circumstances into account when compensating for latency is crucial.

Consider Prediction Errors

Managing prediction errors is critical to designing an effective latency compensation system based on ANNs for video games. Balancing prediction accuracy with real-time responsiveness and adaptability is key to enhancing gaming experience. Thus, developers of games or latency compensation systems based on ANNs should be aware of several critical aspects: First, determine an acceptable level of prediction error or deviation from the actual input. This threshold should be based on the game's specific requirements and the input's sensitivity. For example, in a fast-paced FPS game, the tolerance for prediction errors in computer mouse movement may be lower than in a slower-paced strategy game. Second, the quality and quantity of training data can significantly affect prediction accuracy. Ensure that the ANN is trained on a diverse and representative dataset of player inputs. Periodically updating the training dataset with recent gameplay data can also help improve prediction accuracy. Third, continuously test and validate the latency compensation system with real players in various gaming scenarios. Collect feedback and performance data to identify areas where prediction errors are most problematic and make iterative improvements. Lastly, in extreme cases where prediction errors become too severe or disruptive, consider implementing fallback mechanisms that temporarily disable prediction and revert to traditional input handling until the prediction accuracy improves.

Consider Player Agency

When implementing a latency compensation system based on ANNs that predict a player's next input, such as computer mouse movement, it's essential to consider how such a predictive system can potentially alter the player's agency.

Predictive systems can inadvertently limit a player's sense of control and autonomy. When the system predicts and executes actions on the player's behalf, it may lead to a feeling of reduced agency, especially if the predictions do not align with the player's intentions. Striking a balance between reducing latency and preserving player agency is crucial. Furthermore, developers of commercial video games should provide players with the option to customize or toggle the predictive system. Some players may prefer high prediction and automation, while others may want full manual control. Allowing players to adjust the system's behavior empowers them to tailor the compensation system to their preferences. In the same vein, a real-world latency compensation system based on ANNs should include fail-safe mechanisms that allow players to regain full control in situations where the predictive system's actions are unwanted or detrimental. This can prevent frustration and maintain player agency in critical moments.

In summary, implementing a predictive system based on ANNs for latency compensation in gaming requires careful consideration of player agency and autonomy. The goal is to reduce latency while preserving the player's sense of control and enjoyment. Customizability, transparency, and the ability to adapt to player preferences are key factors to achieve this balance. Pragmatically, game developers always have to consider the player's feeling of agency and control when developing ANN-based compensation systems.

6.3 Limitations and Future Work

In this thesis, we investigate various aspects of latency and how it affects players their gaming sessions. We used this understanding to develop novel latency compensation methods based on ANNs. Despite this thesis's contributions, such as empirical findings, user data, and trained latency compensation ANNs, it still has some limitations, which we discuss in the following. However, each presented limitation is also a possible avenue for future research investigating how to compensate for or how to investigate latency in video games.

Generalization

In Chapter 5, we used ANNs to predict player input in video games. For each study, we tailored a specific ANN. We did this because the interaction with each game differs. If one game requires fast horizontal movement to position the virtual weapon on an enemy, another requires more vertical movement to select units in an RTS game. We showed that this approach is valid in the evaluation of each ANN. However, developing a specialized ANN for each game is resource intensive, not future-proof, limiting its use to a single game, and does not allow generalization to other games. In an ideal scenario, an ANN-based latency compensation technique is game-independent and compensates for latency in all games. Future work should investigate if ANNs can be used to compensate for latency in more general terms. Future work could gather data from players playing games from different genres, such as FPSs and MOBAs, and aim to train the ANN on the aggregated data.

Computational Load

Furthermore, future work should investigate whether latency compensation can be outsourced to another machine that is not running the game. In Sections 5.2 and 5.3, the used workstations were responsible for running the games in high FPS to not introduce a bias by low FPS and providing an inference of the models. This produced an exceptionally high load on the workstation. Particularly, in Study 5.3, the load on the machine was so high that we had to outsource the latency generation (the controlled latency we used in our studies) to an external microcontroller. Controlling for latency, running a graphical demanding video game, such as CS:GO, with high FPS, and providing timely inference from two ANNs (enemy detection and latency compensation) was not possible. It is reasonable to assume that the same problem would arise when using consumer gaming computers to generate inferences from ANNs while playing games. Therefore, for maximized practical applicability, future work should investigate if an ANN-based latency prediction system can be entirely outsourced to a centralized unit, such as a powerful cloud server. In principle, this should be possible. We demonstrated a first step in this direction in Study 5.1. However, in this study, we used proprietary and now deprecated libraries (Tensorflow.JS), which are not optimized for a large number of players, to serve our ANN and model inference online. Using this approach, we reduced the computational load on the players' workstations in the study. Thus, in the age of cloud computing, it seems appropriate and achievable to serve ANN-based compensation systems entirely online.

Complex Network Architectures

A further essential aspect to consider in the context of this thesis is the exclusive use of the feed-forward network architecture. Although we tested different network architectures, such as convolutional neural networks and transformer-based networks in Sections 5.2 and 5.3, we grounded the compensation in all three studies on simple dense feed-forward models. The reasoning for this is trifold: First, more complex architectures, such as the long short-term memory network tested in Section 5.3, required more time to produce an inference, leaving less time to integrate the prediction into the game loop without inferring with the smooth execution of the game. Second, training a more complex network with more neurons, layers, or other features, such as encoder-decoder pipelines, requires substantially more computational resources. A network's training time is proportional to its parameter count, built by the aforementioned factors. Therefore, training complex ANNs is exceedingly hard using limited time and computational power. Lastly, even if we achieved error minimization, and hence, an optimized prediction, as, for example, using the CNN in Study 5.2, it did not increase prediction accuracy. Since more complex architectures did not perform better than the classical feed-forward network, we omitted them for the final evaluation in each study to save computational resources. Nevertheless, it is possible that more complex architecture could perform better. This, for example, could be achieved by using a higher resolution when training the CNN with gameplay images. Hence, future work should keep working on optimizing the ANN-based latency compensation approach using more sophisticated network architectures.

Online vs. Offline Training

Besides the architectural structure of the used models, another limitation is how we trained them. All models presented in this thesis were trained in a classical, data-driven, offline-training approach (Goodfellow et al., 2016). This means that we gathered data from users playing a certain game, defined input and output in the training data (for example, the previous five computer mouse positions and a future position), trained the network, and deployed it combined with the video game from which the data was gathered. While offline-learning is a valid approach in various scenarios, it also entails that the model's weights and parameters are fixed while in use. The model cannot continue to learn. In contrast to the offline approach stands an online learning method. In online learning, a model indefinitely continues to learn. It constantly updates its parameters and weights, learns new patterns (Goodfellow et al., 2016, pp. 25–26), and may even be able to adapt to individual circumstances, such as an individual play style or player, on the spot. Allowing the model to adapt to individual play styles and players seems promising, considering that not one player is like another (Tondello et al., 2016). However, deploying latency compensation based on an online learning ANN system fundamentally differs from the approaches presented in this thesis. Online systems require the ability to train consistently while still providing the game with inference. This produces a high additional computational load. Despite these obstacles, future work should investigate if online systems can be used for latency compensation, given that they offer some advantages over offline systems, such as adaptability to individual players.

Demographics of Sample

The participants investigated in the thesis present another limitation. In all of our studies, except for the studies presented in Section 3.2 and 5.1, participants mainly consisted of university students between the ages of 18 and 30 years. Although this age group reflects a large part of actual video game players (Statista, 2023a), it is not particularly diverse. While we aimed to diversify our samples, the presented samples are not necessarily representative of the overall gaming population. Nowadays, almost everybody plays games, regardless of age, gender, ethnicity, or academic background. It is, for example, possible that the age-related increase in reaction time humans typically undergo (Woods et al., 2015) also changes a human's latency sensitivity. Hence, future work should investigate the effects of latency on PP and GX and how to compensate for its adverse effects using ANNs with a broader and more diverse sample.

Multi-faceted Latency

Lastly, within the context of this thesis, we researched different aspects of latency, how it is perceived, how it is modulated by different game attributes, such as the in-game perspective, and how it is compensated using ANNs. However, while this thesis significantly contributes to the extensive and ever-increasing body of work investigating latency and latency compensation, it is not all-encompassing. Since latency in video games is such a multi-faceted construct, there are still open questions regarding its effects and compensation. For example, recent work by Wiese and Henze (2023) investigates a concept called negative latency. Negative latency refers to a latency that reduces the time between user input and output beyond zero. This is achieved by, similar to the predictive system used in this thesis, predicting a user input farther in the future than the actual latency. Preliminary investigation show that negative latency can improve performance even further than a theoretical zero-latency environment. However, its applicability in a gaming scenario is yet to be investigated.

Overall, it is evident that latency is a complex and highly influential construct. Thus, future work should keep expanding our knowledge about its effects and exploring novel ideas and methods to compensate for it.

6.4 Latency Compensation beyond Games

In the previous section, we discussed the limitations of this thesis. In doing so, we showed new approaches and possibilities to overcome these limitations to better our understanding of latency in video games. However, while video games are undeniably important in the broader picture of Human-Computer Interaction (HCI), they are not the only type of interactive system that is affected by latency and, thus, would benefit from a latency compensation system. One such system that has the potential to introduce a paradigm shift in how we use computers is full-body motion-tracked virtual reality (VR). In a full-body motion-tracked scenario, the user's real-world movement is tracked by several cameras, processed by a workstation, and then displayed on a user-worn head-mounted display (HMD) and thus translated to VR. This allows for complete immersion. The users dive into the VR world. However, this extensive processing pipeline (tracking, processing, rendering, displaying) drastically increases the latency of such systems. Previous work shows that a high latency in such settings leads to, for example, increased motion sickness, decreased immersion, and postural instabilities (Meehan et al., 2003; Kasahara et al., 2017).

Hence, to look at future possibilities and the broader implications of this thesis for HCI, we briefly present two studies investigating compensating for the negative effects of latency in full-body motion-tracked VR. The presented studies were conducted beyond the scope of this thesis, but follow the same approach as the other studies reported in this work.

In the first study (Schwind et al., 2020), we used a similar compensation approach as presented in this thesis. We gathered data from 20 participants while using a full-body motion-tracked VR system in a dedicated data collection study. In the study, participants had to perform several tasks, such as moving slowly in a circle, jumping forward and backward, shadow-boxing with both hands, or waving in certain directions. At the same time, they were tracked and recorded by a motion-tracking system. We used this data to train different ANNs. The ANN's goal was to predict, based on previous movements, the next movement of a user. Similarly to the compensation loop presented in this thesis, this prediction can then be fed to the motion tracking pipeline. Ultimately, this allows the system to anticipate how a user will move next and, hence, to implement a change in the scene faster than with no prediction. Effectively, this reduces the perceived latency by the participant using the full-body motion-tracked VR setting. In two subsequent evaluation studies, we applied our compensation system and controlled for the latency in the setting. Overall, we found that the prediction system did not increase user performance. However, we also found that the perceived accuracy of the body location improved, increasing the users' experiences. In the bigger picture of HCI, our study illustrates that ANN-based latency compensation systems are also suitable for use outside of video games and can reliably predict and anticipate real-world human movement.

The second study also investigates latency compensation in full-body motion-tracked VR (Halbhuber et al., 2023a). However, contrary to the previously presented study and contrary to the work presented in the context of this thesis, we did not use an ANN-based compensation approach in this study. Instead, we investigated a game manipulation approach (cf. Chapter 2) which is built on two pillars: First, previous work shows that both the embodied avatar (the virtual representation of the user in VR) (Kocur et al., 2020; Yee & Bailenson, 2007; Yee et al., 2009; Banakou et al., 2016; Banakou et al., 2018) and latency (Schwind et al., 2020; Meehan et al., 2003; Kasahara et al., 2017) fundamentally change how users experience and behave in VR. Second, previous research in non-VR video games demonstrates that players tolerate higher latency if the controlled avatar better fits the user's mental interaction model (Claypool & Claypool, 2010). However, it was unknown if the avatars' visual appearances in VR can be used to compensate for the adverse effects of latency. Hence, this study investigated how the avatar's visual appearance interacts with latency in full-body motion-tracked VR. Since previous work indicates that an avatar's appearance has a systematic effect on physical performance (Kocur et al., 2020; Kocur et al., 2022; Kocur et al., 2021), we specifically explore the interplay of avatar and latency in two physically demanding tasks. To explore if an avatar's visual appearance can be used to compensate for the negative effects of latency, we conducted a study with 16 participants. In the study, participants performed two physically challenging tasks with two different avatars and two latency levels. While we found a significant effect of latency on performance and experience, our results show no interaction between the avatar's visual appearance and latency. We show that in full-body motion-tracked VR, latency effects do not depend on the embodied avatar. Our study reaffirms the significance of latency in VR experiences. High latency can disrupt user immersion and negatively impact their interaction with virtual environments. This finding underscores the importance of minimizing latency in VR systems to enhance user experiences.

Both studies demonstrate that latency and its role in HCI is highly diverse. While this thesis allows us to better our understanding of latency and its effects and presents several approaches to reduce its adverse effect in video games, there are still various other settings, interaction methods, and systems that potentially interact with latency and are worthwhile to investigate.

In conclusion, the meticulous consideration of latency in HCI is not only paramount for enhancing user experiences but also instrumental in shaping the future of technology and its seamless integration into our daily lives.

The section is partly based on the following articles:

Schwind, V., Halbhuber, D., Fehle, J., Sasse, J., Pfaffelhuber, A., Tögel, C., et al. (2020). "The Effects of Full-Body Avatar Movement Predictions in Virtual Reality Using Neural Networks." In: *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology*. VRST '20. Virtual Event, Canada: Association for Computing Machinery. ISBN: 9781450376198. DOI: 10.1145/3385956.3418941.

Halbhuber, D., Kocur, M., Kalus, A., Angermeyer, K., Schwind, V., & Henze, N. (2023a). "Understanding the Effects of Perceived Avatar Appearance on Latency Sensitivity in Full-Body Motion-Tracked Virtual Reality." In: *Proceedings of Mensch Und Computer 2023*. MuC '23. Rapperswil, Switzerland: Association for Computing Machinery, pp. 1–15. ISBN: 9798400707711. DOI: 10.1145/3603555.3603580.

Bibliography

Ababsa, F., Mallem, M., & Roussel, D. (2004). “Comparison between particle filter approach and Kalman filter-based technique for head tracking in augmented reality systems.” In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*. Vol. 1, 1021–1026 Vol.1. DOI: `10.1109/ROBOT.2004.1307284` (cit. on p. 88).

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). “{TensorFlow}: a system for {Large-Scale} machine learning.” In: *12th USENIX symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283 (cit. on pp. 222, 237, 254).

Abeele, V. V., Spiel, K., Nacke, L., Johnson, D., & Gerling, K. (2020). “Development and validation of the player experience inventory: A scale to measure player experiences at the level of functional and psychosocial consequences.” In: *International Journal of Human-Computer Studies* 135, p. 102370. DOI: `10.1016/j.ijhcs.2019.102370` (cit. on pp. 18, 102, 110, 176, 181, 184, 185, 188, 189).

Accot, J., & Zhai, S. (1997). “Beyond Fitts’ Law: Models for Trajectory-Based HCI Tasks.” In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. CHI ’97. Atlanta, Georgia, USA: Association for Computing Machinery, pp. 295–302. ISBN: 0897918029. DOI: `10.1145/258549.258760` (cit. on p. 46).

Adams, A., Lunt, P., & Cairns, P. (2008). “A qualititative approach to HCI research.” In: Cambridge University Press. ISBN: 9780521870122 (cit. on p. 56).

Ahsan, M. R., Ibrahimy, M. I., & Khalifa, O. O. (2011). “Hand motion detection from EMG signals by using ANN based classifier for human computer interaction.” In: *2011 Fourth International Conference on Modeling, Simulation and Applied Optimization*. IEEE, pp. 1–6. DOI: `10.1109/ICMSAO.2011.5775536` (cit. on pp. 89, 91).

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). "Optuna: A Next-generation Hyperparameter Optimization Framework." In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. DOI: 10.1145/3292500.3330701 (cit. on pp. 237, 254).

Akiduki, H., Nishiike, S., Watanabe, H., Matsuoka, K., Kubo, T., & Takeda, N. (2003). "Visual-vestibular conflict induced by virtual reality in humans." In: *Neuroscience Letters* 340.3, pp. 197–200. DOI: 10.1016/S0304-3940(03)00098-3 (cit. on p. 17).

Amin, R., Jackson, F., Gilbert, J. E., Martin, J., & Shaw, T. (2013). "Assessing the impact of latency and jitter on the perceived quality of call of duty modern warfare 2." In: *International Conference on Human-Computer Interaction*. Springer, pp. 97–106. DOI: https://doi.org/10.1007/978-3-642-39265-8_11 (cit. on p. 111).

Andraszewicz, S., Scheibehenne, B., Rieskamp, J., Grasman, R., Verhagen, J., & Wagenmakers, E.-J. (2015). "An introduction to Bayesian hypothesis testing for management research." In: *Journal of Management* 41.2, pp. 521–543. DOI: 10.1177/0149206314560412 (cit. on pp. 179, 182).

Annett, M., Anderson, F., Bischof, W. F., & Gupta, A. (2014). "The Pen is Mightier: Understanding Stylus Behaviour While Inking on Tablets." In: *Proceedings of Graphics Interface 2014*. GI '14. Montreal, Quebec, Canada: Canadian Information Processing Society, pp. 193–200. ISBN: 9781482260038. DOI: 10.5555/2619648.2619680 (cit. on pp. 17, 24, 49, 196).

Arduino (n.d.). *Leonardo | Arduino Documentation*. https://docs.arduino.cc/hardware/leonardo (cit. on p. 257).

Armitage, G. (2003). "An experimental estimation of latency sensitivity in multiplayer Quake 3." In: *The 11th IEEE International Conference on Networks, 2003. ICON2003.* IEEE, pp. 137–141. DOI: 10.1109/ICON.2003.1266180 (cit. on p. 176).

Arnstein, P., Broglio, K., Wuhrman, E., & Kean, M. B. (2011). "Use of placebos in pain management." In: *Pain Management Nursing* 12.4, pp. 225–229. DOI: 10.1016/j.pmn.2010.10.033 (cit. on pp. 33, 93, 194, 195).

Ashby, F. G., & Isen, A. M. (1999). "A neuropsychological theory of positive affect and its influence on cognition." In: *Psychological Review* 106.3, p. 529. DOI: 10.1037/0033-295x.106.3.529 (cit. on pp. 205, 230).

Attig, C., Rauh, N., Franke, T., & Krems, J. F. (2017). "System latency guidelines then and now–is zero latency really considered necessary?" In: *Engineering Psychology and Cognitive Ergonomics: Cognition and Design: 14th International Conference, EPCE 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part II 14*. Springer, pp. 3–14. DOI: 10.1007/978-3-319-58475-1_1 (cit. on pp. 49, 50).

Aufschnaiter, S., Zhao, F., Gaschler, R., Kiesel, A., & Thomaschke, R. (2021). “Investigating time-based expectancy beyond binary timing scenarios: evidence from a paradigm employing three predictive pre-target intervals.” In: *Psychological Research*, pp. 1–14. DOI: `10.1007/s00426-021-01606-2` (cit. on p. 276).

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). “Segnet: A deep convolutional encoder-decoder architecture for image segmentation.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12, pp. 2481–2495. DOI: `10.48550/arXiv.1511.00561` (cit. on p. 253).

Bae, H.-S., Lee, H.-J., & Lee, S.-G. (2016). “Voice recognition based on adaptive MFCC and deep learning.” In: *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, pp. 1542–1546. DOI: `10.1109/ICIEA.2016.7603830` (cit. on p. 89).

Banakou, D., Hanumanthu, P. D., & Slater, M. (2016). “Virtual embodiment of white people in a black virtual body leads to a sustained reduction in their implicit racial bias.” In: *Frontiers in Human Neuroscience*, p. 601. DOI: `10.3389/fnhum.2016.00601` (cit. on p. 286).

Banakou, D., Kishore, S., & Slater, M. (2018). “Virtually being Einstein results in an improvement in cognitive task performance and a decrease in age bias.” In: *Frontiers in Psychology*, p. 917. DOI: `10.3389/fpsyg.2018.00917` (cit. on p. 286).

Barrera, K. G., & Shah, D. (2023). “Marketing in the Metaverse: Conceptual understanding, framework, and research agenda.” In: *Journal of Business Research* 155, p. 113420. DOI: `10.1016/j.jbusres.2022.113420` (cit. on p. 56).

Bateman, S., Mandryk, R. L., Stach, T., & Gutwin, C. (2011). “Target assistance for subtly balancing competitive play.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2355–2364. DOI: `10.1145/1978942.1979287` (cit. on pp. 78, 91).

Battle, L., Crouser, R. J., Nakeshimana, A., Montoly, A., Chang, R., & Stonebraker, M. (2019). “The role of latency and task complexity in predicting visual search behavior.” In: *IEEE Transactions on Visualization and Computer Graphics* 26.1, pp. 1246–1255. DOI: `10.1109/TVCG.2019.2934556` (cit. on p. 165).

Beecher, H. K. (1955). “The powerful placebo.” In: *Journal of the American Medical Association* 159.17, pp. 1602–1606. DOI: `10.1001/jama.1955.02960340022006` (cit. on pp. 33, 93, 194, 195).

Beigbeder, T., Coughlan, R., Lusher, C., Plunkett, J., Agu, E., & Claypool, M. (2004). “The Effects of Loss and Latency on User Performance in Unreal Tournament 2003®.” In: *Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games*. NetGames

’04. Portland, Oregon, USA: Association for Computing Machinery, pp. 144–151. ISBN: 158113942X. DOI: 10.1145/1016540.1016556 (cit. on pp. 18, 31, 33, 60, 67, 92, 102, 116, 134, 143, 153, 188, 196, 198, 206, 228, 230, 269).

Bernier, Y. W. (2001). “Latency compensating methods in client/server in-game protocol design and optimization.” In: *Game Developers Conference*. Vol. 98033. 425 (cit. on pp. 34, 70, 91, 93, 212).

Bethesda (2022). *The Elder Scrolls Online*. https://www.elderscrollsonline.com/de/home (cit. on p. 140).

Beyer, J., & Möller, S. (2014). “Assessing the impact of game type, display size and network delay on mobile gaming QoE.” In: *PIK-Praxis der Informationsverarbeitung und Kommunikation* 37.4, pp. 287–295. DOI: 10.1515/pik-2014-0030 (cit. on pp. 64, 69).

Beznosyk, A., Quax, P., Coninx, K., & Lamotte, W. (2011). “Influence of network delay and jitter on cooperation in multiplayer games.” In: *Proceedings of the 10th International Conference on Virtual Reality Continuum and its Applications in Industry*, pp. 351–354. DOI: 10.1145/2087756.2087812 (cit. on pp. 59, 60, 69).

Bogon, J., & Halbhuber, D. (2023). *Time and Timing in Video Games: How Video Game and Time Perception Research can benefit each other*. Mensch und Computer 2023 - Workshopband. DOI: 10.18420/muc2023-mci-ws05-439 (cit. on p. 41).

Booth, D. (2004). “The Human-Computer Interaction Handbook: Fundamentals Evolving Technologies and Emerging Applications.” In: *Journal of Occupational Psychology, Employment and Disability* 6.2, pp. 85–87 (cit. on p. 44).

Boudaoud, B., Knowles, P., Kim, J., & Spjut, J. (2021). “Gaming at warp speed: Improving aiming with late warp.” In: *ACM SIGGRAPH 2021 Emerging Technologies*, pp. 1–4. DOI: 10.1145/3450550.3465347 (cit. on p. 74).

Braun, V., & Clarke, V. (2006). “Using thematic analysis in psychology.” In: *Qualitative research in psychology* 3.2, pp. 77–101. DOI: 10.1191/1478088706qp063oa (cit. on p. 56).

Braun, V., & Clarke, V. (2021). “One size fits all? What counts as quality practice in (reflexive) thematic analysis?” In: *Qualitative research in psychology* 18.3, pp. 328–352. DOI: 10.1080/14780887.2020.1769238 (cit. on p. 56).

Brochu, N. (n.d.). *D3DShot*. https://pypi.org/project/d3dshot/ (cit. on p. 252).

Brookhaven National Laboratory (2013). *Tennis For Two come appariva nel 1958*. https://commons.wikimedia.org/wiki/File:Tennis_For_Two_on_a_DuMont_Lab_Oscilloscope_Type_304-A.jpg (cit. on p. 32).

Brun, J., Safaei, F., & Boustead, P. (2006). “Managing latency and fairness in networked games.” In: *Communications of the ACM* 49.11, pp. 46–51. DOI: 10.1145/1167838.1167861 (cit. on p. 189).

Buker, T. J., Vincenzi, D. A., & Deaton, J. E. (2012). "The effect of apparent latency on simulator sickness while using a see-through helmet-mounted display: Reducing apparent latency with predictive compensation." In: *Human Factors* 54.2, pp. 235–249. DOI: 10.1177/001872081142873 (cit. on pp. 89, 91, 212).

Burgess, S., & Katchabaw, M. (2006). "Design and implementation of optimistic constructs for latency masking in online video games." In: *The 2nd annual North American Game-On Conference (GameOn'NA 2006). Monterey, CA, USA*. Citeseer (cit. on pp. 81, 91).

Cairns, P., Cox, A., & Nordin, A. I. (2014). "Immersion in digital games: review of gaming experience research." In: *Handbook of Digital Games*, pp. 337–361. DOI: 10.1002/9781118796443.ch12 (cit. on p. 64).

Card, S., Moran, T., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Crc Press (cit. on pp. 17, 21, 26, 28, 49, 50, 52, 60, 189).

Carlson, E., Fan, T., Guan, Z., Xu, X., & Claypool, M. (2021). "Towards usable attribute scaling for latency compensation in cloud-based games." In: *Proceedings of the Workshop on Game Systems (GameSys' 21)*, pp. 20–25. DOI: 10.1145/3458335.3460964 (cit. on pp. 63, 69).

Caserman, P., Martinussen, M., & Göbel, S. (2019). "Effects of end-to-end latency on user experience and performance in immersive virtual reality applications." In: *Entertainment Computing and Serious Games: First IFIP TC 14 Joint International Conference, ICEC-JCSG 2019, Arequipa, Peru, November 11–15, 2019, Proceedings 1*. Springer, pp. 57–69. DOI: 10.1007/978-3-030-34644-7_5 (cit. on p. 65).

Casiez, G., Conversy, S., Falce, M., Huot, S., & Roussel, N. (2015). "Looking Through the Eye of the Mouse: A Simple Method for Measuring End-to-end Latency Using an Optical Mouse." In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. UIST '15. New York, NY, USA: ACM, pp. 629–636. ISBN: 978-1-4503-3779-3. DOI: 10.1145/2807442.2807454 (cit. on pp. 50, 52, 96).

Casiez, G., Pietrzak, T., Marchal, D., Poulmane, S., Falce, M., & Roussel, N. (2017). "Characterizing Latency in Touch and Button-Equipped Interactive Systems." In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. New York, NY, USA: ACM, pp. 29–39. ISBN: 978-1-4503-4981-9. DOI: 10.1145/3126594.3126606 (cit. on pp. 33, 50, 52, 92).

Chen, K.-T., Chang, Y.-C., Hsu, H.-J., Chen, D.-Y., Huang, C.-Y., & Hsu, C.-H. (2013). "On the quality of service of cloud gaming systems." In: *IEEE Transactions on Multimedia* 16.2, pp. 480–495. DOI: 10.1109/TMM.2013.2291532 (cit. on p. 87).

Chen, L., Chen, G.-C., Chen, H., March, J., Benford, S., & Pan, Z.-G. (2007). "An HCI method to improve the human performance reduced by local-lag mechanism." In: *Interacting with Computers* 19.2, pp. 215–224. DOI: 10.1016/j.intcom.2006.05.009 (cit. on pp. 82, 91).

Chen, Y., Ou, J., & Whittinghill, D. M. (2015). “Cognitive Load in Real-Time Strategy Gaming: Human Opponent Versus AI Opponent.” In: *The Computer Games Journal* 4.1, pp. 19–30. DOI: 10.1007/s40869-015-0002-z (cit. on p. 245).

Cheng, H.-Y., Lin, C.-H., Li, J., & Yang, C.-L. (2010). “Memory latency reduction via thread throttling.” In: *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, pp. 53–64. DOI: 10.1109/MICRO.2010.39 (cit. on pp. 34, 86, 91, 93).

CHI ’92: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (1992). Monterey, California, USA: Association for Computing Machinery. ISBN: 0897915135 (cit. on p. 16).

Chowdary, M. K., Nguyen, T. N., & Hemanth, D. J. (2021). “Deep learning-based facial emotion recognition for human–computer interaction applications.” In: *Neural Computing and Applications*, pp. 1–18. DOI: 10.1007/s00521-021-06012-8 (cit. on p. 89).

Claypool, M. (2018). “Game input with delay—moving target selection with a game controller thumbstick.” In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.3s, pp. 1–22. DOI: 10.1145/3187288 (cit. on p. 64).

Claypool, M., & Claypool, K. (2006). “Latency and Player Actions in Online Games.” In: *Commun. ACM* 49.11, pp. 40–45. ISSN: 0001-0782. DOI: 10.1145/1167838.1167860 (cit. on pp. 18, 20, 23, 25, 58, 97, 98, 113, 139, 140, 143, 151, 152, 169, 170, 176, 188, 190, 196–198, 206, 216, 246).

Claypool, M., & Claypool, K. (2010). “Latency Can Kill: Precision and Deadline in Online Games.” In: MMSys ’10. Phoenix, Arizona, USA: Association for Computing Machinery, pp. 215–222. ISBN: 9781605589145. DOI: 10.1145/1730836.1730863 (cit. on pp. 25, 33, 35, 58, 68, 69, 92, 172, 276, 286).

Claypool, M., Cockburn, A., & Gutwin, C. (2019). “Game input with delay: Moving target selection parameters.” In: *Proceedings of the 10th ACM Multimedia Systems Conference*, pp. 25–35. DOI: 10.1145/3304109.3306232 (cit. on pp. 60, 61, 69).

Claypool, M., Cockburn, A., & Gutwin, C. (2020). “The impact of motion and delay on selecting game targets with a mouse.” In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16.2s, pp. 1–24. DOI: 10.1145/3390464 (cit. on pp. 59, 69).

Claypool, M., & Finkel, D. (2014). “The effects of latency on player performance in cloud-based games.” In: *2014 13th Annual Workshop on Network and Systems Support for Games*. IEEE, pp. 1–6. DOI: 10.1109/NetGames.2014.7008964 (cit. on pp. 18, 19, 31, 153, 170, 191, 198, 246).

Claypool, M., Finkel, D., Grant, A., & Solano, M. (2014). "On the performance of OnLive thin client games." In: *Multimedia Systems* 20.5, pp. 471–484. DOI: 10.1007/s00530-014-0362-4 (cit. on pp. 20, 133, 139, 227, 269, 270).

Cleophas, T. J., & Zwinderman, A. H. (2018). *Modern bayesian statistics in clinical research*. Springer, pp. 1–7. ISBN: 9783319927466 (cit. on pp. 107, 149, 179).

Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017). "EMNIST: Extending MNIST to handwritten letters." In: *International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 2921–2926. DOI: 10.48550/arXiv.1702.05373 (cit. on p. 89).

Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. Routledge, pp. 18–22. ISBN: 9780121790608 (cit. on p. 106).

Colloca, L. (2019). "The placebo effect in pain therapies." In: *Annual Review of Pharmacology and Toxicology* 59, p. 191. DOI: 10.1146\%2Fannurev-pharmtox-010818-021542 (cit. on pp. 194, 195).

Colloca, L., & Barsky, A. J. (2020). "Placebo and nocebo effects." In: *New England Journal of Medicine* 382.6, pp. 554–561. DOI: 10.1056/NEJMra1907805 (cit. on pp. 33, 93, 194, 195).

Cooper, N., Horne, T., Hayes, G. R., Heldreth, C., Lahav, M., Holbrook, J., & Wilcox, L. (2022). "A systematic review and thematic analysis of community-collaborative approaches to computing research." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–18. DOI: 10.1145/3491102.3517716 (cit. on p. 56).

Cowley, B., Charles, D., Black, M., & Hickey, R. (2008). "Toward an understanding of flow in video games." In: *Computers in Entertainment (CIE)* 6.2, pp. 1–27. DOI: 10.1145/1371216.1371223 (cit. on p. 131).

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. Vol. 1990. Harper & Row New York (cit. on pp. 18, 63, 132).

Csikszentmihalyi, M. (1997). *Finding Flow: The psychology of engagement with everyday life*. DOI: 10.1023/A:1010032312178 (cit. on p. 131).

Dabrowski, R., Manuel, C., & Smieja, R. (2014). "The Effects of Latency on Player Performance and Experience in a Cloud Gaming System." In: *Interactive Qualifying Project MLC-AAEZ*. DOI: 10.1109/NetGames.2014.7008964 (cit. on p. 244).

Davis, J., Smyth, C., & McDowell, K. (2010). "The Effects of Time Lag on Driving Performance and a Possible Mitigation." In: *IEEE Transactions on Robotics* 26.3, pp. 590–593. ISSN: 1941-0468. DOI: 10.1109/TRO.2010.2046695 (cit. on p. 51).

Deber, J., Araujo, B., Jota, R., Forlines, C., Leigh, D., Sanders, S., & Wigdor, D. (2016). "Hammer Time!: A Low-Cost, High Precision, High Accuracy Tool to Measure the Latency of Touchscreen Devices." en. In: ACM Press, pp. 2857–2868. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858394 (cit. on p. 96).

Jong, T. de (2019). “A bayesian approach to the correction for multiplicity.” In: PsyArXiv. DOI: https://doi.org/10.31234/osf.io/s56mk (cit. on pp. 108, 148).

Denisova, A., & Cairns, P. (2015a). “First Person vs. Third Person Perspective in Digital Games: Do Player Preferences Affect Immersion?” In: CHI ’15. Seoul, Republic of Korea: Association for Computing Machinery, pp. 145–148. ISBN: 9781450331456. DOI: 10.1145/2702123.2702256 (cit. on p. 171).

Denisova, A., & Cairns, P. (2015b). “The placebo effect in digital games: Phantom perception of adaptive artificial intelligence.” In: *Proceedings of the 2015 annual symposium on Computer-Human Interaction in Play*, pp. 23–33. DOI: 10.1145/2793107.2793109 (cit. on pp. 141, 195).

DESTINYbit (2022). *An exciting free-to-play Real Time Strategy game*. https://empiresapart.net/ (cit. on pp. 234, 236).

Di Luca, M. (2010). “New Method to Measure End-to-End Delay of Virtual Reality.” In: *Presence: Teleoperators and Virtual Environments* 19.6, pp. 569–584. ISSN: 1054-7460, 1531-3263. DOI: 10.1162/pres_a_00023 (cit. on p. 96).

DIN (2022). *DIN EN ISO 9241-220 Ergonomie der Mensch-System-Interaktion - Teil 220: Prozesse zur Ermöglichung, Durchführung und Bewertung menschzentrierter Gestaltung für interaktive Systeme in Hersteller- und Betreiberorganisationen (ISO 9241-220:2019); Deutsche Fassung EN ISO 9241-220:2019*. https://www.din.de/de/mitwirken/normenausschuesse/naerg/veroeffentlichungen/wdc-beuth:din21:289443385 (cit. on p. 16).

Diot, C., & Gautier, L. (1999). “A distributed architecture for multiplayer interactive applications on the Internet.” In: *IEEE Network* 13.4, pp. 6–15. DOI: 10.1109/65.777437 (cit. on pp. 77, 135).

Dong, L. (2013). “A target-predicting dr algorithm in roia cloud platform.” In: *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 25–28. DOI: 10.1109/3PGCIC.2013.11 (cit. on p. 85).

Dossena, F., & Trentini, A. (2022). “OpenLDAT—A system for the measurement of display latency metrics.” en. In: *Journal of the Society for Information Display*. ISSN: 1938-3657. DOI: 10.1002/jsid.1104 (cit. on p. 51).

Du, S., Li, T., Yang, Y., & Horng, S.-J. (2020). “Multivariate time series forecasting via attention-based encoder–decoder framework.” In: *Neurocomputing* 388, pp. 269–279. DOI: 10.1016/j.neucom.2019.12.118 (cit. on p. 256).

Duarte Jr, E. P., Pozo, A. T., & Beltrani, P. (2020). “Smart Reckoning: Reducing the traffic of online multiplayer games using machine learning for movement prediction.” In: *Entertainment Computing* 33, p. 100336. DOI: 10.1016/j.entcom.2019.100336 (cit. on p. 85).

Durnez, W., Zheleva, A., Claypool, M., Maes, M., Bombeke, K., Van Looy, J., & De Marez, L. (2021). "Spaz! The Effects of Local Latency on Player Actions in a Desktop-based Exergame." In: *IEEE Transactions on Games* 14.4, pp. 623–631. DOI: 10.1109/TG.2021.3128714 (cit. on pp. 18, 31, 63–65, 68, 69).

EA (2022). *Apex -Legends-*. https://www.ea.com/de-de/games/apex-legends (cit. on p. 142).

EarlyGame (2020). *Remote esports – is it sustainable?* https://www.earlygame.com/remote-esports-is-it-sustainable/ (cit. on p. 232).

Ebert, J. P., & Wegner, D. M. (2010). "Time warp: Authorship shapes the perceived timing of actions and events." In: *Consciousness and Cognition* 19.1, pp. 481–489. DOI: 10.1016/j.concog.2009.10.002 (cit. on p. 65).

Eg, R., Raaen, K., & Claypool, M. (2018). "Playing with delay: With poor timing comes poor performance, and experience follows suit." In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, pp. 1–6. DOI: 10.1109/QoMEX.2018.8463382 (cit. on pp. 18, 25, 31, 65, 67, 69, 196, 230).

Epic Games (2020). *Fortnite*. https://www.epicgames.com/fortnite/en-US/home/ (cit. on p. 194).

Eurogamer (2020). *Doom Eternal on Stadia looks great - but the lag is just too high*. https://www.eurogamer.net/articles/digitalfoundry-2020-doom-eternal-stadia-looks-the-part-but-lag-is-too-high (cit. on pp. 214, 219).

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). "The pascal visual object classes (voc) challenge." In: *International Journal of Computer Vision* 88, pp. 303–338. DOI: 10.1007/s11263-009-0275-4 (cit. on p. 250).

Everingham, M., Zisserman, A., Williams, C. K., Van Gool, L., Allan, M., Bishop, C. M., Chapelle, O., Dalal, N., Deselaers, T., Dorkó, G., et al. (2006). "The 2005 pascal visual object classes challenge." In: *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment: First PASCAL Machine Learning Challenges Workshop*. Springer, pp. 117–176 (cit. on pp. 250, 251).

Farrer, C., Valentin, G., & Hupé, J.-M. (2013). "The time windows of the sense of agency." In: *Consciousness and Cognition* 22.4, pp. 1431–1441. DOI: 10.1016/j.concog.2013.09.010 (cit. on p. 65).

Feurer, M., & Hutter, F. (2019). "Hyperparameter optimization." In: *Automated Machine Learning: Methods, Systems, Challenges*, pp. 3–33. DOI: 10.1007/978-3-030-05318-5_1 (cit. on p. 220).

Field, A. (2013). *Discovering statistics using IBM SPSS statistics*. sage, pp. 40–44. ISBN: 1446249182 (cit. on p. 104).

Firaxis (2022). *XCOM 2 Collections*. https://xcom.com/de-DE/ (cit. on p. 173).

Fish, C. (2021). *The History of Video Games*. White Owl. ISBN: 1526778971 (cit. on p. 31).

Fitts, P. M. (1954). "The information capacity of the human motor system in controlling the amplitude of movement." In: *Journal of Experimental Psychology* 47.6, p. 381. DOI: https://doi.org/10.1037/h0055392 (cit. on pp. 45, 174).

Flemisch, F., Schieben, A., Schoemig, N., Strauss, M., Lueke, S., & Heyden, A. (2011). "Design of human computer interfaces for highly automated vehicles in the EU-Project HAVEit." In: *Universal Access in Human-Computer Interaction. Context Diversity: 6th International Conference, UAHCI 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part III 6*. Springer, pp. 270–279. DOI: 10.1007/978-3-642-21666-4_30 (cit. on p. 26).

Forbes (2019). *Why 'Anthem' Failed And Why It Was Never Destined To Succeed*. https://www.forbes.com/sites/erikkain/2019/05/30/why-anthem-failed-and-why-it-was-never-destined-to-succeed/ (cit. on pp. 207, 230).

Force, I. E. T. (2020). *Transmission Control Protocol (TCP)*. https://www.ietf.org/rfc/rfc793.txt (cit. on p. 87).

Formosa, J., O'Donnell, N., Horton, E. M., Türkay, S., Mandryk, R. L., Hawks, M., & Johnson, D. (2022). "Definitions of Esports: A Systematic Review and Thematic Analysis." In: *Proceedings of the ACM on Human-Computer Interaction* 6.CHI PLAY, pp. 1–45. DOI: 10.1145/3549490 (cit. on pp. 53, 56).

Fridman, L., Terwilliger, J., & Jenik, B. (2018). "Deeptraffic: Crowdsourced hyperparameter tuning of deep reinforcement learning systems for multi-agent dense traffic navigation." In: *arXiv preprint arXiv:1801.02805*. DOI: 10.48550/arXiv.1801.02805 (cit. on p. 89).

Friston, S., Karlstrm, P., & Steed, A. (2016). "The Effects of Low Latency on Pointing and Steering Tasks." In: *IEEE Transactions on Visualization and Computer Graphics* 22.5, pp. 1605–1615. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2446467 (cit. on pp. 46, 48).

Glorot, X., Bordes, A., & Bengio, Y. (2011). "Deep sparse rectifier neural networks." In: *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323 (cit. on pp. 221, 238).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press (cit. on pp. 88, 237, 282, 283).

Google (2020). *Stadia - Game Streaming*. https://stadia.google.com/ (cit. on p. 214).

Gorisse, G., Christmann, O., Amato, E. A., & Richir, S. (2017). "First-and third-person perspectives in immersive virtual environments: presence and performance analysis of embodied users." In: *Frontiers in Robotics and AI* 4, p. 33. DOI: 10.3389/frobt.2017.00033 (cit. on p. 140).

Gormanley, S. (2013). “Audio immersion in games—a case study using an online game with background music and sound effects.” In: *The Computer Games Journal* 2.2, pp. 103–124. DOI: 10.1007/BF03392344 (cit. on pp. 33, 92, 142).

Grigorik, I. (2013). *High Performance Browser Networking: What every web developer should know about networking and web performance*. " O’Reilly Media, Inc." (cit. on p. 115).

Grimes, S. M., & Feenberg, A. (2009). “Rationalizing play: A critical theory of digital gaming.” In: *The Information Society* 25.2, pp. 105–118. DOI: 10.1080/01972240802701643 (cit. on p. 132).

Grimshaw, M., Lindley, C., & Nacke, L. (2008). “Sound and immersion in the first-person shooter: Mixed measurement of the player’s sonic experience.” In: *Audio Mostly-a conference on interaction with sound*. www. audiomostly. com (cit. on pp. 33, 92, 142).

Griwodz, C., & Halvorsen, P. (2006). “The fun of using TCP for an MMORPG.” In: *Proceedings of the 2006 International workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 1–7. DOI: 10.1145/1378191.1378193 (cit. on pp. 87, 91).

Gutwin, C., Benford, S., Dyck, J., Fraser, M., Vaghi, I., & Greenhalgh, C. (2004). “Revealing delay in collaborative environments.” In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 503–510. DOI: 10.1145/985692.985756 (cit. on pp. 19, 70–72, 91, 212).

habr.com, V. (2019). *Audio over Bluetooth: most detailed information about profiles, codecs, and devices*. URL: https://habr.com/en/post/456182/ (visited on 12/09/2021) (cit. on pp. 139, 156).

Haider, A., Harteveld, C., Johnson, D., Birk, M. V., Mandryk, R. L., Seif El-Nasr, M., Nacke, L. E., Gerling, K., & Vanden Abeele, V. (2022). “MiniPXI: Development and Validation of an Eleven-Item Measure of the Player Experience Inventory.” In: *Proc. ACM Hum.-Comput. Interact.* 6.CHI PLAY. DOI: 10.1145/3549507 (cit. on pp. 259, 261).

Halbhuber, D. (2022). “To Lag or Not to Lag: Understanding and Compensating Latency in Video Games.” In: *Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY ’22. Bremen, Germany: Association for Computing Machinery, pp. 370–373. ISBN: 9781450392112. DOI: 10.1145/3505270.3558364 (cit. on pp. 37, 42).

Halbhuber, D., Fehle, J., Kalus, A., Seitz, K., Kocur, M., Schmidt, T., & Wolff, C. (2019). “The Mood Game-How to Use the Player’s Affective State in a Shoot’em up Avoiding Frustration and Boredom.” In: *Proceedings of Mensch und Computer 2019*, pp. 867–870. DOI: 10.1145/3340764.3345369 (cit. on p. 41).

Halbhuber, D., Henze, N., & Schwind, V. (2021). “Increasing Player Performance and Game Experience in High Latency Systems.” In: *Proc. ACM Hum.-Comput. Interact.* 5.CHI PLAY, pp. 1–20. DOI: 10.1145/3474710 (cit. on pp. 40, 215).

Halbhuber, D., Höpfinger, J., Schwind, V., & Henze, N. (2022a). “A Dataset to Investigate First-Person Shooter Players.” In: *Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play*, pp. 51–56 (cit. on p. 41).

Halbhuber, D., Huber, M., Schwind, V., & Henze, N. (2022b). “Understanding Player Performance and Gaming Experience While Playing a First-Person Shooter with Auditory Latency.” In: *Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY ’22. Bremen, Germany: Association for Computing Machinery, pp. 24–30. ISBN: 9781450392112. DOI: 10.1145/3505270.3558333 (cit. on pp. 38, 142).

Halbhuber, D., Kocur, M., Kalus, A., Angermeyer, K., Schwind, V., & Henze, N. (2023a). “Understanding the Effects of Perceived Avatar Appearance on Latency Sensitivity in Full-Body Motion-Tracked Virtual Reality.” In: *Proceedings of Mensch Und Computer 2023*. MuC ’23. Rapperswil, Switzerland: Association for Computing Machinery, pp. 1–15. ISBN: 9798400707711. DOI: 10.1145/3603555.3603580 (cit. on pp. 41, 286, 287).

Halbhuber, D., Köhler, A., Schmidbauer, M., Wiese, J., & Henze, N. (2022c). “The Effects of Auditory Latency on Experienced First-Person Shooter Players.” In: *Proceedings of Mensch Und Computer 2022*. MuC ’22. Darmstadt, Germany: Association for Computing Machinery, pp. 286–296. ISBN: 9781450396905. DOI: 10.1145/3543758.3543760 (cit. on pp. 39, 155).

Halbhuber, D., Schauhuber, P., Schwind, V., & Henze, N. (2023b). “The Effects of Latency and In-Game Perspective on Player Performance and Game Experience.” In: *Proceedings of the ACM on Human-Computer Interaction* 7.CHI PLAY, pp. 1308–1329. DOI: 10.1145/3611070 (cit. on pp. 39, 171).

Halbhuber, D., Schlenczek, M., Bogon, J., & Henze, N. (2022d). “Better Be Quiet about It! The Effects of Phantom Latency on Experienced First-Person Shooter Players.” In: *Proceedings of the 21st International Conference on Mobile and Ubiquitous Multimedia*. MUM ’22. Lisbon, Portugal: Association for Computing Machinery, pp. 172–181. ISBN: 9781450398206. DOI: 10.1145/3568444.3568448 (cit. on pp. 39, 196).

Halbhuber, D., Schwind, V., & Henze, N. (2022e). “Don’t Break My Flow: Effects of Switching Latency in Shooting Video Games.” In: *Proc. ACM Hum.-Comput. Interact.* 6.CHI PLAY, pp. 1–20. DOI: 10.1145/3549492 (cit. on pp. 38, 115).

Halbhuber, D., Seewald, M., Schiller, F., Götz, M., Fehle, J., & Henze, N. (2022f). “Using Artificial Neural Networks to Compensate Negative Effects of Latency in Commercial Real-Time Strategy Games.” In: *Proceedings of Mensch Und Computer 2022*. MuC ’22. Darmstadt, Germany: Association for Computing Machinery, pp. 182–191. ISBN: 9781450396905. DOI: 10.1145/3543758.3543767 (cit. on pp. 40, 233).

Halbhuber, D., Thomaschke, R., Henze, N., Wolff, C., Probst, K., & Bogon, J. (2023c). "Play with my Expectations: Players Implicitly Anticipate Game Events Based on In-Game Time-Event Correlations." In: *Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia*, pp. 386–397. DOI: 10.1145/3626705.3627970 (cit. on p. 41).

Halbhuber, D., Wolff, C., & Henze, N. (n.d.[a]). "The Effects of Latency in Video Games: A Systematic Review and Thematic Analysis." In: *Currently in preparation* (cit. on pp. 37, 52).

Halbhuber, D., Wolff, C., Seewald, M., Schwind, V., & Henze, N. (n.d.[b]). "Leveling the Playing Field: Adaptive Latency Compensation in First-Person Shooters using Artificial Neural Networks." In: *Currently in preparation* (cit. on pp. 40, 249).

Hassenzahl, M., & Tractinsky, N. (2006). "User experience-a research agenda." In: *Behaviour & Information Technology* 25.2, pp. 91–97. DOI: 10.1080/01449290500330331 (cit. on p. 16).

Hayman, S. (1999). "The mcculloch-pitts model." In: *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339).* Vol. 6. IEEE, pp. 4438–4439. DOI: 10.1109/IJCNN.1999.830886 (cit. on p. 88).

He, D., Liu, F., Pape, D., Dawe, G., & Sandin, D. (2000). "Video-Based Measurement of System Latency." en. In: *International Immersive Projection Technology Workshop* 111, p. 6 (cit. on p. 96).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). "Mask r-cnn." In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969. DOI: 10.48550/arXiv.1703.06870 (cit. on p. 89).

Hendin, J., & Csikszentmihalyi, M. (1975). "Measuring the flow experience in rock dancing." In: *Beyond Boredom and Anxiety*, pp. 102–122 (cit. on pp. 131, 132).

Henze, N., Funk, M., & Shirazi, A. S. (2016). "Software-reduced touchscreen latency." In: *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 434–441. DOI: 10.1145/2935334.2935381 (cit. on pp. 17, 19, 20, 34, 89–91, 93, 121, 135, 212–215).

Henze, N., Halbhuber, D., & Wiese, J. (2023). *Faster Might Not be Fast Enough: Improving Users' Experience and Performance through Negative Latency*. Mensch und Computer 2023 - Workshopband. DOI: 10.18420/muc2023-mci-ws05-405 (cit. on p. 41).

Henze, N., Mayer, S., Le, H. V., & Schwind, V. (2017). "Improving software-reduced touchscreen latency." In: *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 1–8. DOI: 10.1145/3098279.3122150 (cit. on pp. 17, 90, 91, 212, 213).

Hesketh (2022). *Executive Assualt 2*. https://store.steampowered.com/app/510500/Executive_Assault_2/ (cit. on p. 170).

Higuchi, S., Motohashi, Y., Liu, Y., & Maeda, A. (2005). "Effects of playing a computer game using a bright display on presleep physiological variables, sleep latency, slow wave sleep and REM sleep." In: *Journal of Sleep Research* 14.3, pp. 267–273. DOI: 10.1111/j.1365-2869.2005.00463.x (cit. on p. 44).

Holm, S. (1979). "A Simple Sequentially Rejective Multiple Test Procedure." In: *Scandinavian Journal of Statistics* 6.2, pp. 65–70. ISSN: 03036898, 14679469 (cit. on p. 260).

Holmes, R., Tiwari, A., & Kennedy, J. (2016). "Mechanisms of the placebo effect in pain and psychiatric disorders." In: *The Pharmacogenomics Journal* 16.6, pp. 491–500. DOI: 10.1038/tpj.2016.15 (cit. on pp. 141, 194, 195).

Hutchins, E. L., Hollan, J. D., & Norman, D. A. (1985). "Direct manipulation interfaces." In: *Human–Computer Interaction* 1.4, pp. 311–338. DOI: 10.1207/s15327051hci0104_2 (cit. on pp. 21, 26, 27, 30).

IJsselsteijn, W. A., De Kort, Y. A., & Poels, K. (2013). *The game experience questionnaire*. Technische Universiteit Eindhoven (cit. on pp. 66, 119, 122, 123, 146, 157, 160, 162, 163, 198, 202, 205, 223, 226, 229, 241–244).

International Organization for Standardization (2007). *Ergonomics of human—system interaction Part 400: Principles and requirements for physical input devices*. International Organization for Standardization (cit. on p. 48).

Ishibashi, Y., & Kaneoka, H. (2005). "Fairness among game players in networked haptic environments influence of network latency." In: *2005 IEEE International Conference on Multimedia and Expo*. IEEE, 4–pp. DOI: 10.1109/ICME.2005.1521359 (cit. on p. 189).

Ivkovic, Z., Stavness, I., Gutwin, C., & Sutcliffe, S. (2015). "Quantifying and Mitigating the Negative Effects of Local Latencies on Aiming in 3D Shooter Games." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: Association for Computing Machinery, pp. 135–144. ISBN: 9781450331456. DOI: 10.1145/2702123.2702432 (cit. on pp. 23, 78, 79, 91, 101, 120, 145, 175).

Jaleel, J. A., Salim, S., & Aswin, R. (2012). "Artificial neural network based detection of skin cancer." In: *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* 1.3 (cit. on p. 89).

Jerald, J., Fuller, A., Lastra, A., Whitton, M., Kohli, L., & Brooks, F. (2007). "Latency compensation by horizontal scanline selection for head-mounted displays." In: *Stereoscopic Displays and Virtual Reality Systems XIV*. Vol. 6490. SPIE, pp. 568–578. DOI: 10.1117/12.710749 (cit. on p. 88).

Jocher, G., Stoken, A., Borovec, J., Chaurasia, A., Changyu, L., Hogan, A., Hajek, J., Diaconu, L., Kwon, Y., Defretin, Y., et al. (2021). "ultralytics/yolov5: v5. 0-YOLOv5-P6 1280 models, AWS, Supervise. ly and YouTube integrations." In: *Zenodo* (cit. on pp. 249–251).

Jörg, S., Normoyle, A., & Safonova, A. (2012). "How responsiveness affects players' perception in digital games." In: *Proceedings of the ACM symposium on Applied Perception*, pp. 33–38. DOI: 10.1145/2338676.2338683 (cit. on pp. 66, 69).

Jota, R., Ng, A., Dietz, P., & Wigdor, D. (2013). "How Fast is Fast Enough? A Study of the Effects of Latency in Direct-Touch Pointing Tasks." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: Association for Computing Machinery, pp. 2291–2300. ISBN: 9781450318990. DOI: 10.1145/2470654.2481317 (cit. on pp. 17, 24, 48, 49, 51, 196, 276).

K. Lee, S. W., & C. Chang, R. K. (2017). "On "shot around a corner" in first-person shooter games." In: *2017 15th Annual Workshop on Network and Systems Support for Games (NetGames)*, pp. 1–6. DOI: 10.1109/NetGames.2017.7991545 (cit. on pp. 34, 70, 91, 93, 212).

Kaaresoja, T., & Brewster, S. (2010). "Feedback is... Late: Measuring Multimodal Delays in Mobile Device Touchscreen Interaction." In: *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*. ICMI-MLMI '10. Beijing, China: Association for Computing Machinery. ISBN: 9781450304146. DOI: 10.1145/1891903.1891907 (cit. on pp. 17, 50, 96, 114, 142).

Kämäräinen, T., Siekkinen, M., Ylä-Jääski, A., Zhang, W., & Hui, P. (2017). "Dissecting the End-to-end Latency of Interactive Mobile Video Applications." en. In: *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*. Sonoma CA USA: ACM. ISBN: 978-1-4503-4907-9. DOI: 10.1145/3032970.3032985 (cit. on p. 96).

Kanervisto, A., Kinnunen, T., & Hautamaki, V. (2022). "GAN-Aimbots: Using Machine Learning for Cheating in First Person Shooters." In: *IEEE Transactions on Games*. DOI: 10.48550/arXiv.2205.07060 (cit. on p. 269).

Kaptchuk, T. J. (1998). "Powerful placebo: the dark side of the randomised controlled trial." In: *The Lancet* 351.9117, pp. 1722–1725. DOI: 10.1016/S0140-6736(97)10111-8 (cit. on pp. 194, 195, 207).

Kasahara, S., Konno, K., Owaki, R., Nishi, T., Takeshita, A., Ito, T., Kasuga, S., & Ushiba, J. (2017). "Malleable embodiment: changing sense of embodiment by spatial-temporal deformation of virtual human body." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 6438–6448. DOI: 10.1145/3025453.3025962 (cit. on pp. 17, 285, 286).

Kass, R. E., & Raftery, A. E. (1995). "Bayes factors." In: *Journal of the American Statistical Association* 90.430, pp. 773–795 (cit. on pp. 108, 148, 179).

Keetels, M., & Vroomen, J. (2012). "Exposure to delayed visual feedback of the hand changes motor-sensory synchrony perception." In: *Experimental Brain Research* 219, pp. 431–440. DOI: 10.1007/s00221-012-3081-0 (cit. on p. 65).

Kent, S. L. (2010). *The Ultimate History of Video Games, Volume 1: From Pong to Pokemon and Beyond... the Story Behind the Craze That Touched Our Lives and Changed the World*. Vol. 1. Crown (cit. on p. 30).

Kim, J., Knowles, P., Spjut, J., Boudaoud, B., & Mcguire, M. (2020). "Post-render warp with late input sampling improves aiming under high latency conditions." In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 3.2, pp. 1–18. DOI: 10.1145/3406187 (cit. on p. 73).

Kingma, D. P., & Ba, J. (2014). "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980*. DOI: 10.48550/arXiv.1412.6980 (cit. on pp. 221, 238, 255).

Klimmt, C., Blake, C., Hefner, D., Vorderer, P., & Roth, C. (2009). "Player performance, satisfaction, and video game enjoyment." In: *Entertainment Computing–ICEC 2009: 8th International Conference, Paris, France, September 3-5, 2009. Proceedings 8*. Springer, pp. 1–12. DOI: 10.1007/978-3-642-04052-8_1 (cit. on pp. 67, 112).

Kocur, M., Bogon, J., Mayer, M., Witte, M., Karber, A., Henze, N., & Schwind, V. (2022). "Sweating Avatars Decrease Perceived Exertion and Increase Perceived Endurance While Cycling in Virtual Reality." In: *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology*. VRST '22. Tsukuba, Japan: Association for Computing Machinery. ISBN: 9781450398893. DOI: 10.1145/3562939.3565628 (cit. on p. 286).

Kocur, M., Habler, F., Schwind, V., Woźniak, P. W., Wolff, C., & Henze, N. (2021). "Physiological and Perceptual Responses to Athletic Avatars While Cycling in Virtual Reality." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery. ISBN: 9781450380966. DOI: 10.1145/3411764.3445160 (cit. on p. 286).

Kocur, M., Kloss, M., Schwind, V., Wolff, C., & Henze, N. (2020). "Flexing muscles in virtual reality: effects of avatars' muscular appearance on physical performance." In: *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pp. 193–205. DOI: 10.1145/3410404.3414261 (cit. on p. 286).

Könings, K. D., Brand-Gruwel, S., & Van Merriënboer, J. J. (2005). "Towards more powerful learning environments through combining the perspectives of designers, teachers, and students." In: *British Journal of Educational Psychology* 75.4, pp. 645–660. DOI: 10.1348/000709905X43616 (cit. on p. 26).

Kosch, T., Welsch, R., Chuang, L., & Schmidt, A. (2022). "The Placebo Effect of Artificial Intelligence in Human-Computer Interaction." In: *ACM Trans. Comput.-Hum. Interact.* ISSN: 1073-0516. DOI: 10.1145/3529225 (cit. on pp. 33, 78, 93, 102, 141, 177, 194, 195, 206, 207).

Kotaku (2019). *Bug-Riddled Update Shows Why Fallout 76 Needs A Public Test Server*. https://kotaku.com/bug-riddled-update-shows-why-fallout-76-needs-a-public-1836457946 (cit. on pp. 207, 230).

Krafton Inc. (2020). *PUBG's Battlegrounds*. https://na.battlegrounds.pubg.com/ (cit. on p. 207).

Kruskal, W. H., & Wallis, W. A. (1952). "Use of ranks in one-criterion variance analysis." In: *Journal of the American Statistical Association* 47.260, pp. 583–621. DOI: 10.2307/2280779 (cit. on p. 147).

Larian Studios (2022). *Divinity 2: Original Sins*. https://divinity.game/ (cit. on p. 173).

Lavine, M., & Schervish, M. J. (1999). "Bayes factors: What they are and what they are not." In: *The American Statistician* 53.2, pp. 119–122. DOI: https://doi.org/10.2307/2685729 (cit. on pp. 108, 148, 179).

Le, H. V., Schwind, V., Göttlich, P., & Henze, N. (2017). "PredicTouch: A System to Reduce Touchscreen Latency Using Neural Networks and Inertial Measurement Units." In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, pp. 230–239. DOI: 10.1145/3132272.3134138 (cit. on pp. 34, 90, 91, 93, 214, 215).

Lee, C., Bonebrake, S., Bowman, D. A., & Höllerer, T. (2010). "The role of latency in the validity of AR simulation." In: *2010 IEEE Virtual Reality Conference (VR)*, pp. 11–18. DOI: 10.1109/VR.2010.5444820 (cit. on p. 17).

Lee, D., Kim, Y., Seshadri, V., Liu, J., Subramanian, L., & Mutlu, O. (2013). "Tiered-latency DRAM: A low latency and low cost DRAM architecture." In: *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, pp. 615–626. DOI: 10.1109/HPCA.2013.6522354 (cit. on p. 44).

Lee, I., Kim, S., & Lee, B. (2019). "Geometrically Compensating Effect of End-to-End Latency in Moving-Target Selection Games." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, pp. 1–12. ISBN: 9781450359702. DOI: 10.1145/3290605.3300790 (cit. on pp. 19, 34, 61, 69, 70, 79, 80, 91, 93, 212).

Lee, M. D., & Wagenmakers, E.-J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge university press. DOI: 10.1017/CBO9781139087759 (cit. on pp. 108, 148, 149).

Lee, S. W., & Chang, R. K. (2017). “On" shot around a corner" in first-person shooter games.” In: *2017 15th Annual Workshop on Network and Systems Support for Games (NetGames)*. IEEE, pp. 1–6. DOI: 10.1109/NetGames.2017.7991545 (cit. on pp. 62, 69).

Lee, S. W., & Chang, R. K. (2018). “Enhancing the experience of multiplayer shooter games via advanced lag compensation.” In: *Proceedings of the 9th ACM Multimedia Systems Conference*, pp. 284–293. DOI: 10.1145/3204949.3204971 (cit. on pp. 60, 69).

Lee, W.-K., & Chang, R. K. (2015). “Evaluation of lag-related configurations in first-person shooter games.” In: *2015 International Workshop on Network and Systems Support for Games (NetGames)*. IEEE, pp. 1–3. DOI: 10.1109/NetGames.2015.7382997 (cit. on pp. 83, 91).

Li, S., Chen, C., & Li, L. (2007). “A method of using personal habits for path prediction in network games.” In: *The First International Symposium on Data, Privacy, and E-Commerce (ISDPE 2007)*. IEEE, pp. 4143–4143. DOI: 10.1109/ISDPE.2007.113 (cit. on p. 85).

Li, Z., Melvin, H., Bruzgiene, R., Pocta, P., Skorin-Kapov, L., & Zgank, A. (2018). “Lag compensation for first-person shooter games in cloud gaming.” In: *Autonomous Control for a Reliable Internet of Services: Methods, Models, Approaches, Techniques, Algorithms, and Tools*. Springer International Publishing Cham, pp. 104–127. DOI: 10.1007/978-3-319-90415-3_5 (cit. on pp. 77, 135).

Liang, D., & Boustead, P. (2006). “Using Local Lag and Timewarp to Improve Performance for Real Life Multi-Player Online Games.” In: *Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games*. NetGames ’06. Singapore: Association for Computing Machinery, 37–es. ISBN: 1595935894. DOI: 10.1145/1230040.1230047 (cit. on pp. 76, 77, 91).

Liu, S., & Claypool, M. (2021). “EvLag: A Tool for Monitoring and Lagging Linux Input Devices.” In: *Proceedings of the 12th ACM Multimedia Systems Conference*. MMSys ’21. Istanbul, Turkey: Association for Computing Machinery, pp. 281–286. ISBN: 9781450384346. DOI: 10.1145/3458305.3478449 (cit. on p. 99).

Liu, S., Claypool, M., Kuwahara, A., Scovell, J., & Sherman, J. (2021a). “L33t or N00b? How Player Skill Alters the Effects of Network Latency on First Person Shooter Game Players.” In: *Proceedings of the Workshop on Game Systems (GameSys ’21)*. GameSys ’21. Istanbul, Turkey: Association for Computing Machinery, pp. 1–6. ISBN: 9781450384377. DOI: 10.1145/3458335.3460811 (cit. on pp. 61, 66, 68, 69, 101, 112, 152, 153, 155–157, 165, 277).

Liu, S., Claypool, M., Kuwahara, A., Scovell, J., & Sherman, J. (2021b). “The effects of network latency on competitive first-person shooter game players.” In: *13th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, pp. 151–156. DOI: 10.1109/QoMEX51781.2021.9465419 (cit. on pp. 66, 69, 103, 188).

Liu, S., Claypool, M., Kuwahara, A., Sherman, J., & Scovell, J. J. (2021c). “Lower is Better? The Effects of Local Latencies on Competitive First-Person Shooter Game Players.” In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–12. DOI: https://doi.org/10.1145/3411764.3445245 (cit. on pp. 18, 25, 58, 60, 62, 66, 69, 98, 110, 116, 122, 146, 155–158, 165, 166, 168–170, 188, 192, 196, 197, 200, 207, 214, 215, 244, 246).

Liu, S., Kuwahara, A., Scovell, J., Sherman, J., & Claypool, M. (2021d). “Comparing the Effects of Network Latency versus Local Latency on Competitive First Person Shooter Game Players.” In: *OSF Preprints*. DOI: 10.31219/osf.io/bhxcy (cit. on pp. 18, 23–25, 33, 58, 92, 98, 111, 122, 139, 146, 156, 169, 188, 191, 196–198, 200, 207).

Liu, S., Kuwahara, A., Scovell, J. J., & Claypool, M. (2023). “The Effects of Frame Rate Variation on Game Player Quality of Experience.” In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–10. DOI: 10.1145/3544548.3580665 (cit. on p. 31).

Liu, S., Xu, X., & Claypool, M. (2022). “A survey and taxonomy of latency compensation techniques for network computer games.” In: *ACM Computing Surveys (CSUR)* 54.11s, pp. 1–34. DOI: 10.1145/3519023 (cit. on pp. 34, 69, 71, 73–75, 81, 82, 91, 93, 190, 212).

Long, M., & Gutwin, C. (2018). “Characterizing and Modeling the Effects of Local Latency on Game Performance and Experience.” In: *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY ’18. Melbourne, VIC, Australia: Association for Computing Machinery, pp. 285–297. ISBN: 9781450356244. DOI: 10.1145/3242671.3242678 (cit. on pp. 18, 60, 103, 120, 145, 158, 168, 175, 176, 189–192, 200).

MacArthur, C., Grinberg, A., Harley, D., & Hancock, M. (2021). “You’re making me sick: A systematic review of how virtual reality research considers gender & cybersickness.” In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15. DOI: 10.1145/3411764.3445701 (cit. on pp. 53, 56).

MacKenzie, I. S., Sellen, A., & Buxton, W. A. (1991). “A comparison of input devices in element pointing and dragging tasks.” In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 161–166. DOI: 10.1145/108844.108868 (cit. on pp. 48, 51).

MacKenzie, I. S. (2013). *Human-Computer Interaction: An Empirical Research Perspective*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 0124058655 (cit. on pp. 16, 59).

MacKenzie, I. S., & Ware, C. (1993). "Lag as a Determinant of Human Performance in Interactive Systems." In: *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*. CHI '93. Amsterdam, The Netherlands: Association for Computing Machinery, pp. 488–493. ISBN: 0897915755. DOI: 10.1145/169059.169431 (cit. on pp. 17, 22, 24, 45, 48, 51, 96, 157, 158, 168, 192, 196, 200).

Mauve, M., Vogel, J., Hilt, V., & Effelsberg, W. (2004). "Local-lag and timewarp: Providing consistency for replicated continuous applications." In: *IEEE Transactions on Multimedia* 6.1, pp. 47–57. DOI: 10.1109/TMM.2003.819751 (cit. on pp. 75–77, 91).

McClintock, J. (2014). "Stereo Bluetooth and Low Latency Applications." In: *Audio Engineering Society Convention 137* (cit. on p. 156).

Mccoy, A., Ward, T., Mcloone, S., & Delaney, D. (2007). "Multistep-ahead neural-network predictors for network traffic reduction in distributed interactive applications." In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 17.4, 16–es. DOI: 10.1145/1276927.1276929 (cit. on p. 85).

McPhail, I. (2022). *Bluetooth Score and Test - Headphones*. https://www.rtings.com/headphones/tests/connectivity/bluetooth (cit. on p. 146).

Meehan, M., Razzaque, S., Whitton, M. C., & Brooks, F. P. (2003). "Effect of latency on presence in stressful virtual environments." In: *IEEE Virtual Reality, 2003. Proceedings.* IEEE, pp. 141–148. DOI: 10.1109/VR.2003.1191132 (cit. on pp. 17, 285, 286).

Michailidis, L., Balaguer-Ballester, E., & He, X. (2018). "Flow and immersion in video games: The aftermath of a conceptual challenge." In: *Frontiers in Psychology* 9, p. 1682. DOI: 10.3389/fpsyg.2018.01682 (cit. on p. 64).

Michalco, J., Simonsen, J. G., & Hornbæk, K. (2015). "An exploration of the relation between expectations and user experience." In: *International Journal of Human-Computer Interaction* 31.9, pp. 603–617. DOI: 10.1080/10447318.2015.1065696 (cit. on pp. 33, 93, 141, 195).

Microsoft (2021). *Forza 4 Horizon*. https://www.xbox.com/de-DE/games/forza-horizon-4 (cit. on p. 142).

Microsoft Games (2022). *Age of Empire 2 Definitive Edition*. https://www.ageofempires.com/games/aoeiide/ (cit. on p. 236).

Miller, R. B. (1968). "Response Time in Man-Computer Conversational Transactions." In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. AFIPS '68 (Fall, part I). San Francisco, California: Association for Computing Machinery, pp. 267–277. ISBN: 9781450378994. DOI: 10.1145/1476589.1476628 (cit. on pp. 43, 49, 50, 52).

Monteiro, D., Liang, H.-N., Xu, W., Brucker, M., Nanjappan, V., & Yue, Y. (2018). "Evaluating enjoyment, presence, and emulator sickness in VR games based on first-and third-person viewing perspectives." In: *Computer Animation and Virtual Worlds* 29.3-4, e1830. DOI: 10.1002/cav.1830 (cit. on pp. 140, 171).

Montgomery, G., & Kirsch, I. (1996). "Mechanisms of placebo pain reduction: an empirical investigation." In: *Psychological Science* 7.3, pp. 174–176. DOI: 10.1111/j.1467-9280.1996.tb00352.x (cit. on pp. 141, 194, 195).

Morey, R. D., & Rouder, J. N. (2022). *BayesFactor: Computation of Bayes Factors for Common Designs*. R package version 0.9.12-4.4 (cit. on p. 179).

Nabiyouni, M., Scerbo, S., Bowman, D. A., & Höllerer, T. (2017). "Relative effects of real-world and virtual-world latency on an augmented reality training task: an ar simulation experiment." In: *Frontiers in ICT* 3, p. 34. DOI: 10.3389/fict.2016.00034 (cit. on p. 17).

Nerpa Tech (n.d.). *USB Host Library Rev. 2.0*. https://github.com/felis/USB_Host_Shield_2.0 (cit. on p. 258).

Ng, A., Annett, M., Dietz, P., Gupta, A., & Bischof, W. F. (2014). "In the Blink of an Eye: Investigating Latency Perception during Stylus Interaction." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, pp. 1103–1112. ISBN: 9781450324731. DOI: 10.1145/2556288.2557037 (cit. on pp. 17, 49, 51, 220).

Ng, A., & Dietz, P. H. (2013). "The need for speed in touch systems." In: *SID Symposium Digest of Technical Papers*. Vol. 44. 1. Wiley Online Library, pp. 547–550. DOI: 10.1002/j.2168-0159.2013.tb06267.x (cit. on p. 49).

Ng, A., Lepinski, J., Wigdor, D., Sanders, S., & Dietz, P. (2012). "Designing for Low-Latency Direct-Touch Input." In: *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. UIST '12. Cambridge, Massachusetts, USA: Association for Computing Machinery, pp. 453–464. ISBN: 9781450315807. DOI: 10.1145/2380116.2380174 (cit. on pp. 17, 24).

Noordally, R., Nicolay, X., Anelli, P., Lorion, R., & Tournoux, P. U. (2016). "Analysis of internet latency: the reunion island case." In: *Proceedings of the 12th Asian Internet Engineering Conference*, pp. 49–56. DOI: 10.1145/3012695.3012702 (cit. on p. 114).

Norman, K. L. (1991). *The psychology of menu selection: Designing cognitive control at the human/computer interface*. Intellect Books (cit. on pp. 27, 28).

Normoyle, A., Guerrero, G., & Jörg, S. (2014). "Player perception of delays and jitter in character responsiveness." In: *Proceedings of the ACM Symposium on Applied Perception*, pp. 117–124. DOI: 10.1145/2628257.2628263 (cit. on pp. 60, 66, 69).

OpenCV Team (2022). *OpenCV – 4.5.5*. https://opencv.org/ (cit. on pp. 235, 252).

Orland, K. (2022). *Warzone dev says game is losing players over "insane" download sizes*. https://arstechnica.com/gaming/2022/03/warzone-dev-says-game-is-losing-players-over-insane-download-sizes/ (cit. on p. 87).

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., et al. (2021). "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews." In: *Systematic Reviews* 10.1, pp. 1–11. DOI: 10.1016/j.ijsu.2021.105906 (cit. on pp. 52, 53, 68).

Palmér, M. (2022). *pynput 1.7.6*. https://pypi.org/project/pynput/ (cit. on p. 235).

Pantel, L., & Wolf, L. C. (2002). "On the suitability of dead reckoning schemes for games." In: *Proceedings of the 1st workshop on Network and System Support for Games*, pp. 79–84. DOI: 10.1145/566500.566512 (cit. on pp. 34, 84, 91, 93).

Park, K. S., & Kenyon, R. V. (1999). "Effects of network characteristics on human performance in a collaborative virtual environment." In: *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. IEEE, pp. 104–111. DOI: 10.1109/VR.1999.756940 (cit. on p. 82).

Pavlovych, A., & Stuerzlinger, W. (2009). "The tradeoff between spatial jitter and latency in pointing tasks." en. In: ACM Press, p. 187. ISBN: 978-1-60558-600-7. DOI: 10.1145/1570433.1570469 (cit. on p. 48).

PCGamer (2019). *Here's how Stadia's input lag compares to native PC gaming*. https://www.pcgamer.com/heres-how-stadias-input-lag-compares-to-native-pc-gaming/ (cit. on pp. 214, 219).

PCMag UK (2015). *OnLive Streaming Service Shutting Down at End of April*. https://uk.pcmag.com/gaming-systems/40903/onlive-streaming-service-shutting-down-at-end (cit. on p. 214).

Petlund, A., Evensen, K., Halvorsen, P., & Griwodz, C. (2008). "Improving application layer latency for reliable thin-stream game traffic." In: *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, pp. 91–96. DOI: 10.1145/1517494.1517513 (cit. on pp. 87, 91).

Prechelt, L. (1998). "Early stopping-but when?" In: *Neural Networks: Tricks of the trade*. Springer, pp. 55–69. DOI: 10.1007/978-3-642-35289-8_5 (cit. on p. 238).

Quax, P., Beznosyk, A., Vanmontfort, W., Marx, R., & Lamotte, W. (2013). "An evaluation of the impact of game genre on user experience in cloud gaming." In: *2013 IEEE International Games Innovation Conference (IGIC)*. IEEE, pp. 216–221. DOI: 10.1109/IGIC.2013.6659141 (cit. on p. 66).

Quax, P., Monsieurs, P., Lamotte, W., De Vleeschauwer, D., & Degrande, N. (2004). "Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game." In: *Proceedings of 3rd ACM SIGCOMM workshop on Network and System Support for Games*, pp. 152–156. DOI: 10.1145/1016540.1016557 (cit. on pp. 18, 60, 62, 69).

Quinn, J., McEachen, J., Fullan, M., Gardner, M., & Drummy, M. (2019). *Dive into deep learning: Tools for engagement*. Corwin Press. ISBN: 1544361378 (cit. on p. 250).

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria (cit. on p. 179).

R., T. (2021). *Android's Bluetooth latency needs a serious overhaul*. URL: https://www.soundguys.com/android-bluetooth-latency-22732/ (cit. on p. 157).

Raaen, K., & Eg, R. (2015). "Instantaneous human-computer interactions: Button causes and screen effects." In: *Human-Computer Interaction: Users and Contexts: 17th International Conference, HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings, Part III 17*. Springer, pp. 492–502. DOI: 10.1007/978-3-319-21006-3_47 (cit. on p. 65).

Raaen, K., & Grønli, T.-M. (2014). "Latency thresholds for usability in games: A survey." In: *Norsk IKT-konferanse for forskning og utdanning* (cit. on p. 68).

Ravindran, K., Sabbir, A., & Ravindran, B. (2008). "Impact of network loss/delay characteristics on consistency control in real-time multi-player games." In: *2008 5th IEEE Consumer Communications and Networking Conference*. IEEE, pp. 1128–1133. DOI: 10.1109/ccnc08.2007.254 (cit. on p. 63).

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). "You only look once: Unified, real-time object detection." In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 779–788. DOI: 10.48550/arXiv.1506.02640 (cit. on pp. 249, 250).

Rethlefsen, M. L., Kirtley, S., Waffenschmidt, S., Ayala, A. P., Moher, D., Page, M. J., & Koffel, J. B. (2021). "PRISMA-S: an extension to the PRISMA statement for reporting literature searches in systematic reviews." In: *Systematic Reviews* 10.1, pp. 1–19. DOI: 10.1186/s13643-020-01542-z (cit. on p. 53).

Riemer, M., Bogon, J., Rußwinkel, N., Henze, N., Wiese, E., Halbhuber, D., & Thomaschke, R. (2023). *Time and Timing in Human-Computer Interaction*. Mensch und Computer 2023 - Workshopband. DOI: 10.18420/muc2023-mci-ws05-106 (cit. on p. 41).

Ries, M., Svoboda, P., & Rupp, M. (2008). "Empirical study of subjective quality for massive multiplayer games." In: *2008 15th International Conference on Systems, Signals and Image Processing*. IEEE, pp. 181–184. DOI: 10.1109/IWSSIP.2008.4604397 (cit. on pp. 61, 64, 69).

Riot (2021). *League of Legends*. https://www.leagueoflegends.com/de-de/ (cit. on p. 142).

Roboflow (2022). *CSGO TRAIN YOLO V5 Dataset*. https://universe.roboflow.com/new-workspace-rp0z0/csgo-train-yolo-v5. Open Source Dataset (cit. on p. 249).

Rogers, R., Bowman, N. D., & Oliver, M. B. (2015). "It's not the model that doesn't fit, it's the controller! The role of cognitive skills in understanding the links between natural mapping, performance, and enjoyment of console video games." In: *Computers in Human Behavior* 49, pp. 588–596. DOI: 10.1016/j.chb.2015.03.027 (cit. on p. 205).

Rohde, M., & Ernst, M. O. (2013). "To lead and to lag–forward and backward recalibration of perceived visuo-motor simultaneity." In: *Frontiers in Psychology* 3, p. 599. DOI: 10.3389/fpsyg.2012.00599 (cit. on p. 65).

Sabet, S. S., Eg, R., Raaen, K., Oasim, M., Riegler, M., & Halvorsen, P. (2022). "When Every Millisecond Counts: The Impact of Delay in VR Gaming." In: *2022 14th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, pp. 1–4. DOI: 10.1109/QoMEX55416.2022.9900908 (cit. on pp. 64, 67, 69).

Sabet, S. S., Griwodz, C., & Möller, S. (2019a). "Influence of primacy, recency and peak effects on the game experience questionnaire." In: *Proceedings of the 11th ACM Workshop on Immersive Mixed and Virtual Environment Systems*, pp. 22–27. DOI: 10.1145/3304113.3326113 (cit. on p. 189).

Sabet, S. S., Schmidt, S., Griwodz, C., & Möller, S. (2019b). "Towards the impact of gamers' adaptation to delay variation on gaming quality of experience." In: *Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, pp. 1–6. DOI: 10.1109/QoMEX.2019.8743239 (cit. on pp. 33, 92, 166).

Sabet, S. S., Schmidt, S., Zadtootaghaj, S., Griwodz, C., & Möller, S. (2020a). "Delay Sensitivity Classification of Cloud Gaming Content." In: *Proceedings of the 12th ACM International Workshop on Immersive Mixed and Virtual Environment Systems*. MMVE '20. Istanbul, Turkey: Association for Computing Machinery, pp. 25–30. ISBN: 9781450379472. DOI: 10.1145/3386293.3397116 (cit. on pp. 25, 68, 151, 170, 190, 274, 276).

Sabet, S. S., Schmidt, S., Zadtootaghaj, S., Naderi, B., Griwodz, C., & Möller, S. (2020b). "A latency compensation technique based on game characteristics to mitigate the influence of delay on cloud gaming quality of experience." In: *Proceedings of the 11th ACM Multimedia Systems Conference*, pp. 15–25. DOI: https://doi.org/10.1145/3339825.3391855 (cit. on pp. 61, 62, 69, 110, 230).

Salay, R., & Claypool, M. (2020). "A comparison of automatic versus manual world alteration for network game latency compensation." In: *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play*, pp. 355–359. DOI: 10.1145/3383668.3419910 (cit. on pp. 71, 80).

San Roque, L., Kendrick, K. H., Norcliffe, E., Brown, P., Defina, R., Dingemanse, M., Dirksmeyer, T., Enfield, N. J., Floyd, S., Hammond, J., et al. (2015). "Vision verbs dominate in conversation across cultures, but the ranking of non-visual verbs varies." In: *Cognitive Linguistics* 26.1, pp. 31–60. DOI: 10.1515/cog-2014-0089 (cit. on pp. 20, 151).

Sanders, T., & Cairns, P. (2010). "Time perception, immersion and music in videogames." In: *Proceedings of HCI 2010*, pp. 160–167. DOI: 10.5555/2146303.2146327 (cit. on p. 64).

Savery, C., Graham, N., Gutwin, C., & Brown, M. (2014). "The effects of consistency maintenance methods on player experience and performance in networked games." In: *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 1344–1355. DOI: 10.1145/2531602.2531616 (cit. on pp. 66, 83, 91).

Schank, R. C. (1991). "Where's the AI?" In: *AI magazine* 12.4, pp. 38–38. DOI: 10.1609/aimag.v12i4.917 (cit. on p. 26).

Schmid, A., Halbhuber, D., Fischer, T., Wimmer, R., & Henze, N. (2023). "Small Latency Variations Do Not Affect Player Performance in First-Person Shooters." In: *Proc. ACM Hum.-Comput. Interact.* 7.CHI PLAY, pp. 197–216. DOI: 10.1145/3611027 (cit. on pp. 37, 96).

Schmid, A., & Wimmer, R. (2021). *Yet Another Latency Measuring Device*. preprint. Open Science Framework. DOI: 10.31219/osf.io/tkghj (cit. on pp. 96, 100).

Schmid, A., & Wimmer, R. (2023). "Measuring the Latency of Graphics Frameworks on X11-Based Systems." In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI EA '23. Hamburg, Germany: Association for Computing Machinery. ISBN: 9781450394222. DOI: 10.1145/3544549.3585779 (cit. on p. 51).

Schmidt, S., Zadtootaghaj, S., & Möller, S. (2017). "Towards the delay sensitivity of games: There is more than genres." In: *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, pp. 1–6. DOI: 10.1109/QoMEX.2017.7965676 (cit. on pp. 33, 68, 69, 92, 140, 170, 188, 190, 274, 277).

Schmidt, S., Zadtootaghaj, S., Sabet, S. S., & Möller, S. (2021a). "Modeling and understanding the quality of experience of online mobile gaming services." In: *2021 13th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, pp. 157–162. DOI: 10.1109/QoMEX51781.2021.9465431 (cit. on pp. 63, 64).

Schmidt, T., Engl, I., Halbhuber, D., & Wolff, C. (2021b). "Comparing live sentiment annotation of movies via arduino and a slider with textual annotation of subtitles." In: *Post-Proceedings of the 5th Conference Digital Humanities in the Nordic Countries (DHN 2020)*. Vol. 2865. CEUR Workshop Proceedings, pp. 212–223 (cit. on p. 41).

Schmierbach, M., Chung, M.-Y., Wu, M., & Kim, K. (2014). "No one likes to lose: The effect of game difficulty on competency, flow, and enjoyment." In: *Journal of Media Psychology: Theories, Methods, and Applications* 26.3, p. 105. DOI: 10.1027/1864-1105/a000120 (cit. on p. 205).

Schwind, V., Halbhuber, D., Fehle, J., Sasse, J., Pfaffelhuber, A., Tögel, C., Dietz, J., & Henze, N. (2020). "The Effects of Full-Body Avatar Movement Predictions in Virtual Reality Using Neural Networks." In: *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology*. VRST '20. Virtual Event, Canada: Association for Computing Machinery. ISBN: 9781450376198. DOI: 10.1145/3385956.3418941 (cit. on pp. 285–287).

Seow, S. C. (2008). *Designing and engineering time: The psychology of time perception in software*. Addison-Wesley Professional (cit. on p. 49).

Shadow (2020). *Shadow - Verwandle deine Geräte in einen Gaming PC*. https://shadow.tech/ (cit. on p. 214).

Shannon, C. E. (1948). "A mathematical theory of communication." In: *The Bell System Technical Journal* 27.3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x (cit. on p. 45).

Shea, R., Liu, J., Ngai, E. C.-H., & Cui, Y. (2013). "Cloud gaming: architecture and performance." In: *IEEE Network* 27.4, pp. 16–21. DOI: 10.1109/MNET.2013.6574660 (cit. on p. 114).

Sheridan, T., & Ferrell, W. (1963). "Remote Manipulative Control with Transmission Delay." In: *IEEE Transactions on Human Factors in Electronics* HFE-4.1, pp. 25–29. DOI: 10.1109/THFE.1963.231283 (cit. on pp. 43, 46–48).

Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface: Strategies for effective human-computer interaction*. Pearson Education India. ISBN: 0201694972 (cit. on pp. 49, 50, 52).

Simon, L. S., Vimond, A., & Vincent, E. (2013). "Effects of audio latency in a disc jockey interface." In: *Proceedings of Meetings on Acoustics ICA2013*. Vol. 19. 1. Acoustical Society of America, p. 035037. DOI: 10.1121/1.4800187 (cit. on p. 142).

Smed, J., Kaukoranta, T., & Hakonen, H. (2002). "Aspects of networking in multiplayer computer games." In: *The Electronic Library* 20.2, pp. 87–97. DOI: 10.1108/02640470210424392 (cit. on p. 23).

Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K., Cai, S., Nielsen, E., Soergel, D., et al. (2019). "Tensorflow. js: Machine learning for the web and beyond." In: *arXiv preprint arXiv:1901.05350*. DOI: https://doi.org/10.48550/arXiv.1901.05350 (cit. on p. 222).

Smit, F., Liere, R. van, Beck, S., & Fröhlich, B. (2009). "An image-warping architecture for vr: Low latency versus image quality." In: *2009 IEEE Virtual Reality Conference*. IEEE, pp. 27–34. DOI: 10.1109/VR.2009.4810995 (cit. on pp. 73, 91).

So, R. H., & Griffin, M. J. (1992). "Compensating lags in head-coupled displays using head position prediction and image deflection." In: *Journal of Aircraft* 29.6, pp. 1064–1068. DOI: 10.2514/3.46285 (cit. on p. 73).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). "Dropout: a simple way to prevent neural networks from overfitting." In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958. DOI: 10.5555/2627435.2670313 (cit. on pp. 221, 239).

Stadler, P., Schmid, A., & Wimmer, R. (2020). "DispLagBox: simple and replicable high-precision measurements of display latency." en. In: *Proceedings of the Conference on Mensch und Computer*. MuC '20. New York, NY, USA: Association for Computing Machinery, pp. 105–108. ISBN: 978-1-4503-7540-5. DOI: 10.1145/3404983.3410015 (cit. on p. 51).

Statista (2021a). *DOTA 2 The International championships prize pool from 2011 to 2019*. https://www.statista.com/statistics/749033/dota-2-championships-prize-pool/ (cit. on p. 232).

Statista (2021b). *Mobile gaming share of total gaming revenue worldwide from 2012 to 2021*. https://www.statista.com/statistics/260167/mobile-gaming-share-of-total-gaming-revenue-worldwide/ (cit. on p. 114).

Statista (2023a). *Distribution of video gamers in Germany in 2023, by age*. https://www.statista.com/statistics/1112939/video-gamers-age-distribution-germany/ (cit. on p. 283).

Statista (2023b). *Share of digital vs. non-digital revenue of major video game publishers worldwide in most recent fiscal year as of May 2023*. https://www.statista.com/statistics/234664/digital-vs-non-digital-reven (cit. on p. 31).

Stern, M. K., & Johnson, J. H. (2010). "Just noticeable difference." In: *The Corsini Encyclopedia of Psychology*, pp. 1–2. DOI: 10.1002/9780470479216.corpsy0481 (cit. on p. 48).

Studios Drama (2022). *Unrecord*. https://studiosdrama.com/ (cit. on p. 32).

Sun, J., & Claypool, M. (2019). "Evaluating streaming and latency compensation in a cloud-based game." In: *Proceedings of the 15th IARIA Advanced International Conference on Telecommunications (AICT)* (cit. on pp. 75, 76, 91, 214).

Sun, W. (2019). *Adaptive bitrate streaming latency reduction*. US Patent 10,432,982 (cit. on pp. 34, 86, 93).

Sutherland, I. E. (1964). "Sketch pad a man-machine graphical communication system." In: *Proceedings of the SHARE design automation workshop*, pp. 6–329. DOI: 10.1145/800265.810742 (cit. on p. 44).

Suznjevic, M., Skorin-Kapov, L., & Matijasevic, M. (2013). "The impact of user, system, and context factors on gaming QoE: A case study involving MMORPGs." In: *2013 12th Annual Workshop on Network and Systems Support for Games (NetGames)*. IEEE, pp. 1–6. DOI: 10.1109/NetGames.2013.6820606 (cit. on p. 68).

Teather, R. J., Pavlovych, A., Stuerzlinger, W., & MacKenzie, I. S. (2009). "Effects of tracking technology, latency, and spatial jitter on object movement." In: *2009 IEEE Symposium on 3D User Interfaces*, pp. 43–50. DOI: 10.1109/3DUI.2009.4811204 (cit. on p. 48).

Tensor Flow (2015). *TensorFlow*. https://www.tensorflow.org/tutorials/structured_data/time_series (cit. on p. 220).

Thomaschke, R., Wagener, A., Kiesel, A., & Hoffmann, J. (2011). "The scope and precision of specific temporal expectancy: Evidence from a variable foreperiod paradigm." In: *Attention, Perception, & Psychophysics* 73.3, pp. 953–964. DOI: 10.3758/s13414-010-0079-1 (cit. on p. 276).

Tondello, G. F., Wehbe, R. R., Diamond, L., Busch, M., Marczewski, A., & Nacke, L. E. (2016). "The gamification user types hexad scale." In: *Proceedings of the 2016 annual symposium on Computer-Human Interaction in Play*, pp. 229–243. DOI: 10.1145/2967934.2968082 (cit. on p. 283).

Toprac, P., & Abdel-Meguid, A. (2011). "Causing fear, suspense, and anxiety using sound design in computer games." In: *Game sound technology and player interaction: Concepts and developments*. IGI Global, pp. 176–191. DOI: 10.4018/978-1-61692-828-5.ch009 (cit. on p. 142).

Tseng, P.-H., Wang, N.-C., Lin, R.-M., & Chen, K.-T. (2011). "On the battle between lag and online gamers." In: *2011 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*. IEEE, pp. 1–6. DOI: 10.1109/CQR.2011.5996093 (cit. on pp. 67, 69).

Tungsteno (2022). *pyWinhook 1.6.2*. https://pypi.org/project/pyWinhook/ (cit. on p. 235).

Unity 3D (2023). *Cube*. https://docs.unity3d.com/ScriptReference/ (cit. on p. 178).

Vaghi, I., Greenhalgh, C., & Benford, S. (1999). "Coping with inconsistency due to network delays in collaborative virtual environments." In: *Proceedings of the ACM symposium on Virtual Reality Software and Technology*, pp. 42–49. DOI: 10.1145/323663.323670 (cit. on pp. 58, 71, 91).

Valve (2022). *Steam - The ultimate destination for playing, discussing, and creating games.* https://store.steampowered.com/ (cit. on p. 235).

Wel, R. P. van der, Becchio, C., Curioni, A., & Wolf, T. (2021). "Understanding joint action: Current theoretical and empirical approaches." In: *Acta Psychologica* 215, p. 103285. ISSN: 00016918. DOI: 10.1016/j.actpsy.2021.103285 (cit. on p. 25).

Vanden Abeele, V., Nacke, L. E., Mekler, E. D., & Johnson, D. (2016). "Design and preliminary validation of the player experience inventory." In: *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pp. 335–341. DOI: https://doi.org/10.1145/2968120.2987744 (cit. on p. 262).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). "Attention is all you need." In: *Advances in Neural Information Processing Systems* 30. DOI: 10.48550/arXiv.1706.03762 (cit. on p. 253).

Video Games Chronicle (2020). *Warcraft 3: Reforged is now the worst user scored game ever on Metacritic*. https://www.videogameschronicle.com/news/warcraft-3-reforged-is-now-the-worst-user-scored-game-ever-on-metacritic/ (cit. on pp. 207, 230).

Volberg, G., & Thomaschke, R. (2017). "Time-based expectations entail preparatory motor activity." In: *Cortex* 92, pp. 261–270. DOI: 10.1016/j.cortex.2017.04.019 (cit. on p. 276).

Vollebreg, B. (2022). *Auto PY to EXE*. https://pypi.org/project/auto-py-to-exe/ (cit. on p. 235).

Wagener, A., & Hoffmann, J. (2010). "Temporal cueing of target-identity and target-location." In: *Experimental Psychology*. DOI: 10.1027/1618-3169/a000054 (cit. on p. 275).

Wagenmakers, E.-J. (2022). *JASP - A Fresh Way to Do Statistics*. https://jasp-stats.org/ (cit. on pp. 107, 148).

Wagenmakers, E.-J., Love, J., Marsman, M., Jamil, T., Ly, A., Verhagen, J., Selker, R., Gronau, Q. F., Dropmann, D., Boutin, B., et al. (2018). "Bayesian inference for psychology. Part II: Example applications with JASP." In: *Psychonomic Bulletin & Review* 25.1, pp. 58–76. DOI: 10.3758/s13423-017-1323-7 (cit. on pp. 107, 148, 149, 179).

Wahab, A., Ahmad, N., Martini, M. G., & Schormans, J. (2021). "Subjective quality assessment for cloud gaming." In: *J — Multidisciplinary Scientific Journal* 4.3, pp. 404–419. DOI: 10.3390/j4030031 (cit. on pp. 58, 65, 68, 69).

Waltemate, T., Senna, I., Hülsmann, F., Rohde, M., Kopp, S., Ernst, M., & Botsch, M. (2016). "The impact of latency on perceptual judgments and motor performance in closed-loop interaction in virtual reality." In: *Proceedings of the 22nd ACM conference on Virtual Reality Software and Technology*, pp. 27–35. DOI: 10.1145/2993369.2993381 (cit. on p. 65).

Wei, D., Yang, C., Zhang, X. (, & Yuan, X. (2021). "Predicting Mouse Click Position Using Long Short-Term Memory Model Trained by Joint Loss Function." In: *Extended Abstracts of the*

2021 CHI Conference on Human Factors in Computing Systems. CHI EA '21. Yokohama, Japan: Association for Computing Machinery. ISBN: 9781450380959. DOI: 10.1145/3411763.3451651 (cit. on p. 255).

Weiner, B. (1985). "An attributional theory of achievement motivation and emotion." In: *Psychological Review* 92.4, p. 548. DOI: 10.1037/0033-295X.92.4.548 (cit. on p. 67).

Westfall, P. H., Johnson, W. O., & Utts, J. M. (1997). "A Bayesian perspective on the Bonferroni adjustment." In: *Biometrika* 84.2, pp. 419–427. DOI: 10.1093/biomet/84.2.419 (cit. on pp. 108, 148, 150).

Wiese, J., & Henze, N. (2023). "Predicting Mouse Positions Beyond a System's Latency Can Increase Throughput and User Experience in Linear Steering Tasks." In: *Proceedings of Mensch Und Computer 2023*. MuC '23. New York, NY, USA: Association for Computing Machinery, pp. 101–115. ISBN: 9798400707711. DOI: 10.1145/3603555.3603556 (cit. on p. 284).

Wikipedia (2023a). *Dead Reackoning*. https://en.wikipedia.org/wiki/Dead_reckoning (cit. on pp. 84, 91).

Wikipedia (2023b). *Graphics Pipeline*. https://en.wikipedia.org/wiki/Graphics_pipeline (cit. on p. 87).

Wikipedia (2023c). *Linear interpolation*. https://en.wikipedia.org/wiki/Linear_interpolation (cit. on p. 83).

Wikipedia (2023d). *Nintendo 64*. https://en.wikipedia.org/wiki/Nintendo_64 (cit. on p. 31).

Wikipedia (2023e). *Pac-Man*. https://en.wikipedia.org/wiki/Pac-Man (cit. on p. 32).

Wikipedia (2023f). *PlayStation*. https://en.wikipedia.org/wiki/PlayStation (cit. on p. 30).

Wikstrand, G., Schedin, L., & Elg, F. (2004). "High and Low Ping and the Game of Pong." In: *Proceedings of the ACM Workshop on Network and System Support for Games (NetGames'04)* 29, p. 30 (cit. on p. 72).

Wimmer, R., Schmid, A., & Bockes, F. (2019). "On the Latency of USB-Connected Input Devices." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12. DOI: 10.1145/3290605.3300650 (cit. on pp. 22, 23, 50, 96, 100, 114, 172).

Winkler, P., Stiens, P., Rauh, N., Franke, T., & Krems, J. (2020). "How latency, action modality and display modality influence the sense of agency: a virtual reality study." In: *Virtual Reality* 24, pp. 411–422. DOI: 10.1007/s10055-019-00403-y (cit. on pp. 65, 102).

Wobbrock, J. O., Findlater, L., Gergle, D., & Higgins, J. J. (2011). "The aligned rank transform for nonparametric factorial analyses using only anova procedures." In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 143–146. DOI: https://doi.org/10.1145/1978942.1978963 (cit. on pp. 104, 179).

Wobbrock, J. O., & Kientz, J. A. (2016). "Research contributions in human-computer interaction." In: *Interactions* 23.3, pp. 38–44. DOI: 10.1145/2907069 (cit. on p. 35).

Woods, D. L., Wyma, J. M., Yund, E. W., Herron, T. J., & Reed, B. (2015). "Age-related slowing of response selection and production in a visual choice reaction time task." In: *Frontiers in Human Neuroscience* 9, p. 193. DOI: https://doi.org/10.3389/fnhum.2015.00193 (cit. on p. 283).

Xiao, H., Rasul, K., & Vollgraf, R. (2017). "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." In: *arXiv preprint arXiv:1708.07747*. DOI: 10.48550/arXiv.1708.07747 (cit. on p. 89).

Xu, J., & Wah, B. W. (2013). "Concealing network delays in delay-sensitive online interactive games based on just-noticeable differences." In: *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, pp. 1–6. DOI: 10.1109/ICME.2013.6607526 (cit. on p. 79).

Xu, X., Liu, S., & Claypool, M. (2022). "The Effects of Network Latency on Counter-strike: Global Offensive Players." In: *2022 14th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, pp. 1–6. DOI: 10.1109/QoMEX55416.2022.9900915 (cit. on pp. 67–69).

Ye, D., He, J., Hu, W., & Liu, J. (2018). "Measurement and analysis on audio latency for multiple operating systems." In: *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, pp. 2496–2500. DOI: 10.1109/ICIEA.2018.8398130 (cit. on p. 142).

Yee, N., & Bailenson, J. (2007). "The Proteus effect: The effect of transformed self-representation on behavior." In: *Human Communication Research* 33.3, pp. 271–290. DOI: 10.1111/j.1468-2958.2007.00299.x (cit. on p. 286).

Yee, N., Bailenson, J. N., & Ducheneaut, N. (2009). "The Proteus effect: Implications of transformed digital self-representation on online and offline behavior." In: *Communication Research* 36.2, pp. 285–312. DOI: 10.1177/009365020833025 (cit. on p. 286).

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). "A review of recurrent neural networks: LSTM cells and network architectures." In: *Neural Computation* 31.7, pp. 1235–1270. DOI: 10.1162/neco_a_01199 (cit. on p. 253).

Yuan, Y., Lin, L., Huo, L.-Z., Kong, Y.-L., Zhou, Z.-G., Wu, B., & Jia, Y. (2020). "Using an attention-based LSTM encoder–decoder network for near real-time disturbance detection." In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13, pp. 1819–1832. DOI: 10.1109/JSTARS.2020.2988324 (cit. on p. 256).

Zander, S., Leeder, I., & Armitage, G. (2005). "Achieving fairness in multiplayer network games through automated latency balancing." In: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pp. 117–124. DOI: 10.1145/1178477.1178493 (cit. on pp. 62, 63, 69, 189).

Zheng, L., Zhang, L., & Xu, D. (2001). "Characteristics of network delay and delay jitter and its effect on voice over IP (VoIP)." In: *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*. Vol. 1, 122–126 vol.1. DOI: 10.1109/ICC.2001.936286 (cit. on p. 97).

Zhou, S., Cai, W., Lee, B.-S., & Turner, S. J. (2004). "Time-space consistency in large-scale distributed virtual environments." In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 14.1, pp. 31–47. DOI: 10.1145/974734.974736 (cit. on p. 85).

All URLs cited were checked in March 2024.

List of Figures

1.1 Overview and structure of this thesis. 21
1.2 Cycle of execution and evaluation adapted from Hutchins et al. (1985). . 27
1.3 Example screenshots showing the evolution of video games. 32

2.1 Sheridan and Feller's minimal manipulator apparatus (1963). 47
2.2 Prisma flow chart of the literature review. 57
2.3 Number and citation of papers of literature review. 59
2.4 Sun and Claypool's time warp implementation (2019). 76
2.5 Latency compensation via geometrically adaptation by Lee et al. (2019). 80
2.6 Example linear interpolation. 83

3.1 Study I: Two screenshots of Cube 2: Sauerbraten. Shows player viewport and an aerial perspective of the game world. 99
3.2 Study I: Results of local latency measurements of the study apparatus. . 101
3.3 Study I: Results for *Hitrate*, *KD-Ratio*, and *TotalKills*. 105
3.4 Study I: Results for subjective gaming experience measured via Player Experience Inventory (Vanden Abeele et al., 2016). 106
3.5 Study II: Screenshot of self-developed 2D Shoot'em up game. 116
3.6 Study II: Example latency progression and summary of all conditions. . 121
3.7 Study II: Results for *Flow* and *Tension*. 125
3.8 Study II: Results for *Challenge*. 126
3.9 Study II: Results for *Score*. 128
3.10 Study II: Results for *Target Offset*. 129

3.11 Study II: Additional results for *Target Offset*. 131

4.1 Study III: Two screenshots of the self-developed video game showing player view and an aerial overview. 144
4.2 Study IV: Participant playing Counter-Strike in our study. 159
4.3 Study IV: Shows results for Positive Affect, Negative Affect, and Tension. 160
4.4 Study IV: Shows Tension scores given by expert video game players. . . 162
4.5 Study V: Screenshot of self-developed shooting game. 173
4.6 Study V: Three screenshots from the game, each showing a different in-game perspective. 174
4.7 Study V: Aerial view of the in-game world. 178
4.8 Study V: Results for *Reaction Time* and *Accuracy*. 182
4.9 Study V: Results for Ease of Control, Progress Feedback, Challenge, and Mastery. 184
4.10 Study V: Results for Curiosity, Immersion, Autonomy, and Enjoyment. . 185
4.11 Study VI: Two screenshots of Counter-Strike with and without the developed latency overlay. 198
4.12 Study VI: Results for Tension, Competence, and Positive Affect. 202
4.13 Study VI: Results for *Score* and *Kills*. 203
4.14 Study VI: Results for *KD-Ratio*, *Number of Headshots*, and *Damage per Hit*. 204

5.1 Study VII: Aerial view of the self-developed first-person shooter game. 217
5.2 Study VII: Player in-game view while playing the game. 218
5.3 Study VII: Architectural structure of the used artificial neural network. . 221
5.4 Study VII: Flow diagram of the latency compensation pipeline. 223
5.5 Study VII: Results for *MaxScore*. 225
5.6 Study VII: Results for *HitCoef* and *EnemyHit*. 227
5.7 Study VII: Results for *Positive Affect*. 229
5.8 Study VIII: Two screenshot of the real-time strategy games Age of Empires 2 and Empire Earth. 236
5.9 Study VIII: Conceptual structure of the used artificial neural networks. . 240
5.10 Study VIII: Results for Negative Affect and Tiredness. 244
5.11 Study IX: Example output of Yolo's object detection. 251

5.12 Study IX: Screenshot of Counter-Strike coupled with an object detection by Yolo. 252
5.13 Study IX: Architectural structure of the feed-forward network. 255
5.14 Study IX: Architectural structure of the long short-term memory network. 256
5.15 Study IX: Architectural structure of the decoder-encoder network. . . . 257
5.16 Study IX: Example latency progression. 258
5.17 Study IX: Results for Challenge and Control. 262
5.18 Study IX: Results for Autonomy, Curiosity, Immersion, Mastery, and Meaning. 263
5.19 Study IX: Results for overall gaming enjoyment and subjective latency perception. 265
5.20 Study IX: Results for *Kills* and *Assists*. 267
5.21 Study IX: Results for *Deaths* and *Headshots*. 268

List of Tables

1.1 This thesis's research questions. 32

2.1 Number and sources of papers retrieved for the systematic literature review. 54

3.1 Study I: Results of ART-ANOVA. 107
3.2 Study II: Results of local latency measurements. 120
3.3 Study II: Results ANOVA of Game Experience Questionnaire analysis. . 124
3.4 Study II: Results ANOVA or *Score* and *Target Offset* analysis. 127

4.1 Study III: Means and standard deviation of Game Experience Questionnaire and performance scores. 148
4.2 Study III: Results of Bayesian analysis of Game Experience Questionnaire.150
4.3 Study IV: Mean and standard deviation of Game Experience Questionnaire data. 163
4.4 Study IV: Mean in-game scores. 164
4.5 Study V: Mean and standard deviation for the performance measures. . 180
4.6 Study V: Mean and standard deviation for the Enjoyment data. 180
4.7 Study V: Mean Scores Player Experience Inventory. 181
4.8 Study V: Bayesian evidence categories adjusted from Andraszewicz et al. (2015). 182
4.9 Study VI: Mean and standard deviation for the Game Experience Questionnaire data. 201
4.10 Study VI: Mean and standard deviation for the performance measures. . 202

5.1 Study VII: Shapiro-Wilk test results for game experience data. 228
5.2 Study VIII: Results for subjective gaming experience measured via the Game Experience Questionnaire. 242
5.3 Study VIII: Results ANOVA of Game Experience Questionnaire data. . 243
5.4 Study IX: Mean and standard deviation of the mini Player Experience Inventory (Haider et al., 2022) data. 261
5.5 Study IX: Mean and standard deviation for the performance measures. . 266

6.1 Overview of articles included in PRISMA-based literature review (titles partly shorted to increase readability). 329
6.2 Continuation of overview of articles included in PRISMA-based literature review (titles partly shorted to increase readability). 330

List of Acronyms

ADAM adaptive movement estimation

AI artificial intelligence

ANN artificial neural network

AR augmented reality

CHI Conference on Human Factors in Computing Systems

FIFO first in, first out

FPS first-person shooter

GEQ Game Experience Questionnaire

GUI graphical user interface

GX game experience

HCI human-computer interaction

HMD head-mounted display

HRI human-robot interaction

LSTM long short-term memory

MAE mean absolute error

MHP model human processor

MMORPG massively multiplayer online role-playing games

NPC non-player character

OSF Open Science Framework

PGI player-game interaction

PP player performance

PXI Player Experience Inventory

QoE quality of experience

RAM random-access memory

RPG role-playing game

RQ research question

RTS real-time strategy

SRT system response time

TCT task completion time

UP user performance

UX user experience

VR virtual reality

WIMP windows, icons, menus, and pointers

Appendix

Authors	Title	Year
Liu et al.	A survey and taxonomy of latency compensation techniques...	2022
Claypool et al.	A taxonomy for player actions with latency in network games	2015
Baldovino	An overview of the networking issues of cloud gaming	2022
Dick et al.	Analysis of factors affecting players' performance and perception...	2005
Beyer & Moller	Assessing the impact of game type, display size and network delay...	2014
Amin et al.	Assessing the impact of latency and jitter on the perceived quality...	2013
Long & Gutwin	Characterizing and modeling the effects of local latency...	2018
Xu et al.	Compensating for Latency in Cloud-based Game Streaming...	2022
Hirth et al.	Crowd-based study of gameplay impairments and player performance...	2019
Chen et al.	Dude, the source of lags is on your computer	2013
Ries et al.	Empirical study of subjective quality for massive multiplayer games	2008
Lee & Chang	Enhancing the experience of multiplayer shooter games...	2018
Claypool et al.	Game Input with Delay - Moving Target Selection Parameters	2019
Claypool	Game input with delay—moving target selection with a game...	2018
Joerg et al.	How responsiveness affects players' perception in digital games	2012
Hohlfeld et al.	Insensitivity to network delay: minecraft gaming experience...	2016
Claypool & Claypool	Latency and player actions in online games	2006
Liu et al.	Lower is better? The effects of local latencies on competitive...	2021
Schmidt et al.	Modeling and understanding the quality of experience of...	2021
Quax et al.	Objective and subjective evaluation of the influence of small...	2004

Table 6.1: Overview of articles included in PRISMA-based literature review (titles partly shorted to increase readability).

Authors	Title	Year
Lee & Chang	On" shot around a corner" in first-person shooter games	2017
Claypool	On Models for Game Input with Delay—Moving Target Selection...	2016
Quax et al.	An evaluation of the impact of game genre on user experience...	2013
Slivar et al.	Empirical QoE study of in-home streaming of online games	2014
Liu et al.	L33t or N00b? How player skill alters the effects of network latency...	2021
Panel & Wolf	On the impact of delay on real-time multiplayer games	2002
Ham et al.	Do we need a faster mouse? Empirical evaluation of asynchronicity...	2021
Normoyle et al.	Player perception of delays and jitter in character responsiveness	2014
Eg et al.	Playing with delay: With poor timing comes poor performance...	2018
Iykovic et al.	Quantifying and mitigating the negative effects of local latencies..	2015
Bosik	Nova-Using Temporal Scaling for Latency Compensation...	2018
Wahab et al.	Subjective quality assessment for cloud gaming	2021
Claypool & Claypool	Latency can kill: precision and deadline in online games	2010
Moura et al.	Analysis of gameplay in MOBA games under different network...	2019
Lee et al.	Geometrically compensating effect of end-to-end latency...	2019
Gronli et al.	Latency Thresholds for Usability in Games: A Survey	2014
Durnez et al.	Spaz! The Effects of Local Latency on Player Actions...	2021
Fritsch et al.	The effect of latency and network limitations on mmorpgs...	2005
Claypool	The effect of latency on user performance in real-time strategy games	2005
Sabet et al.	Delay sensitivity classification of cloud gaming content	2020
Sheldon et al.	The effect of latency on user performance in Warcraft III	2003
Savery et al.	The effects of consistency maintenance methods on...	2014
Nichols & Claypool	The effects of latency on online madden NFL football	2004
Claypool & Finkel	The effects of latency on player performance in cloud-based games	2014
Beigbeder et al.	The effects of loss and latency on user performance in unreal...	2004
Beznosyk et al.	Influence of network delay and jitter on cooperation in multiplayer...	2011
Xu et al.	The effects of network latency on counter-strike...	2022
Claypool et al.	The impact of motion and delay on selecting game targets with a mouse	2020
Silvar et al.	The impact of network and social context on quality of experience...	2022
Suznjevic et al.	The impact of user, system, and context factors on gaming QoE...	2013
Schmidt et al.	Towards the delay sensitivity of games: There is more than genres	2017
Long & Gutwin	Effects of local latency on game pointing devices..	2019
Ravindran et al.	Impact of network loss/delay characteristics on consistency control...	2008
Jarschel et al.	Gaming in the clouds: QoE and the users' perspective	2013
Sabet et al.	When every millisecond counts: The impact of delay in vr gaming	2022
Tseng et al.	On the battle between lag and online gamers	2011
Zander et al.	Achieving fairness in multiplayer network games...	2005
Liu & Claypool	The impact of latency on navigation in a first-person perspective game	2022
Sabet et al.	A latency compensation technique based on game characteristics to mitigate...	2020
Tseng et al.	On the battle between lag and online gamers	2011
Zander et al.	Achieving fairness in multiplayer network games...	2005
Liu & Claypool	The impact of latency on navigation in a first-person perspective game	2022
Sabet et al.	A latency compensation technique based on game characteristics to mitigate...	2020
Huang et al.	GamingAnywhere: The first open source cloud gaming system	2014
Veron et al.	Matchmaking in multi-player on-line games: studying user traces...	2014
Wu et al.	iCloudAccess: Cost-Effective Streaming of Video Games...	2014
Wong et al.	Network latency classification for computer games	2021
Madanapalli et al.	Know thy lag: In-network game detection and latency measurement	2022
Zha & Zhang	The Effects of Network Latency on Player Gaming Experience	2019
Choy et al.	The brewing storm in cloud gaming: A measurement study...	2012
Saldana & Suznjevic	QoE and latency issues in networked games	2019

Table 6.2: Continuation of overview of articles included in PRISMA-based literature review (titles partly shorted to increase readability).

Erklärung der Urheberschaft

Hiermit versichere ich, dass ich die vorliegende Dissertation selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich aus der Literatur oder aus anderen Quellen übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Die vorgelegten Druckexemplare und die vorgelegte digitale Version sind identisch.

Ort, Datum, Unterschrift