Don't Break my Flow: Effects of Switching Latency in Shooting Video Games

DAVID HALBHUBER, University of Regensburg, Germany
VALENTIN SCHWIND, Frankfurt University of Applied Sciences, Germany
NIELS HENZE, University of Regensburg, Germany

Latency is inherently part of every interactive computing system and particularly important for video games. Previous work shows that constant latency above 25 ms reduces game experience and player performance. However, latency in the wild varies and is never constant due to multiple factors, such as updates in routing tables, users changing their location, or the system's workload. It is unclear if switching latency impairs the gaming experience stronger than a constant high latency. To elucidate, we conducted an experiment with 264 participants playing a shooting video game induced with 0 ms, 33 ms, and 66 ms controlled latency. While playing, the game switched between different latency levels based on three frequencies. Our analysis shows that switching latency significantly impaired the participants' flow. Additionally, we found effects on the perceived tension, the experienced challenge, and the players' performance. We conclude that games should prioritize constant latency, even if that entails artificially adding latency.

CCS Concepts: • **Human-centered computing → Empirical studies in HCI**.

Additional Key Words and Phrases: video games, latency, switching latency

ACM Reference Format:
David Halbhuber, Valentin Schwind, and Niels Henze. 2022. Don't Break my Flow: Effects of Switching Latency in Shooting Video Games. *Proc. ACM Hum.-Comput. Interact.* 6, CHI PLAY, Article 229 (October 2022), 20 pages. https://doi.org/10.1145/3549492

229

1 INTRODUCTION

Latency, the time between an action and the corresponding outcome, negatively influences player performance and gaming experience in video games [4, 9, 16]. Poor gaming experience has ramifications – not only for gamers but also for game developers and publishers. In the worst case, sub-par game experience leads to the game being discontinued [8, 17, 29]. Previous work investigating latency in video games typically treated latency as a constant parameter. Researchers exposed participants to discrete latency in controlled experiments, such as 25 ms, 120 ms, or even 500 ms. This line of work has shown that players score fewer points, need more time to complete tasks, or are unable to solve tasks at all [4, 16, 30, 31]. Similarly, previous work also investigated the effect of packet delay variation [14] in video games, also known as jitter. Jitter is an extremely short-termed latency fluctuation caused by the delay of uni-directional, consecutive packets transmitted between two hosts [38]. It has been shown that jitter also reduces game experience as well as the performance [1, 15, 40]. Jitter, however, is not the only way latency changes. Latency is formed by various factors, such as the used hardware [28, 45], the currently running software, the Internet

Authors' addresses: David Halbhuber, david.halbhuber@ur.de, University of Regensburg, Regensburg, Germany; Valentin Schwind, valentin.schwind@acm.org, Frankfurt University of Applied Sciences, Frankfurt, Germany; Niels Henze, niels.henze@ur.de, University of Regensburg, Regensburg, Germany.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2022 Copyright held by the owner/author(s).
2573-0142/2022/10-ART229
https://doi.org/10.1145/3549492

connection, and the Internet service provider's network routing path [39]. Thus, latency is neither constant nor changes only between consecutive frames but 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 [44] and new gaming paradigms, such as cloud gaming [43]. Mobile players are stronger exposed to switching latency caused by handshakes between cell towers and network load balancing techniques on the carrier's end [19]. However, it is unclear how switching latency affects the players' performance and game experiences in video games.

Previous work investigates two types of latency: (1) constant latency and (2) fast-paced change induced by jitter. However, since latency is caused by multiple factors and is prone to variation, there is a third type: (3) switching latency. Previous work does not account for switching latency and its' effects on players. Our work closes this gap by providing in-depth insights on how switching latency affects game experience and player performance. To achieve this, 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' 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' performance, such as the accuracy and the achieved score. 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 the players' performance and experience. Our work shows 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.

We provide all resources to enable other researchers to replicate and build upon our work. This includes the game, all source code, as well as all gathered, anonymous user data[1].

2 RELATED WORK

The analysis of both latency and latency-related effects is well established in research on interactive systems. Already in 1981, Card [5] had characterized how temporal differences in in- and output impairs the interaction between a user and a system. Card shows that users interact with interactive systems in a continuous feedback loop in his work. The user initiates the interaction by providing data to the target system, for example, by sending a command via the computer mouse. The system receives and processes the input and responds with an output. The user, then, can reply to this output and possibly start another loop cycle. In essence, Card described latency as presented in current research.

In the following, we first describe how latency in interactive systems arises and which effects on the user are caused by it. Next, we show how constant latency and jitter affect video games and players. Finally, we conclude with a summary on why switching latency in video games needs to be investigated.

[1]https://github.com/david-halbhuber/switching-latency

2.1 Latency in Interactive Systems

Latency in interactive systems is not an entirely constant value as it is influenced by multiple factors, such as the used system itself, the system's workload, the used peripheral devices, such as a mouse, keyboard, or monitor, as well as the network [45]. All these factors are grouped in two different forms of latency: (1) network latency and (2) local latency. Network latency comprises multiple components, such as transmission between network hops, response times of servers, or even one's internet connection. In the area of mobile computers, the picture is even more opaque. While upcoming network standards promise ultra-low latency down to 1 ms [6], the current standard *Long-Term-Evolution* (LTE) has a latency of up to 80 ms [18]. However, LTE cell towers have a range of approximately 0.6 km to 1.4 km [24] in an urban area. Considering traveling with an average speed of 60 km/h, either by car or train, this entails switching cell towers approximately every 40 seconds. Every switch updates the routing table - the path network packages have to take - and thus potentially changes the latency. Besides network-related latency, the local latency of devices also adds to the overall latency of the interaction. Local latency is the time arising due to processing and transmission between input and output. Each technical component in this loop, such as but not limited to, the computer, the operating system, the mouse, the keyboard, and the monitor, attributes to local latency [34, 41]. Local latency in real-world setups ranges from 23 to 243 ms [26].

Recent work shows that latency in interactive systems leads to a degradation of user performance. Jota et al. [27] and Annett et al. [2], for example, show that latency above 25 ms leads to degraded user performance. No improvement in user performance was observed for latency less than 25 ms. The authors conclude that users perform best at a latency of 25 ms. Although performance does not increase below a latency of 25 ms, Ng et al. [37] found that users can perceive latency starting at a value of 2 ms. Building on this, Ng et al. [36] show in subsequent work that users in some tasks are even able to notice discrepancies between 1 ms latency and 2 ms latency.

2.2 Latency and Jitter in Games

Video games and gamers are also affected by latency. Negative effects of latency manifest in video games in manifold forms. For example, players score fewer points, require more time to finish tasks, or fail to complete tasks at all [4, 10, 16]. However, Claypool and Claypool [9] show that specific game genres, such as fast-paced shooting games, are more affected by latency than other genres. Some work indicates that the latency tolerance threshold for such games is at 150-180 ms [3], while other work shows that player performance decreases starting at 100 ms latency [40]. In addition to performance degradation, latency is also known to degrade the gaming experience in video games. Liu et al. [31], for instance, show that latency of 150 ms decreases the overall Quality of Experience (QoE) by 25%. In a similar work, Liu et al. [30] found that QoE linearly decreases by 20% when latency rises from 25 ms to 125 ms. However, constant latency is not the only disturbance of Card's human-computer feedback loop. Jitter, caused by the short-termed delay of two uni-directional, consecutive packets transmitted between two hosts [38], is also known to influence game experience and performance negatively. Amin et al. [1] for example found that jitter above 100 ms, which refers to two consecutive network packets being delayed by 100 ms, significantly increases the task completion time in video games compared to the time needed to complete the same task without jitter. Besides the objective player performance, the authors also found that 200 ms of jitter decreases the subjective game experience. Thus, the authors conclude, although jitter decreases the player performance starting at a value of 100 ms, degradation of the gaming experience begins not until 200 ms of jitter.

2.3 Summary

Previous work investigates two types of latency: (1) constant latency and (2) fast-paced change induced by jitter. However, since latency is caused by multiple factors and is prone to variation, there is a third type: (3) switching latency.

Starting at 1 ms, constant latency becomes perceivable by the user and negatively affects user performance beginning at 25 ms [27, 36]. Video games are affected by constant latency as well - players score fewer points, need more time to complete tasks, or are not able to solve a given task at all [4, 10, 16]. Additionally, constant latency is known to affect subjective gaming experience [30, 31] negatively. A similar pattern was found when investigating jitter. Jitter reduces player performance starting at 100 ms and degrades gaming experience at 200 ms. However, both types, (1) constant latency and (2) jitter, are related to either a constant value (1) or extremely fast-paced change (2). Therefore, it is unclear how medium-termed switching latency (3) affects the players' performance and game experiences in video games.

3 GAME DEVELOPMENT

To investigate how switching latency between different constant latency levels affects game experience and player performance, we developed a fast-paced 2D Shoot 'em up game. This type of game features elements found in almost all game genres, such as quick target selection or target tracking, and thus is susceptible to latency. 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.

Fig. 1. Screenshot of the developed 2D Shoot 'em up game. The red box shows the players' 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 crosshairs (center), and large (center) and small targets (center).

3.1 Implementation

Since Shoot 'em up games are fast-paced and require split-second decisions to perform well, they are, as any shooting game [4, 30], susceptible to latency. Therefore, and because they incorporate multiple elements of many different game genres, such as target tracking and target selection, we have defined them as the main subject of our research.

The player's objective is to shoot as many targets as possible to increase the game score before the game time elapses. Each target hit rewards the player with ten points. Figure 1 shows a screenshot of the developed game - the different game elements, such as targets, the 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' viewport do not grant any points. Players do not lose points for missing the targets. To keep players motivated while playing, we increase the difficulty by adding ammunition. 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 *Unity'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 artificially. 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 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. Additionally, playing the game in the browser entails multiple advantages. Firstly, participants do not have to install any software to participate in the study. Secondly, content delivery is straightforward since we only need to provide a hyperlink to our web server. Finally, using the browser allows one 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 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.

3.2.1 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's *Chrome*.

3.2.2 Procedure and Task. We first informed the participants via the crowd-sourcing platform *prolific.co* about the study's purpose and provided a hyperlink to our web server. After giving informed consent to the data collection, the participants could follow the hyperlink to our website. On 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 our institute.

3.2.3 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. The average age was 24.5 years (SD = 4.75 years), with ages ranging from 19 years to 36 years. All participants were screened for prior gaming experience using *prolifc*'s screening interface. This ensured that all participants had a comparable gaming skill.

3.2.4 Descriptive Results. In total, the 24 participants played our game for 3.2 hours. We recorded the participants' mouse movements and click behavior throughout the gaming sessions. Additionally, we also logged the frame rate 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) [25] to quantitatively evaluate the game experience 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. Figure 2 depicts the scores given by the participants in each GEQ sub-scale. 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 the game as the subject for 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.

3.3 Estimating Local Latency

Local latency, the latency caused by ones' own 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 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 [26, 32], 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 the update in the game allowed us

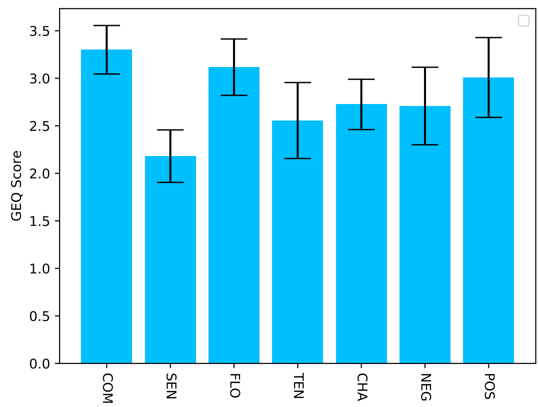


Fig. 2. The scores given by the participants in each GEQ subscale in the pre-study. A maximum of 5 points could be assigned in each subscale. Participants gave 3.29 points (SD = 0.60 points) in the competence (*COM*) subscale, 2.18 points (SD = 0.65 points) in sensory (*SEN*) subscale, 3.11 points (SD = 0.70 points) in the flow (*FLO*) subscale, 2.55 points (SD = 0.94 points) in the tension (*TEN*) subscale, 2.72 points (SD = 0.39 points) in the challenge (*CHA*) subscale, 2.70 points (SD = 0.96 points) in the negative affect (*NEG*) subscale and 3.00 points (SD = 0.99 points) in the positive affect (*POS*) subscale. Errorbars show CI95.

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 frames (SD = 2.28 frames)
Win 10 Home 21H1	Ryzen 7 4800U	16 GB	GTX 1660Ti	DeathAdder V2	14.65 frames (SD = 4.34 frames)
Win 10 Pro 20H2	i5-10500	16 GB	RTX 2070	HP 280 Silent	8.5 frames (SD = 1.47 frames)
Win 10 Home 20H2	Ryzen 3 1200	8 GB	GTX 980Ti	Logitech G203	16.60 frames (SD = 1.65 frames)
Win 10 Pro 21H1	i5-1145G7	8 GB	Intel Iris Xe	Dell MS116-BK	17.30 frames (SD = 3.51 frames)
Win 10 Home 21H1	AMD 3020E	4 GB	AMD Radeon	Logitech M90	11.45 frames (SD = 2.54 frames)

Table 1. 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.)

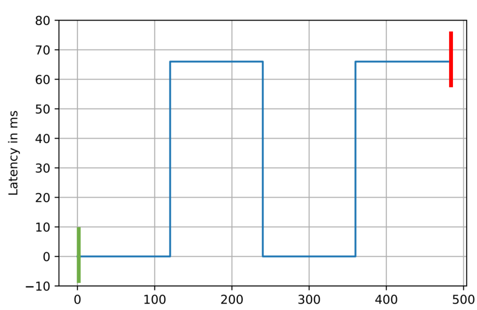
to determine the local latency of 6 different systems. We measured each system 20 times. While playing our developed game, the mean local latency is 13.32 frames (SD = 4.15 frames, n = 120), which translates to a mean local latency of 55.51 ms (SD = 17.29 ms). This latency is considered to be the baseline latency of our game. All further latency values include this baseline without explicitly mentioning it. Table 1 shows further details to all measured systems as well as the mean measured local latency of each system.

4 INVESTIGATING SWITCHING LATENCY IN VIDEO GAMES

To investigate how switching latency between different constant latency levels in a single gaming session affects game experience and player performance, we conducted an in-the-wild study using the developed and tested game. We artificially added multiple latency levels via input buffering to our game to do so. We 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.

4.1 Study Design

To control the latency level and the switches between different latency 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: (1) 0 ms, (2) 33 ms, and (3) 66 ms. LATENCY SWITCHES is likewise factorized in three levels: (1) 0 switches, (2) 3 switches, and (3) 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 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

Fig. 3. The left side shows the latency progression for condition 6 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 to 3 latency switches. The table on the right lists all 11 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[23]. We also examined 33 ms and 0 ms latency to allow detailed analysis. The first level of LATENCY SWITCHES (0 switches) relates to previous work investigating constant latency values [30, 31]. 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. The last level of LATENCY SWITCHES is based on previous work showing that the mean range of LTE is approximately 1.0 km [7] which implies a change of network conditions every 40 seconds (rounded up) at an average travel speed of 60 km/h. At the aspired game time of eight minutes, this corresponds to 12 LATENCY SWITCHES.

We recorded data about the participants' game experience and their performance. To measure participants' 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 [25].

4.2 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*. 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. Each participant played a single version of the game. The final apparatus can be found in the supplementary GitHub repository[2].

4.3 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 participant's goal 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 our institute.

4.4 Participants

We used the crowd-sourcing platform *prolific.co* to recruit a total of 264 participants (55 female, 205 male, two non-binary, 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. 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 participants' average 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 a comparable gaming skill.

5 RESULTS

We evaluated the participants' mouse movement, click behavior, frame rate, system specification, and in-game metrics such as score and accuracy. In total, the 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).

[2]https://github.com/david-halbhuber/switching-latency

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 2. 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.

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

5.1 Game Experiences Questionnaire

Following the authors' guidelines [25], we analyzed each subscale of the GEQ separately. We used a mixed model ANOVA with Latency Switches nesting Start Latency and Target Latency, to analyze each subscale of the GEQ. Table 2 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, which corresponds to the mental state of being highly focused [21, 35], Tukey's test showed that participants' scoring 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 4 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.

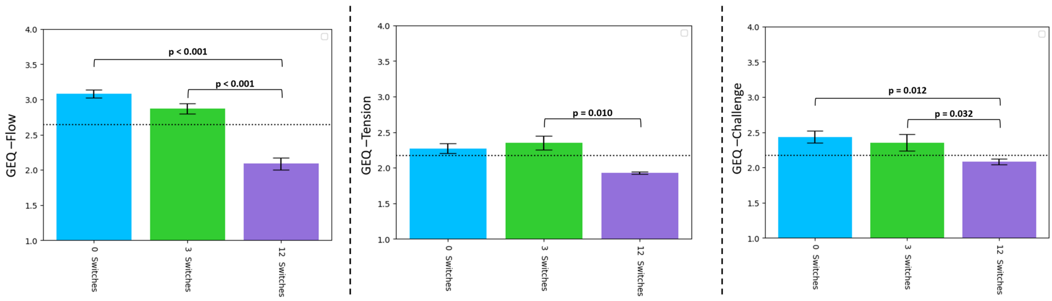


Fig. 4. Depicts *Flow* (left), *Tension* (middle) and *Challenge* (right) of the GEQ assigned by the players. Data is grouped by Latency Switches. Error bars depict the standard error. Additionally, all data is color coded by Latency Switches: Light blue groups 0, green 3, and purple 12 Latency Switches. 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*, *Tension* and *Challenge* 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 4 shows the mean *Tension* scores grouped by Latency Switches(middle). 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 4 shows the mean *Challenge* scores grouped by Latency Switches (right). 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.

5.2 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 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.

5.2.1 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

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. 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*.

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 5 depicts the mean *Score* values for all individual conditions (left), the combined values grouped by Latency Switches (middle) and the statistical significant pairwise comparisons of Latency Switches x Start Latency on *Score* (right).

5.2.2 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 6 shows the mean *Target Offset* values for all conditions (left) and the mean *Target Offset* sorted by Latency Switches (right). 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*).

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 7 (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, η_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

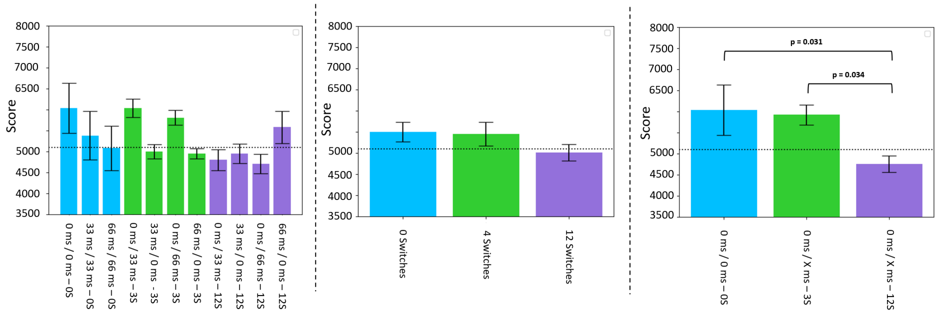


Fig. 5. Left shows mean score values players reached for each condition. Error bars depict the standard error. X-ticks are coded with the independent variables Start Latency / Target Latency - Latency Switches. Additionally, all data is color coded by Latency Switches: Light blue groups 0, green 3, and purple 12 Latency Switches. The middle shows mean *Score* values grouped by Latency Switches and the same color code. Error bars show the standard error. The right side shows the pairwise comparison of Start Latency X Latency Switches. 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.

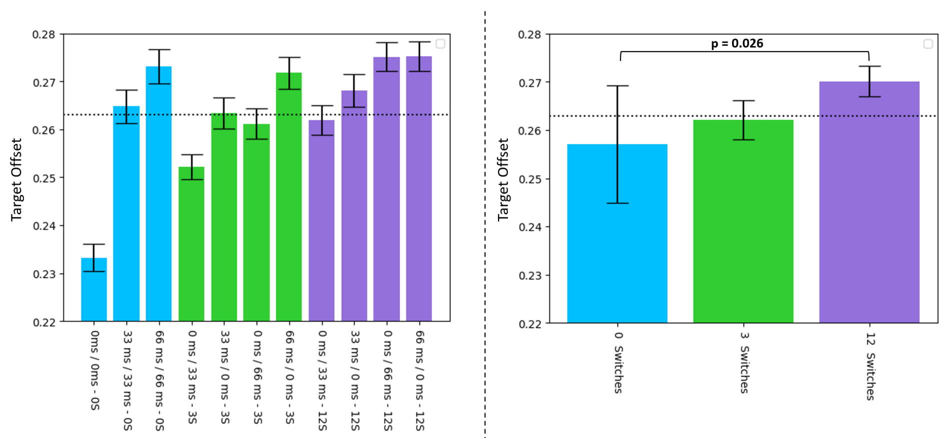


Fig. 6. Left shows mean *Target Offset* values players reached for each condition. Error bars depict the standard error. X-ticks are coded with the independent variables Start Latency / Target Latency - Latency Switches. Additionally, all data is color coded by Latency Switches: Light blue groups 0, green 3 and, purple 12 Latency Switches. The right side shows mean *Target Offset* values grouped by Latency Switches (x-ticks) and the same color code. Error bars show the standard error. Significant differences are highlighted via p-bars. The mean *Target Offset* values is depicted in both plots 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.

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 7 (right) displays the significant differences found in the pairwise comparison.

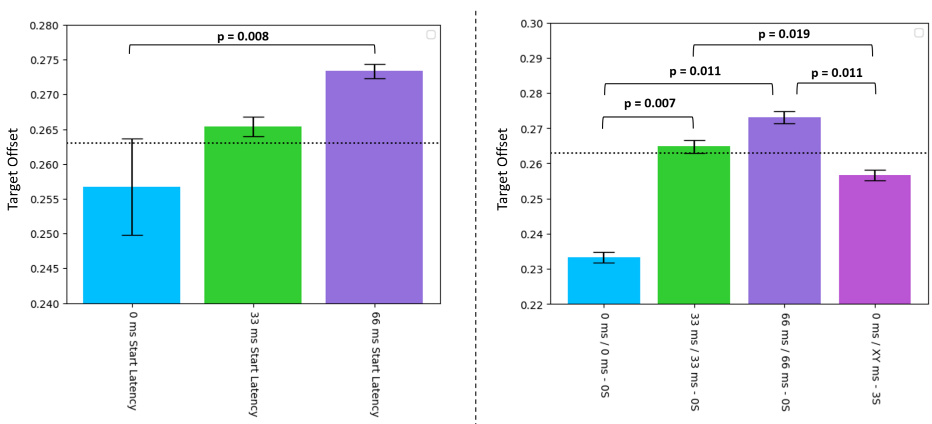


Fig. 7. Left shows the *Target Offset* grouped by Start Latency (x-ticks). Error bars depict the standard error. Additionally, all data is color coded by Latency Switches: Light blue groups 0 ms, green 33 ms and purple 66 ms Start Latency. Participants deviated significantly stronger from an ideal hit 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.

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.

6 DISCUSSION

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

Flow, *Tension*, *Challenge*, *Score* and *Target Offset*. We conclude this section with a discussion of the implications for developers, gamers, and researchers alike.

6.1 Flow, Tension, and Challenge

We found an effect of latency switches on the perceived flow. Flow, firstly described in 1975 by Csikszentmihalyi [22, 35], corresponds to the mental state of *being in the zone*. This state includes being highly focused as well as a strong feeling of immersion with a high level of enjoyment and fulfillment [21]. 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 [12]. 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 [13, 20]. 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 the 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 *GEQ* subscales *Tension* and *Challenge*. Similarly to the flow rating, we found that participants rated both subscales 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 [22], 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, the participants did not see a real challenge in the game while still performing worse than participants in other conditions. Surprisingly, we did not find evidence for this hypothesis in the qualitative feedback data. While not consciously aware of it, participants rated the game with more switches a lower challenge while simultaneously performing worse than participants playing with no switches.

6.2 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 to pose hypotheses that are in line with previous research. Firstly, our analysis revealed *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 of Claypool et al. [11] who found that player performance is stable up to 100 ms latency. Secondly, participants' performance 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; 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.

6.3 Target Offset

Participants playing with 12 Latency Switches aimed more poorly and deviated stronger from the optimal hit point than participants shooting with 0 or 3 Latency Switches. In addition, the start latency has a significant effect on the participants' shooting behavior. 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 [4], 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' accuracy. Nevertheless, we found that the combination of start latency and the number of latency switches creates a significant interaction effect on *Target Offset*. An in-depth analysis of this interaction showed that switching latency might improve player performance 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 cause 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.

6.4 Implications

Our findings have implications for future game design and thus for game developers. To optimize game experience, 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 [23, 42] to fixed values or by artificially adding latency to provide a consistent experience. This may even be applicable in regards to local latency. Local latency negatively and significantly affects game performance and experience [33]. Fixating the local latency to a certain but consistent level could improve performance and experience.

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 high competitive e-sports scenarios, it is important to not only factor in the average latency but also its stability when, for example, evaluating a past gaming performance. 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 the gaming experience.

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 medium-term switches 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, whether discretized or approximated over many samples, potentially leads to the missing of effects on game experience and player performance.

7 CONCLUSION

In this paper, we present a 2D Shoot 'em up game inflicted with switching latency alternating between fixed values in defined intervals. In a study with 264 participants, we found that switching latency significantly affects game experience and player performance. We show that switching latency significantly decreases the perceived flow, effectively preventing participants from *entering the zone*. In addition, we found small effects of switching latency on the experienced tension and the perceived challenge. Furthermore, we showed that participants playing with twelve switches were significantly less accurate than participants playing with no switches.

In summary, we found that switching latency negatively influences game experience and player performance 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.

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 limitation, 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 only 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 player performance, 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, the effects of local latency in a latency-switching environment, and whether there are latency-switching sweet spots.

Furthermore, we found that the starting condition of a gaming session is crucial for the players' 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 *First-Person-Shooter* or racing games. Investigating different genres is valuable not only in the context of this work but also in the context of research in video games. Further deepening our knowledge about latency and its effects on players is crucial to developing games with a high gaming experience.

REFERENCES

[1] Rahul Amin, France Jackson, Juan E Gilbert, Jim Martin, and Terry Shaw. 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, 97–106.

[2] Michelle Annett, Fraser Anderson, Walter F. Bischof, and Anoop Gupta. 2014. The Pen is Mightier: Understanding Stylus Behaviour While Inking on Tablets. In *Proceedings of Graphics Interface 2014* (Montreal, Quebec, Canada) *(GI '14)*. Canadian Information Processing Society, CAN, 193–200. https://doi.org/10.5555/2619648.2619680

[3] Grenville Armitage. 2003. An experimental estimation of latency sensitivity in multiplayer Quake 3. In *The 11th IEEE International Conference on Networks, 2003. ICON2003*. IEEE, 137–141.

[4] Tom Beigbeder, Rory Coughlan, Corey Lusher, John Plunkett, Emmanuel Agu, and Mark Claypool. 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* (Portland, Oregon, USA) *(NetGames '04)*. Association for Computing Machinery, New York, NY, USA, 144–151. https://doi.org/10.1145/1016540.1016556

[5] Stuart Card. 1981. The model human processor: A model for making engineering calculations of human performance. In *Proceedings of the Human Factors Society Annual Meeting*, Vol. 25. SAGE Publications Sage CA: Los Angeles, CA, 301–305.

[6] Kwang-Cheng Chen, Tao Zhang, Richard D Gitlin, and Gerhard Fettweis. 2018. Ultra-low latency mobile networking. *IEEE Network* 33, 2 (2018), 181–187.

[7] Yi Hua Chen, Chang Lueng Chu, and Kai Jen Chen. 2017. Measurement and analysis of LTE cell range and downlink throughput in suburban area. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 611–614.

[8] 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/. Accessed on 2021-07-05.

[9] Mark Claypool and Kajal Claypool. 2006. Latency and Player Actions in Online Games. *Commun. ACM* 49, 11 (Nov. 2006), 40–45. https://doi.org/10.1145/1167838.1167860

[10] Mark Claypool and David Finkel. 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, 1–6.

[11] Mark Claypool, David Finkel, Alexander Grant, and Michael Solano. 2014. On the performance of OnLive thin client games. *Multimedia systems* 20, 5 (2014), 471–484.

[12] Ben Cowley, Darryl Charles, Michaela Black, and Ray Hickey. 2008. Toward an understanding of flow in video games. *Computers in Entertainment (CIE)* 6, 2 (2008), 1–27.

[13] Mihaly Csikszentmihalyi and Mihaly Csikzentmihaly. 1990. *Flow: The psychology of optimal experience*. Vol. 1990. Harper & Row New York.

[14] Carlo Demichelis and Philip Chimento. 2002. RFC3393: IP packet delay variation metric for IP performance metrics (IPPM).

[15] Matthias Dick, Oliver Wellnitz, and Lars Wolf. 2005. Analysis of factors affecting players' performance and perception in multiplayer games. In *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*. 1–7.

[16] Ragnhild Eg, Kjetil Raaen, and Mark Claypool. 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, 1–6. https://doi.org/10.1109/QoMEX.2018.8463382

[17] 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/. Accessed on 2021-07-05.

[18] César Augusto García-Pérez and Pedro Merino. 2018. Experimental evaluation of fog computing techniques to reduce latency in LTE networks. *Transactions on Emerging Telecommunications Technologies* 29, 4 (2018), e3201.

[19] Ilya Grigorik. 2013. *High Performance Browser Networking: What every web developer should know about networking and web performance.* " O'Reilly Media, Inc.".

[20] Sara M Grimes and Andrew Feenberg. 2009. Rationalizing play: A critical theory of digital gaming. *The information society* 25, 2 (2009), 105–118.

[21] David Halbhuber, Jakob Fehle, Alexander Kalus, Konstantin Seitz, Martin Kocur, Thomas Schmidt, and Christian Wolff. 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*. 867–870.

[22] J Hendin and M Csikszentmihalyi. 1975. Measuring the flow experience in rock dancing. *Beyond boredom and anxiety* (1975), 102–122.

[23] Niels Henze, Markus Funk, and Alireza Sahami Shirazi. 2016. Software-reduced touchscreen latency. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 434–441.

[24] Harri Holma and Antti Toskala. 2009. *LTE for UMTS: OFDMA and SC-FDMA based radio access*. John Wiley & Sons.

[25] Wijnand A IJsselsteijn, Yvonne AW de Kort, and Karolien Poels. 2013. The game experience questionnaire. *Eindhoven: Technische Universiteit Eindhoven* (2013), 3–9.

[26] Zenja Ivkovic, Ian Stavness, Carl Gutwin, and Steven Sutcliffe. 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* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 135–144. https://doi.org/10.1145/2702123.2702432

[27] Ricardo Jota, Albert Ng, Paul Dietz, and Daniel Wigdor. 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* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2291–2300. https://doi.org/10.1145/2470654.2481317

[28] Topi Kaaresoja and Stephen Brewster. 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* (Beijing, China) *(ICMI-MLMI '10)*. Association for Computing Machinery, New York, NY, USA, Article 2, 8 pages. https://doi.org/10.1145/1891903.1891907

[29] 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. Accessed on 2021-07-05.

[30] Shengmei Liu, Mark Claypool, Atsuo Kuwahara, Jamie Sherman, and James J Scovell. 2021. 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*. 1–12.

[31] Shengmei Liu, Atsuo Kuwahara, James Scovell, Jamie Sherman, and Mark Claypool. 2021. Comparing the Effects of Network Latency versus Local Latency on Competitive First Person Shooter Game Players. (2021).

[32] Michael Long and Carl Gutwin. 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*. 285–297.

[33] Michael Long and Carl Gutwin. 2019. Effects of local latency on game pointing devices and game pointing tasks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[34] I. Scott MacKenzie and Colin Ware. 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* (Amsterdam, The Netherlands) *(CHI '93)*. Association for Computing Machinery, New York, NY, USA, 488–493. https://doi.org/10.1145/169059.169431

[35] Csikszentmihalyi Mihaly. 1997. Finding Flow: The psychology of engagement with everyday life.

[36] Albert Ng, Michelle Annett, Paul Dietz, Anoop Gupta, and Walter F. Bischof. 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* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1103–1112. https://doi.org/10.1145/2556288.2557037

[37] Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. 2012. Designing for Low-Latency Direct-Touch Input. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 453–464. https://doi.org/10.1145/2380116.2380174

[38] Ripe NNC. 2021. FAQ: Test Traffic Measurements. https://www.ripe.net/analyse/archived-projects/ttm/test-traffic-measurements/what-is-ip-packet-delay-variation-ipdv-also-called-jitter. Accessed on 2021-07-05.

[39] Réhan Noordally, Xavier Nicolay, Pascal Anelli, Richard Lorion, and Pierre Ugo Tournoux. 2016. Analysis of internet latency: the reunion island case. In *Proceedings of the 12th Asian Internet Engineering Conference*. 49–56.

[40] Peter Quax, Patrick Monsieurs, Wim Lamotte, Danny De Vleeschauwer, and Natalie Degrande. 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*. 152–156.

[41] Andreas Schmid and Raphael Wimmer. 2021. Yet Another Latency Measuring Device. (2021).

[42] Valentin Schwind, David Halbhuber, Jakob Fehle, Jonathan Sasse, Andreas Pfaffelhuber, Christoph Tögel, Julian Dietz, and Niels Henze. 2020. The Effects of Full-Body Avatar Movement Predictions in Virtual Reality using Neural Networks. In *26th ACM Symposium on Virtual Reality Software and Technology*. 1–11.

[43] Ryan Shea, Jiangchuan Liu, Edith C-H Ngai, and Yong Cui. 2013. Cloud gaming: architecture and performance. *IEEE network* 27, 4 (2013), 16–21.

[44] Statista. 2021. 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/. Accessed on 2021-08-01.

[45] Raphael Wimmer, Andreas Schmid, and Florian Bockes. 2019. On the Latency of USB-Connected Input Devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

Received February 2022; revised June 2022; accepted July 2022