Last Stand – A First Person Shooter Game for User Studies on the Effects of Network Delay on Players

Samin Shahriar Tokey
sstokey@wpi.edu
WPI, Worcester, Massachusetts, USA

Mark Claypool
claypool@wpi.edu
WPI, Worcester, Massachusetts, USA

ACM Reference Format:
Samin Shahriar Tokey and Mark Claypool. 2018. Last Stand – A First Person Shooter Game for User Studies on the Effects of Network Delay on Players. In *Proceedings of Foundations of Digital Games 2024.* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

First-person shooter (FPS) games demand players to make split-second decisions and have quick reflexes. Delays from player input until game result output can make FPS games less enjoyable and unfair [3]. These delays might come from local sources, like input or output devices, or from network delays, which affect how quickly data travels between the player's game and the server. Higher delay can make the game feel less responsive for players, degrading their performance and overall gaming experience [2, 5]. In multiplayer online games, differences in network delays among players, often due to their geographic proximity to the server, can lead to unfair advantages, particularly disadvantaging those with higher delay. Latency compensation techniques can mitigate some of the adverse effects of network delay on gaming and are deployed by many commercial games [1, 4].

Despite numerous studies of delay and games and wide-spread deployment of latency compensation techniques to deal with network delay, there is more research to be done. Additional studies are needed to understand the impact of delay over the broad range of player-game interactions and in-game conditions even with a well-studied genre (e.g., FPS games). While commercial game developers may assess implemented latency compensation techniques internally, their analysis is rarely disseminated to the broader public, limiting the ability of researchers from learn from their engineering. In order to better understand and improve upon latency compensation technologies, we need studies of these techniques over a range of latencies and game conditions.

Despite the prevalence of previous research and commercial games supporting network gameplay, better tools are needed for latency researchers. Study of commercial games obfuscate latencies effects with latency compensation and opaque ("black box") implementations. This leads to many researchers "reinventing the wheel" to make new games for their studies. While some of these are single-player by design (FPSci)[1], this poses a challenge to researching network latency where client-server interactions are required. Moreover, most bespoke games lack high-end graphics and animations, making it possible that they may not accurately represent player experience with commercial games.

Latency researchers need a client-server networked FPS game featuring a client for each player connected to an authoritative server to mimic typical commercial game architectures. Basic FPS actions need to be supported (e.g., moving and shooting) with high-end graphics and sound that provide for an immersive experience. Game parameters need to be configurable to support the breadth of FPS game conditions. An framework is needed to allow designers to easily set per-round parameters for a range of possible experimental conditions. Extensive logging is needed for all player actions as well as summary performance statistics each round. Lastly, configurable subjective opinion prompts are essential for assessing Quality of Experience (QoE) effectively.

To meet these rquirements, we have designed and developed "Last Stand" – a two-player networked FPS game implemented with the Unity game engine, specifically designed to study the impacts of latency and latency compensation techniques on player performance and QoE. The gameplay features rounds of deathmatch 1v1 play with unlimited lives. The player character is equipped with fluid procedural animations for actions such as sprinting, jumping, aiming, and leaning. The game equips players with a single-fire rifle that holds a customizable number (default 11) of bullets per magazine (unlimited magazines) and a customizable fire rate (default 250 rounds per minute).

A central feature of the game is the configurable experimental harness. This feature allows for launching game rounds with different amounts of delay for each player. Also included is a logging system that captures player data every game tick, provides round summaries, and records details of each projectile shot, facilitating detailed analysis of player performance under different latency conditions. After each round, the game shows customizable qualitative questions to players – the defaults inquires about perceived lag and acceptability. User answers are logged in the round summary.

We have used Last Stand in a study of FPS games and a latency compensation technique (time delay). Our experience suggests it can be beneficial for other researchers, too, in their latency studies and so we have made Last Stand available as open source on GitHub[2].

2 LAST STAND

This section describes a demonstration video, key implementation details, and configuration options.

Foundations of Digital Games 2024, May 21st to 24th, 2024, Worcester, Massachusetts, USA

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

[1] FPSci: https://github.com/NVlabs/FPSci
[2] Last Stand GitHub repository: https://github.com/Tokey/LastStand

2.1 Demonstration

The game demo[3] illustrates two rounds of Last Stand. The demo starts by showing animations regarding movement and aiming. Each player can detect the proximity and direction of the other player by looking at the yellow indicators on the sides of the screen. If the indicator is on the top, it means the opponent is in front of the player, and similarly for both the sides as well. The indicator also grows larger as the player approaches the opponent. Starting from *0:28*, the opponent's animations are displayed, indicating that animations are replicated across the network. At *0:44*, the player secures a confirmed kill, increasing both the kill count and score. After that regular combat takes place, highlighting death events, respawns, and headshots. After the end of the round players are presented with two consecutive qualitative questions at *1:22*. During round 2, regular combat is shown followed by the same questionnaire at the end.

2.2 Implementation

The networking of Last Stand is done using Unity's Netcode for Game Objects[4]. The game runs in a peer-to-peer configuration, where one computer acts as the host and authoritative server and the other is the client. Animations and player positions are continuously replicated for both. Hit detection is done on the client and then replicated over the network. The server controls the projectile spawning and projectile control. Latency is added to each player action by delaying the player input by that set latency by having the game loop check keep an input queue of action and only execute them at scheduled times (i.e., after fixed added delay). All of the round configurations and session control are read, transferred, and executed by the host/server.

2.3 Configuration

After downloading,[5] Last Stand needs to be configured before experiments are run. At least 2 computers are required to run the game, where 1 will act as server/host and 1 as a client. Inside the `Data/` folder there are two configuration files. The first (`GameConfig.csv`) is for setting the game parameters and takes in 1 row of data, each with 4 columns defining the global game configuration: Is Host, IP, Port, Enable Ping Display.

The second file (`PlayerConfig.csv`) defines the configuration for both of the players' delay settings each round. It can have n rows, indicating $n + 1$ rounds. The first row of the game is played twice: initially as the "Practice Round" and then again at a random point in the session. Before each session, the round settings are shuffled to ensure variability in the experiences of the players. This file is read only on the host side and the settings are propagated to the client at the start of each round. There are 8 columns in each row. For the columns, these are: host start delay, host end delay, host adaptive time delay increase rate, host adaptive time delay decrease rate. The next 4 columns are similar, but for the Client.

[3]Last Stand Demo Video: https://youtu.be/5AMLja4qZkI
[4]Unity Netcode for Game Objects: https://docs-multiplayer.unity3d.com/netcode/current/about/
[5]Last Stand Download: https://github.com/Tokey/LastStand/releases/download/v24.03.24/Last.Stand.zip

3 EXAMPLE RESULTS

This section shows preliminary results from an ongoing study using Last Stand. The results are meant to illustrate a type of data and analysis that can come from user studies using Last Stand, but we note there is far more analysis that could be done.

We had 26 pairs of users (so far) play short (80 seconds) rounds of Last Stand using a weapon with the default settings. Players had different amounts of delay added each round, ranging from 0 ms to 200 ms.

Figure 1 shows the performance of players and QoE versus latency. The x-axis shows latency the y axis on the left shows QoE and right shows Score. Each point is the mean value of all users across the indicated conditions, shown bound by 95% confidence intervals.

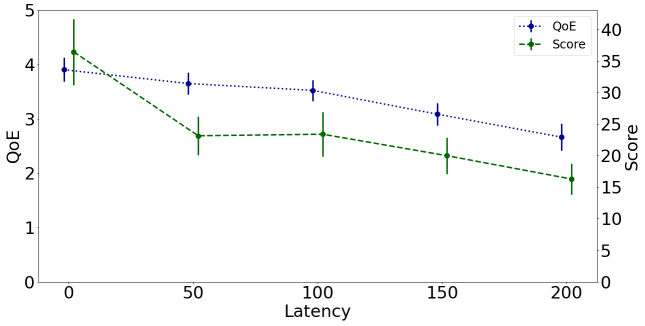

Figure 1: Mean Score and QoE versus latency.

For calculating the score, each confirmed hit gives the player 1 point, head-shots are 5 points and each confirmed kill is 10 points. QoE (1-low to 5-high) was provided from the in-game prompts at the end of each round. From the graph, player performance and QoE drop as latency increases.

4 SUMMARY

Research on delay and FPS games is needed to design better games and game systems, improve latency compensation techniques, and apply technologies beyond just games for entertainment. To that end, we have developed Last Stand a customizable FPS game specifically designed for user studies with delay.

REFERENCES

[1] Yahn W. Bernier. 2001. Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization. In *Proceedings of the GDC*. San Francisco, CA, USA.
[2] Wai-Kiu Lee and Rocky K. C. Chang. 2015. Evaluation of Lag-related Configurations in First-person Shooter Games. In *Proceedings of NetGames*. Zagreb, Croatia.
[3] Shengmei Liu, Atsuo Kuwahara, James Scovell, Jamie Sherman, and Mark Claypool. 2021. The Effects of Network Latency on Competitive First-Person Shooter Game Players. In *Proceedings of QoMEX*. Virtual Conference.
[4] Shengmei Liu, Xiaokun Xu, and Mark Claypool. 2022. A Survey and Taxonomy of Latency Compensation Techniques for Network Computer Games. *ACM Comput. Surv.* 54, 11s, Article 243 (Sept. 2022), 34 pages.
[5] P. Quax, P. Monsieurs, W. Lamotte, D. De Vleeschauwer, and N. 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 the NetGames'04*.