Lead Rush: A First-Person Shooter for User Studies and Understanding Effects of Frame Time Spikes

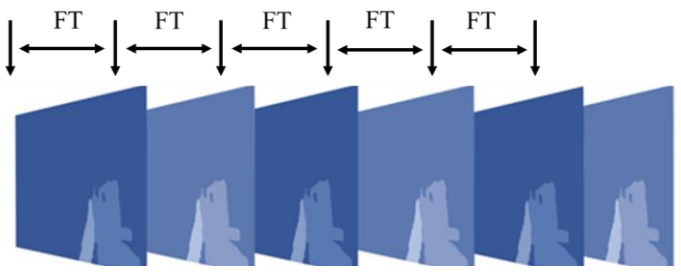
Samin Shahriar Tokey, Ben Boudaoud, Joohwan Kim, Josef Spjut, Peter Xenopoulos, Mark Claypool

QoMEX 2025

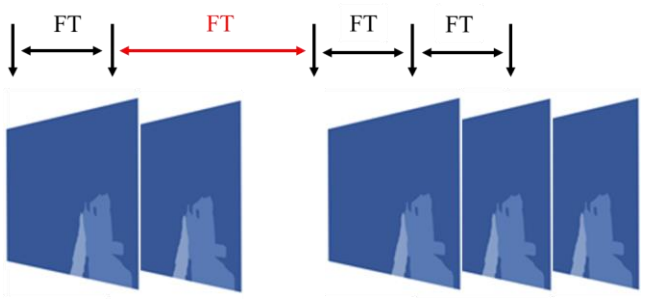
Introduction

- Commercial FPS games lack flexibility for controlled user studies
- Lead Rush: a custom FPS tool designed for research
- Runs at high frame rate (500 Hz) consistently with precise frame time spikes
- Spikes triggered during reload, aiming, or fast mouse movement
- Performance and QoE logging per tick, round and during eliminations
- Public dataset: 38 participants, 20 rounds each

Frame Time Stutters



No variation in frame time



Large variation in frame time

Related Works

Frame Time Variation

- Klein et al.: Large frametime deviations reduced perceived motion smoothness (perceptual effects).
- Liu et al.: CPU-induced frametime variation in Valorant degraded QoE.
- Xu & Claypool: Frametime variation + interruptions explained ~90% of QoE; average FPS was unreliable.

Frame Rate Studies

- Tokey et al.: Player performance plateaued at 60 fps, but perceived smoothness improved up to 500 fps.
- Claypool et al.: Higher framerates improved aiming and player performance in FPS games.

Adaptive Display Techniques

- Watson et al.: G-Sync improved player performance in Battlefield 4.
- Denes et al.: Adaptive refresh rates (perceived speed-based) yielded smoother visuals than fixed rates but couldn't remove large spikes.

Prior work studied random frametime spikes, but event-specific spikes during player actions had not been studied — our study addresses this gap.

Lead Rush

- Custom FPS built in Unity 2023, designed to run at 500 Hz
- Enemy: orb charging the player using NavMesh AI
- Weapon: hit scan based with procedural recoil, sway, ADS, reloads and 750 rounds/min fire rate
- Frame time spikes added during game events
- Performance and QoE logged each round

Gameplay Video

Methodology

Event Based Stutters

Three gameplay events were chosen: **Reload (low impact), Mouse Speed (medium), and On-Target Aiming (high).**

- **Reload:** Spike added whenever the gun reloads (auto/tactical).
- **Mouse Speed:** Spike when mouse motion exceeds a threshold; cooldown prevents repeats.
- **On-Target:** Spike when crosshair enters a collider around the enemy; collider size oscillates; with a cooldown.

Frames timed at nanosecond precision; frames were delayed using a busy loop to match spike.

Experimental Design

- **20 rounds** per participant (2 practice, 15 main, 3 control)
- **60 seconds** per round
- End-of-round questions:
  - **Rate the visual smoothness of the round** – (Very Choppy) 1 to 5 (Very Smooth), slider
  - **Was the smoothness of the round acceptable?** (Yes/No)
- Configs randomized across sessions using latin square
- Frame Time Conditions:
  - **Practice**: 0 ms, 675 ms on-target
  - **Control**: 3 rounds without spikes (Rounds 3, 11, 20)
  - **Reload**: 6 rounds (2×75 ms, 2×225 ms, 2×675 ms)
  - **Mouse Speed:** 3 rounds (1×75 ms, 1×225 ms, 1×675 ms)
  - **On-Target**: 6 rounds (2×75 ms, 2×225 ms, 2×675

Dataset

Contents

- **Player logs**: actions, shots, hits, misses, score, deaths, etc.
- **Round logs**: configs, spike events, QoE responses etc.
- **Enemy logs**: spawns, positions, status, angles, etc.
- **Reaction time test** per participant

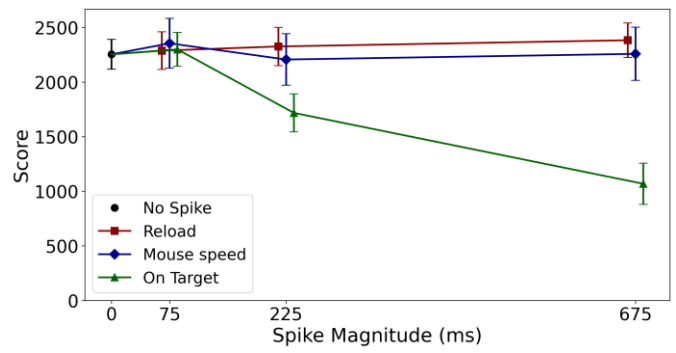
Per session data organized by folder.

Dataset Download

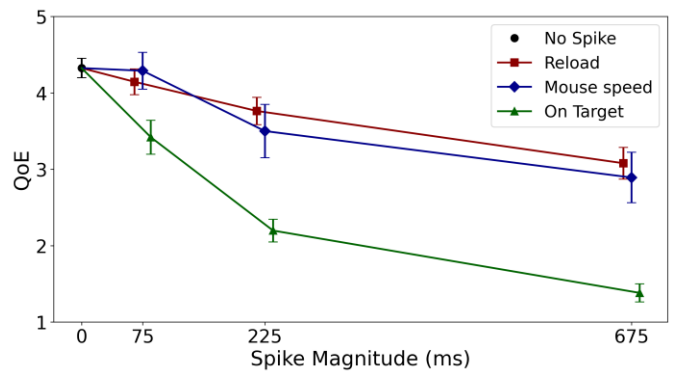
Demographics

User	Age	Gender	Gaming Skill (1-5)	FPS Skill (1-5)	Reaction Time (ms)
38	20.4 (2.7)	♂30 ♀5 ☿3	3.5 (0.7)	3.9 (0.8)	194.5 (78.9)

Performance



QoE



On-target spikes had higher impact on performance and QoE; reload and mouse-speed spikes had lower impact — showing that when spikes happen matters.

Future Work

Short Term

- Compare event-based spikes (reload, aiming, movement) using additional metrics such as reaction time, accuracy, miss-angle, time-to-hit and time-to-kill
- Explore if graphical fidelity has any correlation with frametime spike or not.

Long Term

- Extend experiments to other game genres (platformers, MOBAs, racing etc.)
- Develop models that predict QoE using both spike magnitude and event/task
- Mitigate effects of spikes with adaptive hitbox scaling.