FirstPersonScience: An Open Source Tool for Studying FPS Esports Aiming

Ben Boudaoud
bboudaoud@nvidia.com
NVIDIA
Durham, USA

Josef Spjut
jspjut@nvidia.com
NVIDIA
Durham, USA

Pyarelal Knowles
pknowles@nvidia.com
NVIDIA
Santa Clara, USA

Arjun Madhusudan
amadhusudan@ncsu.edu
NCSU/NVIDIA
Durham, USA

Joohwan Kim
sckim@nvidia.com
NVIDIA
Santa Clara, USA

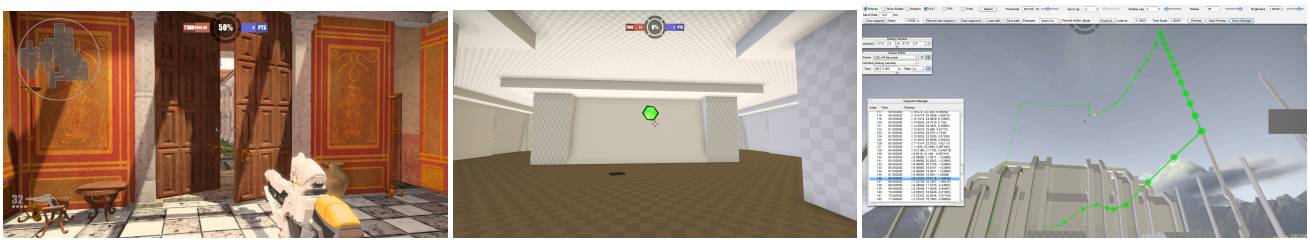

Figure 1: Screenshots of various experimental configurations of FPSci including (left) a high fidelity visual experience, (center) a typical high-performance experiment, and (right) a more advanced waypoint editor mode for world-space targets.

ABSTRACT

First-person shooters (FPS) games are dominant in the competitive gaming and esports community. However, relatively few tools are available for experimenters interested in studying mechanics of these games in a controlled, repeatable environment. While other researchers have made progress with one-off applications as well as custom content and mods for existing games, we are not aware of a general purpose application for empirically studying a broad set of user interactions in the FPS context. For the past few years our team has developed, maintained, and deployed First Person Science (FPSci), a tool for controlled user studies in FPS gaming. FPSci experimenters configure their desired base environment, as well as conditions and user preferences using a simplified JSON-esque set of input configurations, and results are stored in an SQLite database. By allowing finer grained parametric control of the environment together with frame-wise logging of player state and performance metrics, we achieve a level of granularity of control not offered by other solutions. FPSci is available as an open source project [1] under a CC BY-NC-SA 4.0 license.

CCS CONCEPTS

• **Human-centered computing** → **User studies**; • **Applied computing** → **Computer games**.

KEYWORDS

latency, first person targeting, games, aiming

[1]Source available at https://github.com/NVlabs/FPSci

SIGGRAPH '22 Talks, August 8–11, 2022, Vancouver, CA

ACM Reference Format:
Ben Boudaoud, Josef Spjut, Pyarelal Knowles, Arjun Madhusudan, and Joohwan Kim. 2022. FirstPersonScience: An Open Source Tool for Studying FPS Esports Aiming. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '22 Talks), August 8–11, 2022, Vancouver, CA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3450550.3465347

1 INTRODUCTION

FPSci grew from a desire to study FPS gaming in a non-performance limited environment. The application was originally designed for a single experiment that required high run time performance, removing any system bottlenecks that might impact user experience. This experiment(s) focused on independent control of frame rate and latency, demonstrating that, when effectively decoupled, the benefits of reduced latency far outweigh those of increase frame rate [Spjut et al. 2019]. However, with a growing set of interests around various interactions in the FPS context, we set out to improve the generality of the platform via its ability to be easily configured to match a wide subset of FPS experiences without compromising performance. The resulting FPSci platform supports a wide range of parameters and has been used for over a dozen experiments studying a wide range of questions for FPS gaming performance.

2 RELATED WORK

For users interested in improving their aiming performance, aim trainers such as KovaaK's [2] and AimLab [3] provide interesting opportunities to create scenarios, measure performance, and guide improvement. Similarly PLAYMASTER [4] from Logitech seeks to enable improved data collection within the context of Counter Strike: Global Offensive (CS:GO) environment. Alternatively, some

[2]More information available at https://themeta.com/kovaaks/
[3]More information available at https://aimlab.gg/
[4]More information availabe at https://playmaster.gg/

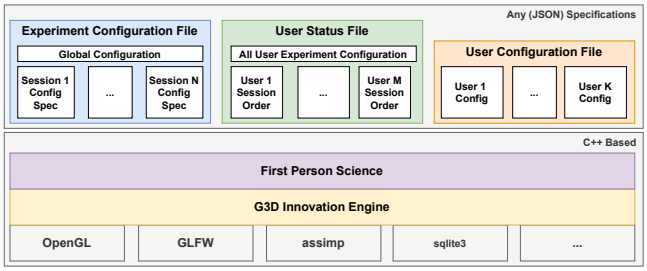

Figure 2: FPSci configuration and software architecture.

researchers in the field have opted for analyzing *replay files*, or summaries of player actions recorded from real games, for their work [Makarov et al. 2017]. This solution has the downsides that real, in-game situations can be hard to control and repeat, and conclusions drawn from these situations risk confounding additional high-level, game-specific skills and strategy with particular mechanics researchers often seek to study. To our knowledge, none of these solutions provide the level of environmental control nor the granularity of logged results provided by FPSci.

3 DESIGN

FPSci is written in C++ on top of the G3D Innovation Engine [5] using OpenGL for rendering. The project uses G3D's Any file specification, a super set of JSON that allows for text-based description of many variables loaded at runtime. System, user, and experiment-level parameters are all specified using a wide range of configuration. Many experiments using FPSci require little to no C++ code modification, being specified entirely using the Any syntax.

FPSci experiments are designed around a repeated-trial paradigm. The *experiment* is a set of global configuration, agnostic of individual users, provided as a single Any file to FPSci. Within an experiment an array of *sessions* define conditions that each user will experience (possibly in different orders) with some parameter(s) of interest changing between sessions. Each session consists of repeated *trials* which do not vary in configuration, but may produce different behavior randomized within ranges, for example target position/motion or weapon behavior. A trial uses one or more targets with specified model, size, color, sound, and motion parameters.

User configuration is provided via two more configuration files. A generic user configuration file provides experiment-agnostic user information including an ID, mouse sensitivity, DPI, and preferred reticle parameters. An experiment-specific user status file holds the remaining experiment-contingent, per-user information including the currently selected user as well as (optionally randomized) session order. The user status file is updated as the user completes sessions to track and save their progress through the experiment.

3.1 Configuration Parameters

FPSci supports a wide range of user input parameters including (but not limited to):

- Experiment design and flow control
- Target model, color, size, position, and motion
- Scene geometry and spawn location
- Player camera and rendering properties
  - FoV, AA, depth of field, tone mapping, gamma, motion blur
  - Frame rate/time and delay (w/ randomization)
  - Resolution for 2D, 3D, and composited content
  - Screen-space shaders for 2D, 3D, and composite pass
- Trial timing, counts, feedback, and invalidation criteria
- Various audio clip and volume control
- Player movement rate and jump physics
- Weapon configuration
- HUD controls (progress, ammo, health, weapon status, etc)
- Logging controls for various results
- In application menu controls
- Feedback questions prompted for after various sessions
- Arbitrary commands to be run around session/trial start/end

For additional info refer to the FPSci documentation.

4 APPLICATIONS

We have used FPSci in a number of studies of performance impacts in FPS gaming. Some examples include:

- Independent frame rate and latency [Spjut et al. 2019]
- Applications of late warp to FPS gaming [Kim et al. 2020]
- Impacts of low end-to-end latency [Spjut et al. 2021]
- Impacts of audio/visual latency
- Understanding mouse sensitivity [Boudaoud et al. 2022]
- Display size and resolution [Spjut 2022]
- Bullet drop and frame rate effects

5 CONCLUSION

FPSci is a highly configurable application intended for broad FPS performance and perception experimentation. From graphics and rendering research to user experience and system-level optimizations FPSci presents interesting opportunities for exploring relevant questions. We encourage anyone with FPS experiment ideas to try FPSci and contact us for any support or collaboration.

REFERENCES

Ben Boudaoud, Josef Spjut, and Joohwan Kim. 2022. Mouse Sensitivity in First-person Targeting Tasks. In *2022 IEEE Conference on Games (CoG)*. IEEE, 1–8.

Joohwan Kim, Pyarelal Knowles, Josef Spjut, Ben Boudaoud, and Morgan Mcguire. 2020. Post-Render Warp with Late Input Sampling Improves Aiming Under High Latency Conditions. *Proc. ACM Comput. Graph. Interact. Tech.* 3, 2, Article 12 (aug 2020), 18 pages. https://doi.org/10.1145/3406187

Ilya Makarov, Dmitry Savostyanov, Boris Litvyakov, and Dmitry I Ignatov. 2017. Predicting winning team and probabilistic ratings in "Dota 2" and "Counter-Strike: Global Offensive" video games. In *International Conference on Analysis of Images, Social Networks and Texts*. Springer, 183–196.

Josef Spjut. 2022. Improving aiming time on small FPS targets with higher resolutions and larger screen sizes. https://developer.nvidia.com/blog/improving-aiming-time-on-small-fps-targets-with-higher-resolutions-and-larger-screen-sizes/

Josef Spjut, Ben Boudaoud, Kamran Binaee, Jonghyun Kim, Alexander Majercik, Morgan McGuire, David Luebke, and Joohwan Kim. 2019. Latency of 30 Ms Benefits First Person Targeting Tasks More Than Refresh Rate Above 60 Hz. In *SIGGRAPH Asia 2019 Technical Briefs* (Brisbane, QLD, Australia) *(SA '19)*. Association for Computing Machinery, New York, NY, USA, 110–113. https://doi.org/10.1145/3355088.3365170

Josef Spjut, Ben Boudaoud, and Joohwan Kim. 2021. A Case Study of First Person Aiming at Low Latency for Esports. https://doi.org/10.31219/osf.io/nu9p3

[5] More information available at https://casual-effects.com/g3d/