Proceedings of the 2020 Annual General Donald R. Keith Memorial Capstone Conference
West Point, New York, USA
April 30, 2020
A Regional Conference of the Society for Industrial and Systems Engineering

Assessing System Latency in Virtual Reality Head-mounted Displays: A Benchmark Comparison between Unity and PsychoPy

Noah Faurot[1], Peter Shevchenko[1], Christian Barentine[2], and Anthony Ries[2]

[1]Department of Systems Engineering
[2]Warfighter Effectiveness Research Center
United States Air Force Academy, Colorado Springs, CO

Corresponding Author: anthony.j.ries2.civ@mail.mil

Author Note: Cadet 1st Class Noah Faurot is an undergraduate student at the United States Air Force Academy. He is majoring in Systems Engineering with a focus on Human Factors. Post-graduation he is scheduled to commission in the Air Force and attend Remote Pilot Aircraft Operator training. Cadet 1st Class Peter Shevchenko is an undergraduate student at the United States Air Force Academy. He is majoring in Systems Engineering with a focus on Human Factors. Post-graduation he is pursuing a commission in the United States Space Force as a Space Operations Officer. Christian Barentine is a computer programmer working with Dr. Anthony Ries on eye tracking research. He has a B.A. in Philosophy from New Mexico State University, and is currently earning his Masters in Computer Science - Games and Media Integration from University of Colorado, Colorado Springs. He specializes in Python, the Unity Game engine, and VR. Dr. Anthony Ries is a Research Psychologist at the U.S. Army Research Laboratory in Aberdeen, MD. He is currently a visiting researcher in the Warfighter Effectiveness Research Center (WERC) at the U.S. Airforce Academy. He received the B.S. degree in Psychology from Northwest Missouri State University in 2000 and the M.A. and Ph.D. degrees in Cognitive Neuroscience from the University of North Carolina at Chapel Hill in 2003 and 2007.

Abstract: Virtual reality (VR) head mounted display (HMD) systems provide a higher degree of realism for experimentation. However, this comes with a cost as VR systems inherently introduce system lag due to heavy processing loads. This system lag can disrupt the ability to assess the timing of human behaviors relative to experimental manipulations. Our proposed solution was to quantify and account for the system latency by comparing the timing of session events generated with Unity and PsychoPy, popular VR and psychophysics programming languages, to times measured with a photo diode. We leveraged system engineering approaches paying particular attention to the system analysis process. Applying this process, the team quantified the timing data in MATLAB to directly address DoD stakeholder interests. The results indicated a variable lag present within the Unity-based VR system which was much less pronounced when using PsychoPy.

Keywords: Unity, PyschoPy, Psychophysics, Systems Engineering, Photo Diode

1. Introduction

Recent developments in virtual reality technology, which has largely been driven by the gaming community, is now becoming more prevalent in training and clinical environments. Additionally, VR is beginning to provide researchers a unique opportunity to conduct experimentation in an immersive environment while still preserving necessary experimental control (Clay et al, 2019). While VR has the advantage of providing a 360 degree, fully immersive experience for experimentation, the software used to create VR environments is not designed to have millisecond precision necessary for most studies of eye tracking and/or neurological measurements of human cognition. DoD programs are already seeing the potential of using VR and similar technology such as augmented reality (AR) in both basic and applied experimentation such as Army's Integrated Visual Augmentation System (IVAS) and the Air Force's Pilot Training Next (PTN); however, their use cases must be validated and tested to ensure reliability specifically as it relates to temporal precision in experimentation.

In this paper, we describe the systems engineering approach we used to calculate temporal latency of a popular VR headset, HTC Vive, with embedded eye-tracking hardware from Tobii Technologies. Specifically, we developed identical versions of a stimulus presentation paradigm using either Unity software, which is the gold standard for VR development, or PsychoPy, software used in many psychophysics experiments requiring precise experimental and timing control. PsychoPy operates at a set frame rate, meaning that the time between one update and the next is consistent, and extremely stable. Unity on the other hand, has an unlocked frame rate by default. This means that when Unity is finished updating, it will immediately move on to the next update, regardless of how long the previous update took. This technique allows Unity to provide a smoother rendering to the screen, at the potential cost of unpredictable timing.

2. Systems Analysis

In order to test the latency of the VR system we first stepped through the design definition process as a means to outline the required system elements. Consulting the vision science literature enabled us to determine how best to measure temporal precision. The most commonly used approach uses a photo diode attached to a monitor to determine the instant a luminance change is detected on the screen (Plant et al., 2004). This time is then compared to the time at which the computer or system requested the luminance change. With this approach in mind, we were able to define the system elements required

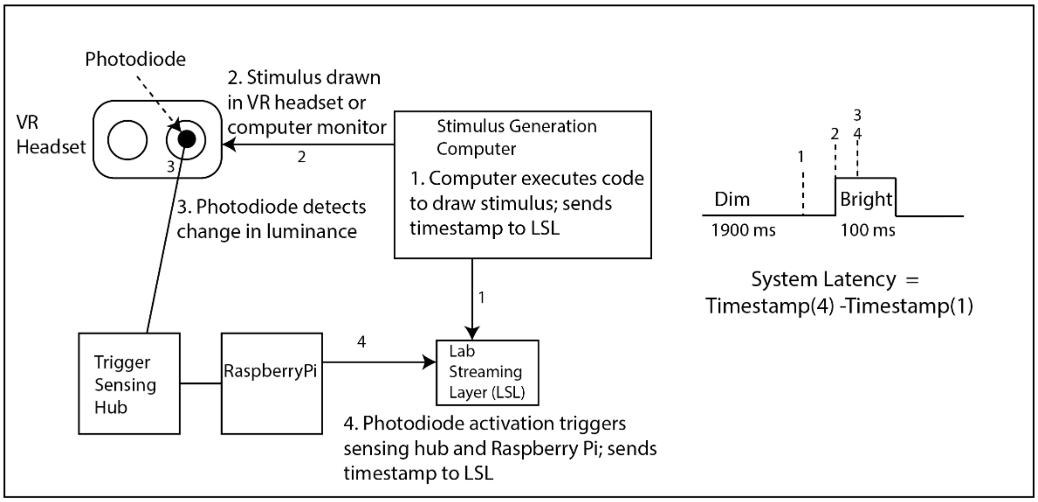

Figure 1. Flow chart of the system design used to measure the temporal precision of the VR and benchmark desktop system

for this study (Wiley, 2015). The first element being a simple program within both Unity and PsychoPy software that would transition the monitor from black to white every 2 seconds. The code requested the white, bright screen flash duration of 100ms with 1900 ms between flashes. Once the code was called to present the stimulus (dark screen to bright), it sent an event marker with corresponding timestamp to Lab Streaming Layer (LSL), a network based recording software designed to integrate multiple data streams with sub millisecond precision (Kothe, 2014). The code also sent a marker and timestamp when the stimulus was removed (i.e. when it went from bright to dark). The second element of our system focused on the photo diode. We acquired a trigger hub (Wearable Sensing) which uses a photo diode attached to a suction cup as an input. When the photodiode detected a luminance change it sent a 4.5 V pulse as output. To detect the output and record the time of the flash at the monitor, we first used the audio input port to record the photodiode values. However, after evaluation and further research we realized the audio port itself was introducing a significant lag in system measurement. This led us to use a Raspberry Pi, which consists of a low-latency, high sampling rate and more robust and stable approach to measure the photodiode event. The photodiode event was also marked and timestamped in LSL. The final element within our system consisted of a VR headset to evaluate the system latency. We measured the time between the timestamp in which the stimulus display was requested to the timestamp at which the photo diode detected the change in luminance in the VR headset, when run in Unity, or on the monitor, when run with PsychoPy. Figure 1 shows a design layout realized at the end of the integration process which allowed us to assess the precision of the VR system compared to traditional desktop approach.

3. Methods and Measures

Table 1. Median Summary Statistics.
Inter-quartile range (IQR) in parentheses. SOA – Stimulus Onset Asynchrony

	Event Log SOA (ms)	Event Log Duration (ms)	PhotoDiode SOA (ms)	PhotoDiode Duration (ms)	System Latency (ms)
UNITY	2011.9 (9.9)	104 (9.5)	2010.7 (11.2)	115.4 (.001)	35.43 (7.29)
PsychoPy	1997.5 (.29)	99.9 (.28)	1997.5 (.01)	124.95 (.001)	28.31 (.38)

For each test we performed a 10 minute recording of the flash events. Flash events occurred every two seconds. The same stimuli and timing parameters were employed using Unity software with the photodiode placed in the VR headset and with PsychoPy with the photo diode placed on the computer monitor. We evaluated multiple measures of temporal precision within each test. We measured the time between the onset to offset of the bright stimulus, i.e. its duration using both the log file event codes generated by the software as well as those detected by the photodiode. A similar measure was taken to assess stimulus onset asynchrony (i.e. time from bright stimulus onset to the next bright stimulus onset). Central to our focus was the system latency which we defined as the time between when the software requested the bright stimulus to be shown to when the photo diode actually detected the stimulus. Summary statistics from these measures are presented in Table 1.

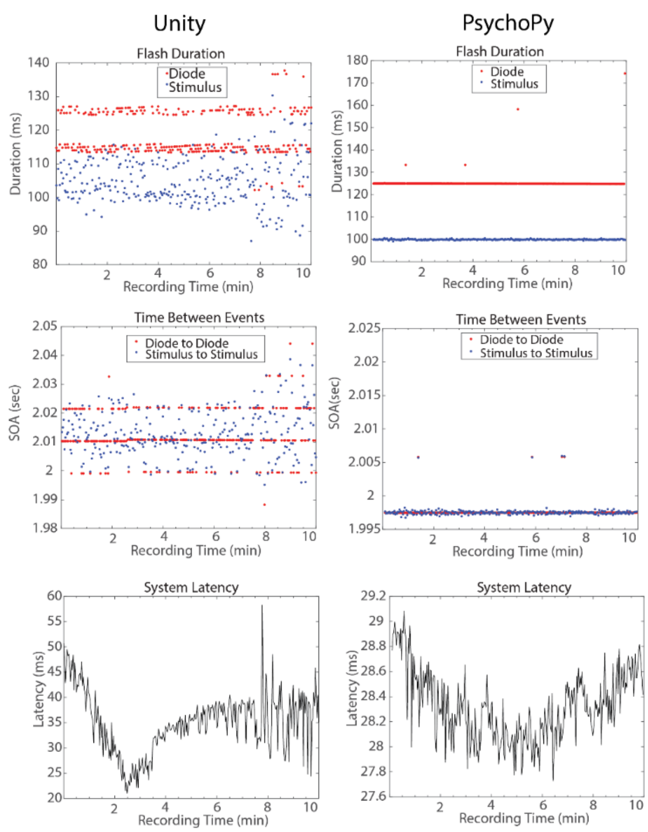

Figure 2. Latency results. Flash duration (top), SOA (middle), System latency (bottom)

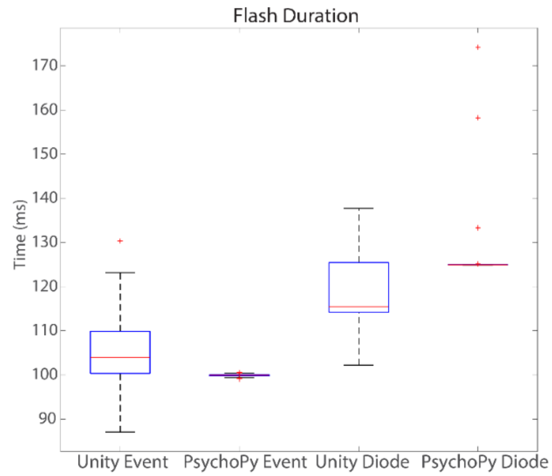

Figure 3. Calculated stimulus durations from Unity and PsychoPy using system event logs and photo diode

The results of the requested duration of 100ms show that, when evaluating the software event logs, PsychoPy presented values closer to the requested duration when compared to Unity and that the variability was much higher for Unity than PsychoPy. Figure 2 top shows the calculated duration from the software log file and photo diode using both Unity and PsychoPy software. Each point represents one sample (difference between offset and onset times) over the 10 minute recording period. Figure 3 shows the summary of the flash stimulus duration data. A similar finding was also found when evaluating the stimulus onset asynchrony (SOA) between bright stimuli as seen in the middle of Figure 2 and Figure 4. Similar to the duration the SOA shows more variability within the Unity presentation compared to PsychoPy. Additionally, our system calculations suggest Unity often lags behind the specified durations as most values are above 100ms and 2 seconds for duration and ISI respectively. Critically, system latency was markedly different between the two software systems with higher system latency measured with Unity with respect to PsychoPy (see bottom of Figure 2 and Figure 5).

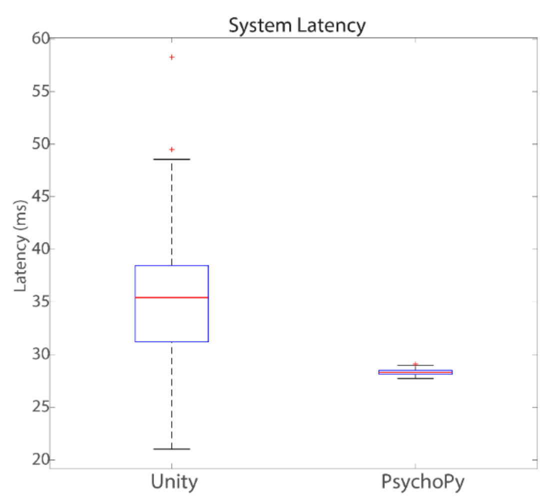

Figure 4. Calculated stimulus to stimulus durations.

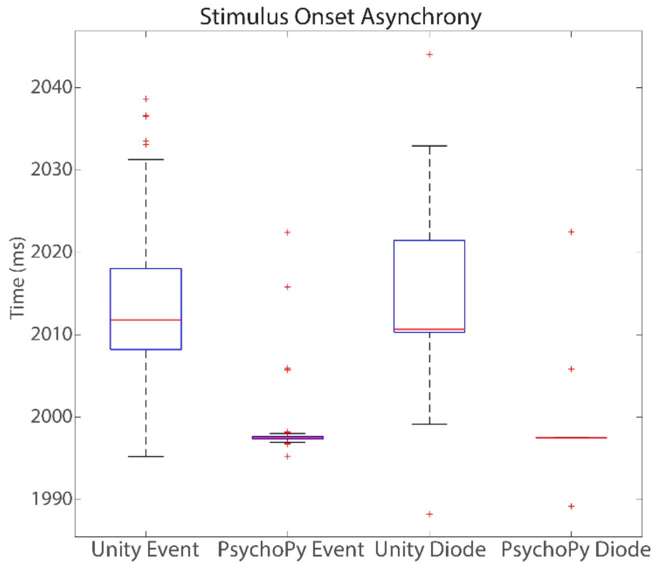

Figure 5. Calculated stimulus to stimulus durations

4. Conclusions

This paper examined the observed system latency of a head mounted display system when running Unity compared to PsychoPy software. Utilizing systems engineering principles we were able to first establish DoD stakeholder interests and then subsequently proceed through multiple systems engineering (SE) technical processes to a functioning system that accomplishes the stakeholder needs (Wiley, 2015). Our results indicate that PsychoPy returned latency values that were considerably more accurate and precise to those obtained from VR-based Unity software. Unity proved to have a greater standard deviation between samples which contributed to a higher mean latency time. However, this research is limited to the capacity of the photo diode. The original photo diode used provided unusual data and warranted a systematic evaluation to determine the cause which eventually led to a more advanced photo diode being used. Additionally, future research in this area could include utilizing different coding routines for presenting stimuli in Unity as well as different commercial brands of VR headsets to determine if there are significant differences in measured latency between different products.

5. References

Clay, V., et al. (2019). Eye Tracking in Virtual Reality. *Journal of Eye Movement Research*, vol. 12, no. 1, ser. 3, 2019. *3*.

Kothe, C. (2014). Lab Streaming Layer (LSL). Retrieved from *GitHub*: github.com/sccn/labstreaminglayer/search?q=Kothe&type=Code Allen, T. (1974). *Vanishing wildlife of North America.* Washington, D.C.: National Geographic Society.

Boorstin, D. (1992). *The creators: A history of the heroes of the imagination.* New York: Random House.

Devitt, T. (2001, August 2). *Lightning injures four at music festival*. Retrieved January 23, 2002, from The Why? Files: http://whyfiles.org/137lightning/index.html

GVU Center. (n.d.). Retrieved August 8, 2000, from 8th WWW user survy: http://www.cc.gatech.edu/gvu/usersurveys/survey1997-10/

Harlow, H. (1983). Fundamentals for preparing psychology journal articles. *Journal of Comparative and Physiological Psychology, 55*, 893-896.

Henry, W. I. (1990, April 21). Making the grade in today's schools. *Time*, *135*, pp. 28-31.

Kalette, D. (1986, July 21). California town counts town to big quake. *USA Today*, *9*, p. A1.

Nicol, A., & Pexman, P. M. (1999). *Presenting your findings: A practical guide for creating tables.* Washington, D.C.: American Psychological Association.

Searles, B., & Last, M. (1979). *A reader's guide to science fiction.* New York: Facts on File, Inc.

The safety of genetically modified food crops. (2002, February). Retrieved March 22, 2005, from Health Canada: http://www.hc-sc.gc.ca/english/protection/biologics_genetics/gen_mod_foods/genmodebk.html

Plant, R., et al. (2004). Self-Validating Presentation and Response Timing in Cognitive Paradigms: How and Why?. *Behavior Research Methods, Instruments, & Computers*, vol. 36, no. 2, 2004.

Wiley. (2015). *INCOSE Systems Engineering Handbook*. John Wiley & Sons, 2015.