How to demonstrate delay? Let's play!

Ragnhild Eg
Westerdals - Oslo School of Arts,
Communication and Technology
Faculty of Communication
Oslo, Norway
egrag@westerdals.no

Kjetil Raaen
Westerdals - Oslo School of Arts,
Communication and Technology
Faculty of Technology
Simula / University of Oslo
raakje@westerdals.no

ABSTRACT

The speedy interactions of many computer games demand prompt reactions from player and system alike. This demonstration invites participants to experience in-game latency. We present variations of two familiar games that use either mouse or key inputs to control actions seen on the monitor. While playing the games, participants can make changes to the temporal offset between a motor input and a resulting visual action, providing them with a direct experience of motor-visual delays. In turn, they will gain an understanding of how performance may suffer when trying to compensate for such delays.

CCS Concepts

•**Human-centered computing** → **Laboratory experiments;** *Graphical user interfaces;* Pointing devices; •**Software and its engineering** → Interactive games;

Keywords

interface delay, experiment demonstration, human-computer interaction

1. INTRODUCTION

When you aim to score, you would hope that the moment you hit the button, the ball will actually bounce. Unfortunately, this is not always the case. Computer games are highly dependent on network and system operations, and delays can arise in the form of game actions that lag behind player inputs [1]. In the physical world, actions have predictable and nearly immediate consequences. Because the digital world often serves as a representation of the physical one, it follows that we expect the same immediacy from many of our digital actions. Yet, network and computational limitations can hinder the fulfilment of our expectations. In the digital world, the link between an input and an output can be weakened by temporal separation. The example game in figure 1 illustrates delay as a temporal separation between a key input and a visual output in the form of a ball bounce. Many computer games are especially contingent on the timing of events. In action games, players must react quickly to perform well, but a successful outcome is equally dependent on swift responses from the game system.

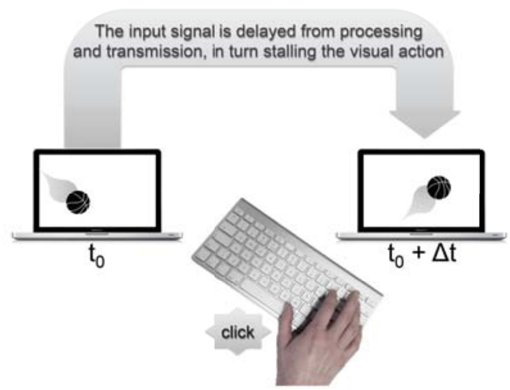


Figure 1: Imagined time-dependent game with delay between motor input and visual action.

To provide first-hand experiences with visual actions that lag behind motor commands, we have designed two games that we present in a technical demonstration. The games give players a feel of motor-visual delays and allow players to explore how these delays can affect their performance.

1.1 Doing and seeing at different times

While we see, hear and feel performed actions almost at the same time, the perceptual system does not require absolute synchrony for subjectively coherent experiences. Due to the different speeds of light and sound, corresponding visual and auditory signals are unable to trigger sensory receptors simultaneously. In addition, sensory signals travel along different neural pathways and are processed in specialised cortical regions. Still, the perceptual system is able to align and synchronise signals from different modalities, and it does so with a buffer [2]. For instance, a sound that follows a sight can lag by hundreds of milliseconds and still go unnoticed [3]. Similar buffers have been observed for de-

MMSys'16 May 10-13, 2016, Klagenfurt, Austria

ACM ISBN 978-1-4503-4297-1/16/05.
DOI: http://dx.doi.org/10.1145/2910017.2910628

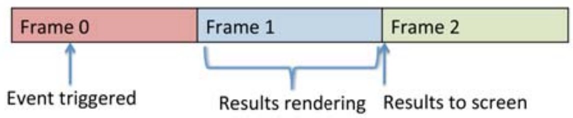


Figure 2: Timeline for events with double-buffering.

lays between physical actions and resulting visual events, such as a button push that triggers a flash [4]. In other words, the human perceptual system tolerates small inconsistencies in the timing of sensory events, it will even cover them up. Yet, all the while this mechanism goes unnoticed, it may still affect behaviour and performance.

Alongside perceptual processes, humans actively use cognitive mechanisms when interacting with the environment. We can predict the when and where of a movement when we act on an object and we can point to a flying ball's landing point with a fair degree of accuracy. These skills exemplify how we attribute an effect to its cause, despite separations in time [5]. People can even learn to compensate for lagging visual feedback when moving an object past obstacles [6], at least with short temporal offsets. In a study by Cunningham and colleagues [6], delays longer than 235 ms would impair performance on the obstacle avoidance task. Similarly, MacKenzie and Ware studied participants' abilities to move a mouse cursor to a target, under various delay conditions, and they noted significant reductions in performance at both 75 ms and 225 ms delays [7]. These two studies address one type of temporal adaptation where visual delays follow motoric inputs, which leads to poorer performance in obstacle-avoidance [6] and in target-acquisition [7] tasks.

The findings illustrate how important timing is in highly interactive tasks and computer games are perfect examples in this respect. Moreover, computer games are directly affected by the network and system limitations that translate to latency (most recently with the introduction of cloud gaming services [8]), which in turn affect the game play [9, 10]. Research in this domain has established that network delays are particularly detrimental to precision in first-person shooting and racing games [11]. Despite numerous efforts to establish a delay threshold for impaired performance on a game task, previously published results are too variable for comparison [6, 7, 9, 10]. Furthermore, the outlined studies have applied experimental setups that predominantly rely on mouse-cursor movements. The majority have also refrained from introducing delays shorter than 100 ms and do not report inherent delay in their setup.

Delays in interactions have many potential sources, not all predictable. Among these are delays introduced by display devices such as screens, input devices such as mice, graphics hardware as well as the applications itself. Among the more important sources of delay is double-buffering. When double-buffering is used, rendering follows the sequence shown in figure 2: Assume frame 0 is the frame during which an event from an input device is registered. Frame 1 contains the result of the event, and at the time of frame 2 the result is sent to screen. This gives a minimum of one and a maximum of two full frame times from input event to result on screen. Depending on frame-rate this translates to around 17 - 33 ms of delay. Further potential sources of delay are described in earlier work [1].

1.2 Detecting delays and playing with delays

Our research in this field commenced with an investigation into how much delay a system can operate with between a motor input and a visual output before it becomes noticeable. We ran a discrimination experiment with 51 participants that we equipped with a button device. On a monitor in front of them, participants observed a rotating black disc that changed direction in response to a push of the button and they made judgements on the change being immediate or delayed. Delays introduced between the button push and the rotation change ranged from 0 ms to 500 ms. However, when interpreting our results, we made sure to account for the internal system delay, measured to be approximately 50 ms. Consequently, we established a discrimination threshold close to 200 ms, which represents the point where participants are equally likely to detect a delay as they are to miss it. If this result can be generalised to the population at large, we could conclude that the average person is sensitive enough to detect motor-visual delays above 200 ms at better than chance levels. Still, individual variations are large.

This initial investigation yielded informative thresholds for subjective points of detection. However, it provided no information on how motor-visual delays affect the behaviour or performance of a person engaged in a game, or a similarly dynamic computer interaction. Hence, our next line of investigation will look into how delays at various levels, above and below the 200 ms threshold, influence player performance. Importantly, we commence this work with attention to inherent system delays and caution in recording and compensating for these. Moreover, we put focus on more than one type of interaction and include both mouse movements and key presses. We plan a series of experiments where participants play different games at various levels of delays, using distinct motor inputs, while we measure and record their game performance. This demonstration presents the first step in our endeavour.

2. GAME DESIGN

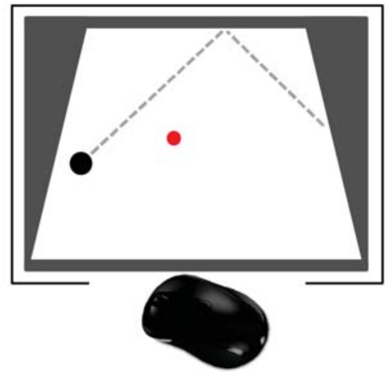
We designed two separate games to investigate how user performance is affected by interface delay; both assess the subjective ability to compensate for visual lags. The first game, "Puck hunt", is modelled after [7] and is also inspired by Nintendo's "Duck hunt". The second game, "Bounce & break", took shape from the commercial "Breakout" games; we removed the bricks, but kept the bouncing ball. Combined, the two games cover several aspects of human-computer interactions, where both require anticipation of future locations. One relies on the computer mouse for inputs, incorporating precision movements with timed key-presses, while the other focuses on predictive timing using indirect controls of visual movements. To minimise the inherent latency of the designed systems, the games are written in C++ using OpenGL. With vertical screen-synchronisation disabled, the hardware can run at full capacity. Because of the simplicity of the games, modern hardware is able to run them at many hundreds of frames per second. Accordingly, we achieved minimum delays of less than 40 ms.

2.1 "Puck hunt" game

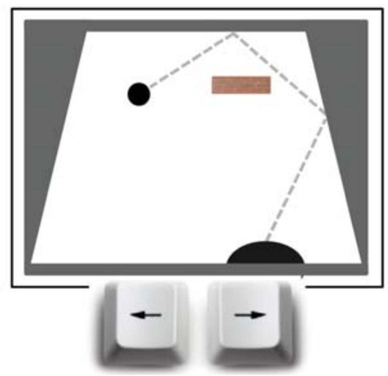
The game progresses over several short trials, each trial commences as the mouse cursor, represented by a small red ball, appears together with a larger black ball, or puck. For each trial these appear at new, arbitrary locations. An example of a starting trial is portrayed in figure 3a. The speed and trajectory of the black ball will vary from trial to trial, but the goal is consistent, use the mouse to move the red ball until it overlaps the moving black ball and click the button as quickly and as accurately as possible. The selected delay sets the time it takes for the cursor to follow the movement of the mouse, and for the registration of the click. This demands that participants must compensate for the distance covered by the black ball during the lag by targeting its future location. Game performance is assessed on the speed of the response, and the accuracy in targeting the center of the black ball.

2.2 ”Bounce & break” game

In this game, the ball moves in a straight line and rebounds from the walls with an angle that corresponds to the incoming angle, relative to the surface. Compared to a true physical system, the ball reacts in a predictable manner. The challenge of this game is to aim the ball towards a target, using the paddle at the bottom of the screen, as illustrated in figure 3b. The arrow keys move the paddle back and forth, at a constant velocity, and the chosen delay defines the time it takes for the paddle to respond. This implies that participants must compensate for the delayed movement of the paddle and plan its placement ahead of time. Performance is assessed on how long it takes to hit the target, and the number of bounces required.



(a) Puck hunt



(b) Bounce & break

Figure 3: Illustrations of the ”Puck hunt” and ”Bounce & break” games. In the first, the ball moves around the confined area and rebounds off the surfaces, the goal is to move the cursor with the mouse and target the center of the ball. The challenge of the second game is to bounce the ball off the paddle, using the arrow keys, in order to hit the brick target.

3. DEMONSTRATION SETUP

This technical demonstration comprises two interactive games and interested parties are invited to try their luck in one or both. We first show participants how to adjust and select a delay value, so that they can choose and observe how many milliseconds pass between the input action and the visual output. They thereafter receive short instructions on the controls and aim of the game, before they commence to play.

4. PRELIMINARY RESULTS

To add perspective on what participants in this demonstration will experience, we ran a pilot experiment with the ”Puck hunt” game, using a subset of the delay values we plan to implement in the full experiment. Six participants agreed to partake in the pilot, which we ran in the Mobile Tech Lab at Westerdals Oslo ACT. With a total of 54 trials, comprised of six delay values, three ball speeds, and three repetitions, the pilot took less than 10 minutes to complete. To compensate for inherent delay in the experimental equipment, we used an external setup to measure baseline delay. All results are presented with this baseline added.

The findings from the pilot are presented in Figure 4 and show that performance on this interactive game task does deteriorate with motor-visual delay and that the impact on performance increases with the delay value. From this, we surmise that the effect of delays on performance is a topic worth pursuing in a full-factorial experiment.

5. DISCUSSION AND APPLICABILITY

In this demonstration, participants have the opportunity to explore their own sensitivity to motor-visual delays and to try out the effects as in-game latency. This sensitivity varies among people, some can spot lags shorter than 50 ms, whereas others cannot notice delays that exceed 200 ms [12]. While a few studies have looked into how performance is affected by visual lags, none have considered subjective variability and looked at thresholds for detectability of delays, nor have they compared different input actions. Moreover, this line of research requires precautions with respect to inherent system delay, which we measure and compensate for in our experiment setup. Our preliminary results from the pilot run of the ”Puck hunt” game indicate that game performance is not markedly affected when delays are too short to be noticed, but we have yet to assemble and assess the complete picture.

The game aspects of the demonstration are arenas for introducing latency, but also serve as relevant applications that are affected by the challenge of synchronising digital events. The two games are designed to investigate the consequences of visual lags on player performance. For the sake of demonstration, participants will have both the knowledge and the experience of the potentially detrimental effects. Trying one or both demonstration games, players can experience the full range of delay values, from the undetectable

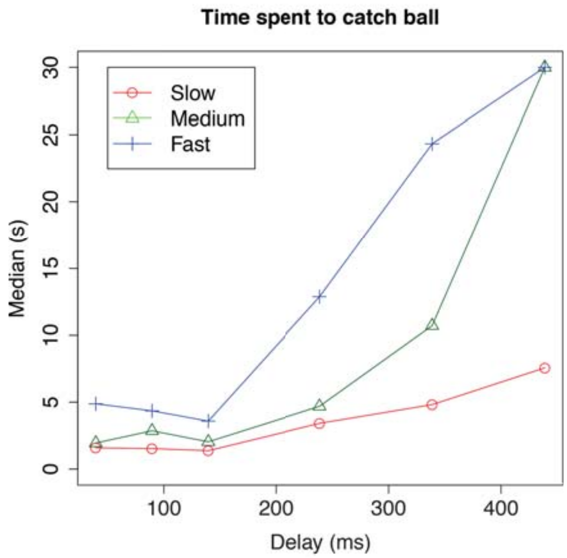


Figure 4: Results collected from a pilot run of the "Puck hunt" game, portraying how the ball becomes more difficult and takes longer time to catch (y-axis) with increasing delay (x-axis) and with faster ball speed (separate lines).

to the clearly visible. In the "Puck hunt" game, performance is assessed on participants' abilities to compensate for visual lag by predicting the cursor's future location at the time the mouse-click registers. Similarly, the "Bounce & break" game requires that players can compensate for the delayed reaction and movement of the paddle, in order to place it in the right location at the right time.

We plan to apply the games in future studies of performance and compensatory behaviour, using consistent visual lags. Latency is a problem in many human-computer interactions, but they are particularly noticeable in operations that depend on a system's swift response. Thus, our setup fulfils two purposes. First, it presents participants with an entertaining task, presumably increasing motivation and task focus. Second, the design incorporates one application we wish to explore, namely game performance as a contingency of delayed visual actions. We seek to establish thresholds for compensatory and detrimental visual delays. Moreover, we aim to find out whether game difficulty and detectability of delays will influence these thresholds.

Considering how many human-computer interactions depend on inputs and commands provided by pressing a button, a mouse, or a keyboard, this line of experimentation could be extended to most workflows that require speedy outcomes. Useful applications include practice control stations and simulators for planes, military systems, and nuclear reactors, as well as software-based tasks such as design work.

6. REFERENCES

[1] K. Raaen and A. Petlund, "How much delay is there really in current games?" *Proceedings of the 6th Annual ACM SIGMM Conference on Multimedia Systems (MMSys)*, pp. 2–5, 2015.

[2] J. Vroomen and M. Keetels, "Perception of intersensory synchrony: A tutorial review," *Attention, Perception & Psychophysics*, vol. 72, no. 4, pp. 871–884, 2010.

[3] M. Zampini, S. Guest, D. I. Shore, and C. Spence, "Audio–visual simultaneity judgments," *Perception & Psychophysics*, vol. 67, no. 3, pp. 531–544, 2005.

[4] M. Rohde, M. Scheller, and M. O. Ernst, "Effects can precede their cause in the sense of agency." *Neuropsychologia*, vol. 65, pp. 191–196, Dec. 2014.

[5] C. Stetson, X. Cui, P. R. Montague, and D. M. Eagleman, "Motor-sensory recalibration leads to an illusory reversal of action and sensation." *Neuron*, vol. 51(5), pp. 651–659, Sep. 2006.

[6] D. W. Cunningham, V. A. Billock, and B. H. Tsou, "Sensorimotor adaptation to violations of temporal contiguity." *Psychological Science*, vol. 12(6), pp. 532–535, Nov. 2001.

[7] I. S. MacKenzie and C. Ware, "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*, Apr. 1993, pp. 488–493.

[8] W. Cai, M. Chen, and V. Leung, "Toward gaming as a service," *Internet Computing, IEEE*, vol. 18(3), pp. 12–18, May 2014.

[9] C. Savery, T. C. N. Graham, and C. Gutwin, "The human factors of consistency maintenance in multiplayer computer games," in *Proceedings of the 16th ACM International Conference on Supporting Group Work*, 2010, pp. 187–196.

[10] D. Stuckel and C. Gutwin, "The effects of local lag on tightly-coupled interaction in distributed groupware," *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, pp. 447–456, Nov. 2008.

[11] M. Claypool and K. Claypool, "Latency can kill: Precision and deadline in online games," in *Proceedings of the 1st Annual ACM SIGMM Conference on Multimedia Systems (MMSys)*, Feb. 2010.

[12] K. Raaen, R. Eg, and C. Griwodz, "Can gamers detect cloud delay?" in *Proceedings of the 13th Annual Workshop on Network and Systems Support for Games (NetGames)*, vol. 200, Nagoya, 2014.