WPI-CS-TR-16-03 July 2016

The Effects of Delay on Game Actions: Moving Target Selection with a Mouse

by

Mark Claypool
Ragnhild Eg
and Kjetil Raaen

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

The Effects of Delay on Game Actions – Moving Target Selection with a Mouse

Mark Claypool
Worcester Polytechnic Institute – Computer Science
Worcester, MA, 01609, USA
claypool@cs.wpi.edu

Ragnhild Eg
Westerdals – Oslo School of Art, Communication and Technology
Box 9215 Grønland, 0134 Oslo
egrag@westerdals.no

Kjetil Raaen
Westerdals – Oslo School of Art, Communication and Technology
Box 9215 Grønland, 0134 Oslo
raakje@westerdals.no

ABSTRACT

In modern computer systems, user input, particularly for computer games, is affected by delay from both local systems and networks and servers. While general awareness of the degradation effects of delay to player performance and player quality of experience are well known, a detailed understand quantifying how player actions are impacted is missing. This work presents a detailed user study that gathers data on player actions for a range of latency and game conditions for the fundamental action of moving target selection using a mouse. Analysis shows sensitivity to delays in all conditions, with particular sensitivity when targets are fast. We derive a simple analytic model from the data that is a promising step for a broadly applicable tool to better understand and compensate for delay in games and other interactive applications.

Author Keywords

delay; mouse; lag; game

ACM Classification Keywords

H.5.2 Information interfaces and presentation (e.g., HCI): User Interfaces:Input devices and strategies (e.g., mouse)

INTRODUCTION

Computer games increasingly run on a wide range of hardware and software platforms, from mobile phones and hand-held game devices to dedicated game consoles and high-end gaming computers. Moreover, gaming systems are increasingly networked, with players competing and collaborating with others connected by a network and, in some cases, with the game itself being played in the cloud and gameplay sent as video to the player.

This system heterogeneity adds variability in the processing time of user actions where some actions have an almost immediate response from the computer, while other actions are delayed or may not even have an apparent response. Even temporal delays in milliseconds can hamper the interplay between a users' actions and the intended results, particularly for time-sensitive actions. In particular, real-time games require players to make many time-sensitive actions that can suffer when computer responses lag behind player input. For example, lag when moving and clicking a mouse can make it difficult for a player to aim and hit a target in a shooting game, hurting the player's score and degrading the quality of experience.

While there exist many methods to compensate for delays, including system-level treatments (e.g., real-time priorities), network latency compensation algorithms (e.g., dead reckoning) and even game design with latency in mind (e.g., delayed attacks), an understanding of how latency affects fundamental player actions in games is critical in order to decide on the most appropriate and effective delay compensation techniques to use.

Previous research has explored the time to complete a motor-skill task based on difficulty, including robust models that incorporate the most relevant input parameters. In particular, Fitts' Law, an ergonomic model for the time it takes for a user to select a target of a given size and distance [13], has been applied to different tasks [18], many of them related to user interactions with a computer [23]. While the original Fitts' model has been extended to alternate environments and tasks, such as moving targets [21, 20, 8], two-dimensional spaces [29, 16] and even new user interfaces [24], such works have not incorporated the effects of delay. Work that has studied the effects of delay on remote manipulation tasks in relation to Fitts' Law [19] has not incorporated moving targets nor 2d computer pointing tasks (e.g., a computer mask), in addition to studying delays generally much larger than those found on modern computer systems.

There has been considerable work measuring the effects on delay and games [4, 9, 10, 3, 7], investigating the impact on player performance as well player quality of experience. However, much of the work fails to account for end-system delays, only considering network delay. Moreover, generally in-game performance is the focus, without necessarily understanding the impact delay has on the player action.

So, while foundational studies on user actions (e.g., Fitts' Law) have shown promise to better understand the effects of modern computer systems on games, such studies have not incorporated system delay and typical game interactions (e.g., selecting a moving target with a mouse). Conversely, game studies have incorporated system and network delays and focus on game genres (e.g., first person shooters) but fail to provide an understanding of the fundamental effects of delay on player inputs appropriate for games.

Our work incorporates approaches from psychophysical experiments, multimedia and gaming research, as well as Quality of Experience (QoE). We design and implement a simple game that allows for control of the delay between the player input and the resulting game action, as well as control of the game difficulty. In the game, called *Puck Hunt*,[1], players select a moving target with the mouse, where delay may be added to both the mouse movement and the selection and target speed changes between rounds. The game is used in large user study with over 30 participants, with added delays ranging from 0 to 400 milliseconds and target speeds ranging from 150 to 450 pixels/second.

Analysis of the results shows the time to select a moving target with the mouse increases polynomially with added delay – this is in contrast to earlier work that showed a linear relationship. The time to select the target does not vary with target speed for low delays, but there is a pronounced interaction effect between added delay and target speed for high delays – in other words, the time to select a fast target at high delay is larger than either parameter alone would suggest. User opinions on the quality of experience based on the responsiveness of the game shows a more pronounced decrease than performance, even for modest delay increases. We present a derived analytic model for the time to select the target based on the delay and the target speed that explains 95% of the variation. The model is polynomial with delay and includes a linear interaction term for delay and target speed.

The rest of this paper is organized as follows: Section 2 provides background research on work related to this paper, including work by cognitive researchers as well as game-specific systems research; Section 3 describes our methodology to study the effects of delay on moving target selection with a mouse, includes development of a custom game and a user study; Section 6 analyzes the user study data through graphs and an analytic model; and Section 8 summarizes our conclusions and presents possible future work.

[1]A pun on the classic game *Duck Hunt* (Nintendo, 1984).

BACKGROUND

Delay affects both player performance and quality of experience in games as well as other realtime interactive scenarios. The question of *how* can be approached from various angles.

At the most basal level, cognitive researchers and neuroscientists investigate the temporal separation between sensory and efferent signals. Some study temporal order or how much time must pass before two signals cease to be perceived as simultaneous [15, 28, 30]. Others focus on adaptation and how large a temporal window the human perceptual system can compensate for after continuous exposure to signal delay [12, 17, 31]. Because many of these studies use isolated and static stimuli, their results are difficult to generalise to applied scenarios such as games. However, in more recent work, some have shifted focus to more complex interactions. For instance, in a series of experiments, [28] explored participants' ability to track a moving target by drawing with a pen on a graphics tablet. Participants could not see their hands, but they could see a cursor on a screen that represented their drawing. With a 200 ms feedback delay, they found that participants performed better when they could predict the movement of the target.

In the field of computer games, focus is often on network latency.

Studies on how network latency affects players often follow a common pattern [1, 11, 5]. Researchers let participants play a game and introduce varying degrees of latency. Then they rate the players' performance in game, usually by some metric included in the game such as points. In addition it is common to rate the players' experience using a Quality of Experience metric.

These studies have built a comprehensive empirical list of how much latency affects games depending on the type of game as well as different actions within each game. However, they have not attempted to find or explain the mechanisms by which latency affects players. Thus, if a developer makes a new game, and wants to know how much latency players can tolerate, they will have to find the most similar game in the literature and use results from that. Ideally, we would like a theory that can predict sensitivity to delay based on easily obtainable metrics describing the game.

A related but separate field centres around the basics of interactive controls. They have build theories to describe the complexity of tasks. One such theory is *Fitts' law* [14], which predicts the difficulty of moving your finger to hit a target based on distance to target and width of target. This has since been expanded to two dimensions [29] and moving targets [20].

This work indicates a useful approach for games, though games face an even more complex problem. Many games allow movement in three dimensions for both player and targets. This could add complexity, but both controls and views are usually projected down to two dimensions which means 2D models might model the situation quite accurately. Further, input in computers does never produce output immediately. This input delay severely impacts both task performance and quality of experience for users [22], and is not covered in the previous modeling work.

As mentioned, in-game latency affects not only player performance, but also the play experience. For instance, in a research-oriented Unreal Tournament, 12 players played 20 different game scenarios with varying levels of jitter and/or delay [26]. Unsurprisingly, the network quality scores provided by players correlated with both the jittery and the delayed game scenarios. In fact, the results indicated negative consequences of delay already at 60 ms. Similar detriments to experienced quality have been reported for a study using the game Call of Duty [2] and for an experimental game focusing on the responsiveness of a third person avatar [25]. Acknowledging that a player's performance is only one aspect of the game experience, we also assess the QoE for the game implemented in this study. With this, we wish to explore whether the experienced game quality goes down along with performance or whether the QoE is affected at larger delay values.

METHODOLOGY

To investigate the effects of delay on player game actions, we deployed the following methodology:

1. Design and develop a game (*Puck Hunt*) that isolates player actions and controls amount of added delay (Section 4).
2. Conduct a user study to evaluate impact of delay on user actions, measuring user performance and quality of experience (QoE) (Section 5).
3. Analyze the results of the user study (Section 6).
4. Develop an analytic model for user performance and delay (Section 6.4).

PUCK HUNT

We designed a game, called *Puck Hunt*, that allows for study of a single player action in isolation and control over the amount of delay between the user input for the action and the action resolution. Puck Hunt is simple so as to give consistent results, while still providing an interaction that mimics more full-fledged games. The action chosen is one common to many games – selection of a 2-d, moving target with a mouse.

In Puck Hunt, depicted in Figure 1, the player proceeds through a series of short trials, where each trial has a large black ball, the puck, starting in a random location with a random velocity bouncing around the screen. The player moves the mouse cursor, represented by a small red ball, and attempts to "hit" the puck by moving the mouse over the puck and clicking the mouse button. Once the player has successfully hit the puck, the puck disappears and a notification pops up, telling the player to prepare for the next trial. Upon pressing any key, a new trial starts. The player is scored via a timer that counts up from zero at the beginning of each trial, stopping when the player has successfully hit the puck.

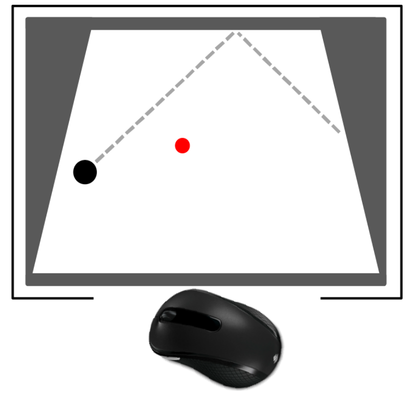

Figure 1. *Puck Hunt* – Player tries to click on moving target (the puck) with mouse.

Speed	
Slow	150 pixels/sec
Medium	300 pixels/sec
Fast	450 pixels/sec

Table 1. Puck speeds for user study.

Each trial, the puck starts with one of three possible speeds, shown in Table 1, selected randomly. Effectively, these speeds create different difficulties for the player. The game also adds a controlled amount of delay the amount selected from a set of 11 possible values, shown in Table 2. The set of delays is chosen so as to explore in detail delays up to 200 ms (common in many broadband networks and systems, while allowing some exposure to larger delays (common in some

Delay
0 milliseconds
25 milliseconds
50 milliseconds
75 milliseconds
100 milliseconds
125 milliseconds
150 milliseconds
175 milliseconds
200 milliseconds
300 milliseconds
400 milliseconds

Table 2. Delays for user study.

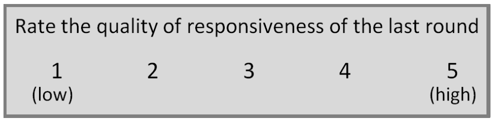

Figure 2. Quality of experience prompt to player.

wireless and wide-area networks). The delay is added to all mouse movements and mouse button clicks for the duration of the trial. Each delay + speed combination is only present a fixed number of times, controlled by a configuration parameter, but are shuffled so as to appear in a random order.

Every 30 trials, the game stops for a minimum of 20 seconds, with a countdown timer shown to the player via a popup window. This pause is to allow the player to rest in order to retain his/her concentration for the next set of trials.

Exactly once for each combination of delay and ball speed, the player is asked to rate the quality of the responsiveness of the round. The game pauses until the player selects a choice, 1–5. Figure 2 depicts the popup window.

Puck Hunt runs in fullscreen mode, at 1080p resolution (1920x1080 pixels). The puck is 100 pixels in diameter and the mouse cursor (the red ball) is 25 pixels in diameter.

To minimize the latency inherent in the software, Puck Hunt is written in C++ using OpenGL with support from the Angel 2D game engine.[2]

USER STUDY

The user study was conducted in a basement lab with bright, fluorescent lighting. A picture of the lab layout is in Figure 3. The lab is equipped with Dell PCs with Intel Core i7-4790 4.0 GHz processors, 4 GB GeForce GTX 960 graphics cards and 16 GB of RAM. The PCs run Windows 7. Monitors are 24" Dell U2412M with a native resolution of 1920x1200 pixels and a refresh rate of 59p Hz.

Potential users were solicited through WPI email aliases. Incentives included a raffle for a $25 Amazon gift card for participating and a $25 Amazon gift card for the player with the highest score. Students in an advanced game development class also received 1 extra point on the final exam for participating in the user study.

Users first were read a scripted brief about the study and signed an Institute Review Board (IRB) consent form at the researcher's position (see Figure 3). Next, users were seated at one of the computers in the lab, separate from any other user participating in the study. Users were then asked to make themselves comfortable, adjusting the chair height and monitor angle/tilt so as to be looking directly at the center of the screen. Users

[2]`http://angel2d.com/`

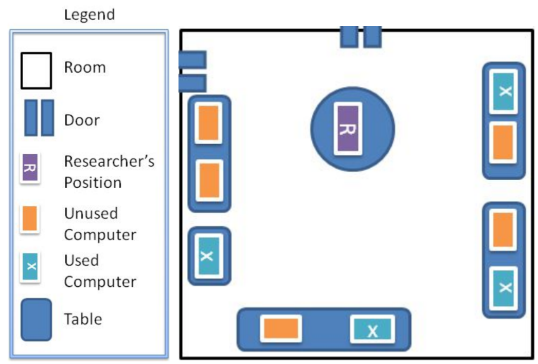

Figure 3. Lab for user study.

were encouraged to shift the mouse to whichever had was preferred.

Users logged into the computer using their WPI credentials and were asked to open a Web browser with a survey coded using the Qualtrics survey tool[3]. The survey asked questions about demographics and gaming experience.

After completion of the survey, the Puck Hunt game and incentive options were described followed by launching the game.

Play commenced immediately, but the first two trials of the game were used for "practice" and the results were not recorded. Play then proceeded through 5 iterations of all combinations of puck speed and delay, all ordered randomly, with one QoE question for each delay-speed combination and a forced pause every 30 trials (see the "Puck Hunt" section). In total, players played 165 recorded trials, which took about 15 minutes including answering questions and pausing.

The delays in Table 2 added by Puck Hunt are in addition to any delays inherent in the base computer system. Since such base delays have been shown to be significant [27], we measured the base delay for mouse actions on our test system. using a Blur-busters type technique to measure input lag.[4] A bread board with an led was connected via a wire soldered to a mouse so that the led lit up when the button was clicked. Using the mouse on the lab computers, a high speed camera (Casio EX-ZR200) filmed the player clicking on the QoE prompt, recording the action at 1000 f/s. By manually examining the individual video frames, the frame number when the light appeared with the button click is subtracted from the frame number from when the QoE prompt shows the input.

Figure 4 depicts the measurement method. The mouse is poised over the QoE prompt in frame 5175. In frame 5176, the button has been pressed indicated by the lit led on the breadboard. The input is not displayed on the QoE prompt until frame 5177. Since there is one

[3]`https://www.qualtrics.com/`

[4]`http://www.blurbusters.com/gsync/preview2/`

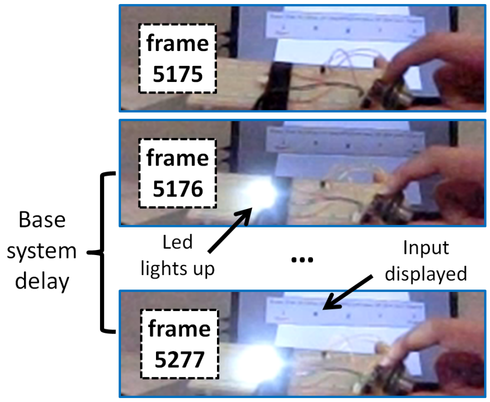

Figure 4. Measuring base delay in system.

video frame each millisecond, subtracting 5716 from 5277 gives a base delay of 101 milliseconds.

The measurement method was repeated 5 times, resulting in base delay values of 93, 99, 101, 101 and 112 milliseconds. Hence, 100 milliseconds is added to all subsequent data when analyzing the impact of delay.

RESULTS

This section presents demographics, objective results, subjective results, a model and discussion.

Demographics

Thirty-two subjects participated in the users study. Ages ranged from 18-26 years with a mean and median of 21. Twenty-three identified as male, 8 as male and 1 did not specify. Twenty-seven indicated they were right-handed, 4 left-handed and 1 ambidextrous, but all said they used a computer mouse right-handed. Twenty-one needed corrective lenses to see clearly and only 2 did not have them.

The mean self-rating as a computer gamer (scale 1–5) was a 3.5, showing a slight skew to having "high ability". Self-ratings for PC gaming and network gaming had similar trends as for computer gamer. Exactly half played 6+ hours of video games per week, about the same fraction that used a computer (PC/Mac) with a mouse 6+ hours per week. Most studied Computer Science, Game Development or Engineering.

Objective – Game Performance

Puck Hunt is designed to isolate the fundamental action of target selection with a mouse. As such, we assess Puck Hunt player performance, where the players' score is the time it takes to hit the puck – the lower the number the better.

Figure 5 depicts one graph of player performance. The x-axis is the input delay (added delay + system base delay) and the y-axis is the time to hit the puck, in milliseconds. There are three trend-lines, one for each puck speed tested. Each point is the mean hit time for all users for that speed and delay combination, shown

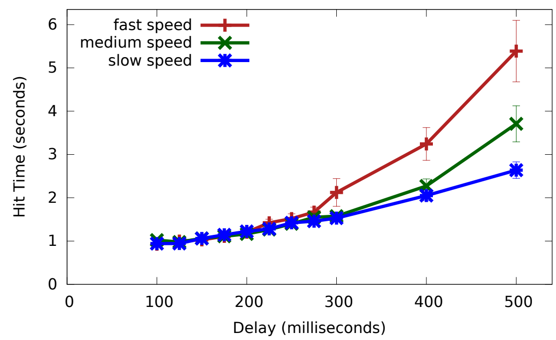

Figure 5. Player performance – Hit time versus delay, grouped by puck speed.

with a 95% confidence interval. Overall, there is an increase in hit time as the delay increases, left to right. This increase appears polynomial or exponential over the range of delays tested. For delays under 200 milliseconds, the speed of the puck does not matter in terms of average hit time. However, starting at delays of 225 milliseconds (for fast pucks) and 400 milliseconds (for medium pucks), the faster speed pucks become harder to hit than the slow speed pucks. At the extreme (500 milliseconds) delay, the fast pucks take 5x longer to hit than when there is minimal (100 milliseconds) delay and even the slow pucks take over 2.5x longer to hit.

Figure 6 depicts another graph of player performance using the same data but analyzing by speed. The x-axis is the speed in pixels per second and the y-axis is the time to hit the puck, in milliseconds. There are five trend-lines, one for total delays (added delay + system base delay) of 100 – 500 milliseconds.[5] Each point is the mean hit time for all users for that speed and delay combination, shown with a 95% confidence interval. Overall, there is an increase in hit time as the puck speed increases, left to right. This increase appears linear for the most part over the range of speeds tested. Delay impacts the hit time for all puck speeds, but is most pronounced for the highest puck speeds as seen by the diverging lines. As seen in alternate form in Figure 6, for delays of 200 milliseconds and under, the lines are flat – the speed of the puck does not impact average hit time.

A secondary measure of performance is the number of mouse clicks it takes for a player to hit the puck. The minimum number of mouse clicks is 1, so any mouse click greater than 1 means the player "missed" the puck, clicking the mouse button when the mouse was not over the target, possibly due to lag and/or the target speed. While the extra mouse clicks do not factor into the player's score, for many games, missing a target with a mouse click could matter (e.g., expending ammunition in a shooting game).

[5]The other delays tested are not shown to keep the graph readable.

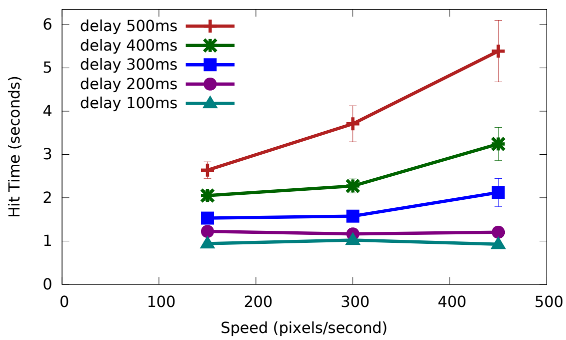

Figure 6. Player performance – Hit time versus speed, grouped by added delay.

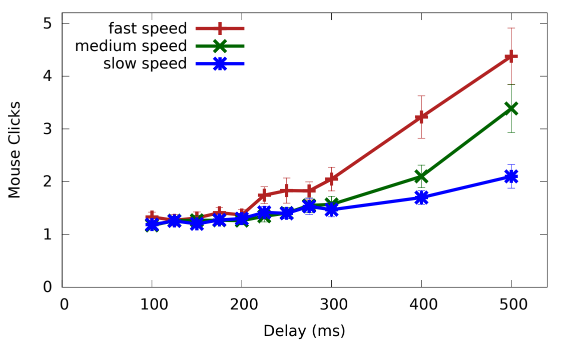

Figure 7. Player performance – Mouse clicks versus delay, grouped by puck speed.

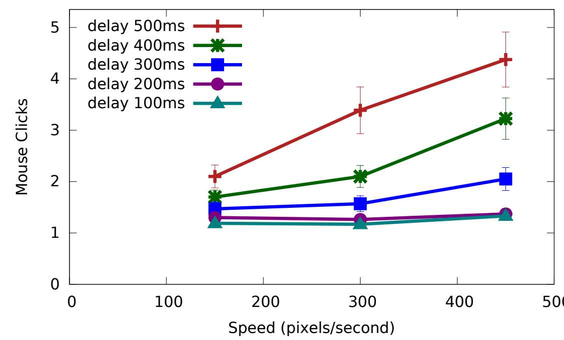

Figure 8. Player performance – Mouse clicks versus speed, grouped by added delay.

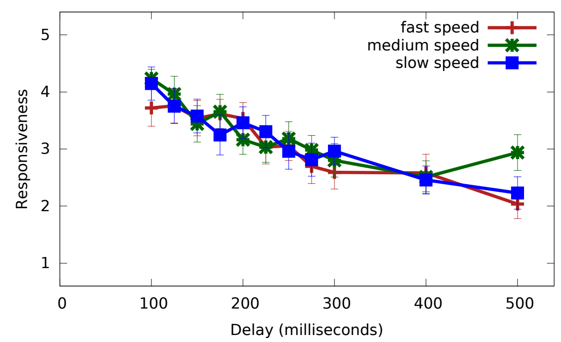

Figure 9. Quality of Experience – Responsiveness versus delay, grouped by puck speed.

Figure 7 depicts one graph of player performance based on mouse clicks. The x-axis is the total input delay and the y-axis is the number of mouse clicks needed to hit a puck. The three trend-lines are for each puck speed, with points being the mean mouse clicks for all users for the given speed and delay combinations, shown with 95% confidence intervals. Overall, the trends in Figure 7 look similar to those in Figure 5 with separation of the trend lines based on puck speed at about 200 milliseconds and 400 milliseconds for fast and medium pucks, respectively. Overall, the mean puck speed is slightly above 1, even for the slowest pucks and no delay, suggesting the game is challenging enough that players sometimes miss the puck. At the most extreme At the extreme (500 milliseconds) delay, the fast pucks are missed on average more than 3 times before being hit.

Figure 8 depicts player performance for mouse clicks analyzed by speed. The x-axis is the speed in pixels per second and the y-axis is the number of mouse clicks needed to hit a puck. The trend-lines are one for each total delay, 100 – 500 milliseconds, and each point is the mean clicks for all users, shown bracketed by a 95% confidence interval. Overall, the trends in Figure 8 look similar to those in Figure 6, with a linear increase in clicks as puck speeds increase. The slope of the lines are flat for the lowest delays – the puck speed does not impact the number of clicks – but increases as delay increases.

Subjective – Quality of Experience

While player's opinions of a game often correlate with their performance, subjective measures of the experience can ascertain the quality of the experience beyond just the game score. For Puck Hunt, for each latency and puck speed combination, players were asked to rate the responsiveness of the game (Figure 2).

Figure 9 depicts a graph of the responsiveness versus delay. The x-axis is the total input delay and the y-axis is the quality of experience – here, the responsiveness of the trial. There are three trend-lines, one for each puck speed tested. Each point is the mean rating for all users for that speed and delay combination, shown with a 95% confidence interval. From the graph, there is an observable downward trend in QoE with delay, indicating players perceive the added latencies. However, unlike for performance, there is no noticeable separation of QoE values with puck speed suggesting players are equally able to gauge responsiveness based on latency independently of the game action difficulty.

Model

While trends in player performance and opinions with delay provide valuable insights for game developers, more useful is a general relationship in the form of an analytic model. As a step towards such a model, we modeled mean player performance – time to select a moving target with a mouse – with delay.

Based on the previous analysis, there is a clear upward trend in mean hit time with delay, possibly a linear relationship but with a polynomial more likely to capture the observed curvature. The time trend with puck speed is less clear – there is little effect of puck speed on performance with low delay, however there is for high delays. Thus, there seem to be important interactions between speed and delay.

Thus, we propose modeling the time to select a target with a mouse (T) as a quadratic polynomial for delay only, with an interaction term for delay (D) and speed (S):

$$T = a + bD + cD^2 + dD \cdot S \quad (1)$$

where a, b, c and d are constants determined empirically through the user study experiments. Fitting this regression model in R yields an R^2 0.95, F-stat 118 and $p < 2.511e - 16$ with the simplified final model:

$$T = 1 - 0.005D + 0.00002D^2 + 0.000009D \cdot S \quad (2)$$

where T is the mean time to select a target in seconds, D is the total input delay in milliseconds, and S is the target speed in pixels/second.

DISCUSSION

The observable impact of delay on target selection with a mouse for fast targets suggests delays of 100 milliseconds added by network and game servers impacts players. Moreover, the decrease in QoE for all puck speeds as delay increases suggests players notice delays before it impacts their performance. This indicates every effort should be made to minimize local delays, and network and server delays, where appropriate, and compensating for delay [6], where possible.

While the results presented are for a specific game, Puck Hunt, the task of selecting a target with a mouse is common to many games. Most notably, the popular first person shooter (FPS) genre (e.g., *Call of Duty*, Activision, 2003) has target selection with the mouse as the primary method of aiming and shooting. Likewise, the newer multiplayer online battle arena (MOBA) genre (e.g., *League of Legends*, Riot Games, 2009) uses target selection for moving avatars and casting spells. The results may hold for player actions with other input devices, such as aiming with an analog joystick in a game controller, but additional studies should be done.

While the task of moving target selection is common to many games, the final model (Equation 2), as presented, likely holds primarily for the conditions tested. The size of the target is known to affect target selection time, most famously studied through Fitts' Law [13]. While target size was not varied in our experiments, combining Fitts' Law with our contributions in measuring delay may yield a unified general model. Such modeling should consider both the absolute target size in pixels and also the target size relative to the screen resolution.

The results presented are relevant to all forms of input delay, whether from the local system (e.g., operating system and hardware) or from the network. In particular, cloud games – where all player input is sent to the cloud for rendering – will have all mouse actions, both movement and clicking, delayed by the local system, network and server. However, traditional network games – where mouse movement is processed and rendered by the local client – will only suffer local delay for mouse movement but will suffer from additional delays for mouse clicking since the latter incurs network and server processing delays.

CONCLUSION

The variety of platforms, some distributed, used by computer games adds different amounts of delay to user input. Understanding the effects of delays on user inputs can help game developers and researchers develop designs and solutions to possibly reduce or at least mitigate the effects of delay on players. While some previous work has measured the effects of delay on games and other work has modeled user input for some fundamental tasks without delay, there has yet to be a thorough exploration of the effects of delay on fundamental player actions.

This paper presents work in progress towards a model for player action with delay. We present results of a large user study wherein users played a custom game that required target selection with a mouse with various amounts of delay, the game difficulty controlled by the target speed. Over 30 users participated in the study, providing data for delays ranging from 100 to 500 milliseconds across 3 target speeds – in total, over 5000 observations of player performance. In addition, players provided over 1000 subjective quality assessments for each of the different delay + speed combinations.

Analysis of the results shows a measurable increase in the time to select a moving target even for low amounts of delay and a sharp increase in selection time for higher delays and fast targets. While target speed is not a factor for low delays, subjective opinions show users are sensitive to even modest amounts of delay. An derived analytic model with provides a good fit for the average time to select a moving target, with quadratic terms for delay, no terms for the target speed, and an important interaction term that captures the effects of target speed combined with delay.

While promising, there are several key areas for continued work. The analysis can be continued, providing additional analytic models for mouse clicks and quality of experience. Task selection over a wider range of target speeds would help the results pertain to a broader set of games. More general models could incorporate target size and screen size (distance). Other forms of player input that involves target selection (e.g., analog controller, touch on mobile/tablet) or even keyboard or game controller button pressing could be explored.

REFERENCES

1. Rahul Amin, France Jackson, JuanE. Gilbert, Jim Martin, and Terry Shaw. 2013a. Assessing the Impact of Latency and Jitter on the Perceived Quality of Call of Duty Modern Warfare 2. In *Human-Computer Interaction. Users and Contexts of Use*, Masaaki Kurosu (Ed.). Lecture Notes in Computer Science, Vol. 8006. Springer Berlin Heidelberg, 97–106. DOI: http://dx.doi.org/10.1007/978-3-642-39265-8_11

2. R Amin, F Jackson, J Gilbert, J Martin, and T Shaw. 2013b. Assessing the impact of latency and jitter on the perceived quality of Call of Duty Modern Warfare 2. In *Human-Computer Interaction. Users and Contexts of Use: 15th HCI International, Proceedings, Part III*. 97–106.

3. Rahul Amin, France Jackson, Juan E. Gilbert, Jim Martin, and Terry Shaw. 2013c. Assessing the Impact of Latency and Jitter on the Perceived Quality of Call of Duty Modern Warfare 2. In *Proceedings of the International Conference on Human-Computer Interaction – Users and Contexts of Use*. Las Vegas, NV, USA, 97–106.

4. Grenville Armitage. 2003. An Experimental Estimation of Latency Sensitivity in Multiplayer Quake 3. In *Proceedings of the 11th IEEE International Conference on Networks (ICON)*. Sydney, Australia.

5. Tom Beigbeder, Rory Coughlan, Corey Lusher, John Plunkett, Emmanuel Agu, and Mark Claypool. 2004. The effects of loss and latency on user performance in unreal tournament 2003. In *Proceedings of 3rd Workshop on Network and system support for games (NetGames) (NetGames '04)*. ACM, New York, NY, USA, 144–151. DOI: http://dx.doi.org/10.1145/1016540.1016556

6. Yahn W. Bernier. 2001. Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization. In *Proceedings of the Game Developers Conference (GDC)*. San Francisco, CA, USA.

7. K. Chen, P. Haung, G. Wang, C. Huang, and C. Lee. 2006. On the Sensitivity of Online Game Playing Time to Network QoS. In *Proceedings of IEEE Infocom*. Barcelona, Spain.

8. T. Chiu, C. Lin, K. Young, C. Lin, S. Hsu, B. Yang, and Z. Huang. 2011. A Study Of Fitts' Law on Goal-Directed Aiming Task with Moving Targets. *Journal of Perceptual and Motor Skills* 113, 1 (Aug. 2011), 339–352.

9. Mark Claypool. 2005. The Effect of Latency on User Performance in Real-Time Strategy Games. *Elsevier Computer Networks, Special Issue on Networking Issues in Entertainment Computing* 49, 1 (Sept. 2005), 52–70.

10. Mark Claypool and Kajal Claypool. 2006. Latency and Player Actions in Online Games. *Commun. ACM* 49, 11 (Nov. 2006).

11. Mark Claypool and Kajal Claypool. 2010. Latency Can Kill : Precision and Deadline in Online Games. *Proceedings of the First ACM Multimedia Systems Conference* (2010).

12. D W Cunningham, A Chatziastros, M von der Heyde, and H H Bülthoff. 2001. Driving in the future: temporal visuomotor adaptation and generalization. *Journal of Vision* 1, 2 (Jan. 2001), 88–98. DOI:http://dx.doi.org/10:1167/1.2.3

13. Paul M. Fitts. 1954. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology* 47, 6 (June 1954), 381–391.

14. Paul M Fitts and James R Peterson. 1964. Information capacity of discrete motor responses. *Journal of experimental psychology* 67, 2 (1964), 103.

15. Waka Fujisaki and Shin'ya Nishida. 2009. Audio-tactile superiority over visuo-tactile and audio-visual combinations in the temporal resolution of synchrony perception. *Experimental Brain Research* 198, 2-3 (Sept. 2009), 245–259. DOI:http://dx.doi.org/10.1007/s00221-009-1870-x

16. Abir Al Hajri, Sidney Fels, Gregor Miller, and Michael Ilich. 2011. Moving Target Selection in 2D Graphical User Interfaces. In *Proceeding of the IFIP TC International Conference on Human-Computer Interaction (INTERACT)*. Lisbon, Portugal.

17. James Heron, James V M Hanson, and David Whitaker. 2009. Effect before Cause: Supramodal Recalibration of Sensorimotor Timing. *PLoS ONE* 4, 11 (2009), e7681. DOI: http://dx.doi.org/10.1371/journal.pone.0007681

18. Errol Hoffmann. 1991. A Comparison of Hand and Foot Movement Times", journal = "Ergonomics. 34, 4 (1991), 397–406.

19. Errol Hoffmann. 1992. Fitts' Law with Transmission Delay. *Ergonomics* 35, 1 (1992), 37–48.

20. ERROL R. HOFFMANN. 1991. Capture of moving targets: a modification of Fitts' Law. *Ergonomics* 34, 2 (1991), 211–220. DOI: http://dx.doi.org/10.1080/00140139108967307

21. R. Jagacinski, D. Repperger, S. Ward, and M. Moran. 1980. A Test of Fitts' Law with Moving Targets. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 22, 2 (April 1980), 225–233.

22. Michael Jarschel, Daniel Schlosser, Sven Scheuring, and Tobias Hoßfeld. 2011. An Evaluation of QoE in Cloud Gaming Based on Subjective Tests. *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* (jun 2011), 330–335. DOI: http://dx.doi.org/10.1109/IMIS.2011.92

23. I. S. MacKenzie, A. Sellen, and W. Buxton. 1991. A Comparison of Input Devices in Elemental Pointing and Dragging Tasks. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*. New Orleans, LA, USA, 161–166.

24. Michael J. McGuffin and Ravin Balakrishnan. 2005. Fitts' Law and Expanding Targets: Experimental Studies and Designs for User Interfaces. *ACM Transactions on Computer-Human Interaction* 12, 4 (Dec. 2005), 388–422.

25. Aline Normoyle, Gina Guerrero, and Sophie Jörg. 2014. Player perception of delays and jitter in character responsiveness. In *Proceedings of the ACM Symposium on Applied Perception*. Vancouver, 117–124.

26. Peter Quax, Patrick Monsieurs, Wim Lamotte, Danny De Vleeschauwer, and Natalie 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 3rd Workshop on Network and system support for games (NetGames) (NetGames '04)*. ACM, New York, NY, USA, 152–156. DOI: http://dx.doi.org/10.1145/1016540.1016557

27. Kjetil Raaen and Andreas Petlund. 2015. How Much Delay Is There Really in Current Games? *Proceedings of the 6th ACM Multimedia Systems Conference* (2015), 2–5.

28. Marieke Rohde, Loes C J van Dam, and Marc O Ernst. 2014. Predictability is necessary for closed-loop visual feedback delay adaptation. *Journal of vision* 14, 3 (2014), 1–23. DOI: http://dx.doi.org/10.1167/14.3.4

29. I Scott Mackenzie and William Buxton. 1992. EXTENDING FITTS' LAW TO TWO-DIMENSIONAL TASKS INTRODUCTION Since the advent of direct manipulation. *ACM CHI'92 Conference* (1992), 219–226. DOI: http://dx.doi.org/10.1145/142750.142794

30. Charles Spence, Roland Baddeley, Massimiliano Zampini, Robert James, and David I Shore. 2003. Multisensory temporal order judgments: When two locations are better than one. *Perception & Psychophysics* 65, 2 (Feb. 2003), 318–328.

31. Chess Stetson, Xu Cui, P Read Montague, and David M Eagleman. 2006. Motor-sensory recalibration leads to an illusory reversal of action and sensation. *Neuron* 51, 5 (Sept. 2006), 651–659. DOI:http://dx.doi.org/10.1016/j.neuron.2006.08.006