A Discussion on Time Optimization for Games in Mobile Computing

Simon T. Perrault
Yale-NUS College
Singapore
simon.perrault@yale-nus.edu.sg

ABSTRACT

Interactive systems are usually evaluated in terms of both accuracy and speed. This trade-off between speed and accuracy, in which the user tries to maximize both quantities, has been extensively studied in psychology [4] and HCI. Instead of such a max-max tradeoff, one can also see the problem as a min-min tradeoff between time and error. Time, e.g. reaction or execution time, is often seen as a quantity that should tend towards zero. However, the notion of "time" is quite broad and not every time measurement should be a quantity to nullify or minimize. In this paper, I will discuss different definitions of time and highlight a global time trade-off in which some quantities should be minimized and other maximized in order to improve the engagement time of users in a game.

CCS Concepts

•**Human-centered computing** → **Mobile computing;**

Keywords

Time, Reaction Time, Execution time, Trade-off, Mobile, Games, Wearable Computing

1. INTRODUCTION

As a dependent variable, time is often seen as a quantity that should tend towards zero. In controlled experiments, a good design/interaction technique is usually a solution that implies the shortest reaction and/or execution time. Psychology and HCI researchers also worked on trying to find models to predict the time needed for interaction. Two specific laws are currently used: one that predicts reaction time (Hicks-Hyman law) and the other one that predicts execution time (Fitts law). In many cases, time is used as a tie-breaker to choose between two alternatives after accuracy is taken into account: technique A will be deemed as better than technique B if it is more accurate than B, or in the case both A and B have comparable accuracy rates, if technique A is faster than B, where faster can be both in terms of reaction or execution time. However, the time difference is sometimes so small that one may wonder if that difference matters, which ultimately raises the following two questions:

1. What is an optimal value for a time quantity for an interactive system?
2. How to optimize a specific time quantity?

Through different examples and scenarios, I will now discuss several time quantities: reaction, execution, system (including latency, frame display time, global cooldown), engagement time, and time to notice. Each of them can have its own order of magnitude as well as potential optimal values in different scenarios.

MobiGames'16, June 26 2016, Singapore, Singapore

ISBN 978-1-4503-4325-1/16/06...$15.00

DOI: http://dx.doi.org/10.1145/2934646.2934647

2. REACTION TIME

When presented to a specific stimulus, a human subject will usually not perform any action for some time. This short interval is used by the brain to process the incoming information and decide which action to take and is called reaction time.

Reaction time has been extensively studied in both HCI and Psychology [6]. In a mobile context, where the user is usually interacting with the system while performing a primary task, e.g. walking, a short reaction time is especially crucial danger may occur at any time.

2.1 Reaction time on Head Mounted Displays

Head Mounted Displays are devices worn usually on the head which offer a small screen that may or may not be transparent. In a mobile context, a see-through display is valuable as it can provide additional information on top of what the user actually sees, e.g. showing directions to the desired location or displaying notifications and incoming messages. Google Glass are one of the latest state-of-the art optical see-through head-mounted display (OST-HMD).

A good OST-HMD should provide information to its user as fast as possible, while making sure that it does not cause too much distraction to the user. The screen of the Google Glass is by default located on the top-center field of view of the user (elevation of $+12.5°$) and offers a $16/9$ aspect ratio for a viewing experience similar to a 25" screen located 2.4 meters away from the viewer.

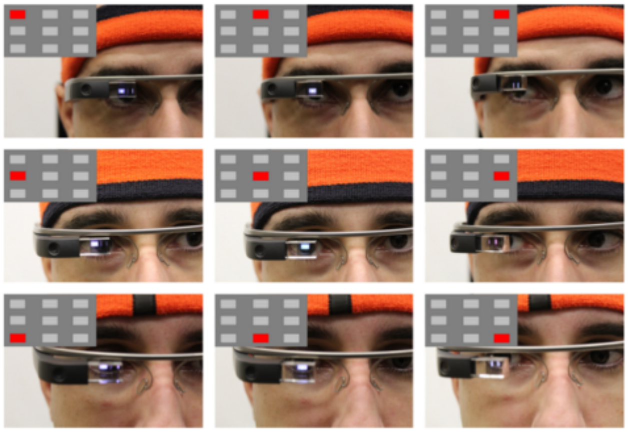

Figure 1: The nine display positions investigated in the study. The red rectangle shows the display position and size from users' point of view.

2.2 Display Positions

This default position forces the viewer to look up, which makes it difficult to use for extended periods of time [1]. We thus decided to investigate other possible locations for the display but manipulating both elevation (-12.5°, 0°, +12.5°) and azimuth (-17.28°, 0°, +17.28°) angles (see Figure 1). As a primary task, we used a simple driving simulator. While driving, participants would receive random stimuli of three types: *colors*, *icons* and *short text messages*. After noticing a stimulus, participants would press a button and answer a question about the stimulus (which color, what was the content of the text message). More details can be found in [2].

2.3 Study Results

The participants did overall recognize most of the stimulus correctly (accuracy of 97%) without any significant differences between the display positions. As seen in Table 1, we noticed however some significant differences in terms of reaction time, specifically between middle center on one hand, and top left, top right and top center. While the difference might seem minimal (1.32 seconds vs 1.6 seconds), it is just an average between different conditions. The time difference observed is much higher when it comes to text messages: the display position with the lowest reaction time is still middle center (1.63s), while the one with the highest reaction time is top left (2.29s), which represents a difference of 660 milliseconds.

2.4 Meaningful Reaction Time Difference

One may consider the average time difference observed as negligible (roughly 300 milliseconds). Let us put that difference into perspective. Bob, a glass user, is currently driving home on the highway at a speed of 130 kilometers per hour (36 meters per second). Bob is a bit tired at the end of the day and is barely paying a close attention to the car in front of him which happens to be breaking in emergency. Thankfully, its HMD displays a warning.

[1]http://www.adweek.com/socialtimes/wearing-google-glass-gets-pretty-tiring-pretty-fast-ux-designer-admits/132634

Table 1: Reaction time (in seconds) averaged over the types of stimuli for each display position. α, β and γ represent statistical significance ($p < .05$).

	Left	Center	Right
Top	1.64^{α}	1.56^{β}	1.55^{γ}
Middle	1.51	$1.32^{\alpha\beta\gamma}$	1.52
Bottom	1.54	1.41	1.55

In such a scenario, it usually takes 1 second [5] for a person to react. In this second, the car will move 36 meters forward. The 300 milliseconds difference observed between the best and worst display position may increase that reaction time by 30%, causing the car to move 12 additional meters forward, greatly increasing the chance of a collision.

We may also consider the case of somebody playing a Real Time Strategy game, such as Starcraft (1 or 2) or Warcraft III. In this kind of game, the player must input a large amount of commands (or action) and react to any incoming stimulus as fast as possible. The best Starcraft players are known to be able to input up to 400 actions per minute [7], or one action every 150 milliseconds. A loss of 300 milliseconds would potentially mean not retreating a wounded unit which could have huge consequences on the current game.

In both scenarios, we can see that even though the difference might seem minor. Thus, reaction time should be minimized to a value as close to zero as possible. It is of course not possible to have a reaction time equal to zero, but decreasing it even by a few hundreds of milliseconds is crucial both for wearable devices and/or game applications.

3. TIME TO NOTICE

In the previous section, I discussed reaction time, which is the time it takes a user to process an external stimulus after noticing it, and how it is important to minimize it even by a few dozens or hundreds of milliseconds. However, in some cases, it may take additional time for the user to notice the stimulus. This additional time is called here ”time to notice”.

3.1 Noticing Stimuli on a Smart Ring

To investigate the time to notice on a wearable device, we designed five rings. Each ring can provide a stimulus of a specific type (see Figure 2). The types of stimulus are: *vibration*, *sound*, *light*, *poke (through the use of a solenoid)* and *heat*. Because of hardware limitation, the results of the *heat ring* will not be discussed further. We chose the ring as a form factor as the hand is one of the most sensitive area on the body for haptic and thermal stimuli, as well as one of the most visible.

We asked 25 participants to wear these rings and perform five different levels of physical activity: *lying down*, *sitting*, *standing*, *walking (2.5 km/h)* and *running (7.5 km/h)*. We randomly sent stimuli to the participants in short 5 minutes session and asked them to press a button whenever they

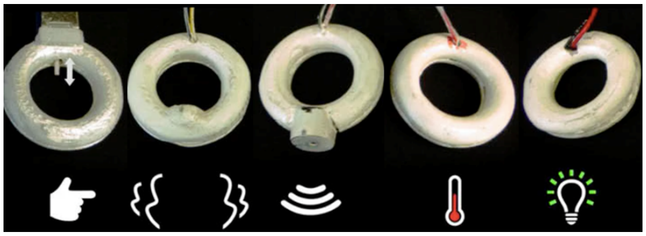

Figure 2: The five different rings used in the NotiRing study.

would notice something on the ring. Our participants had **20 seconds** to react to a stimulus. More details about the experiment can be found in this paper [8].

3.2 Study Results

Table 2 shows a summary of the results (physical activity levels were aggregated). Overall, the notice rate for each type of stimulus was near 100%, except for Light, which had a 95% notice rate only ($p < .01$).

Table 2: Average Time (including Time to Notice and Reaction Time) for each type of stimuli. α, β **and** γ **represent statistical significance (**$p < .05$**).**

	Light	Sound	Vibration	Poke
Time (s)	$2.87^{\alpha\beta\gamma}$	1.68^{α}	1.46^{β}	1.71^{γ}

More importantly, the results show a significant difference in terms of time to notice between light and every other condition. The difference ranges from 1.18 seconds (light vs. poke) to 1.42 seconds (light vs. vibration). We can also see that it takes roughly 100% more time to notice a blinking light than a vibration.

3.3 Discussion

We previously discussed about reaction time and how even a gain of 300 milliseconds can make a huge difference. However, that gain should never be done at the cost of increasing the time to notice an incoming stimulus: the time difference in that case is one order of magnitude higher (from 10^2 to 10^3 milliseconds). Thus, the type or modality used to convey information in an interactive system is a critical choice. As a general guideline, time to notice some incoming information should be decided and evaluated first before proceeding to reaction time optimization.

4. EXECUTION TIME

Whenever a notification or important event happens, e.g. a text message or even an enemy unit coming in a game, users will usually perform an action as a reaction to that information. The time to perform an action depends on the complexity of the action: pressing a button can be done within a few dozens of milliseconds, while performing a complex input sequence (see Figure 3) can take up to several hundreds of milliseconds.

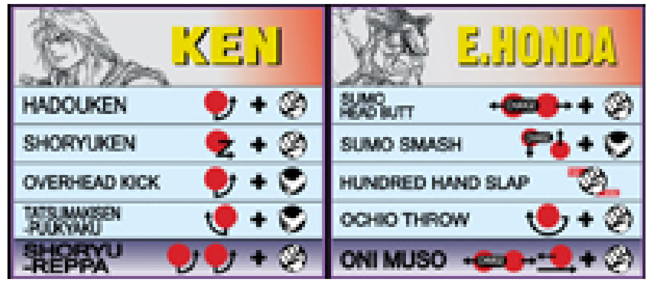

Figure 3: Details of input combination on Street Fighter II.

One could think that execution time should tend towards zero as much as possible. This may be true for any application but a ludic one: if inputting even a complex command is simple and fast, then users will tend to only use powerful and complex commands, making simpler one simply useless and potentially reducing the fun and interest of the game, leading a sharp decrease of engagement of users on the application.

This is why execution time has two optimal values: zero in the general case, or as low as possible while still taking into account the potential complexity difference between commands in a game.

5. REFRESH/SYSTEM TIME

Any interactive system needs to process a lot of information. In particular, games are extremely computation intensive applications as they require complex computation of 3D models to render smooth and high resolution graphics. Consoles and PCs are trying to offer a smooth experience for players by refreshing the image shown to the user either 30 or 60 times per second (30-60 frames per second, or 30-60 FPS). There has been a lot of debate lately in the gaming community, as well as in the scientific community about the number of FPS that a human eye can perceive. Since many games are played online, they may also require a connection to a server. The communication itself adds some latency to the system, as the communication between the client and the server may take a few tens of milliseconds to hundreds in a worst case scenario. Finally, the specific client-server architecture of many games (including Massively Multiplayer Online games) requires the developpers to prevent potential cheating, with a modified client sending erroneous information to the server.

5.1 FPS

"The human eye cannot perceive more than 30 FPS" (or similar) is a sentence commonly heard. The number of FPS for movies is for example between 25 (old PAL/SECAM standards) and 30 (NTSC standard). As such, the frame rate of many video games is usually around 30 FPS. Lower frame rate usually results in slow and jerky animation. Games on the newest generation of game consoles (PlayStation 4 and XBox one) usually offer frame rates of 30-60 FPS, with some editors arguing that 30 and 60 FPS look the same.

There is no clear answer on the actual number of FPS a human can perceive. It mostly depends on training. Specifically, some gamers would simply not buy a game if the frame rate is too low as they claim to see a clear difference. The optimal frame rate value greatly depends on the training of the person watching: US Air Force pilots are rumored to perceived up to 220 FPS, and some scientific works, such as Davis et al. [3] suggest that some people may perceive flickering artifacts at 500 FPS with 50-90 FPS being a comfortable frame rate. Andreev [1] also proposed a simple technique to convert 30 FPS animation to 60 FPS.

This would suggest that the current evolution in terms of FPS will go and games in the next few years should offer 60 FPS frame rates, and potentially 120 FPS or more. The refresh time of frame is thus definitely a quantity that should be minimized as much as possible ideally towards zero. The current order of magnitude for the display time of a frame is thus between 8 and 33 milliseconds, i.e. an order of magnitude of $10^1 ms$.

5.2 Latency and Global Cooldown

Video games have been multiplayer even since the 70s (Pong) and offered "local" multiplayer options ever since then, with multiple players staying in the same room. In the late 90s, the first games taking advantage of networks appeared: Doom (1993, see Figure 4), one of the most famous first-person shooter offered a multiplayer mode over a network (using the IPX protocol). The multiplayer mode was so popular that it tended to congest companies' networks. A few years later, the first Massively Multiplayer Online Games reached the market, e.g. Ultima Online one of the first MMORPG. Since earlier games were played locally, the latency was not a big issue, given the physical proximity and good bandwitdh of local networks at that time.

Figure 4: Doom was one of the first FPS game offering multiplayer using IPX.

With online games played on the Internet, latency became problematic: a lot of people only have limited dial up connections at home (bandwidth of 28.8 to 56 kbits/s), and players experiencing a latency of hundreds of milliseconds. Early online games also did not include control mechanisms on the server, relying on the clients to check whether some actions were possible or not. This allowed some players to modify their own clients and cheat. Many online games were thus plagued with bots which could play independently without have any human controlling them.

When Blizzard released their first MMORPG called World of Warcraft (WoW) in 2004-2005, they included many anti-bots mechanisms on their server. They also included the global cooldown mechanism. In WoW, a player can only input a command every 1 or 1.5 second. This has two main positive effects:

1. This gives the players more time to react between two inputs. As we saw, reaction times can be up to one or more seconds. This will also prevent users from spamming buttons as players are limited to either 40 to 60 Actions per Minute (APM). Novice players can thus get engaged with the game more rapidly as they do not need to train in order to reach very high APM values.

2. Because the number of inputs from player is limited, the required bandwidth for a smooth experience is quite low. Svoboda et al. [9] showed that a comfortable bandwidth is 3-4 kbps for uplink traffic and 40 kbps for downlink, which in turns potentially reduces the latency.

Ultimately, the WoW example shows that limiting the number of APM of a player in an online game can reduce the latency of the system and make the game more enjoyable both for novice players and more experienced users. Latency time is thus a time that should be minimized and usually scores in the 10s of milliseconds ($10^1 ms$). On the other hand, an optimal global cooldown value is yet to be determined. WoW uses a value of 1000 to 1500 milliseconds ($10^3 ms$).

6. ENGAGEMENT TIME

A successful application is an application on which users will have an enjoyable experience. This will lead them to:

1. Spend more time on the application.

2. Invite their friends to play with them.

3. In the case of a game, perform microtransactions via in-app purchases or other in-game shops.

4. Stay engaged for long periods of time.

Engagement time is thus the cumulative amount of time a user spent playing a game. This last point is crucial: many games which are time consuming may have their users play for hundreds of hours, weeks, months or even years (in cumulated in-game time). To illustrate that point, the author personally spent a total of more than 400 days in-game time in WoW, which is roughly $3.456 \times 10^{10} ms$. This quantity of time is definitely one that should be maximized.

7. CONCLUSION: EMERGENCE OF A GLOBAL TIME TRADE-OFF

Time can have multiple definitions and this paper discusses a few of them. In this paper, we discussed five of them: reaction, notice, execution, system and engagement. The main goal of a system or game designer is to ensure a long engagement time of its users. This quantity thus needs to be maximized. However, engagement also depends on the quality of the user's experience which itself depends on minimizing the four other time quantities we covered in that paper. For an HCI researcher, minimizing these quantities requires to know:

1. The usual order of magnitude of each quantity.
2. The order of magnitude of a potential optimization.
3. The optimal value for each quantity.

Table 3: Summary of each time quantity discussed. Orders of magnitude are in milliseconds (ms). Note: Optimal Value may depend on the type of software/game.

	Usual Order of Magnitude	Optimization Order of Mag.	Optimal Value
Reaction	10^3	10^2	0
Notice	10^3	10^3	0
Execution	Variable	Variable	Variable
Frame	10^1	10^0	0
Latency	10^1	10^0	0
GCD	10^3	**?**	**?**
Engagement	Up to 10^{10}	Up to 10^{10}	**Max**

Table 3 shows "variable" values for both orders and the optimal value of execution time. This specific case was discussed earlier and in some kind of games, e.g. fighting games, include both simple commands and more powerful and complex ones. Execution time must thus reflect that complexity and power difference, which is done by asking the user to perform more actions to perform actions. Offering a simple way to perform complex commands would lead players to only use them and in longer term would reduce the interest of the game as the challenge disappears (WoW's "I win" button).

To the best of my knowledge, while global cooldown (GCD) has been proven an efficient way to reduce latency and help novice users get used to a game, there does not seem to be many studies on which value is optimal for GCD. This would definitely be an interesting direction to consider for further research. Very likely, this specific value would depend on the kind of game as well as the expected level of skills an average user should reach.

The results presented in this paper also suggest that the time to notice a stimulus also offers a huge optimization potential: the optimization is within the same order of magnitude of its usual value (10^3 ms), which makes it one of the priority for game designers. Frame display time and latency should also be considered, however, they depend more largely on the hardware itself, on which programmers and designers cannot perform huge optimization. Newer technologies should be considered, as the potential limits of the human eye are yet to be reached for frame rates and latency peak may also happen in a mobile context.

From all the time quantities discussed in that paper, engagement time is the one that depends on all the other ones. It is also the only quantity that should be maximized, whereas other ones should be minimized (not always to zero though!). We can thus see the emergence of a max-min trade-off between engagement time on one hand, and reaction, notice, execution, frame, latency and GCD on the other hand.

8. REFERENCES

[1] D. Andreev. Real-time frame rate up-conversion for video games: Or how to get from 30 to 60 fps for "free". In *ACM SIGGRAPH 2010 Talks*, SIGGRAPH '10, pages 16:1–16:1, New York, NY, USA, 2010. ACM.

[2] S. H. Chua, S. T. Perrault, D. J. C. Matthies, and S. Zhao. Positioning glass: Investigating display positions of monocular optical see-through head-mounted display. In *Proceedings of the Fourth International Symposium of Chinese CHI*, Chinese CHI '16, pages 1–6, New York, NY, USA, 2016. ACM.

[3] J. Davis, Y.-H. Hsieh, and H.-C. Lee. Humans perceive flicker artifacts at 500 hz. *Scientific Reports*, 5:1–4, 2015.

[4] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, June 1954.

[5] M. Green. "how long does it take to stop?" methodological analysis of driver perception-brake times. *Transportation Human Factors*, 2(3):195–216, 2000.

[6] R. Hyman. Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45(3):188–196, March 1953.

[7] J. M. Lewis, P. Trinh, and D. Kirsh. A corpus analysis of strategy video game play in starcraft: Brood war. In *Proceedings of the 33rd annual conference of the cognitive science society*, pages 687–692, 2011.

[8] T. Roumen, S. T. Perrault, and S. Zhao. Notiring: A comparative study of notification channels for wearable interactive rings. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2497–2500, New York, NY, USA, 2015. ACM.

[9] P. Svoboda, W. Karner, and M. Rupp. Traffic analysis and modeling for world of warcraft. In *2007 IEEE International Conference on Communications*, pages 1612–1617, June 2007.