Waiting to Play – Measuring Game Load Times and their Effects on Player Quality of Experience

Shengmei Liu, Eren Eroglu, Miles Gregg, Federico Galbiati
{sliu7,ezeroglu,mgregg,fgalbiati}@cs.wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

Atsuo Kuwahara, James Scovell
{atsuo.kuwahara,james.j.scovell}@intel.com
Intel Corporation
Hillsboro, OR, USA

Mark Claypool
claypool@cs.wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

ABSTRACT

Before playing, gamers must wait for the game to load. While the effects of waiting on user quality of experience is well-studied for some domains, the effects of wait times on game players is not known, nor is the impact of computer system components, such as the processor or graphics card, on game loading times. We present results from a user study that evaluates the impact of game loading time on quality of experience using a custom tool that simulates game loading and collects player ratings. Analysis of the results shows game loading time has a pronounced effect on player quality of experience, but differs based on the individual game time and game load content. Results from our subsequent measurement experiment show the potential to reduce game load times through hardware upgrades – type of processor and graphics card have significant effects on game load times, but type of storage device less so.

CCS CONCEPTS

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

KEYWORDS

gamer, games, user study, QoE, hardware, measurement

ACM Reference Format:
Shengmei Liu, Eren Eroglu, Miles Gregg, Federico Galbiati, Atsuo Kuwahara, James Scovell, and Mark Claypool. 2024. Waiting to Play – Measuring Game Load Times and their Effects on Player Quality of Experience. In *Proceedings of the 19th International Conference on the Foundations of Digital Games (FDG 2024), May 21–24, 2024, Worcester, MA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3649921.3649937

1 INTRODUCTION

Computer systems often have moments where users must wait for the application to gather data before users can interact with it. Common examples include waiting for a file to download, a Web page to load or a video to finish buffering before the user experience can commence. Early studies of computer applications converged on a commonly agreed "acceptable" waiting threshold of about ten seconds, beyond which users find waiting reduces the quality of the experience [2, 14]. Longer waiting times can be sources of boredom, at best, and anxiety and frustration at worst. While prior research has demonstrated the relationship between user waiting time and decreased user satisfaction [3, 11, 13], the severity of this degradation depends upon the application [8] and type of feedbackmobile-load-23, swipes-loading-23.

Computer game players often face considerable wait times since the game needs to load before play can commence. During game loading, the computer system reads and converts data needed by the game engine and initializes various sub-systems, such as physics and graphics, needed to run the game. For the player, loading a game happens during two phases, depicted in Figure 1. The player must first wait while the game is launched (labeled "Game Launch"), much as is done for any application started on a computer. Once the game has been launched, the player interacts with a game menu to select the version of the game to play (e.g., choice of game mode and character selection). Once the player has selected all game settings, the player must again wait while the game level loads (labeled "Level Load"). Only once the game level has loaded can play commence.

To mitigate the negative effects of waiting on users, applications can provide feedback on progress, widely accepted as a means to improve a user's waiting experience [5, 14, 17]. Feedback during game loading can take many forms such as progress bars, text messages, and animated graphics and videos. While it is generally known that waiting decreases a users experience, and feedback can lessen the impact of waiting, the extent to which feedback, and other content that is shown during game loading for commercial games (e.g., splash screens), impacts player quality of experience (QoE) is not well-known. In addition, while decreasing the waiting time for some Internet-based applications can be done by increasing network bitrates [7], much of what a computer game loads is not related to the network capacity. It is unknown how standard hardware upgrades, such as improved processor, storage device or graphics card, can decrease game load times.

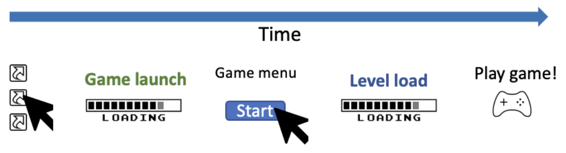

Figure 1: The player's perspective of game loading

This paper seeks to better understand game load times and QoE via a user study that assesses the QoE for users waiting for a game to load. We selected 12 popular games and encoded their loading screens as videos, re-encoding them to different lengths. We then recruited 54 users that had experience playing at least 3 of the games and had each participant use our custom application to simulate loading each game and provide an assessment of their QoE for each. Analysis of the results shows a marked decrease in QoE with increased game load times with results that vary from game to game. Models based on game loading time, possibly supplemented with visual content, have the potential to accurately predict QoE for loading games in general without needing to measure the QoE of loading individual games. Based on these results, we measured game load times on different computer systems in order to better understand and inform which hardware components may have the most impact on game load times. Analysis of the measurements shows that processor and graphics card have a significant effect on game load times, while storage device type less so. With the results, players and computer system developers may see benefits to player QoE from decreasing game load times and better decide how to upgrade the systems.

The rest of the paper is organized as follows: Section 2 provides background on game launching and level loading; Section 3 describes work related to this paper; Sections 4 and 5 detail our methods and results for our user study and hardware measurement experiments, respectively; Section 6 discusses implications of our findings; Section 7 mentions some limitations of our work and suggests possible future work; and Section 8 summarizes our conclusions.

2 GAME LOADING BACKGROUND

During game loading, the game system must transfer and convert game data from long-term storage, such as a hard disk drive or a solid state drive, to memory so it can be used by the game software. The largest volume of game data is typically from art assets, such as models, animations, and textures, but also includes sound effects and music. These assets usually need to be converted from their storage format, typically compressed to save space, to an in-memory format usable by the game engine. Other aspects of game loading include bringing in and parsing game data such as virtual world maps and game object attributes, and code and data for game systems and interfaces to control them. Processing during loading also includes pre-calculations and pre-rendering for visuals, and initialization of the many different sub-systems that run the game, such as the graphics, physics and AI engines, as well as networking and logging. Garbage collection may also run in order to free up code and assets that are used during initialization but can subsequently be discarded.

For the player, the above process of loading a game happens during two phases, depicted in Figure 1. The player must first wait while the game is launched (labeled “Game Launch”), much as is done for any application started on a computer. Once the game has been launched, the player enters a home screen which has an interface to select the version of the game to play (e.g., game mode and character selection). Once the player has configured all game settings, the player must again wait while the game level loads (labeled “Level Load”). Only once the level had loaded can play commence.

So, the player must wait during two phases of game loading: game launch and level load. Game launch happens only once, when the game is first started and does not need to happen again until the player exits the application and restarts at a later time. Level load, however, needs to happen each time the player starts or re-starts a game level which can be as frequently as once every few minutes for some games or as infrequently as once every few hours, for others.

During both phases of game loading – game launch and level load – the game often shows visuals, such as splash screens with company logos from the game studios, publishers or systems, or images of game characters or game scenes, or even cinematic renditions of gameplay, sometimes accompanied by sound effects or music. Some games, albeit far fewer, even provide a lightly-interactive interface for waiters. In some cases, the cinematics and animations can be skipped by the player with the press of a button or key. Some load screens show a progress indicator, such as a loading bar or animated loading icon, providing feedback to the player that the game system is getting the game ready to play.

3 RELATED WORK

This section describes related work in three areas: user quality of experience while waiting (3.1), feedback to mitigate waiting cost (3.2), and cognitive load and waiting (3.3).

3.1 Wait Time and Quality of Experience

Egger et al. [7] discussed the psycho-physics basis for a model of quality of experience and human time perception (e.g., delay when buffering a video during initial playout). The authors described a set of studies that lay out the basis for a logarithmic relationship for waiting time and user satisfaction ratings. They found this relationship held for several tasks, including file downloading – i.e., that user satisfaction with file downloads decreased logarithmicly with download time.

Hoßfeld et al. [8] quantified the impact of wait time on user quality of experience for different application scenarios by means of subjective laboratory and crowdsourcing studies. They found user QoE for the waiting time depended on the application, with users waiting for a video to start playing being more tolerant of waiting than users logging into a social network.

Allard et al. [1] studied the tradeoffs between initially waiting for a streaming video to start playing and interrupts (waiting) in the middle of a video playout. They showed users’ annoyance with waiting increased logarithmically with the time it takes a video to start playing, with annoyance greater for interrupts.

Ip and Jacobs [9] carried out a user study in which focused on various game characteristics of rally games. They found that games rated as “good” had typical load times of around 15 seconds, whereas games rated as “poor” had typical load times of around 40 seconds.

Clincy and Wilgor [6] conducted a study where users played a first-person shooter game on the OnLive cloud-based game streaming service. After play, the users subjectively rated their experience into 8 categories, one of which was level load time. Users rated the level load times they experienced an average of 3.5 (on a 1-5 scale), without measuring the actual load times nor assessing how different load times impacted QoE.

3.2 Feedback While Waiting

Branaghan and Sanchez [3] found that users waiting for a simulated movie to start had the most satisfaction with duration indicators shown via a progress bar. Nah [13] examined how feedback while waiting improved user satisfaction and showed the presence of display feedback increased how long a user is willing to wait. Lallemand and Gronier [11] confirmed these results and showed users more satisfied with a waiting interface that provided more information. However, too much feedback with details on progress can make a wait seem longer as the user interprets every event as taking time, and more events feels like more waiting time [3, 11].

Kim et al. [10] assessed the impact of different types of feedback provided while users waited for a video to load, considering progress and duration indicators and shape and embellishment distractors. Their results showed that duration and progress indicators affected the viewers waiting time perception, but shape and embellishment distractors did not.

Nimwegen and Rijn [15] compared differences in activities while waiting, divided into ‘no activity’ (e.g., a progress bar), ‘passive waiting’ (e.g., reading) and ‘active waiting’ (e.g., doing something). A user study with a mobile website (emulating purchasing a train ticket) and the 3 conditions assessed perceived time and enjoyment while waiting. The ‘no activity’ and ‘active waiting’ conditions were thought to be faster than the ‘passive waiting’ condition, while the ‘passive’ and ‘active’ waiting conditions were more enjoyable than the ‘no activity’ condition.

Cheng et al. [5] examined the effects of interactive loading screens on the waiting experience compared to a passive loading screen on mobile phones (the commonly-used ‘rotating ring’ was the baseline). Participants compared the baseline passive animation with grayscale and color-changed interactive animations, answering questions about experience and time perception. The results indicated waiting time had a significant impact on the loading screen preference and perceived time was shorter when an interactive loading screen was used.

3.3 Cognitive Load While Waiting

Lallemand and Gronier [11] used cognitive models of time perception, varying the cognitive workload and informational feedback, to study impact on satisfaction and perceived waiting time for users. They found a link between cognitive workload and waiting time perception. Users judged shorter waiting times more positively with a decrease in satisfaction that was linear with time.

Ledbetter [12] assessed perceived waiting times for an amusement park ride. He tested users playing various interactive math games while waiting for a virtual roller coaster, measuring their perception of the time spent waiting and their experience. His findings suggested adding an simple gameplay would shorten how long the wait time felt to the users.

Cheng et al. [4] studied player experiences with a mobile game that had an interactive loading screen. Twenty-three users participated in a two-year study using a UX Curve and QUIS measurement tools to assess player experience with an interactive loading screen, updated periodically by the publishers. Users found the interactive loading screen achieved a good waiting experience, and that changes to interactive loading screen were needed to keep the experience fresh.

3.4 Summary

The references in this section are to works related to our own in that they pertain to users that are ready to go, but must wait until the system is ready before they can play. While previous studies include browsing the Web, watching videos and waiting for rides, with the exception of Ip and Jacobs [9] that study only rally games and only level loading and Cheng et al. [4] that have only one waiting-time interface, our work is the first we know of that compares waiting for games to load across games – the task (interactive play) and the game load screens (visuals) are quite different than those for other events. Feedback while waiting has been demonstrated to be effective for mitigating wait times and we also consider visual style and progress elements that games provide as a step towards understanding users’ experiences while waiting for games to load.

4 QUALITY OF EXPERIENCE

In order to assess the effects of game load time on player quality of experience, we selected games to cover a range of genres, designed and implemented an application to simulate game loading, recruited participants for a user study, had each participant load games and rate their quality of experience, and analyzed the results.

4.1 Methodology

We selected twelve (12) games based on their popularity to make it more likely participants in our sample pool would have some familiarity with them, while also covering a range of game types and loading screens. Table 1 shows the games in alphabetic order along with publisher, year published and genre. The twelve games cover six (6) genres: four first-person shooter games, two turn-based strategy games, two battle royale games, two role-playing games, one multiplayer online battle arena (MOBA) game and one sandbox game. For each game, we record a video of the game launch and the level load on the same PC in order to provide for a single point of reference. The reference PC is an Alienware with a Samsung NVMe SSD, an Intel i7-6700 CPU @3.4 GHz with 16 GB RAM and an NVIDIA GeForce GTX 1070 graphics card.

In order to assess how changes to game loading times might impact QoE (e.g., if the user purchased a faster computer system or if the game developer reduced the game load time), we scaled the lengths of the videos for each game: 0.25, 0.5, 0.75, 1.0 and 1.25. So, for example, a 10 second game load video scaled to 0.25 would run to completion in 2.5 seconds. Note, our scales deliberately concentrate on decreasing load times from our reference PC since hardware trends tend to improve performance over time, and users and system developers are presumably interested in how upgrades may impact QoE. When scaling, we took care to make time reductions only for non-animated portions of the videos in order to minimally distort the visuals.

In order to mimic the game loading experience, we developed a stand-alone application that had users launch the game (e.g.,

Table 1: Games studied

	Game	Publisher	Year	Genre
AL	Apex Legends	EA	2019	First-Person Shooter
CV	Civilization VI	2K	2016	Turn-based Strategy
CS	CS:GO	Valve	2012	First-Person Shooter
FN	Fortnite	Epic	2017	Battle Royale
G5	Grand Theft V	Rockstar	2013	Role-Playing Game
HS	Hearthstone	Blizzard	2014	Turn-based Strategy
LL	League Legends	Riot	2009	MOBA
MC	Minecraft	Mojang	2011	Sandbox
OW	Overwatch 1	Blizzard	2016	First-Person Shooter
PG	PUBG	Microsoft	2017	Battle Royale
R6	Rainbow 6 Siege	Ubisoft	2015	First-Person Shooter

double-click on the game icon on the desktop) and load the level (e.g., select "start" and pick the intended map to play) as if they were about to play the game. Participants were instructed to do just that – to launch the game and select the game level as if they were about to play. For each case – game launch and level load – the application plays the pre-recorded video, pausing when input is needed by the user to proceed. When this happens, the application highlights where the user must click (e.g., the "start" button). When the video finishes playing, the application pops up a survey to assess the player's quality of experience via two questions: a) a Mean Opinion Score (MOS) "Please rate your experience" with a text box for 1.0 to 5.0 point numeric entry, shown along with scale: Excellent, Good, Fair, Poor, Bad; and b) a yes/no question "Is the experience acceptable?"

Since a user's opinion of a game load's time may depend upon their familiarity with the game, users were invited to participate in the study based on how much and how recently they played each game in Table 1. Invited users were familiar with at least 3 of the games, chosen so as to provide an approximately equal number of users for all games. The three games each individual was most familiar with was used in their test session. After testing, the number of users for each game was: Assassin's Creed (15), Apex Legends (15), Civilization VI (18), Counter-strike: Global Offensive (19), Fortnite (18), Grand Theft Auto V (18), Hearthstone (16), League of Legends (17), Minecraft (19), Overwatch (17), Player Unknown's Battlegrounds (16), and Rainbow 6 Siege (17).

For each participant, the application randomly shuffled the order of the games loaded. For each game, users first assessed all the game launches and then assessed all the level loads (again, told to prepare as if they were going to play the game), with the scale lengths randomly shuffled. Thus, we had a between-subjects design, where each user loaded 3 (of the possible 12) games, each with two conditions (game launch and level load) and 5 lengths for each condition ($3 \times 2 \times 5$) for a total of 30 game loads.

The user study was conducted in a dedicated, on-campus computer lab. Our test computer was a Windows 10 Alienware with an Intel i7-4790K CPU @4 GHz with 16 GB RAM and an Intel HD 4600 graphics card. While some users may have more powerful PCs to play the game, our game load simulation application is lightweight, needing only processing akin to that needed for a streaming video player. However, in order to provide for fast input and display, the PC was equipped with a gaming mouse and high refresh rate monitor: a 25" Lenovo Legion monitor, 1920x1080 16:9 pixels @240 Hz with AMD FreeSync and a 1 ms response time; and b) a Logitech G502 mouse, 12k DPI with a 1000 Hz polling rate.

After completing all the game rounds, users were given an additional questionnaire with demographics questions about overall gamer experience – average time spent playing games and self-rated expertise with computer games.

In summary, the procedure each user followed was:

(1) Submit screener to ensure familiarity with the games.
(2) For invited participants, arrive at the dedicated lab at a scheduled time and sign the consent form.
(3) Adjust the computer chair and monitor so as to be comfortably looking at the center of the screen.
(4) Read the instructions regarding the application and controls.
(5) Launch the game and, when done, fill out the corresponding QoE survey. Repeat for each scaled length (shuffled).
(6) Load the level and, when done, fill out the corresponding QoE survey. Repeat for each scaled length (shuffled).
(7) Repeat the previous 2 steps for each of the 3 games (shuffled).
(8) Complete a final demographics questionnaire.

The study length depended upon the games tested, but was under 30 minutes total in nearly all cases. A user study proctor was available for questions and trouble-shooting for the duration.

The study was approved by our university's Institute Review Board (IRB). Study participants were solicited via university email lists. All users received a $10 USD Amazon gift card upon completion of the study, and many users received academic credit for relevant classes in which they were enrolled.

4.2 Analysis

This section first summarizes participant demographics (Section 4.2.1) then presents the core results – QoE versus game loading time (Section 4.2.2). It then describes derived models of QoE based on game loading time and loading visual content (Section 4.3).

Table 2: Participant demographics

Users	Age (yrs)	Gender	Gaming per Week (hours)	Gamer Self-rating
54	19.3 (1.5)	40 ♂ 13 ♀ 1 other	10.4 (8.3)	3.4 (1.1)

4.2.1 Demographics. Table 2 summarizes the demographic information for the user study participants. Gamer self-rating is on a five-point scale, 1-low to 5-high. For age and gamer self-rating, the values are means with standard deviations in parentheses. Our user study had 54 participants, ranging from 17-24 years. Gender breakdown is predominantly male (40 males out of 54 users), which reflects the sample pool of students at our university. Half of the participants played 10 or more hours of computer games per week. User self-rating experience as a gamer skews slightly above the mid-point (mean 3.4). Most participants majored in Robotics Engineering, Computer Science, or Game Development.

4.2.2 *Quality of Experience.*
QoE was assessed from user responses to the MOS question filled out at the end of each round. Responses are on a 5 point scale, from 1-low to 5-high.

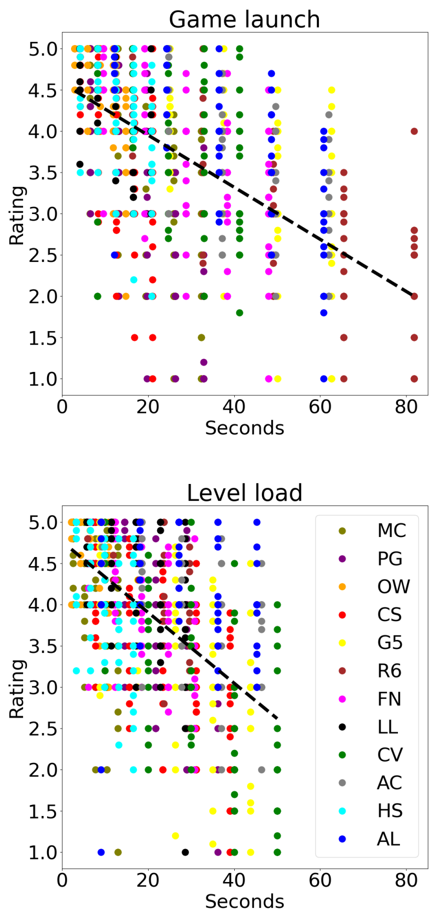

Figure 2: QoE versus game load time – All

Actual game load time. Figure 2 depicts scatter plots of QoE ratings versus game load times for game launch and level load. The x-axes are times in seconds and the y-axes are QoE ratings on a 5-point scale (1-low to 5-high). Each dot is the QoE value for one user rating one game load. The black dashed lines are linear regression trendlines through all data in each graph. The Pearson's correlation (R) between rating and time is -0.56 for game launch and -0.6 for level load, indicating a medium strength of downward trend. As a take away, on average, player experience degrades 1.9 points and 2.6 points (on a 5-point scale) for each minute for game launching and level loading, respectively.

Scaled game load time. Since each game load video was scaled relative to its initial length, we analyze how QoE changes with relative game load times – in other words, how much does decreasing (or increasing) the game load time improve (or degrade) QoE. Figure 3 depicts the results. The y-axis is the QoE rating and the x-axis is the scale relative to the initial game load time – i.e., 100% is the original time, with values below this being faster and values above this being slower. For each game load type (launch and level), there are 5 scales ranging from 25% to 125%. The green squares (game launch) and blue circles (level load) are mean values for all users across all games, shown with 95% confidence intervals. The lines are linear regression fits through the mean values. The linear regressions fit the data well for both game launch and level load with R^2 near 1. As a take away, a 50% decrease in game loading time improves player experience by about 0.75 on a 5-point scale, with a slightly larger impact on game launch than on level load.

Threshold and QoE. After each game load, in addition to a QoE rating, users were asked if the experience was acceptable. Figure 4 depicts the relationship between acceptable and QoE. The x-axis is the QoE rating collected by the MOS question, and the y-axis is the fraction that unacceptable was answered (i.e., "no"). Each dot is the unacceptable fraction for all corresponding QoE values grouped by 0.5 point bins. The green squares are for game launch and the blue circles are for level load. In general, QoE values less than 2 are nearly always unacceptable and QoE values above 3.5 are always acceptable and in-between there is a steep change from unacceptable to acceptable. The "above 3.5" threshold can be used in conjunction with QoE models in order to estimate how system or software improvements that result in reduced game load times are pertinent to the user. From the regression models shown in Figure 2, game launch times should be kept to under 27 seconds and level load times under 33 seconds in order to make the wait time acceptable (average QoE 3.5+).

4.3 Modeling

This section presents different approaches to modeling QoE for game loading. Such models can be helpful for game and system developers to predict how improvements to game loading time might benefit player experience.

4.3.1 *Individual models.*
While the relationship between QoE and game load time can be coarsely modeled with the regression depicted in Figure 2, the individual games themselves may not adhere closely to this relationship.

We model QoE for game launch and level load individually for each game. Figure 5 depicts per-game models, with axes as in Figure 2. The circles are mean values for all users across the 5 different time scales, and the lines are linear regressions through the mean values. Each color represents data from one game. The regressions fit the individual games well, with adjusted R^2 from 0.88 to 1.00 (mean 0.94, SD 0.03) for game launch and adjusted R^2 from 0.87 to 1.00 (mean 0.90, SD 0.20) for level load. The exception is for the Apex Legends level load with an adjusted R^2 of 0.28.

These individual models represent a sort of "best case" in that the model may accurately reflect QoE for that specific game, but may not generalize to other games. Put it another way, use of a per-game model means all games would need to go through a process akin to ours – a user study – before the specific effects of game load for that game could be determined.

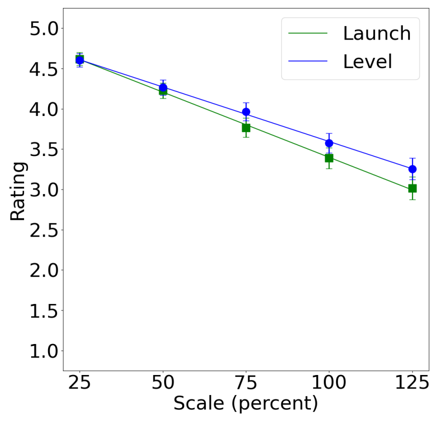

Figure 3: QoE versus game load scale

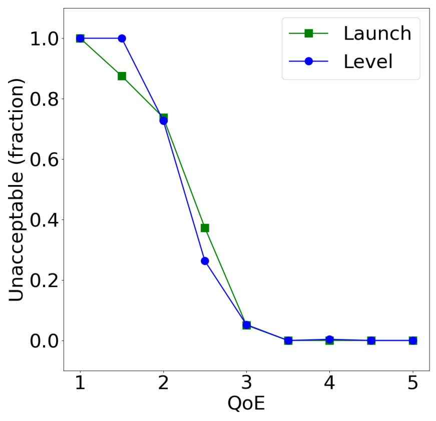

Figure 4: Unacceptability and quality of experience

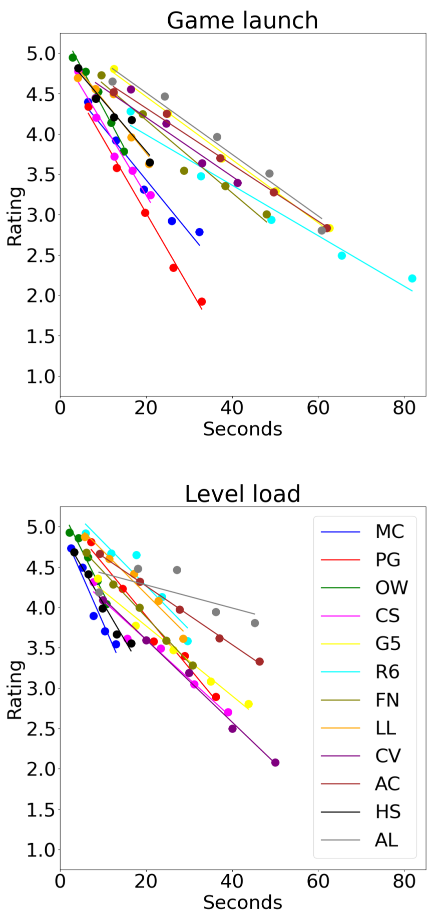

Figure 5: QoE versus game load time – All

4.3.2 Unified models. In order to generalize QoE models to games that have not been tested (and may not have even been invented yet), we look for alternatives to the individual game-specific models.

We use the length of the original game load video (unscaled) captured on our reference system PC as a model parameter to derive a "unified model" for the QoE for game launch (Q_g) and level load (Q_l) that can be applied to all games:

$$Q_g = 4.90 + 0.07 \cdot b \cdot l - 0.07 \cdot l \quad (1)$$
$$Q_l = 4.85 + 0.06 \cdot b \cdot l - 0.06 \cdot l \quad (2)$$

where b is the base encoding time in seconds and l is the actual encoded time in seconds. The unified model fit for game launch has an adjusted R^2 of 0.70 and for level load has an adjusted R^2 of 0.66, an improvement over a regression through all the data points (see Figure 2), but not as accurate as the per-game models.

Since the differences in QoE across games does not appear to be only based on game load times, we use visual content as parameters in the model. The ITU recommended [16] visual measures of spatial perceptual information (SI) and temporal perceptual information (TI) are used as model inputs – for each video, we compute SI and TI for each frame, then average them for the entire game load video:

$$Q_g = 4.59 + 0.11 \cdot b \cdot l - 0.08 \cdot l + 0.07 \cdot b + 0.003 \cdot \text{SI} + 0.11 \cdot \text{TI} - 0.0003 \cdot \text{SI} \cdot l + 0.004 \cdot \text{TI} \cdot l \quad (3)$$

$$Q_l = 4.70 + 0.10 \cdot b \cdot l - 0.09 \cdot l - 0.35 \cdot b + 0.01 \cdot \text{SI} - 0.05 \cdot \text{TI} - 0.0001 \cdot \text{SI} \cdot l + 0.08 \cdot \text{TI} \cdot l \quad (4)$$

with variables as for the previous model. The resulting unified plus SI and/or TI models have an adjusted R^2 of 0.80 for game launch and 0.81 for level load. Using these models on an untested game would require first capturing the game load content and running it through a tool[1] to compute the SI and TI.

Some game load videos show just one or two scenes for the entire load, whereas others have several different scenes to provide

[1] e.g., https://github.com/Telecommunication-Telemedia-Assessment/SITI

visual variety for the user. Similarly, some game load videos show progress indicators (e.g., a loading bar) that provide feedback to the user about when the game loading might complete, and prior work has shown the value in providing feedback to users that are waiting (see Section 3.1). We manually count scene changes and encode progress indicators (0 means none and 1 means one or more) for each game load video and use that as an additional input to the unified model:

$$Q_g = 4.90 + 0.07 \cdot b \cdot l - 0.10 \cdot l + 0.50 \cdot b + 0.11 \cdot \mathrm{p} - 0.03 \cdot \mathrm{s} + 0.002 \cdot \mathrm{p} \cdot l + 0.004 \cdot \mathrm{s} \cdot l \quad (5)$$

$$Q_l = 5.30 + 0.003 \cdot b \cdot l - 0.08 \cdot l + 0.44 \cdot b - 0.30 \cdot \mathrm{p} - 0.13 \cdot \mathrm{s} + 0.003 \cdot \mathrm{p} \cdot l + 0.01 \cdot \mathrm{s} \cdot l \quad (6)$$

where p indicates there is a progress bar and s is the number of scene changes. Other variables are as in the previous equations. Scene changes provide only modest benefit (adjusted R^2 of 0.71 for game launch and adjusted R^2 of 0.70 for level load) as do progress bar indicators (adjusted R^2 of 0.70 for game launch and adjusted R^2 of 0.69 for level load), but together they provide slightly more improvement (adjusted R^2 0.74 of game launch and adjusted R^2 of 0.77 for level load). Using these models on an untested game requires first manually watching and scoring the video for scene changes and progress indicators.

Finally, all of SI, TI, scene changes and progress indicators can be used along with the unified model:

$$Q_g = 4.67 + 0.10 \cdot b \cdot l - 0.09 \cdot l + 1.52 \cdot b - 0.004 \cdot \mathrm{SI} + 0.29 \cdot \mathrm{TI} + 0.33 \cdot \mathrm{p} - 0.12 \cdot \mathrm{s} - 0.0003 \cdot \mathrm{SI} \cdot l - 0.004 \cdot \mathrm{TI} \cdot l - 0.002 \cdot \mathrm{p} \cdot l + 0.006 \cdot \mathrm{s} \cdot l \quad (7)$$

$$Q_l = 5.31 + 0.16 \cdot b \cdot l - 0.11 \cdot l - 1.66 \cdot b + 0.01 \cdot \mathrm{SI} - 0.16 \cdot \mathrm{TI} - 0.50 \cdot \mathrm{p} + 0.05 \cdot \mathrm{s} - 0.0001 \cdot \mathrm{SI} \cdot l + 0.01 \cdot \mathrm{TI} \cdot l + 0.01 \cdot \mathrm{p} \cdot l - 0.005 \cdot \mathrm{s} \cdot l \quad (8)$$

The combined model provides for an adjusted R^2 of 0.82 for game launch and an adjusted R^2 0.84 for level load, within 10% of the "best-case" per-game load models.

4.3.3 Model Summary. Table 3 summarizes the models ordered by their increasing adjusted R^2 values and decreasing Root Mean Square Error (RMSE) values. The per-game models are averaged across their models, with the standard deviation shown in parentheses. The *All* model uses a single equation for QoE based on the game load time regardless of the game. As such, it is the easiest to use, but the least accurate. At the other end are the *Per-game* models, one per game, which are quite accurate for their respective game, but for untested games require user studies to assess quality with scaling making them impractical, in general. In between are generalized, *Unified* (U) models that consider the original game load time on a reference system as a parameter and then predict QoE based on the scaled game load times the player experiences. These unified models can be enhanced with options to add scene changes (S) and progress encoding (P) – both of which requires manual scoring – and spatial information (SI) and temporal Information (TI) – both of which require capturing then analyzing via code the spatial/temporal quantities. For use on an untested game, the unified models require measuring the game load time on a reference system akin to the one used to capture the game load videos in our study (see Section 4).

Table 3: QoE models. *All* - one model through all individual data points. *Unified* - single models parameterized by time (U), S - scene changes, P - progress indicators, SI - spatial information, TI - temporal information. *Per game* - separate model for each game.

		Game Launch		Level Load	
	Model	Adj R^2	RMSE	Adj R^2	RMSE
	All	0.60	0.47	0.65	0.38
Unified	U	0.70	0.39	0.66	0.37
	U, S	0.71	0.38	0.70	0.34
	U, P	0.70	0.39	0.69	0.35
	U, S, P	0.74	0.36	0.77	0.30
	U, SI	0.70	0.39	0.65	0.37
	U, TI	0.78	0.34	0.73	0.33
	U, SI, TI	0.80	0.32	0.81	0.27
	U, S, P, SI, TI	0.82	0.29	0.84	0.25
	Per-game	0.94 (0.04)	0.10 (0.03)	0.90 (0.20)	0.09 (0.05)

5 HARDWARE MEASUREMENTS

Since user QoE improves with a decrease in game load times, and game load times presumably can be reduced with upgraded hardware, we designed experiments to assess how game load times vary for different hardware components – potentially helpful for game players in deciding upon a system upgrade, hardware developers as they target next generation systems, or game developers as they consider the experience their players may have on different PC configurations. We setup a hardware testbed to facilitate adjusting configurations, implemented scripting software to automatically launch games and load their levels while measuring hardware performance, and ran repeated runs of our scripts for all games and different processors, graphics cards and storage devices.

5.1 Methodology

We designed and implemented Python scripts in order to automatically launch each game and load each level while recording game load times and hardware performance.

For each game, screenshots are captured for each button that the automated script needs to click in the proper order. Then, the automated script iterates through each button screenshot, polling every 0.5 seconds until the button is displayed on the screen and pausing a minimum of 1.5 seconds between button presses. This design allows for relatively easy update to the script when a publisher's game update changes the load sequence – when this happens, the old screenshot just needs to be replaced with one or more new screenshots for the script to follow.

In cases where the loading screen can be skipped (e.g., some game cut scenes allow the player to bypass them), the scripts are set to always skip them (i.e., the game loading is done as fast as possible).

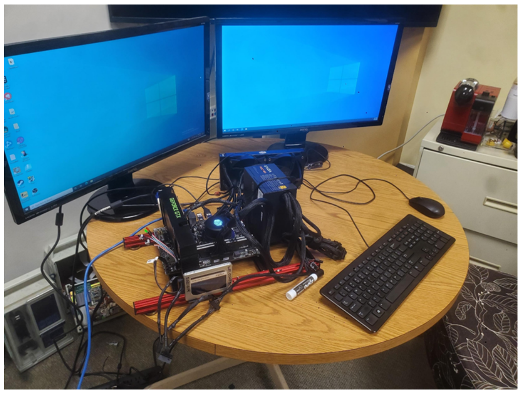
Figure 6: Hardware measurement testbed

Figure 6 shows a photo of our hardware testbed. Our testbed has an open-air frame to allow for easy swapping of components and to facilitate cooling to avoid performance degredations from overheating. All configurations used: A) the all-in-one Cooler Master MasterLiquid Lite 240 evaporative water cooler; B) an EVGA 700 GD 80+ GOLD power supply; and C) Team T-Force Vulcan Z 16 GB (2x8 GB) DDR4-3000 CL16 RAM. The testbed has two different motherboards which were swapped in as needed: the ASUS H110M-K with the LGA 1151 CPU socket and the H410M-A with the LGA 1200 CPU socket. The LGA 1151 CPU socket supports the 6th and 7th generation Intel processors, and the LGA 1200 socket the 10th generation Intel processor. The independent variables of interest and their parameters are shown in Table 4.

Table 4: Hardware Parameters

Component	Year	Type
CPU	2015	Intel i5-6500 CPU @ 3.20GHz (6th Gen)
	2017	Intel i5-7600 CPU @ 3.50GHz (7th Gen)
	2020	Intel i5-10400 CPU @ 2.90GHz (10th Gen)
graphics	2017	Intel CPU integrated UHD Graphics 630
	2015	NVIDIA GeForce GTX 960
	2020	AMD Radeon RX 5600XT
storage	2011	Seagate BarraCuda ST2000DM001 HDD
	2018	Crucial MX500 500GB 3D NAND SSD

Base system: Our base system has the most recent CPU, most powerful GPU and fastest storage device in our set: 10th generation Intel i5, Crucial MX500 SSD, and the AMD Radeon graphics card. From that base system, we varied each component independently: CPU, graphics card and storage.

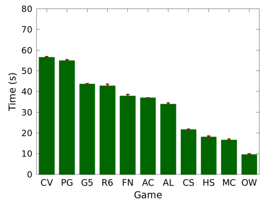

Figure 7: Game launch times

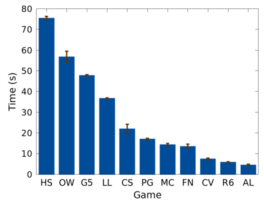

Figure 8: Level load times

During game load, a script collects hardware metrics and usage data every second using Open Hardware Monitor.[2] An evaluation of the script's overhead shows it contributes less than 3% to the CPU load. All of the tests on all systems were conducted within a couple of days in March 2022 and there were no game updates between the trials on each system.

State-of-the-art system: In addition, we measured one additional hardware configuration: a ROG Strix Z590-E motherboard with an Intel i9-11900K @ 3.50GHz (11th Gen) CPU, Corsair Vengance RGB 32GB DDR4 3200 RAM, a Samsung 970 Evo Plus solid state drive, and an NVIDIA GeForce RTX 2080 graphics card. This represents a more "state of the art" game system in terms of our performance evaluation.

During collection, the testbed used a single monitor at 1920x1080p. The same resolution was maintained throughout all tests so that the size and location of each target button matches the pre-recorded screenshots. For level loading for multiplayer network games (e.g., League of Legends), we used single player/offline modes with bots so as not to have the load times dependent upon other player's networks. Each game load was executed 10 times in order to understand variation across runs. One round of 10 iterations took about 6 hours, after which we archive the data and swap out a single hardware component (e.g., change the graphics card), and then repeat.

Assassin's Creed level load and League of Legends game launches failed on our base system so are excluded from all level load and game launch analysis, respectively. During our measurements on our "state of the art" system, League of Legends and Assassin's Creed had been updated, breaking our scripts so those games are excluded completely from analysis on that platform only.

5.2 Analysis

This section analyzes the game load times for the different components of interest: CPU, graphics card and storage device. These latter results are also compared to our "state of the art" system.

Figure 7 and Figure 8 depict the game load times on our base system for game launch and level load, respectively. The x-axes have the games, abbreviated (see Table 1). The y-axis is the time in seconds. The bars are the mean game load times for the 10 iterations on the base system, ordered high to low, and shown with a 95% confidence interval.

Games that load with a high CPU utilization (e.g., Minecraft) may benefit more from a more powerful CPU, while games that load with a high GPU (e.g., PUGB level load) may benefit more from a more powerful graphics card. Games with light CPU and GPU

[2]https://openhardwaremonitor.org/

Table 5: Total CPU and GPU usage percent for base system

		Game Launch		Level Load	
	Game	CPU (%)	GPU (%)	CPU (%)	GPU (%)
AL	Apex Legends	20.5	32.9	16.4	25.9
CS	CS:GO	12.8	17.1	15.1	41.7
CV	Civ VI	9.1	16.2	11.6	36.5
FN	Fortnite	17.2	12.4	21.9	17.0
G5	GTA5	8.9	15.0	8.8	20.9
HS	Hearthstone	9.9	18.6	3.2	16.2
MC	Minecraft	67.9	15.0	86.9	14.4
OW	Overwatch	24.0	20.1	10.7	77.4
PG	PUBG	18.7	28.2	41.2	81.9
R6	R6S	15.8	12.3	14.2	29.8

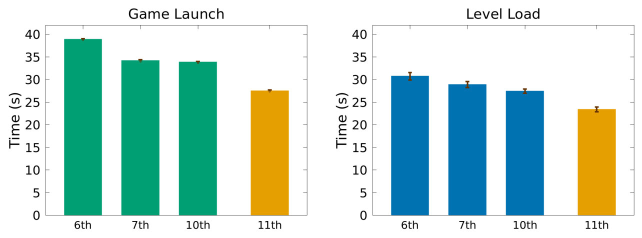

Figure 9: Game load time versus CPU generation

load (e.g., Hearthstone) may not improve much with an upgrade to hardware. Table 5 shows the overall breakdown of CPU and GPU usage recorded during game launch and level load. The units are a percentage reported by Open Hardware Monitor, averaged over all iterations. Overall, given the sizeable CPU and GPU loads observed, we would expect CPU and GPU to both affect game load times overall. In addition, when the CPU and GPU are not busy, we expect the game loading to be waiting on information from storage. Thus, overall, we expect an improved storage device – an SSD versus an HDD – to also improve game load times.

To assess the impact of hardware components on game load times, we measure the time it takes for game loading for one iteration of all games and then divide that by the number of games to get the average game loading performance. We do this separately for each of game launch and level load, for 9 iterations of each.

Figure 9 depicts the game loading time versus CPU generation. There are two graphs depicted: on the left is the average game launch performance and on the right is the average level load performance. The y-axes are the game load times in seconds and the x-axes are the CPU generations, ordered oldest to newest. Each bar is the mean game load time shown with a 95% confidence interval. For this and subsequent graphs, we show the game load time for our "state of the art" system on the right, separated by a bit of space and with a different color. This system has different (better) components than our base system so is not directly comparable for the individual components, but provides a useful reference point for where our system performs relative to an overall higher-end system.

From the graphs, there is a statistically significant difference (the confidence intervals do not overlap) in the game load times, with a reduction to average game load time for an improved CPU. Game launch times decrease about 4% per CPU generation and level load times decrease about 3% per CPU generation.

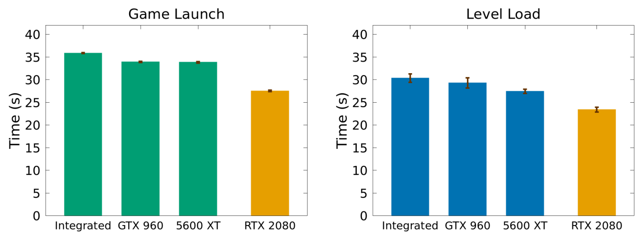

Figure 10: Game load time versus graphics card

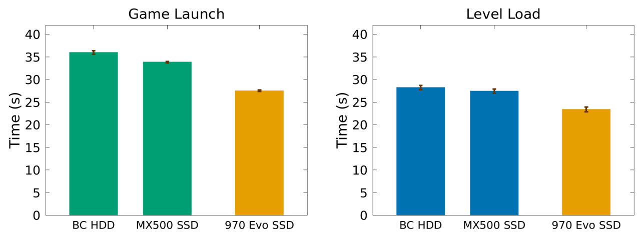

Figure 11: Game load time versus storage device type

Figure 10 depicts the same data, but for different GPUs on the x-axes, ordered from least powerful (a GPU integrated with the CPU) to most powerful (a 5600 XT). From the graph, the GPU also makes a statistically significant difference (the confidence intervals do not overlap) to performance with a decrease in game load times with an increase in GPU power. Times decrease about 10% from the integrated graphics card to the best graphics card for game launch and decrease 6% for level load.

Figure 11 depicts the same game load performance but with two different different storage devices: a BaraCuda hard disk drive (HDD) compared to an MX500 solid state drive (SDD). The storage device also makes a statistically significant difference to performance, albeit not as much – the SDD only decreases game load times by about 3%.

6 DISCUSSION

From the analysis in Section 4.2, waiting time is clearly a dominant determinant to the player experience while waiting to play a game – i.e., how long the game load takes matters the most – but other aspects such as visual content and feedback influence a player's experience, as well.

The models summarized in Table 3 present a variety of options to predict player QoE for a particular game load. Each model is potentially of use, depending upon the situation: *All* can be used for a general assessment of QoE with game load time, *Unified* is appropriate where base load times can be measured with refinements made, as needed, based on the visual information, and *Per-game* models are viable for when specific game load predictions are needed (e.g., by a game developer). Platform developers – those that seek to improve game load times in general – would do well to use the Unified model coupled with tools that automatically measure game load times, compute SI and TI, and classify progress indicators and scene changes, without manual intervention. Any predicted QoE values can be combined with our unacceptability analysis in Figure 4 to determine if players are likely to be satisfied with particular loading times. For "acceptable" QoE, developers would be advised to keep game start times to less than 27 seconds and level load times to less than 33 seconds.

The relationships observed between game load times and QoE appear linear over the range of times tested. This linear trend with time is different than the logarithmic trend with time observed for other applications, such as Web browsing (see Section 3.1). The differences in trends may stem from the range of values tested where much larger game load times could show a lessening of the impact of time on QoE – in fact they must as they approach the floor value (1) of the 5-point QoE rating scale. Other differences in the trend shape with time could come from the feedback that game load screens provide to users that helps temper their experience.

The game load times measured in Figure 7 and Figure 8 show there is a wide range of game launch and level load times, respectively. However, given the relatively small improvements to overall game load times shown for better CPUs, graphics cards or storage devices, it is clearly challenging to just upgrade a computer's hardware and have sizably reduced times. In other words, users cannot just upgrade their computer and expect to have dramatically lower game load times, nor can developers expect their users to do the same. Instead, the game developers themselves likely have the the greatest chance of making differences to reduce game load times, or at least add visual feedback to make waiting more palatable for their players.

7 LIMITATIONS AND FUTURE WORK

The 54 users in our study provided a sample size large enough for statistically significant results for user QoE with game load time, but more users may help with challenging QoE predictions for games like Apex Legends. Similarly, potentially assessing more game load times within a game could help determine where a linear relationship does and does not hold.

Our methodology intentionally had users watch videos of game loading instead of actual launching and playing games in order to minimize human errors and ensure reproducibility of the study. While the process requires the same user interactions (e.g., mouse clicks) as an actual game loading experience and users were instructed to use their familiarity with the game combined with their expectation to play, but how the results differ from actual game loading experience is unknown.

Our sample is skewed towards males and university students, so has a narrow age range (18-22 years). Similarly, our study of 12 games across 7 genres is reasonably broad, yet some genres are not tested, nor is the game loading experience assessed on devices other than a PC (e.g., mobile).

Future work is to validate our models with more users and games and, once validated, use the models to simulate game loading experience for wide range of games. In addition, instead of linear regression, we may look to other shapes for an even better fit.

8 CONCLUSION

People are increasingly turning to games for entertainment evidenced by the growth in the game and esports industries. While users may be eager to play, waiting for game loading is inevitable and, unfortunately, the waiting time can degrade player quality of experience (QoE). Up to now, how much QoE degrades while waiting for a game to load is not known.

This paper presents results from a user study that assesses player QoE with different game loading times. We recorded game loading videos and built a self-contained application to mimic the actual game loading experience. By re-encoding the game loading videos to be faster and slower and embedding them into our application, we are able to measure how game loading time directly impacts QoE. We setup a user study in a dedicated lab where 54 participants each launched and loaded 3 games (selected from 12 games) across 9 different game load times, providing subjective opinions on their experience via surveys.

Analysis of user study results shows both game launch and level load times have significant impact on player QoE. Across the range of game load times studied (about 10 to 80 seconds), a 50% decrease in game loading time improves player experience by about 0.75 points on a 5-point scale – an amount that can improve the player experience from unacceptable to acceptable. Models of QoE with game load times suggest simple linear regression can be improved by considering visual content.

This paper also presents results from experiments that measure game load times for different processors, graphics cards and storage devices. We design and implement scripts to automatically load games and record performance and run repeated iterations of loading and launching the 12 games, individually swapping out a single hardware component at a time to assess its impact.

Analysis of the measurement results shows considerable variation in game launch times and level load times, differing by 6-fold for game launch and 15-fold for level load. CPU and GPU use during game loading is similarly varied, but all games use some of each, suggesting improvements to hardware can reduce game load times. Supporting this, our analysis shows better CPUs and graphics cards can reduce game load times by about 5% per generation, with a slightly smaller benefit to game load times (3%) for upgrading from a hard disk drive to a solid state drive.

REFERENCES

[1] Josh Allard, Andrew Roskuski, and Mark Claypool. 2020. Measuring and Modeling the Impact of Buffering and Interrupts on Streaming Video Quality of Experience. In *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia (MoMM)*. Chiang Mai, Thailand.

[2] Anna Bouch, Allan Kuchinsky, and Nina Bhatti. 2000. Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. The Hague, The Netherlands.

[3] Russell Branaghan and Christopher Sanchez. 2009. Feedback Preferences and Impressions of Waiting. *Human Factors* 51, 4 (2009), 528 – 538.

[4] Anping Cheng, Dongming Ma, Hao Qian, and Younghwan Pan. 2023. Exploring the Long-Term User Experience of an Interactive Loading Screen Using UX Curve and QUIS. *International Journal of Human–Computer Interaction* (May 2023).

[5] Anping Cheng, Dongming Ma, Hao Qian, and Younghwan Pan. 2023. The Effects of Mobile Applications' Passive and Interactive Loading Screen Types on Waiting Experience. *Behaviour and Information Technology* (2023).

[6] V. Clincy and B. Wilgor. 2013. Subjective Evaluation of Latency and Packet Loss in a Cloud-Based Game. In *Proceedings of the 10th International Conference on Information Technology: New Generations*. Las Vegas, NV, USA, 473 —- 476.

[7] Sebastian Egger, Peter Reichl, Tobias Hoßfeld, and Raimund Schatz. 2012. 'Time is Bandwidth'? Narrowing the Gap between Subjective Time Perception and Quality of Experience. In *Proceedings of the IEEE International Conference on Communications (ICC)*. Ottawa, ON, Canada.

[8] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen. 2012. Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea. In *Proceedings of IEEE QoMEX*.

[9] Barry Ip and Gabriel Jacobs. 2004. Quantifying Game Design. *Design Studies* 25, 6 (Nov. 2004), 607–624.

[10] Woojoo Kim, Shuping Xiong, and Zhuoqian Liang. 2017. Effect of Loading Symbol of Online Video on Perception of Waiting Time. 33, 12 (2017), 1001 – 1009.

[11] Carine Lallemand and Guillaume Gronier. 2012. Enhancing User Experience During Waiting Time in HCI: Contributions of Cognitive Psychology. In *Proceedings of the Designing Interactive Systems Conference (DIS)*. Newcastle Upon Tyne, UK.

[12] Jonathan Ledbetter. 2016. *Interactive Amusement Park Queues: Examining the Indirect Effect of Task Work Load on Guests' Perception of Wait Duration Through Task Immersion.* Ph.D. Dissertation. University of Central Florida (UCF), Gainesville, FL, USA.

[13] Fiona Fui-Hoon Nah. 2004. A Study on Tolerable Waiting Time: How Long are Web Users Willing to Wait? *Behaviour & Information Technology* 23, 3 (2004), 153–163.

[14] Jakob Nielsen. 1993. *Usability Engineering*. Morgan Kaufmann, San Francisco, CA, USA.

[15] Christof Van Nimwegen and Emiel Van Rijn. 2023. Time Swipes When You're Having Fun: Reducing Perceived Waiting Time While Making it More Enjoyable. *Behaviour and Information Technology* 42, 3 (2023), 307–315.

[16] ITU-T Rec. 2008. Subjective Video Quality Assessment Methods for Multimedia Applications. (Accessed September 5, 2021).

[17] Dominique Scapin and J. M. Christian Bastien. 1997. Ergonomic Criteria for Evaluating the Ergonomic Quality of Interactive Systems. *Behaviour & Information Technology* 16, 4-5 (1997), 220 – 231.