A Proposed Taxonomy for the Design Qualities of Video Game Loading Interfaces and Processes

David Antognoli
Columbia College Chicago
Interactive Arts and Media Department
Chicago, IL USA
312-369-6927
dantognoli@colum.edu

Joshua A. Fisher
Ball State University
Center for Emerging Media Design and Development
Muncie, IN USA
765-285-8901
Joshua.fisher@bsu.edu

ABSTRACT
Though ubiquitous, the design and development of loading interfaces and processes has not received the critical attention in games studies that their presence deserves. As interfaces, they illustrate a designer and developer's desire to create game experiences that push the technical limits of available computing hardware. While loading screens are well known, loading interfaces span myriad forms. From an archaeogaming perspective, this paper examines how the design of video game loading interfaces and processes is a response to ever-increasing demands for higher fidelity gaming experiences on the behalf of players and designers in the face of hardware limitations. This histography of loading interfaces and processes is one of technical and design innovations that demonstrate the ethos and telos of designers. Through an Interface Study, the following design qualities of loading screens were derived: hypermediacy and transparency, diegetic and non-diegetic, passive and interactive, and pedagogic and misdirection. We conclude the paper by looking at case studies that exemplify the derived design qualities of loading interfaces and processes.

Keywords
Game design, archaeogaming, loading screens, game user experience design

INTRODUCTION
A lack of critical attention has been paid to the design and implementation of video game loading interfaces and processes. Perhaps related to Brendan Keogh's conception of the purity complex in games studies, the absence of critical scholarship is noted and "non-play elements are ignored" (2014). Yet, such elements enable a greater fidelity of play and impact states of flow for players (Iacovides et al. 2015). Further, players themselves create fan sites and forum posts around loading screens ("Loading Screen Games" n.d.; TheRittyl 2014; "Loading Screens" n.d.), and industry periodicals write about them (Chitty n.d.; Kantor 2021; Gilbert 2013). The apparent popular attention but lack of critical discourse illustrates that loading interfaces, as well as their accompanying processes, demand our attention in the historiography of games. While

Proceedings of DiGRA 2023

loading screens are iconic interfaces of loading, myriad other forms and approaches exist. A history of design strategies for loading interfaces signifies a history of innovation in the evolving loading processes of games and the representation of those processes for users through user interfaces. As artifacts within games, examining loading interfaces helps us understand the ethos (credibility) and telos (purpose) of their designers by learning how others have historically navigated loading.

This article proposes essential and distinguishing qualities of video game loading design strategies. From these qualities, we elucidate a taxonomy through an archaeogaming perspective that uses an Interface Study methodology to explore 139 games. These games were chosen through convenience sampling. As no official archive exists, these games were chosen because their loading interfaces and processes were documented and/or a discussion of them was available in popular media, on fan sites, in forum posts, or on social media. Of these 139 games, we selected some emblematic and historical examples to illustrate patterns that led to the taxonomy's themes. The taxonomy identifies loading interfaces and their accompanying strategies along four different scales. First, hypermediacy to transparency; second, diegetic to non-diegetic; third, passive to interactive; fourth, pedagogic to misdirection. We recognize that each of these design choices are rooted in a design aesthetic necessitated by hardware constraints, the ethos of the designers, and the telos of their design[1].

Defining loading, the loading dilemma, and the history of loading interfaces

Loading in video games occurs when game software performs prerequisite work for the proper functioning of the play experience. This often involves game data being moved from a storage medium, such as a hard drive or disc media, into a computing platform's memory. However, loading encompasses a wide variety of scenarios, such as downloading new game files from the internet or procedurally generating content. All loading is transitional, marking a movement from one game state to another. This transition can involve states and content that exist outside the game's narrative world, such as the loading that occurs when a game applies an update to a newer version. Or the transition can bridge adjacent sections within a game's narrative context. The narrative adjacency surrounding loading can be physical, as in the loading transition between adjoining playable spaces in *Half-Life* (Valve 1998), or temporal, as in the loading that occurs when Link travels between the past and future in *The Legend of Zelda: Ocarina of Time* (Nintendo EAD 1998).

The loading dilemma includes two key challenges for game developers. First, loading takes time, and a game cannot function properly until loading has completed. Without a strategy for addressing this problem, a game will appear blocked, frozen, or incomplete until loading finishes. In addition, even if users understand that the game is loading, they may become frustrated by frequent or long loading times. Second, game developers commonly desire to include more content in their games than host hardware platforms have available memory to support. This means only a portion of a game's content can be loaded into memory at once. Running out of memory at runtime causes software to suddenly stop working (i.e., "crash"). To address these challenges, game developers must limit content or engineer loading solutions palatable for players. In this way, loading signifies a designer's desire to expand a game experience beyond the technical limits of computing hardware.

Game developers commonly organize games into distinct playable sections, levels, rooms, or areas. These distinct parts often feature unique environment themes, obstacles, items, characters, and scripted sequences. This organization enables designers to control pacing by designing a sequence of content with narrative and play progression. It also creates a natural way for organizing labor on a project, where a set of developers can work on one level or section without worrying about stepping on the toes of developers working on other levels or sections. This increases development efficiency as developers must only load the sections of the game they are currently working with. From a player perspective, games following this pattern often load as the player transitions from one gameplay section to the next. This facilitates the unloading of unused content from the previous sections and allows each distinct section to include as much unique content as can be fit into memory. Higher fidelity content, such as high polygon count models and high-resolution textures, typically requires a larger memory footprint. With this organizational method, rather than budgeting the amount and quality of content so that everything in an entire game is squeezed into memory at once, only the content that the player is currently experiencing needs to be loaded. This strategy enables the possibility of larger and higher fidelity game worlds.

As interactive systems, games involve a looping flow of user input and system output. When this flow is interrupted the system loses responsiveness (Karhulahti 2012). Responsiveness is critical to how games achieve immersion and evoke game feel, thus unresponsive game systems have a litany of negative consequences for player experience (Swink 2009). While games are loading, they are not completely functional. If game content is simple enough relative to the capacities of the host hardware, it can load so quickly that players cannot perceive the load. However, with more ambitious (or inefficient) game content, loading can take long enough for players to perceive, and a game can become unresponsive until loading completes. Steve Swink, game developer and author of the book *Game Feel* (2009), explains that games will feel unresponsive after just 240 milliseconds of delay between player input and system feedback. For this reason, designers need to convey the loading status to players to explain pauses in responsiveness and prevent suspicion that the game has stopped functioning (Nielsen 1994; Nielsen and Molich 1990). This usability-based need led to the invention of loading interfaces, and some modern game console platforms require developers to provide animated loading feedback for any extended pauses in gameplay (Strat 2017).

While loading feedback dates to the physical status lights on mainframe computers of the 1960s (CuriousMarc 2017) and early computer operating systems included countdowns and other loading progress updates (Moshix 2017), video games drove the cultural significance of loading screens in the 1980s with engaging art and sound that spurred lasting fandom (Onaretrotip 2021; Cundy 2009; Let's Talk Retro 2017; Pickford 2004).

The forms of video game loading interfaces and processes have evolved with games as developers have navigated the loading dilemma across ever-shifting hardware and software platforms. These interfaces and processes reflect a tension between consumer demands for larger and higher fidelity game experiences and their impatience with loading. Examining hardware changes over successive game console generations reveals intersecting factors of cost, speed, and storage space that influence loading. Early game consoles used cartridge-based game media with low storage and quick loading. In the fifth generation of game consoles, the Playstation introduced higher storage, cheaper to manufacture, optical disc media that enabled greater fidelity content than previous cartridge-based consoles. However, the mechanical operation of its disc drive introduced longer loading times. As a result, PlayStation versions of 2D games that loaded instantly on previous console platforms take longer to load (Wittkamp n.d.). With the seventh generation of game console hardware, hard disk drives became standard (Palomba 2016). This introduced a new loading trade-off for gamers as games began to require lengthy installs up front with the promise of faster loading in game after the install process. In addition, the inclusion of built-in storage and internet connectivity created new loading scenarios as developers created downloadable content and patches for games. In the ninth generation of game consoles, speedier solid-state drives replaced hard disks. The faster loading potential of these drives paved the way for more immersive open worlds with smaller or even invisible loading seams.

METHODOLOGY

Archaeogaming is the chosen perspective for this project. Archaeogaming is material agnostic and approaches artifacts whether they exist in physical media (like a CD) or are downloaded from Steam (Reinhard 2018). Our focus then is agnostic and not attached to any singular platform's loading processes and interfaces. Additionally, we are not focusing on the cultural or societal context of loading screens. Instead, we are focusing on their implementation to answer, "where are these old ideas useful or in use today?" (Aycock 2016, 206). Game Archaeologist Andrew Reinhard's recognition that, "the video game archaeologist can now study hardware and software and how they combine for gameplay" (2018) draws our attention to the design methods that addressed loading processes and interfaces. Each loading interface becomes an opportunity to interrogate just how designers of an interface grapple with the capacities of hardware to support the play experience.

We utilize the Interface Study method used in previous archaeogaming scholarship. This qualitative methodology, established by Mia Consalvo and Nathan Dutton, defines the interface as, "any on-screen information that provides the player with information concerning the life, health, location or status of the character(s), as well as battle or action menus, nested menus that control options such as advancement grids or weapon selections, or additional screens that give the player more control over manipulating elements of gameplay." (Consalvo and Dutton 2006). We extend this understanding to include the long-ignored loading screen that can provide information that may alter a player's approach to gameplay. Further, we felt such an approach was appropriate considering Consalvo and Dutton's use of the method to comment on the ideology behind UI design choices and the ethics of designers.

Much like Consalvo and Dutton's Object Inventory method discussed in the same paper, we developed a database of interfaces to direct our analysis. We employed an undergraduate research assistant to develop our list of 139 games and their loading interfaces. For each entry we made notes on the elements that made up the interface, how those elements were composed, and whether they were interactive. These 139 artifacts were chosen through convenience sampling. This method of sampling was chosen because there are no formal archives or databases of loading screens. The information exists in discussions available online through reviews, blog posts, social media, and fan sites.

For our analysis, we utilized the previously established definition of loading. This definition includes several transparent mechanics that still encompass a loading process. These transparent mechanics include the dynamic loading or streaming of new areas, such as in massively multiplayer online games and single player open world experiences (Chen et al. 2005; Clark 1976; Good 2013). Focusing on the process as part of our analysis informs our understanding of the composition, design, and implementation of a game's loading interface. The focus represents the understanding that loading processes evolve along with hardware capacities and software demands. As new methods of loading become available, such as those discussed in (Bhojan et al. 2020; Kang, Kim, and Kang 2017; Brusca 2021; Harper 2022), the way in which they are experienced from a player's perspective evolves.

Our collection of data points on loading processes and interfaces encompasses four decades of using hardware to marshal platform resources to support game experiences. Each interface and process represents a programming and design team's attempt to create a novel game experience. Our presented framework identifies the ways in which these interfaces and processes have been designed to achieve that particular effect. Our inventory, located at www.loadinginterfaces.space, allows games researchers to interrogate the effectiveness of these solutions, in what context they exist, and how they achieve their purpose. Designers and developers will be able to consider how to use these lessons from the past in their own work.

VIDEO GAME LOADING QUALITIES TAXONOMY

Solving the loading dilemma has given rise to a wide variety of design techniques and strategies for loading interfaces. This variety makes a taxonomy of video game loading design strategies valuable to game designers and researchers. We propose qualities understood from the player perspective, as opposed to the developer perspective. Our explanation of each quality does not relate to the technical underpinnings of how games work beyond what is necessary to understand each quality. These qualities can overlap or exist to various degrees in each loading interface or process.

Hypermediacy / Transparency

While all loading is transitional, the transition can be described on a scale between hypermediacy and transparency. Hypermediated loading intentionally presents and calls attention to itself, while transparent loading attempts to go unnoticed (Bolter and Grusin 2000). Loading screens, the emblematic interfaces of loading, are traditionally opaque elements that call attention to a video game's mediation of a user experience.

Hypermediated loading screens can help orient the player and process a change in narrative time or place. For example, games that feature a day/night cycle simulating the passing of time often include content that only exists at certain in-game times. Since waiting in real-time for the in-game time to pass can be boring, these games commonly offer a way to jump to a specific time. During such transitions, it would be jarring for players to see sudden dramatic lighting changes and characters with behavior and locations based on in-game time of day abruptly disappearing or appearing. As a result, these loads commonly take a hypermediated approach that includes some indication of the passing time. The game *Elden Ring* (FromSoftware 2022), for example, fades to a black loading screen with a dial that shows the changing state of the sun before fading back in to reveal the new game state (Waldrop 2022). This helps players process the change in time. In addition, many games with large worlds often include a fast travel system for quickly teleporting the player character to a previous location rather than requiring players to manually pilot characters across great in-game distances. Such games often employ hypermediated loading for player character location transitions triggered through fast travel systems. Notably, many games that otherwise attempt to create large seamless worlds with transparent loading still use hypermediated loading screens for fast travel and time change scenarios (e.g. *Elder Scrolls V: Skyrim* (Bethesda Game Studios 2011), *Elden Ring*, *Pokémon Scarlet* (Game Freak 2022), etc.). Even if the transition could technically be handled instantly, the loading screen interface helps players orient themselves and process the scene change.

In this way, hypermediated loading is analogous to scene changes in a theatre production. When the stage needs to transition from one time or place to another, the lights turn off or a curtain is used to mask the work taking place as the set is rearranged. This preserves the active creation of belief (Murray 2017) and helps audiences process the narrative transition in time and/or place. In this analogy, the loading screen functions as a curtain, hiding the loading work being done behind it (Laurel 2014). The functionality of the loading screen as a theater curtain is highlighted by the interactive drama game *Façade* (Mateas and Stern 2003), which uses a curtain visual motif for its loading screen. The curtain as loading screen visual metaphor is used in other games as well, such as *Paper Mario: The Thousand Year Door* (Intelligent Systems 2004) and

BattleBlock Theatre (The Behemoth 2013). Just as a theater curtain is emblematic of stage productions, loading screens are signifiers of video games.

Alternatively, some video games prioritize immediacy as a mechanism for creating immersion. Bolter and Grusin explain that technology seeking immediacy should closely reflect the real world to create a sense of presence. Immediacy demands the transparency of an interface that erases itself to put the user "in an immediate relationship to the contents of the medium" (2000, 318). In contrast to hypermediated loading where the loading process is announced by an opaque signifier, transparent loading seeks to make the loading process invisible to players. By making loading invisible to players, games with transparent loading keep players grounded in the game world and avoid reminding them that they are engaged in a mediated experience. These games eschew UI-based transitions in favor of transparent solutions.

With transparent loading, players seamlessly experience real-time ludic and/or narrative content. There are technical and design implications to this strategy. Without the loading screen "curtain" to hide transitions behind, the playable space must be arranged in a way that prevents the player from seeing the parts of the game that are not currently loaded in memory. When players can access content before it is fully loaded, the glimpse behind the curtain disrupts immersion and often results in frustration (WCK619 2012; Devil Dog 2020). Designers commonly employ strategic level layouts such as air locks or U-shaped curves to facilitate transparent loading and block visibility of unloaded content (Yang 2013). Game developers also commonly employ fog that increases in opacity further from the game camera to hide unloaded content. The game *Turok: Dinosaur Hunter* (Iguana Entertainment 1997) on the Nintendo 64 provides an example of the competing design considerations of loading. The game's sprawling outdoor environments marked a departure from the constricted level design of corridor shooters of the era such as *Goldeneye 007* (Rare 1997) and *Doom 64* (Midway Studios San Diego 1997). However, these game environments required a notoriously dense fog to hide unloaded content and preserve performance. Another common strategy is to use lower-detail versions of game visual elements while those elements are far away from the game camera and transparently load higher-detail versions as the camera nears (Clark 1976; Unity Technologies 2022).

The chunks of content that are dynamically loaded can be further subdivided beyond spatial considerations. For example, in *The Conduit* (High Voltage Software 2009), which boasted higher fidelity art assets than typical Wii games of the time, the assets for all the weapons could not fit in the Wii's limited memory along with the environment art for any given level. As a result, each level in the game only included a subset of the game's weaponry. In *Conduit 2* (High Voltage Software 2011), to support the designers' desire for the use of any weapon in any level, a system was devised to transparently load and unload weapon assets as needed independently from the environment art. The loading takes place while the player's equipped weapons are off camera during an animation that plays when switching weapons. This enabled all the game's weapons to be usable in any level while avoiding a jarring pause in the action when the player switches weapons.

Transparent loading is often used in open-world games that feature less rigidly defined levels where the space between levels is playable. Since it takes time to load new content, intermediary sections connecting distinctly loaded sections must be large enough (or the player character's move speed must be slow enough) for the game to complete the load before the player can view the new area. A good example can be seen in *The Legend of Zelda: Wind Waker* (Nintendo EAD 2002) on the GameCube, where large oceans separated detailed levels. Interestingly, when the game was rereleased on the more advanced WiiU hardware platform, the developers included a movement speed upgrade in response to the criticism that the extended sailing portions wore on players. This speed upgrade would have broken the original GameCube version as player character movement could outpace loading, allowing access to areas before they completely loaded. On the Wii U, however, greater available memory meant more of the game's world could be processed at once, enabling the faster movement speed (Good 2013). The *Wind Waker* example demonstrates how hardware with more memory and hardware capable of faster loading enables more transparent loading gameplay scenarios. This is reflected in the preponderance of open-world gameplay featured in games designed for contemporary generations of console hardware.

Diegetic / Non-diegetic

The distinction between diegetic and non-diegetic has been discussed by game theorists and game designers in relation to game user interfaces (Genette 1980; Bordwell and Thompson 1993; Fagerholt and Lorentzon n.d.; Stonehouse 2014). In this context, diegetic elements of a game exist within the game's narrative world. Characters within the narrative world can perceive diegetic elements. Accordingly, in video games, diegetic loading interfaces are blended into the environmental context of the game's fictional world. For example, in *Metroid Prime* (Retro Studios 2002), loading occurs behind closed doors that exist in the 3D game world. These doors will not open until the loading has completed ("Metroid Prime/Gamecube vs. Wii Loading Discrepancies" 2020). In this way, *Metroid Prime's* doors play the role of a loading screen, blocking the player from seeing the loading happening behind them. It is understood that Samus (the player character) and the other characters of the game's narrative world can perceive these doors; though they do not perceive the fourth wall-breaking concept of loading, which would acknowledge the fictionality of their world.

Non-diegetic elements, on the other hand, exist outside of the narrative world. Characters in the narrative world are not aware of and cannot perceive non-diegetic elements (without breaking the fourth wall). The classic loading screen is an example of a non-diegetic loading interface. The loading screen is intended to be perceived only by players beyond the fourth wall of the narrative experience. For example, the loading screen for *Armalyte* (Cyberdyne Systems 1988) on the Commodore 64 includes the game title and publisher. This information is presented to players and is not meant to exist within the game's narrative or be perceived by its characters.

Notably, not all diegetic loading is transparent. Diegetic and non-diegetic can be considered on a spectrum, like hypermediacy and transparency. For example, the game *Deus Ex: Mankind Divided* (Eidos Montréal 2016) features a diegetic loading sequence where the player character is on a train car moving between playable areas. However, while rendered by the game engine in real-time, this "screen" is not interactive and displays an animated icon to help players understand that the game is loading. Reflecting the transitional quality of loading, diegetic depictions of travel during loading is a common motif, as seen in other games such as Marvel's *Spider-Man: Miles Morales* (Insomniac Games 2020), *Star Wars: Jedi Fallen Order* (Respawn Entertainment 2019), *Mass Effect 2* (BioWare 2010), *Metroid Dread* (MercurySteam and Nintendo EPD 2021), and *XCOM 2* (Firaxis Games 2016).

Passive / Interactive

Loading screens historically originated as passive elements. Players could not influence or interact with them in any way. As discussed, loading traditionally interrupts the responsiveness of video games as interactive systems. This downtime comes at the expense of players who expect interactivity and entertainment from video games. Some games have become notorious for long or frequent loading. For example, the PS2 game *Crash Bandicoot: The Wrath of Cortex* (Traveller's Tales 2001) has been pilloried for featuring gameplay levels with playtimes shorter than their loading times (Sinha 2017). The game *Sniper Ghost Warrior 3* (CI Games 2017) on the PS4 took so long to load when the game was first released that one reporter demonstrated it was possible to complete an entire race in *Mario Kart 8*, catch multiple Pokémon in *Pokémon GO*, or complete four speed runs of the entire game *Gone Home* during the load time (Higton 2017).

However, players' tendency to play other games while waiting for a game to load has not been overlooked by game developers, spurring the innovation of the interactive loading screen. As Brenda Laurel puts it, "A key premise of the mobile-technology game industry is that the pleasure of interactivity is preferable to boredom." (Laurel 2005). Following this philosophy, some games include simple interactions on their loading screens. For example, on the *Elden Ring* loading screen, players can press a button to advance through a list of tips. The loading screen in Fire *Emblem: Three Houses* (Intelligent Systems and Tecmo 2019) allows players to move a simplified pixel art representation of the player character from one side of the screen to another with gyroscope input and jump by pressing a button. In *Skyrim*, players can rotate 3D models of props from the game world with directional input.

While in the former examples, there is no gameplay objective connected to these interactions, other games employ full on minigames to entertain players during loading. This practice was born on the Commodore 64 platform, when games were loaded into the computer's memory from a notoriously slow tape medium in a process that could take five to 10 minutes. A playable minigame could load in one minute or less. In 1984's *Skyline Attack* (Argonaut Games 1984), players could play a clone of the game *Snake* in what is considered the first instance of a loading screen minigame (Onaretrotip 2021). Infamously, over a decade later, Namco successfully filed a patent for "auxiliary games" to be played while a main game loads, preventing an "unnecessary wastage of time" (Hayashi 1998). There are even loading minigames where the outcome of the minigame can impact the main game state, such as in *Rayman Legends* (Ubisoft Monpellier 2013) and *Okami* (Clover Studio 2006).

Pedagogic / Misdirection

Another set of loading interfaces are pedagogical and educate the player about a variety of topics. In addition, a smaller set of loading interfaces engage in misdirection. By misdirection, we mean that the interface does not teach the player but instead seeks to distract the player or misinform them. Pedagogical interfaces may teach the player about the game world, play controls, gameplay hints, and even reflexively about themselves. Misdirection alternatively either distracts, misinforms, or acts as a superfluous spectacle that does not parallel loading processes. In short, the difference between pedagogy and misdirection is that the former intentionally helps the player have a more educated play experience and the latter intentionally occludes or obstructs that experience to achieve a secondary goal.

Pedagogic loading screens are incredibly common. In the *Total War* series (Creative Assembly n.d.) players experience world building pedagogical information that is either historically or lore accurate. Loading screens that provide tips for gameplay are also common, especially in complex games. *Europa Universalis IV*, an expansive and complicated strategy game, (Paradox Development Studio 2013) contains tips for gameplay to help the player digest its vast ruleset over time. A compelling example of reflexive pedagogical loading screen comes from *Spec Ops: The Line* (Yager Development 2012). These loading screens ask the player to reflect on video games, murder, and state-sanctioned violence. One screen has the line "To kill for yourself is murder. To kill for your government is heroic. To kill for entertainment is harmless." Another says, "The US military does not condone the killing of unarmed combatants.

But this isn't real, so why should you care?" Both lines may motivate a critical reflection on the military-industrial-media-entertainment complex and the player's complicity as a consumer-participant.

Misdirection in loading screens is more insidious. It is rife in mobile games wherein loading may be intertwined with advertisements for in-game microtransactions or other mobile games from the same developer. Another example of misdirection is the insertion of ads that cannot be skipped or other marketing content such as in *NBA 2K21* (Visual Concepts 2020). Sometimes, this misdirection serves pedagogy as in Guerilla Game's *Horizon Forbidden West* (2022) where loading times are artificially extended to provide more time to read the hints (Gratton 2022). The misdirection occurs in the sense that the loading process is already completed, the screen is superfluous but for the pedagogical goals of the designers. They do, however, provide the user with the agency necessary to start the game when they are ready. This form of misdirection might be classified as a benevolent deception (Adar, Tan, and Teevan 2013).

APPLYING THE LOADING QUALITIES TAXONOMY

Having defined the separate qualities, this section now demonstrates how the proposed taxonomy may be applied in analysis. Through the analysis, it will become clear that each case study makes use of the defined qualities of video game loading processes and interfaces. The way in which designers have implemented these qualities and to what degree differs considerably from game to game. We have chosen games that are exemplars of particular qualities. However, these experiences still have other qualities present.

Mass Effect: Transparent, Diegetic, Interactive

Figure 1: Screenshot of an elevator loading sequence in *Mass Effect*. The news subtitle is voice acted.

The sci-fi roleplaying game *Mass Effect* (2007) marked the first BioWare title for the seventh generation of game consoles and their first game using the then new Unreal Engine 3 game engine. It was the first entry of a new IP and introduced a variety of new design ideas enabled in part by the more advanced technology. One such innovation was the adoption of transparent loading. As seen in Figure 1, *Mass Effect's* level designers used elevators to disguise loading without interrupting the normal gameplay camera. The elevator accomplished the necessary visual restrictions to hide the previous section unloading and the new section loading while providing a natural moment of transition familiar to most players. Dusty Everman, senior level designer at BioWare, explained the studio's then design philosophy that "Working elevators and airlocks are a much nicer and [more] immersive experience than load screens."(Totilo 2013). The elevator loading sequences also have a strong diegetic quality. During elevator rides, fictional audio news clips or advertisements play out that help construct the game's narrative world. Additionally, characters strike up conversations that supplement the plot and help with character development. In *Mass Effect*, the player can swap in and out party members, providing a variety of unique conversations that can play out in elevators. Everman explains, "The elevator banter was added to make the load time of the elevator entertaining. This was a fantastic 'problem-tunity.' The writers were able to build character and flesh out the universe." (Totilo 2013).

***Mass Effect 2*: Hypermediated, Diegetic, Passive**

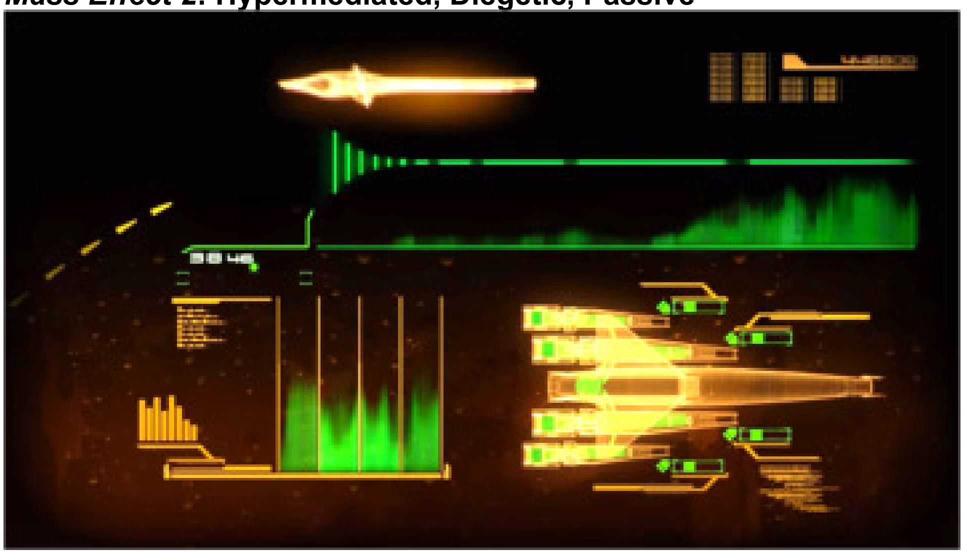

Figure 2: Screenshot of a loading screen in *Mass Effect 2*.

In contrast to its predecessor, *Mass Effect 2* (BioWare 2010) uses passive, hypermediated loading screens to mask loading. As seen in Figure 2, these loading screens have a diegetic quality, with imagery depicting the futuristic ship's diagnostic information like what one of the game's shipboard computers might display. Notably, this information is not legible and has no pedagogic quality.

Given BioWare's own explanation of the enhanced immersion of transparent loading, one might assume that transparent loading is always more desirable than discrete loading screens. However, the *Mass Effect* series demonstrates a popular and critical rejection of transparent loading in favor of the hypermediacy of the loading screen. The elevator sequences of *Mass Effect* gained notoriety for their long durations and were critically panned. The studio responded by switching to hypermediated loading screens in *Mass Effect 2*. Notably, the load times in *Mass Effect 2* were still prevalent and lengthy, with a 35 second delay between transitioning from one section of the player character's spaceship to another. However, unlike the load times in *Mass Effect's* elevators, the durations of *Mass Effect 2's* loading screens did not become a major criticism of the game. The hypermediacy of a loading screen, with its presentation as a mediated experience, may result in gamers accepting loading times as necessary, while the immersive transparency of the elevators caused gamers to scrutinize the narrative logic of the scenario. Game journalist Stephen Totilo explains, "To get from one level to the next [in an elevator], it takes 26 seconds, which is about the same time it takes [the player character's ship] the Normandy to fly from one corner of the galaxy to the other, at least when the game's running on my Xbox 360." He adds that watching the loading screen during this wait was "…way less annoying since you're not watching [the player character] Shepard stand in a slow-moving elevator." (2013).

Resident Evil 2 (1998) and the "Double Zombie Door" Loading Screen: Hypermediated, Diegetic, Passive, Misdirection

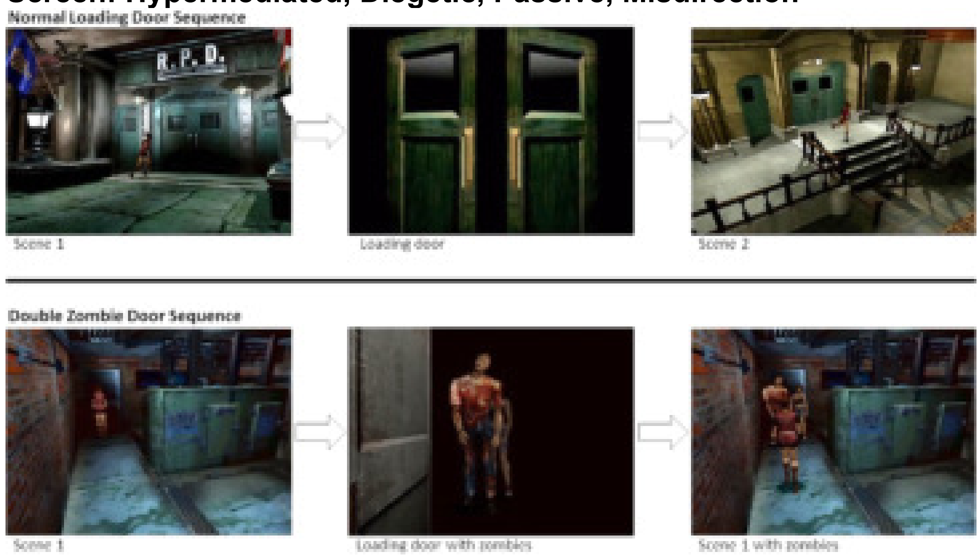

Figure 3: The top row shows the typical loading door sequence of *Resident Evil 2* where a new scene is loaded. The bottom row shows the misdirection of the game's "Double Zombie Door" loading sequence.

Resident Evil 2 (Capcom 1998) continued the series' iconic loading sequences that separate physically adjacent areas. The game's playable environments are prerendered with fixed camera angles. However, its loading sequences feature a black screen with a real-time rendered, 3D representation of the diegetic physical threshold between the two distinctly loaded sections (commonly a doorway or gate). In these loading sequences, an animated camera moves the player's view through the threshold. The sequence employs a first-person perspective, contrasting from the third-person perspective used during playable sections. Typically, a door or gate opening animation and sound effect punctuate the transition. This case is interesting in its blend of simultaneous diegetic and hypermediated qualities. Each instance of loading in the game uses distinct 3D models and sound effects to match the environmental context of the connected scenes. While this diegetic quality maintains an element of narrative grounding and spatial immersion, the vacant black background, jarring change in perspective, and predictable pacing plainly announces to players: "this is a loading screen." The loading screens call so much attention to themselves that fans have created top lists and even fan made versions (Ink Ribbon 2020; Tillian 2013).

Of particular interest is the *Resident Evil 2* loading screen known as the "Double Zombie Door." As described, *Resident Evil* loading screens (prior to the series reimagining with *Resident Evil 4*) follow a distinct formula, and they have a passive quality. This gives players a brief respite to dry their sweaty palms between interactive sections that can be stressful or frightening. Players of the series would have encountered dozens of these loading screens before encountering the Double Zombie Door and understand the formal affordance of this pattern. However, the Double Zombie Door loading sequence interrupts the pattern with two enemy zombies that invade the loading screen. The player is led to believe a new room is loading, but the game is actually loading two new enemies in the current room. The game's designers telegraph safety through the passive loading interface only to subvert player expectations with a scenario that demands quick interaction, creating a surprise scare befitting the game's horror genre. This unique loading sequence shows how designers can leverage the misdirection quality in loading interfaces to create genre-specific moments for play.

Assassin's Creed: Hypermediated, Diegetic, Interactive, Pedagogic

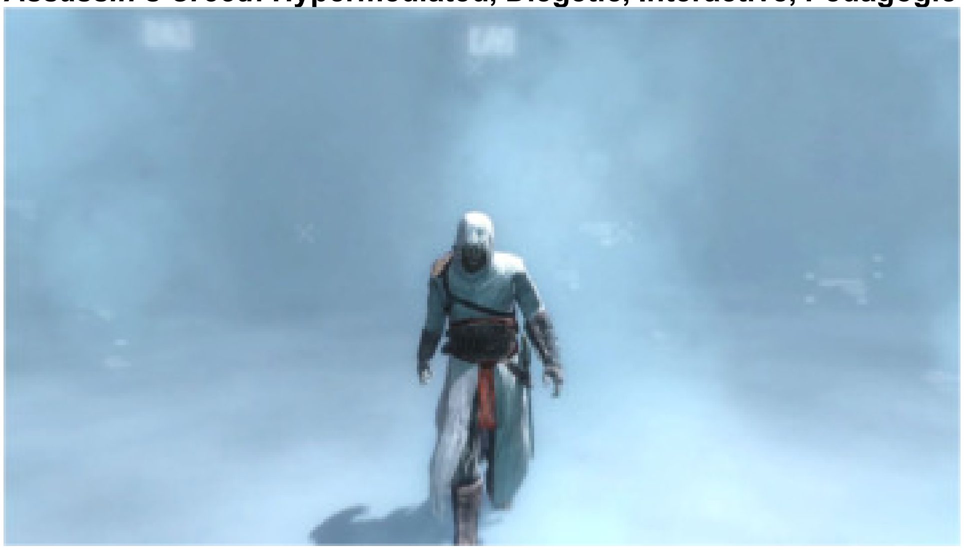

Figure 4: Screenshot of the playable "Animus" loading sequence from Assassin's Creed.

Upon release, *Assassin's Creed* (Ubisoft Montreal 2007) surprised players with its frame story narrative structure, wherein the player character existed in a technologically advanced present and used a virtual reality device to access a simulated version of the 12th century middle eastern setting shown in the game's prerelease marketing materials. The game leveraged this frame story structure in its innovative loading strategy. The game loads as it transitions from the narrative present in Rome to the simulated past in various middle eastern cities. This loading sequence features a unique mix of qualities that has proven memorable to players (Zwiezen 2020; GameCompareCentral 2020). The loading interface is diegetic, as the narrative scenario involves the player character experiencing the immersive simulation loading. The loading sequence is also interactive, giving the player full control over their avatar in a white digital expanse as they wait for the 12th century simulation to load. The player has a new avatar while in the simulation, with new abilities. The playable void expanse of the loading sequence, shown in Figure 4, allows players a safe space to orient themselves and gain familiarity with the new character's controls. This lends *Assassin's Creed's* loading interface a pedagogical quality.

It is more complicated to plot this loading interface on the scale between transparency and hypermediacy. It has the hallmarks of transparency as the game streams the next playable area without interrupting gameplay. However, the narrative context of a computer simulation permits a hypermediated representation of loading without losing the diegetic quality. Despite the game maintaining its normal interactive loop between player input and audiovisual feedback during loading, it opaquely signals the player that loading is occurring to better illustrate its narrative premise. The game leverages hypermediacy in additional areas within its narrative context to explain other loading-related game conventions, such as the boundaries of levels themselves and reloading after failing. *Assassin's Creed* demonstrates how a narrative context of computers and simulations provides game designers an opportunity for sleekly incorporating diegetic loading that evokes the hypermediated loading interface tropes of human-computer interaction and game design.

CONCLUSION

Loading processes and interfaces exist in the limits and margins of video game experiences. They are peripheral; yet they are essential, representing microcosms of problem-solving and innovation in game development. While new technology enables faster loading processes, it simultaneously fuels demand for ever-increasing scale and fidelity that requires more loading. Despite solid state drives promising fast loading, loading screens are still present in games on contemporary consoles like the PlayStation 5 and Xbox Series X. Beyond this, as demonstrated by *Assassin Creed*, loading has become a key signifier of video games, simulations, and human-computer interaction. As a result, loading interfaces may exist in future game experiences regardless of technical requirements for them (reflecting the misdirection quality). Though players commonly resent the interruption of loading, the ubiquitous nature of loading within video games has given rise to its status as a symbol of gaming itself, coinciding with loading interface fan sites and popular media attention. This importance highlights the need for tools for describing, understanding, and analyzing video game loading processes and interfaces.

In this article, we proposed a taxonomy of design qualities of loading interfaces and processes derived from an Interface Study of 139 games, accessible online at www.loadinginterfaces.space. These qualities took the form of four scales: hypermediacy to transparency, non-diegetic to diegetic, passive to interactive, and pedagogic to misdirection. Examining loading interfaces as artifacts within games helps us understand the ethos and telos of their designers and learn from how others have historically navigated the loading dilemma. While we adopted an archaeogaming perspective, it is worth noting that Platform Studies provides a compelling route to understand how a singular family of hardware engages with these very same ideas. Further, Platform Studies would provide an intriguing cultural and societal context for these processes. While such an approach is out of scope for this project, it warrants mentioning and future work. We hope that this framework for understanding and analyzing video game loading interfaces and processes is valuable to game developers and scholars alike.

ENDNOTES

1 The ethos of a loading interface's design points to the authority of the designer and the familiarity of the interface. Telos relates to the purpose of the screen, its goal, and what it is meant to accomplish. A designer's telos, for example, may be to educate.

ACKNOWLEDGEMENTS

We would like to thank Karen Spriggs, the undergraduate student who helped us source and organize our data.

BIBLIOGRAPHY

Adar, Eytan, Desney S Tan, and Jaime Teevan. 2013. "Benevolent Deception in Human Computer Interaction." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1863–72.

Argonaut Games. 1984. *Skyline Attack*. Century Software.

Aycock, John. 2016. *Retrogame Archaeology*. Springer.

Bethesda Game Studios. 2011. *The Elder Scrolls V: Skyrim*. Bethesda Softworks.

Bhojan, Anand, Siang Ping Ng, Joel Ng, and Wei Tsang Ooi. 2020. “CloudyGame: Enabling Cloud Gaming on the Edge with Dynamic Asset Streaming and Shared Game Instances.” *Multimedia Tools and Applications* 79 (43): 32503–23.

BioWare. 2007. *Mass Effect*. Electronic Arts.

———. 2010. *Mass Effect 2*. Electronic Arts.

Bolter, Jay David, and Richard Grusin. 2000. “Immediacy, Hypermediacy, and Remediation.” *Remediation: Understanding New Media*, 20–50.

Bordwell, David, and Kristin Thompson. 1993. *Film Art: An Introduction*. 4th ed. New York: McGraw-Hill. https://archive.org/details/filmartintroduct0000bord/mode/2up.

Brusca, Victor G. 2021. *Introduction to Video Game Engine Development*. Springer.

Capcom. 1998. *Resident Evil 2*. Capcom.

Chen, Jin, Baohua Wu, Margaret Delap, Björn Knutsson, Honghui Lu, and Cristiana Amza. 2005. “Locality Aware Dynamic Load Management for Massively Multiplayer Games.” In *Proceedings of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 289–300.

Chitty, Nelson. n.d. “15 Video Games With The Best Loading Screens.” FandomSpot. Accessed April 7, 2022. https://www.fandomspot.com/best-video-game-loading-screens/.

CI Games. 2017. *Sniper Ghost Warrior 3*. CI Games.

Clark, James H. 1976. “Hierarchical Geometric Models for Visible Surface Algorithms.” *Communications of the ACM* 19 (10): 547–54. https://doi.org/10.1145/360349.360354.

Clover Studio. 2006. *Ōkami*. Capcom.

Consalvo, Mia, and Nathan Dutton. 2006. “Game Analysis: Developing a Methodological Toolkit for the Qualitative Study of Games.” *Game Studies* 6 (1). http://gamestudies.org/0601/articles/consalvo_dutton.

Creative Assembly. *Total War*. Sega. Accessed April 7, 2022. https://www.totalwar.com/.

Cundy, Matt. 2009. “A Celebration of C64 Loading Screens.” GamesRadar+. August 12, 2009. https://www.gamesradar.com/a-celebration-of-c64-loading-screens/.

CuriousMarc. 2017. “IBM System/360 Front Panel.” YouTube. February 1, 2017. https://www.youtube.com/watch?v=WS-WtjwAAO0&ab_channel=CuriousMarc.

Cyberdyne Systems. 1988. *Armalyte*. Thalamus Ltd.

Devil Dog. 2020. “Textures Not Loading.” Steam Forums. August 31, 2020. https://steamcommunity.com/app/870780/discussions/0/2914346777823449475/.

Eidos Montréal. 2016. *Deus Ex: Mankind Divided*. Square Enix.

Fagerholt, Erik, and Magnus Lorentzon. n.d. “Beyond the HUD User Interfaces for Increased Player Immersion in FPS Games.” Accessed January 3, 2023.

Firaxis Games. 2016. *XCOM 2*. 2K Games.

FromSoftware. 2022. *Elden Ring.* Bandai Namco Entertainment.

Game Freak. 2022. *Pokémon Scarlet.* Nintendo, The Pokémon Company.

GameCompareCentral. 2020. “Evolution of Assassin’s Creed Animus Loading Screen (2007-2020).” YouTube. December 4, 2020. https://www.youtube.com/watch?v=MbTXuwblFsc&t=80s.

Genette, Gérard. 1980. *Narrative Discourse: An Essay in Method (Translated by Jane E. Lewin)*. Ithaca, New York: Cornell University Press.

Gilbert, Henry. 2013. “More Games Need Interactive Loading Screens.” GamesRadar+. September 5, 2013. https://www.gamesradar.com/more-games-need-rip-these-interactive-load-screens/.

Good, Owen. 2013. “Here’s Why Sailing Took So Damn Long In Legend Of Zelda: Wind Waker.” Kotaku. June 14, 2013. https://www.kotaku.com.au/2013/06/heres-why-sailing-took-so-damn-long-in-legend-of-zelda-wind-waker/.

Gratton, Kyle. 2022. “Horizon Forbidden West’s Fake PS5 Load Screens Are Wasting Your Time.” ScreenRant. February 22, 2022. https://screenrant.com/horizon-forbidden-west-ps5-fake-loading-screens-pointless/.

Guerrilla Games. 2022. *Horizon Forbidden West*. Sony Interactive Entertainment.

Harper, Christopher. 2022. “What Is Steam Deck Shader Pre-Caching?” Gamers Director. August 12, 2022. https://www.gamersdirector.com/steam/what-is-steam-deck-shader-pre-caching/.

Hayashi, Yoichi. 1998. “US5718632A - Recording Medium, Method of Loading Games Program Code Means, and Games Machine.” United States Patent. February 17, 1998. https://patents.google.com/patent/US5718632.

High Voltage Software. 2009. *The Conduit*. Sega.

———. 2011. *Conduit 2*. Sega.

Higton, Ian. 2017. “Watch: Sniper Ghost Warrior 3 Takes 5 Minutes to Load. Here’s What You Can Do in That Time.” Eurogamer. April 24, 2017. https://www.eurogamer.net/articles/2017-04-24-watch-sniper-ghost-warrior-3-takes-5-minutes-to-load-heres-what-you-can-do-in-that-time.

Iacovides, Ioanna, Anna Cox, Richard Kennedy, Paul Cairns, and Charlene Jennett. 2015. “Removing the HUD: The Impact of Non-Diegetic Game Elements and

Expertise on Player Involvement." *CHI PLAY 2015 - Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, October, 13–22. https://doi.org/10.1145/2793107.2793120.

Iguana Entertainment. 1997. *Turok: Dinosaur Hunter*. Acclaim Entertainment.

Ink Ribbon. 2020. "Top 10 Resident Evil Doors." YouTube. January 1, 2020. https://www.youtube.com/watch?v=fSklp-nlV5Y.

Insomniac Games. 2020. *Marvel's Spider-Man: Miles Morales.* Sony Interactive Entertainment.

Intelligent Systems. 2004. *Paper Mario: The Thousand Year Door.* Nintendo.

Intelligent Systems, and Koei Tecmo. 2019. *Fire Emblem: Three Houses*. Nintendo.

Kang, Yunhee, HoJung Kim, and JungJoo Kang. 2017. "Docker Based Computation Off-Loading for Video Game Based Mobile VR Framework." In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 123–25. IEEE.

Kantor, Jonathan. 2021. "10 Loading Screens So Good You Forgot You Were Waiting To Play." WhatCulture. October 18, 2021. https://whatculture.com/gaming/10-loading-screens-so-good-you-forgot-you-were-waiting-to-play.

Karhulahti, Veli-Matti. 2012. "Double Fine Adventure and the Double Hermeneutic Videogame." *Proceedings of the 4th International Conference on Fun and Games*. https://www.academia.edu/1928502/Double_Fine_Adventure_and_the_Double_Hermeneutic_Videogame.

Keogh, Brendan. 2014. "Across Worlds and Bodies: Criticism in the Age of Video Games." *Journal of Games Criticism* 1 (1). http://www.gamescriticism.org.

Laurel, Brenda. 2005. "'Piercing the Spectacle.'" In *Game Design Reader: A Rules of Play Anthology*, edited by Katie Salen Tekinbaş and Eric Zimmerman. Cambridge, MA: MIT Press.

———. 2014. *Computers as Theatre*. 2nd ed. Upper Saddle River, NJ: Addison-Wesley.

Let's Talk Retro. 2017. "Our Top 10 Commodore 64 Loading Screens." YouTube. February 18, 2017. https://www.youtube.com/watch?v=09LBqtf7xuQ&t=19s.

"Loading Screen Games." n.d. Fandom Ōkami Wiki. Accessed April 7, 2022. https://okami.fandom.com/wiki/Loading_screen_games.

"Loading Screens." n.d. Fandom Fortnite Wiki. Accessed February 1, 2022. https://fortnite.fandom.com/wiki/Loading_Screens.

Mateas, Michael, and Andrew Stern. 2003. "Façade: An Experiment in Building a Fully-Realized Interactive Drama." In *Game Developers Conference*, 2:4–8.

MercurySteam, and Nintendo EPD. 2021. *Metroid Dread*. Nintendo.

"Metroid Prime/Gamecube vs. Wii Loading Discrepancies." 2020. Metroid Prime Speedrunning Wiki. October 6, 2020. https://wiki.metroidprime.run/wiki/Metroid_Prime/Gamecube_vs._Wii_Loading_Discrepancies.

Midway Studios San Diego. 1997. *Doom 64*. Midway Home Entertainment.

Moshix. 2017. "Running the IBM MVT Operating System - M7." YouTube. June 24, 2017. https://www.youtube.com/watch?v=ZtK_p06br9k&t=785s&ab_channel=moshix.

Murray, Janet. 2017. "Agency." In *Hamlet on the Holodeck*, 2nd ed., 159–89. Cambridge: MIT Press.

Nielsen, Jakob. 1994. "Heuristic Evaluation." In *Usability Inspection Methods*, edited by Jakob Nielsen and Robert L Mack. New York: John Wiley & Sons.

Nielsen, Jakob, and Rolf Molich. 1990. "Heuristic Evaluation of User Interfaces." *Conference on Human Factors in Computing Systems - Proceedings*, March, 249–56. https://doi.org/10.1145/97243.97281.

Nintendo EAD. 1998. *The Legend of Zelda: Ocarina of Time*. Nintendo.

———. 2002. *The Legend of Zelda: Wind Waker*. Nintendo.

Onaretrotip. 2021. "Playable Loading Screens - A Brief History - YouTube." YouTube. November 16, 2021. https://www.youtube.com/watch?v=d2tOnf9pa4c&ab_channel=onaretrotip.

Palomba, Anthony. 2016. "Brand Personalities of Video Game Consoles." *American Communication Journal* 18 (1).

Paradox Development Studio. 2013. *Europa Universalis IV*. Paradox Interactive.

Pickford, Ste. 2004. "Loading Screens." Zee-3 Digital Publishing Blog. June 28, 2004. http://www.zee-3.com/pickfordbros/archive/loadingscreens.php.

Rare. 1997. *Goldeneye 007*. Nintendo.

Reinhard, Andrew. 2018. *Archaeogaming: An Introduction to Archaeology in and of Video Games*. Berghahn Books.

Respawn Entertainment. 2019. *Star Wars Jedi: Fallen Order*. Electronic Arts.

Retro Studios. 2002. *Metroid Prime*. Nintendo.

Sinha, Ravi. 2017. "15 Games That Have Frustratingly Long Loading Times." GamingBolt. July 24, 2017. https://gamingbolt.com/15-games-that-have-frustratingly-long-loading-times/6.

Stonehouse, Anthony. 2014. "User Interface Design in Video Games." Game Developer. February 2014. https://www.gamedeveloper.com/design/user-interface-design-in-video-games.

Strat. 2017. “Inside Look at the Certification Process.” Gems of War Community. 2017. https://community.gemsofwar.com/t/inside-look-at-the-certification-process/28759.

Swink, Steve. 2009. *Game Feel: A Game Designer’s Guide to Virtual Sensation*. Elsevier/Morgan Kaufmann.

The Behemoth. 2013. *BattleBlock Theater*. Microsoft Game Studios.

TheRittyl. 2014. “Games with Entertaining Loading Screens.” Reddit. July 5, 2014. https://www.reddit.com/r/gaming/comments/29w8iq/games_with_entertaining_loading_screens/.

Tillian. 2013. “RESIDENT EVIL DOOR ANIMATIONS [FAN MADE].” YouTube. October 22, 2013. https://www.youtube.com/watch?v=0_AVI1WZs9Q.

Totilo, Stephen. 2013. “An Oddly Long, Banter-Filled History of Mass Effect Elevators.” Kotaku. April 12, 2013. https://kotaku.com/an-oddly-long-banter-filled-history-of-mass-effect-ele-472630862.

Traveller’s Tales. 2001. *Crash Bandicoot: The Wrath of Cortex*. Universal Interactive Konami.

Ubisoft Monpellier. 2013. *Rayman Legends*. Ubisoft.

Ubisoft Montreal. 2007. *Assassin’s Creed*. Ubisoft.

Unity Technologies. 2022. “Level of Detail (LOD) for Meshes.” Unity Manual. November 17, 2022. https://docs.unity.cn/2023.1/Documentation/Manual/LevelOfDetail.html.

Valve. 1998. *Half-Life*. Sierra Studios.

Visual Concepts. 2020. *NBA 2k21*. 2k Sports.

Waldrop, Cameron. 2022. “Elden Ring: How to Wait and Pass Time.” Twinfinite. February 28, 2022. https://twinfinite.net/2022/02/elden-ring-wait-pass-time-how/.

WCK619. 2012. “Low Textures and Pop In? - Gears of War 3.” GameFAQs Forum. 2012. https://gamefaqs.gamespot.com/boards/991468-gears-of-war-3/60458705.

Wittkamp, Steve. n.d. “The Ultimate Guide to Which Version of Chrono Trigger You Should Play.” Geek to Geek Media. Accessed April 7, 2022. https://geektogeekmedia.com/geekery/best-version-of-chrono-trigger/.

Yager Development. 2012. *Spec Ops: The Line.* 2k Games.

Yang, Robert. 2013. “Let’s Play: The First Room of Anomalous Materials in Half-Life 1.” YouTube. April 29, 2013. https://www.youtube.com/watch?v=iGH41AgcjP4&t=253s.

Zwiezen, Zack. 2020. “Running Around During Loading Screens Is An Important Part Of Assassin’s Creed.” Kotaku. December 13, 2020. https://kotaku.com/running-around-during-loading-screens-is-an-important-p-1844170039.