

“Weekends became something other people did”: Understanding and intervening in the habitus of video game crunch

Dr. Amanda C. Cote, acote@uoregon.edu (Corresponding Author)

Mr. Brandon Harris, bharri10@uoregon.edu

University of Oregon

School of Journalism and Communication

This is an Accepted Manuscript of an article published by Sage in *Convergence: The International Journal of Research into New Media Technologies* on March 26, 2020, available online: <https://journals.sagepub.com/doi/full/10.1177/1354856520913865>.

Abstract

“Crunch”— a period of unpaid overtime meant to speed up lagging projects— is a common labor practice in the video game industry and persists despite many costs to developers. To understand why, we conducted a critical discourse analysis of *Game Developer* magazine (2000-2010) to explore how industry members perceive and discuss gamework 1) in a publication for developers, by developers and 2) during the first decade in which serious conversations about labor emerge in the games industry. Our analysis found that many gameworkers treat crunch as “inevitable” due to three specific themes: games as an unmanageable creative industry, an anti-corporate ethos, and a stereotypical developer identity based on passion and perfectionism. These constructions— combined with the industry’s project-based nature and cultures of passion and secrecy— build crunch into the *habitus* of gamework, helping reproduce exploitative labor practices. However, habitus can and does change over time, providing interested employees, companies, and labor organizers a means to intercede in existing work practices. We suggest a multi-pronged intervention that could build a healthier, more sustainable habitus of gamework.

Keywords

Video games, media industries, labor practices, habitus, critical discourse analysis, press analysis, game studies, crunch

Introduction

“All-nighters and seven-day weeks are a fool’s game. Nothing superb ever gets done by overtired people whose lives have been turned upside down.” (January 2002, pg. 38)

Game Developer offered this advice to readers in 2002 as part of an article on managing large-scale game production. Despite this, “crunch”, or a period of extended working hours meant to speed up lagging projects, remains a common practice in the video game industry. Crunch is supposed to be temporary, but companies often require protracted periods of unpaid overtime even when projects have consistently met deadlines (Hoffman, 2004; Rockstar Spouse, 2010). This practice has become, for many developers, a taken-for-granted part of the job (Consalvo, 2008; Legault & Weststar, 2016; Peticca-Harris et al., 2015).

Although researchers, industry members, and game-related press have all pointed out the costs of extended working hours, and companies have faced expensive class-action lawsuits due to this practice, crunch continues. Because recognizing the costs of crunch, discussing quality of life (QoL), and bringing lawsuits has not been enough to alter work practices, other factors must be at play. Existing research in this area has primarily attributed crunch’s persistence to the games industry’s project-based nature (e.g. Legault & Weststar, 2016; Peticca-Harris et al., 2015) as well as to its culture of secrecy (O’Donnell, 2014), whereby companies’ non-disclosure agreements are so wide-ranging that industry members hesitate to share even general best practices developed while working on a project.

In this article, we argue that these explanations are correct but not sufficient to explain crunch’s persistence fully. Rather, these factors combine with industrial discourse to create an overall *habitus* in which crunch is normalized and taken for granted. We defend this claim

through a critical discourse analysis of video game press directed at gameworkers, which reveals that developers are not only conditioned to accept and perpetuate crunch by the project-based, secretive nature of the games industry, but also by discourses that present crunch as inevitable, that forward an anti-corporate view of the industry, and that normalize certain understandings of developer identity. These all create a “particular class of conditions of existence” (Bourdieu, 1990, p. 53) in which crunch becomes a logical choice, building a framework for gameworkers’ continued reliance on it.

Our analysis is deliberately historical and longitudinal in nature, assessing issues of *Game Developer* from 2000-2010. Discussions of crunch and labor practices have tended to prioritize specific, supposedly watershed events, rather than addressing work practices holistically (O’Donnell, 2014). This sample provides a more comprehensive view regarding if or how understandings of labor, QoL, and crunch change over time. Further, 2000-2010 represents a theoretically interesting moment, as it is when the International Game Developers Association (IGDA), the industry’s largest membership group, first formed a QoL Committee to address issues around crunch. Thus, these years form the foundation upon which any future interventions into crunch must build.

We find that developers at all levels are inundated with discourses that normalize crunch, to the extent that it becomes a standard part of gamework’s habitus. For each conversation about improving QoL, there are multiple, more prominent discourses that portray crunch as an inevitable part of the game development process. As such, developers make their decisions about labor practices in the face of ingrained beliefs that support and often reproduce crunch as a practice. This is both unfortunate and potentially positive in that, while some contributors to crunch, like games’ project-based nature, are unlikely to change, habitus can and does shift over

time. As such, it provides a location through which intervention into games' dangerous labor practices can occur.

Literature Review

What is the “game industry”?

Before addressing how the video game industry understands labor practices, we must clarify what we mean by “the video game industry”. While it is common to discuss games collectively, “the industry” can best be understood as a loose assemblage of different *industries*, each with their own goals, audiences, and practices (Keogh 2015; Kerr, 2017). This project focuses on the industry segment known as triple-A or AAA. This sector includes gaming's largest companies, such as Electronic Arts, Bungie, and Rockstar, which produce high-tech, high-budget games for consoles and PCs. As such, while other segments of the industry are growing, AAA remains both the ideological foundation of the video game industry, as it is what most people think of first when they think “video games”, and the driver behind many industrial practices due to its reach and power.

AAA has also been at the heart of past QoL issues. For instance, both Electronic Arts and Rockstar have been called out by developers' families for overexerting their employees, crunching even though projects were on schedule, and under-compensating workers for overtime (Hoffman, 2004; Rockstar Spouse, 2010). Although these individuals' public letters to their spouses' employers helped lead to a series of class-action lawsuits that awarded employees unpaid overtime, the problems they raised have not subsided. Recent updates show similar issues (Cross, 2018; Schreier, 2014; 2016; 2018; Sinclair, 2012)¹. Thus, AAA labor practices appear

¹ Some developers refute claims of crunch (e.g. Good, 2018) and some evidence shows the practice declining in frequency (Legault & Weststar, 2016). However, crunch-related stories are still common enough to be concerning, and the intensity and duration of crunch periods remain high (ibid.)

sorely in need of an intervention. Further, as AAA is at the heart of the industry, changes here could resonate throughout other industry segments as well.

Crunch and its Costs

“Crunch” refers to labor practices that commonly force developers, artists, programmers, and others working on a software project to work extensive, unpaid overtime. This practice has both individual and systemic effects on the industry. As established in industries like software development, extended overtime leads to exhaustion, mental and physical health problems, stress on families, and a high rate of burnout for employees (Fujigaki, 1996; Kuutila et al., 2018; Nishikitani et al., 2005; Wright and Cropanzano, 2000). In games, for instance, most developers leave the industry in three to nine years, stating they “want a better quality of life” or are “burned out” (Edwards et al., 2014, p. 18). Overall, many employees find the industry’s working practices unsustainable². Diverse contributors have brought attention to the costs of crunch in the hopes of changing the industry’s reliance on precarious work practices (e.g. Deuze et al., 2007; Johns, 2006; Kerr, 2011; Schreier, 2016; Legault & Weststar 2016). We begin by describing these factors before discussing their influence on gamework’s habitus.

The “Iron Triangle”, Secrecy, and Passion

Within AAA, video game production involves several steps and players³. A game begins in preproduction, where a small team of artists, designers, and engineers hammer out its main concept, storyline, and aesthetic. This team is generally employed by the studio producing the game. Once initial details are settled, developers seek funding for full-scale production.

² Although we lack the space to address it here, it is worth noting that burnout disproportionately affects older and female employees (Peticca-Harris et al., 2015). As the game industry is extremely male-dominated (Weststar, O’Meara & Legault, 2018), labor practices’ contributions to this inequality should be addressed.

³ This explanation of game development emerges from our magazine analysis, but interested readers can find further details on the process in Dyer-Witthford & de Peuter (2006), Johns (2006), Keogh (2015), Kerr (2006, 2011, 2017), and O’Donnell (2014).

Alternative or independently funded games are possible, but AAA games are more commonly bankrolled through a publishing company that provides financing and marketing and distribution assistance, in exchange for a degree of control over production and rights over the finished game. Because AAA games have very high production budgets, only a small number of successful publishers can meet these funding needs, which often grants them incredible power in the industry (more on this below). Funded games move to the production stage, where workers design levels, create art assets, flesh out storylines and dialogues, and more. Marketing and promotion also begin at this time, with developers creating trailers and demos for advertisements and industry conventions. Finally, the game is released and, if successful, its profits help fund preproduction on the studio's next endeavor.

In reality, development is never quite this smooth, and different processes change order, overlap, and lengthen or accelerate as needed. This means that AAA emphasizes post-industrial, flexible work patterns. As preproduction requires only a small, core team of creators, while production demands many specialized roles, development is characterized by a system of uneven working hours, just-in-time production, and contract-based employment (Deuze et al., 2007; Dyer-Witthford and de Peuter, 2006; Huntemann, 2010; Legault & Weststar, 2016). Careers are inherently uncertain, with studios hiring or firing in line with the skill sets needed for each step of a project (Deuze et al., 2007; Huntemann, 2010). This precarity is one of the factors that contributes to crunch. Developers are always aware of the need to get the next job, and working overtime can build their reputation as a team player. Because jobs are often obtained through informal networks, reputation can be essential to career success (Legault & Weststar, 2016; Peticca-Harris et al., 2015).

The “Iron Triangle”, or the combined influence of budget concerns, deadlines, and product specifications (Peticca-Harris et al., 2015; Legault & Weststar, 2016), dictates much of the production process in project-based industries like game development. In AAA, publishers often set these three factors, while gameworkers’ “task is to complete the project to the client’s satisfaction by any means necessary” (Peticca-Harris et al., 2015, p. 573). Developers must meet frequent milestones to reassure their publishers that the project is on schedule to secure continued funding. Losing funding can result in mid-development bankruptcy; fear of this contributes to unbalanced relationships where “publishing companies exercise their position of power... at the expense of developers’ time and effort” (O’Donnell, 2014, p. 192). Publishers may request that developers work on associated materials, such as trailers and demos, and can even require that features be added to or removed from the game. Studios must often rethink their schedules, move team members around, or, unsurprisingly, move into extended crunch to account for these ongoing demands. Under the Iron Triangle, where workers are dependent on publishers for their company’s success and survival, employees’ “non-work time becomes a reserve of emergency time where the emergency is defined by external deadlines, demands or changes” (Legault & Weststar, 2016, p. 88). Given the costs inherent in losing funding, crunch becomes a preferable alternative.

O’Donnell (2014) argues that the game industry’s culture of secrecy intensifies these issues. Because many companies defend their projects with complicated non-disclosure agreements, gameworkers are hesitant to share details about their work processes. As a result, gameworkers end up solving the same problems over and over again due to the lack of shared knowledge about game technologies, programming techniques, management strategies, and more. This wastes significant amounts of time, and it also helps perpetuate a system in which

real improvement is impossible, as the conversations around which change could happen simply do not occur.

The above factors act as a “soft-coercive system” (Whitson, 2013, p. 124) where economic constraints and the nature of gamework encourage developers to work overtime; developers themselves can then exacerbate the problem. The games industry is filled with passionate people who have grown up playing video games and love being able to develop them for a living (Consalvo, 2008; Deuze et al., 2007; Dyer-Witthford & de Peuter, 2006; Harvey & Shepherd, 2016; Kerr, 2011). This passion can trap developers into unhealthy relationships with work. As a participant told O’Donnell (2014), “There is a lot of history in the game industry as rooted in nerds in garages and doing it for the love of it. That has just filtered down into places expecting employees to just go above and beyond without question, all the time” (p. 136). The interviewee described this as a “culture of overtime,” indicating crunch’s taken-for-granted nature. Passion also leads to a greater supply of employees than demand, making it easy for companies to replace workers, increase competition for positions, and decrease salaries (Hoffman, 2004). Workers who manage to get jobs, especially at large studios, may consider themselves lucky to be employed, limiting their ability to resist unrealistic demands.

Finally, employees have few avenues for addressing issues other than public complaints and class-action lawsuits. In their 2014 QoL survey, the IGDA found that most employees were only able to raise concerns “directly to their managers” (Edwards et al., 2014, p. 31). Over 40% stated that they would rather have the ability to approach issues collectively through a union or professional organization (p. 31). Despite support for potential unionization, though, little serious discussion around the possibility has occurred until very recently. While employees recognize unions’ potential benefits, they see significant barriers to creating one. Because crunch, for

instance, remains an industry standard, employees who want to speak out against long, unrealistic hours fear that they will suffer repercussions for doing so (Sinclair, 2012). But without collective organization or changes to the industry, working practices remain the same.

The Role of *Habitus*

Habitus refers to “systems of durable, transposable dispositions, structured structures predisposed to function as structuring structures” (Bourdieu, 1990, p. 53). Put more simply, habitus describes how individuals and social groups internalize their backgrounds, experiences, and environments, then subconsciously draw on these to interpret new situations and make decisions. In this way, social structures can become self-perpetuating, as new structures are always built through the lens of existing ones. Particular beliefs, and their resultant actions, come to “go without saying” (ibid, p. 71) due to their naturalization through individuals’ shared environments and habits. In the case of crunch, we can see habitus functioning in the contributing factors discussed above. The cultures of secrecy and overwork O’Donnell describes, as well as how the Iron Triangle encourages specific labor practices, all contribute to “a particular class of objective regularities” (Bourdieu, 1990, p. 55) or a framework in which crunch becomes a logical solution to a persistent problem.

At the same time, habitus may be *determining*, but it is not *determinant*; it does not completely dictate how decisions are made nor is it unchanging. Habitus persists “to the extent (*and only to the extent*) that the objective structures of which they are the product are prolonged in the structures within which they function” (Bourdieu, 1977, p. 73, emphasis original). In this article, we recognize that games’ project-based nature, one of the key contributors to crunch, is unlikely to change. However, by examining other aspects of crunch culture, such as the discourses that circulate within the game industry, we can denaturalize that which habitus current naturalizes, preventing the structured structures of crunch from becoming *structuring* structures

and highlighting other potential solutions. In doing so, we can perhaps change the taken-for-granted nature of overtime, secrecy, and crunch to instead promote better QoL for gameworkers.

Methods

We conducted an inductive, critical discourse analysis (Fairclough, 2012) of all *Game Developer* (GD) issues published from 2000-2010 (125 total). While game labor has been investigated through observation (O'Donnell, 2014) as well as surveys and interviews (e.g. Legault & Weststar, 2016), discourse analysis extends this conversation by showing how developers speak about labor and crunch in public-facing, but still targeted, venues. GD is widely available, but it is a publication by developers, for developers. Articles focus on the day-to-day work of producing games, and current gameworkers pen most columns. GD thus provides a naturalistic view of developers' perspectives on crunch and can help us understand the overall environment in which games are produced. Assessing the magazine over time also reveals if/how these discourses change. As O'Donnell (2014) indicates, conversations around crunch often focus on individual events, which can draw "analytic attention away from the broader issue" (p. 159). Our approach responds to O'Donnell's call for more holistic industrial research.

We selected 2000-2010 because these years represent a significant decade of industrial growth. Developers grappled with rapid technological changes, increasing expectations for game and graphic quality, and the industry's ongoing professionalization. Further, the early 2000s saw extremely high rates of crunch (Legault & Weststar, 2016) as well as significant backlashes against these (Hoffman, 2004; Rockstar Spouse, 2010). Finally, although unionization efforts ramped up in the late 2010s⁴, the IGDA instituted their Quality of Life (QoL) Committee and began openly addressing labor issues in 2004 (Bonds et al., 2004). Thus, if the seeds of

⁴ E.g. the Game Workers Unite movement.

unionization exist in the industry, they were planted in the early 2000s. Analyzing magazine issues from 2000-2010 allows us to assess the long-term foundation any continuing QoL movements will have to build on and/or overcome.

Given our overall focus on labor, critical discourse analysis is appropriate as it “does not simply describe existing realities but also evaluates them, assesses the extent to which they match up to various values, which are taken (more or less contentiously) to be fundamental for just or decent societies” (Fairclough, 2012, p. 9). This study intends both to explain how the industry approaches questions of labor and to provide suggestions for overcoming systemic problems. Further, CDA forefronts developers’ own voices, but it also provides space to critically assess what those voices, and patterns within them, mean more broadly.

We read each issue and used the NVivo qualitative analysis program to identify articles’ component themes, then generated relevant categories from these smaller units (for details on this process, see Lindlof and Taylor, 2002). We assessed each issue fully but paid particular attention to the magazine’s “Postmortem” articles, where developers laid out five things that went right and five things that went wrong during a game’s creation. These articles provide detailed and often quite personal insights into the process of creating a game and are also emblematic of broader industrial discourses. The ways in which developers discuss their successes and failures reveals how they understand development overall; if postmortems frequently mention crunch but rarely present meaningful solutions, that demonstrates how crunch becomes ingrained into the industry’s habitus. However, if companies avoid crunch consistently, that is also educational. As a result, the conclusions of this piece emerge primarily from postmortems, but they also take into account other articles like developer interviews, reflections

on common talking points mentioned throughout the years of postmortems, and editorials on crunch and QoL.

Results

Analysis revealed three main discourses that excuse and normalize crunch: 1) perceptions of games as an unmanageable creative endeavor, 2) an overall anti-corporate ethos, and 3) the characteristics and identities of developers. These themes confirm many previous research findings, but the dataset also extends these through its focus on how developers discuss issues amongst themselves, rather than with a researcher, and through its longitudinal nature. That these discourses remained consistent across a decade further shows how deeply crunch has been ingrained into this industry. Crunch is part of a complicated habitus that will need significant intervention if better QoL is to be obtained.

Games as Unmanageable

GD strongly espoused “the widely held belief that game development is different, just unmanageable” (O’Donnell, 2014, p. 139) across the decade. Developers attributed this unmanageability to three factors— games' status as a creative enterprise, their reliance on ever-changing technology, and publishers’ influence over the development process— and used these to explain why projects went off schedule, or even to justify under planning the project in the first place. These themes strongly align with previous researchers’ discussion of the Iron Triangle, or how budget concerns, deadlines, and product specifications lead companies to rely on workers’ overtime hours as an emergency resource (Legault & Weststar, 2016; Peticca-Harris et al., 2015). However, our analysis extends work on the Iron Triangle by showing how it becomes a structuring structure, normalizing crunch as an industrial practice. In other words, although crunch is only “one strategy among other possible strategies” (Bourdieu, 1977, p. 73),

developers' past use of crunch as a solution to failed scheduling normalized their continued reliance on this practice. Scheduling problems and crunch came to be seen as regular parts of the industrial process, rather than an issue.

Many articles and postmortems reiterated the traditional rhetoric that game development as a creative endeavor was impossible to manage. It was common to find passages like “you cannot plan when inspiration will strike” (June 2002, p. 32) or “Game development was once described to me as building a house on top of a moving train. In space. With dinosaurs. It’s never an easy process, and you have to learn to deal with the unexpected” (January 2010, p. 8). These comments assume that it is impossible to accurately schedule the creative innovation a successful game requires. The data did show that game features required considerable time to design, implement, and test (described further below). However, it matters less whether games are actually unmanageable or not; rather, because developers *perceive* them as such, this belief proceeds to structure how they make their decisions and the ways in which crunch often goes unquestioned. For instance, the discourse that games were “unmanageable” often encouraged gameworkers to under plan, as they felt setting a detailed schedule would be wasted effort when it was only going to change.

Unfortunately, this cavalier approach to making games required management to leverage their employees' labor to make up the difference. Teams that saw development as unmanageable tended to solve problems using crunch instead of finding alternative solutions. Take the 2009 postmortem for *Tomb Raider: Underworld*, where creative director Eric Lindstrom highlighted clear issues that occurred throughout the production process, including scheduling failures, an overly ambitious design, and challenges working with new technology. However, by the end of the piece, he had dismissed these issues as “the usual delays and schedule slips” (June/July 2009,

p. 38) and wrote, “Creative endeavor by its very nature is chaotic” (p. 40). Lindstrom also pointed out that, although the Tomb Raider team crunched on this project, it was less than on previous projects, which he described as a win. This shows how deeply ingrained crunch is as a structuring structure, in that developers see crunch as inevitable and focus simply on minimizing rather than eliminating it. Further, crunch’s taken-for-granted nature means that little is done to address its contributing causes, even when developers have identified other issues that could be remedied.

Developers also excused scheduling issues and explained away crunch due to the challenge of producing games for constantly improving technologies, including new console generations, growing development budgets and teams, and different demands for games. For instance, many articles and postmortems emphasized how games’ rapid innovation altered their standards of quality. GD often promoted near-cinematic realism in graphics and animation as the goal for game development, which put extensive pressure on gameworkers. Artists and art directors, in particular, felt that the demands of their jobs were rapidly expanding “as games begin to deliver the visual quality of movies and television and also approximate the visceral experience of live theater” (April 2003, p. 20). Postmortems often showed that these demands were inadequately accounted for during hiring, meaning artists worked long hours to meet new standards.

Developers also generally produce games for proprietary hardware systems that each offer different tech and performance specifications, delivery formats, and more. Gameworkers can rarely rest on their laurels; even if they are developing sequels to existing properties, they must learn a new system and much of the game’s background work is therefore different. Adding further challenges, a new platform’s specs and format could change multiple times before its

actual release, meaning developers felt they “worked under a lot of uncertainties. When would the PSP launch? Were these ‘final’ PSP specs really final?” (May 2006, p. 28). These types of questions were common in the “What went wrong” sections of postmortems. Even after console specs solidified, switching technologies often caused unexpected issues. For instance, moving from the original PlayStation to the PlayStation 2 meant developers switched from CD-ROM games to DVD games, requiring both new technology and time. As the *Ratchet and Clank* team testified in one postmortem (June 2003), simply creating a new game copy became a multi-hour process, and that was if everything went correctly. Unexpected technological hurdles threw off schedules, leading developers to turn to crunch as a solution.

Developers used these discourses around the difficulties of fully planning a game’s production to justify scheduling failures and crunch throughout the decade. At the same time, these challenges are nothing new; developers even tend to discuss technological changes similarly across time. For instance, discourses around the transition from the PlayStation 2 to the PlayStation 3 strongly resemble those around the shift from the Super Nintendo Entertainment System to the Nintendo 64. Each demanded higher quality graphics, more expansive storylines and art styles, and new approaches to memory management. Therefore, experienced gameworkers should, over time, learn to manage the vagaries of testing new tech and schedule accordingly. The fact that they don’t seem to do so reflects not only how games’ culture of secrecy (O’Donnell, 2014) makes it difficult to develop standardized practices, but also how the perception of games as unmanageable creates persistent issues with crunch.

Finally, developers also under planned their projects because they felt powerless in the face of publishers’ influence. As described above, publishers provide development studios with funding, establish and execute marketing strategies, and set milestones throughout the

development cycle. It was this latter trend that the dataset painted as problematic, especially when publishers dictate a games' platforms and release schedule. In particular, developers struggled when "a publisher demands the widest possible release imaginable" (May 2003, p. 28), necessitating laborious translation processes, or when "tasked with simultaneously releasing worldwide on both the Xbox 360 and PlayStation 3, something the studio had never before accomplished" (April 2009, p. 18). Releasing a game on multiple platforms and in multiple languages boosts its global appeal (and profits), but it simultaneously requires considerable developer labor. Developers often viewed crunch as a preferable alternative to failing to meet publishers' expectations. Publisher demands thus overlapped with and exacerbated the technological issues described above, contributing to the pervasiveness of crunch, while also leaving developers uninvested in managing schedules themselves since they felt they had little say in the overall process.

Anti-Corporate Ethos

This leads directly into another persistent theme throughout GD— the existence of an anti-corporate ethos within the industry that ultimately weaved crunch into the habitus of game development. By "anti-corporate ethos", we refer to the ways in which developers' nostalgia for games' early era of garage development led them to reject organizational structures and systems. This is not to say that the developers were not already involved in corporate capitalist systems; rather, we argue that they resisted seeing themselves as part of these systems, preferring instead to hold on to gaming's hobbyist past. This created a form of cognitive dissonance whereby how developers viewed themselves varied dramatically from their actual role. This disconnect often acted as a form of self-sabotage in that it led workers to forego many interventions that could reduce the need for crunch and improve their QoL.

The dataset collectively showed a reluctance from developers, artists, and coders to embrace formal business structures. Over time, it became evident that gameworkers idealized the industry's origins in hobbyist programming and waxed nostalgic over "the days of the garage developer" (June 2002, p. 32) or "the garage atmosphere most people in the game industry thought died out a decade ago" (May 2005, p. 28). GD even celebrated the fact that many developers lacked formal education in their field as evidence of the industry's continued openness to "entrepreneurial, DIY culture" (April 2008, p. 14). In contrast, developers framed corporate policies as anathema to the innovation on which good games relied, arguing they would "torpedo creativity" (June 2002, p. 32), and rejected these despite games' growing size and significance. This bred resistance to corporate behaviors and practices.

Simultaneously, developers revealed that a lack of business structures negatively affected many games' development. Small developer Irrational Games, for instance, found that as they expanded, they needed to institute set work hours and rules regarding the use of office technology. Company co-founder Jonathan Chey wrote, "One developer who prefers to start work in the afternoon, for example, or another who spends the afternoon playing LAN games, can cause problems not only through personal behavior but also by undermining other people's work practices" (May 2002, p. 47). Many studios also failed to coordinate inter-departmental communications effectively, often leading to wasted work, and developers even highlighted multiple instances where they failed to hire in line with the needs of different project stages. For example, when TimeGate Studios was expanding to become a multi-project company, the production team severely underestimated the human capital needed to make the shift effectively. Their postmortem said, "We might have had an easier time if we had expanded the team prior to beginning the projects." (November 2004, pg. 37). Four out of five "What Went Wrong" points

in this postmortem dealt with management and corporate structure issues, indicating how developers struggle to see themselves and function as professionals.

Examples such as these illustrate that developing a game and running a business are entirely different skills, and developers largely seem to concentrate on the former. While some managers proved the exception by addressing organizational shortcomings, the data showed little to suggest any widespread efforts to eliminate crunch⁵. This is unfortunate, as the few moments within the magazine where developers employ corporate processes show many benefits. For instance, when two mid-sized studios— Backbone Entertainment and The Collective— merged in March 2005, the studio postmortem explained, “We realized that as much as we all respect our ‘garage band’ roots, publishers are not looking for garage band developers these days” (June/July 2005, p. 29). Becoming a larger, more organized company gave them access to greater resources, more negotiating power with publishers, and a foundation for developing unique new properties. In other instances, developers simply stated things like “all I can say is that things worked a lot better for us when we stopped working silly hours and started treating video game development like a proper job” (March 2006, p. 49). These examples show that developers who embraced some degree of structure clearly benefited. However, these instances were by far the exception to the rule; the vast majority of postmortems showed developers struggling with organization and corporate policies, as how they viewed their work did not align with the work itself. Therefore, GD discourses from this period show how an overall anti-corporate ethos was built into game development’s habitus in lieu of more organized practices.

Developer Identity

⁵ A few managers began to discuss the benefits of adjusting design methodologies and scheduling around 2008, likely in response to the recession. We discuss this in the conclusion.

Finally, in line with previous findings (e.g. Consalvo, 2008; Deuze et al., 2007; Dyer-Witthoford & de Peuter, 2006; Harvey & Shepherd, 2016; Kerr, 2011), discourses around developer identity contributed to crunch's persistence. Throughout the sample decade, developers were portrayed as passionate and committed to the industry "because they love the work" (May 2002, p. 47).

Accordingly, GD articles often framed long hours and excessive overtime as an employee's *choice*, rather than a company requirement or a failing of the whole industrial system. In one of the most extreme examples, Cyber Connect 2 CEO Hiroshi Matsuyama described the self-imposed crunch he employed when he started in the industry. He stated,

"For three years I basically lived at work. I paid rent on an apartment that was empty; that I never lived at. Because of the fact that I was an amateur entering into a world where there were other professionals working, I had to work three times as hard as everybody else... So I was there all night long. But, what I would do is, I would stick my head under the kitchenette's warm water spout, and wash my hair there, and strip down and take a towel and take a sponge bath. And since there's nobody there, I can completely just take it all off. But that's just how I lived, day after day!" (May 2009, p. 5)

Most employees did not live at work for three years, but late hours and overnights— framed as an employee's choice— were common. For instance, when the *Unreal Tournament* team offered their workers flexible scheduling, "The open hours often saw team members working a 24-hour day, sleeping on a couch for six hours, and then working another 24-hour day" (May 2000, p. 50). Comparable anecdotes appear constantly throughout the decade. Crunch thus becomes framed as a natural extension of an employee's love of games and their drive for perfection. In

other words, the data set consistently shows the perception of developers as passionate and ambitious justifying the persistence of crunch as an industry practice.

Similarly, developers often described crunch using an ends-justify-the-means discourse, where workers could dismiss the crunch's costs provided their final game was a success. Nearly every postmortem that discussed crunch also included some form of "it was ok in the end because...." rationale⁶. "Post-project catharsis and the feeling of having survived and solved all the creative puzzles is also a powerful influence in the overall control mechanism of project-based work, particularly when the project is a success (Ó'Riain, 2001)" (Peticca-Harris, 2015, p. 578). This mentality encourages developers to focus on the product at their own expense, dismissing the costs of crunch as an inevitable part of creating a great game.

Finally, many developers not only accepted crunch but actually valorized it as a sign of their commitment to producing high-quality games. For instance, the 2010 postmortem for *Deadly Premonition* listed "96: continuous hours without sleep", "3,806.75 gallons: Coffee consumed", and "338,040: Cigarettes smoked" (August 2010, p. 26) as part of their "development by the numbers" stats. Other representative comments ranged from "As proud as the team is of the final incarnation of the game, some remain more proud of having survived the development that led up to it" (July 2002, p. 41) to "Half the cool stuff that happens [in development] is because someone sacrifices something, and some other group of people is veering off the budget, the schedule, and the corporate goals" (September 2007, p. 39). Crunch is held up by these developers as a sign that they will do whatever it takes in pursuit of greatness. As such, it becomes a goal rather than a problem.

⁶ E.g. postmortems for *Black & White* (June 2001), *Star Wars: Rogue Leader* (March 2002), *Jak II* (January 2004) and *Age of Booty* (November 2008).

Each of the above contributed to a social environment in which crunch is a normal, expected practice, allowing it to continue structuring future choices. It's important to note, of course, that not all articles supported the idea of crunch as inevitable. These articles stated things like, "there's a difference between developers who willingly give overtime, and managers who demand it" (October 2007, p. 28), recognizing that developers who worked long hours were often not *choosing* to do so. Therefore, the solution is not for workers to be less passionate, but for industrial mechanisms to help guard against overwork. The creation of a new habitus, with different built-in labor expectations, could be one solution.

Discussion

The analysis we offer here has a number of limitations. For instance, our choice to focus on 2000-2010, although relevant to questions of crunch, makes it difficult to draw conclusions about why conversations around unionization are rising in game spaces now. We also focus only on one magazine, without validating our themes via means like interviews with developers. Therefore, our analysis may not reflect all the discourses circulating in game development spaces, and it's possible that other publications discuss crunch differently. However, we contend that GD was the correct choice for beginning this analysis as it was the most popular development-targeted games magazine throughout its existence. Further, based on our analysis of the magazine, we found it often presented multiple sides of an issue, foregrounded developers' voices, and balanced positive and negative views on the industry. It is of course true that this publication, by developers for developers, is constructed within the existing habitus of game development and therefore has many of its own blind spots. But as providing an external eye that seeks to recognize and denaturalize those is one of the main goals of this project, GD offers a clear path towards starting a process of change.

Further, the fact that many of our findings mirror those of different studies— including ones that rely on workplace observation (e.g. O’Donnell, 2014) or surveys and interviews (e.g. Legault & Weststar, 2016; Peticca-Harris et al., 2015)— suggests that these limitations are not enough to render the results useless. Rather, our findings replicate and extend existing work by proposing that, although the project-based nature of game development might not change, the industry’s habitus— “the continuous, unconscious conditioning that is exerted through conditions of existence as much as through explicit encouragements or warnings” (Bourdieu, 1990, pg. 50)— could. To this end, our analysis revealed three distinct yet overlapping discourses— the idea that games were an unmanageable creative industry, a palpable anti-corporate ethos, and game developers’ stereotypical identity— that normalize game development’s habitus regarding crunch.

Every employee, regardless of rank, is deeply embedded in a social environment in which crunch becomes a logical strategy, due to shared discourses, project-based work, and developer identities. Then, when faced with new projects and new decisions about management, structure, and labor, gameworkers continually draw on the structures and experiences they have employed in the past. As O’Donnell (2014) states, “[gameworkers] are not oblivious to the structures in which they all work, but despite their frustration with the industry, they continue to invest themselves in projects” (p. 22). Therefore, any attempts to promote better labor practices within the industry must take a multi-pronged approach, making issues more transparent across industrial levels, initiating informed discussions, and promoting various solutions to each component of crunch.

QoL issues are ongoing in the games industry, and future work should focus on how crunch is discussed and understood in more recent years. For instance, although it is beyond the

scope of this piece, the contemporary Game Workers Unite movement (“About Us”) has taken off in the late 2010s. Organized and run entirely by gameworkers, GWU pushes back against the normalization of crunch and advocates for better labor organization and unionization efforts within game development. Further, GWU reaches across different segments of the games industry, with both AAA and indie members (Milner, 2018). Therefore, its members may not all be inured to AAA development’s existing habitus, allowing them to question and break it down more effectively. This movement thus needs further comparison to earlier efforts like the IGDA’S QoL white papers in order to understand their differing foundations and impacts.

While this ongoing research occurs, academics, industrial organizations, and unionization movements must continue to denaturalize games’ culture of overwork. One possible intervention is referring to crunch as what it is— unpaid overtime— rather relying on a misnomer. Showing the costs of unpaid labor and discussing alternative industrial models, in which employees have a better work/life balance, can also work to break down crunch’s taken-for-granted status. Within GD, there were a few articles and postmortems where development teams successfully avoided crunch. For instance, Blue Fang Games President Hank Howie argued, “Keeping people’s hours reasonable must be a company priority. I submit that it should be one of your company’s essential values. Otherwise, and especially if it hasn’t been a key value to date, when confronted with a difficult situation, it’ll be too easy to backslide into old habits” (January 2005, p. 20). Blue Fang maintained a reputation for producing successful games while preserving employee QoL. Therefore, alternative development methods are possible, but rare, indicating how they must be normalized the same way crunch has been. Building these different structures into the habitus of gamework will change the lens through which future decisions are made and could emphasize work-life balance as a structuring structure for the industry.

Further, those who set production schedules, both in studios and publishing companies, must reconsider the commonly held notion that games are unmanageable and work to build clearer development processes. Whether development is actually unmanageable or not, our dataset clearly showed that the *perception* of games as unmanageable meant that teams often under planned their projects, in particular shortchanging their preproduction phase. One memorable piece wrote, “Our guess is that 80 percent of mistakes in game development are the direct result of things done— or not done— in preproduction” (June 2002, P. 32). Because developers hesitated to take the time to lay out a game’s foundation fully, due to the expectation that it would change, they often lacked critical details needed to inform later hiring, development steps, and priorities. This frequently led to crunch. In contrast, the few postmortems that showed greater efforts to manage production faced fewer issues. Developers should be attentive to this evidence.

In the latter half of the decade, several studios drew on the broader software industry’s “agile” and “scrum” development methods in acknowledgement of the game industry’s maturing position. As part of a multi-billion-dollar enterprise with rising game budgets, AAA managers and executives faced greater pressure to complete projects on time. New design methodologies were a response to this pressure, and many resulted in positive outcomes. For instance, the 2009 *Brütal Legend* postmortem stated, “We were drawn to the advantages of agile software development and decided to adopt Scrum... the initial payoffs were impressive” (December 2009, p. 26). Although not explicitly intended to reduce crunch, these efforts could have that effect, as they show that the game production process can be managed. Indeed, the fact that different studios successfully turned to different management approaches indicates that developers even have options to draw on. To be clear, we are not recommending one specific

methodology over another, as all have their pros and cons. However, we are saying that managers must recognize what has been effective for teams or projects like theirs in the past and use these to generate solutions rather than falling back on existing labor structures and the excuse that games are “unmanageable”. New forms of management can then serve “as principles which generate and organize practices” (Bourdieu, 1990, p. 53), adapting future game-making decisions and reducing the need for crunch.

Another useful change to games’ habitus would be to remove the romanticized notion that garage development is inherently more creative or better than corporate development. This discourse routinely led employees to balk at any type of policy that suggested corporatization, be it a dress code, limitations on gaming in the workplace, or mandatory working hours.

Unfortunately, scorning corporate policy often acted as a form of self-sabotage where workers rejected many of the very decisions that could improve their QoL. Gameworkers should be encouraged to see themselves as part of a mature industry; accepting organizational structures like Human Resources, set work hours, and inter-departmental communication protocols would likely facilitate game studios’ day-to-day operations and improve the lives of employees.

Questioning existing assumptions regarding developer identity could create additional opportunities to introduce meaningful QoL discussions into the industry. Employees perform the bulk of the labor that generates millions in revenue for their studios and publishers. Helping developers, coders, designers, artists, and programmers realize the value of their labor should be a top priority for anyone interested in breaking the cycle of crunch; doing so will hopefully help workers resist unrealistic demands on the part of their employers as well as the discourses that argue their unpaid overtime on a game is justified if the product is a success. Drawing attention

to the problematic aspects of developer identity that emphasize passion, commitment, and perfectionism is a step in building a better habitus for these employees.

Many aspects of the video game industry that contribute to crunch, such as development's project-based nature, are unlikely to change. Therefore, if better QoL is to be obtained, interventions need to aim at areas that could be altered, such as the sociocultural environments of gamework. Game development's current habitus will likely be difficult to overcome. However, given the costs that crunch inflicts on workers, their families, and the industry, it must change. There are many possible paths to this result, including unionization, and we recognize that efforts to improve the video game industry's labor practices are ongoing. What our research contributes is a deeper understanding of the challenges that these efforts face; crunch is currently supported by a diverse set of social and economic forces that frame unpaid overtime as a natural, inevitable outcome of the development process itself. Because of this, any interventions must be multi-tiered and longitudinal, with different approaches employed to address the unique challenges faced by different types of employees.

Works Cited

- “About Us”. (n.d.) Game Worker Unite. Available at: <https://www.gameworkersunite.org/about-us>
- Bonds, S, Briant, J, Clingman, D, Howie, H, Laramée, FD, LoPiccolo, G, Luckey, A and McShaffry, M (2004) Quality of Life in the Game Industry: Challenges and Best Practices. Available at: www.igda.org/qol.
- Bourdieu, P (1977) *Outline of a Theory of Practice*. Translated by R. Nice. Cambridge, UK: Cambridge University Press.
- Bourdieu, P (1990) *The Logic of Practice*. Translated by R. Nice. Stanford, CA: Stanford University Press.
- Consalvo, M (2008) Crunched by passion: Women game developers and workplace challenges. In Kafai Y, Heeter C, Denner J and Sun JY (Eds) *Beyond Barbie and Mortal Kombat*. Cambridge, MA: MIT Press, pp. 117-190.
- Cross, K (2018) What will be left of the people who make our games? *Polygon*, 17 October. Available at: <https://www.polygon.com/>.
- Deuze, M, Martin, CB, & Allen, C (2007) The professional identity of gameworkers. *Convergence* 13(4): 335–353.
- Dyer-Witheford, N and de Peuter, GS (2006) “EA Spouse” and the Crisis of Video Game Labor: Enjoyment, Exclusion, Exploitation, and Exodus, *Canadian Journal of Communication*, 31(3).
- Edwards, K, Weststar, J, Meloni, W, Pearce, C, and Legault, M (2014). Developer satisfaction survey 2014: Summary report. Report for the IGDA, 25 June. Available at: www.igda.org.

- Fairclough, N (2012) Critical discourse analysis. In Gee JP and Handford M (eds) *The Routledge Handbook of Discourse Analysis*. New York, NY: Routledge, pp. 9–20.
- Fujigaki, Y (1996). Time series investigation of job-events and depression in computer software engineers. *Industrial Health* 34(2): 71–79.
- Good, OS (2018) Red Dead Redemption 2 developers open up about work conditions at Rockstar Games. *Polygon*, 18 October. Available at: <https://www.polygon.com/>
- Harvey, A & Shepherd, T (2017) When passion isn't enough: gender, affect and credibility in digital games design. *International Journal of Cultural Studies* 20(5): 492–508.
- Hoffman, E (2004) EA: The human story. In: ea_spouse blog. Available at: <http://ea-spouse.livejournal.com/274.html>.
- Huntemann, N (2010) Irreconcilable differences: Gender and labor in the video game workplace. *Flow*. Available at: <http://flowtv.org/>
- Johns, J (2006) Video games production networks: Value capture, power relations and embeddedness. *Journal of Economic Geography* 6(2): 151–180.
- Keogh, B (2015) Between triple-A, indie, casual and DIY: Sites of tension in the videogames cultural industries. In Oakley, K and O'Connor, J (eds) *The Routledge Companion to the Cultural Industries*. New York, NY: Routledge, pp. 152–162.
- Kerr, A. (2006) *The business and culture of digital games: Gamework and gameplay*. London: Sage Publications.
- Kerr, A (2011) The culture of gamework. In Deuze, M (ed) *Managing Media Work*. London, UK: Sage, pp. 225–236.
- Kerr, A (2017) *Global games: Production, circulation and policy in the networked era*. New York, NY: Routledge.

- Kuutila, M, Mäntylä, MV, Maëlick, C, and Elovainio, M (2018) Daily questionnaire to assess self-reported well-being during a software development project. In: *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, Gothenburg, Sweden, pp. 39–43. ACM Press.
- Legault, M-J and Weststar, J (2016) Videogame developers among “extreme” workers: Are death marches over? *The E-Journal of International and Comparative Labour Studies* 6(3): 73–99.
- Lindlof, TR and Taylor, BC (2002) *Qualitative communication research methods (2nd ed)*. Thousand Oaks, CA: Sage.
- Milner, D (2018, December 21) Game Workers Unite: The fight to unionize the video game industry. *Game Informer*. Available at: <https://www.gameinformer.com>.
- Nishikitani, M, Mutsuhiro N, Kanae, K, Kyoko, N, and Yano, E (2005) Influence of overtime work, sleep duration, and perceived job characteristics on the physical and mental status of software engineers. *Industrial Health* 43(4): 623–629.
- O’Donnell, C (2014) *Developer’s dilemma: The secret world of videogame creators*. Cambridge, MA: MIT Press.
- Peticca-Harris, A., Weststar, J., & Mckenna, S. (2015). The perils of project-based work: Attempting resistance to extreme work practices in video game development. *Organization*, 22(4), 570–587.
- Rockstar Spouse (2010) Wives of Rockstar San Diego employees have collected themselves. *Gamasutra*, 7 January. Available at: <http://www.gamasutra.com/>
- Schreier, J (2014) Why game developers keep getting laid off. *Kotaku*, 5 June. Available at: <http://kotaku.com/>

- Schreier, J (2016) The horrible world of video game crunch. *Kotaku*, 26 September. Available at:
<http://kotaku.com/>
- Schreier, J (2018) Inside Rockstar Games' culture of crunch. *Kotaku*, 23 October. Available at:
<https://kotaku.com/>
- Sinclair, B (2012) Why would anyone ever want to be a AAA game developer? *GameSpot*, 3
July. Available at: <http://www.gamespot.com/>
- Weststar, J, O'Meara, V, & Legault, M-J (2018) Developer satisfaction survey 2017: Summary
report. Report for the IGDA. Available at: www.igda.org
- Wright, TA, and Cropanzano, R (2000) Psychological well-being and job satisfaction as
predictors of job performance. *Journal of Occupational Health Psychology* 5(1): 84–94.