

Praxis

The everyday lives of video game developers: Experimentally understanding underlying systems/structures

Casey O'Donnell

Grady College at the University of Georgia, Athens, Georgia, United States

[0.1] *Abstract*—This essay examines how tensions between work and play for video game developers shape the worlds they create. The worlds of game developers, whose daily activity is linked to larger systems of experimentation and technoscientific practice, provide insights that transcend video game development work. The essay draws on ethnographic material from over 3 years of fieldwork with video game developers in the United States and India. It develops the notion of creative collaborative practice based on work in the fields of science and technology studies, game studies, and media studies. The importance of, the desire for, or the drive to understand underlying systems and structures has become fundamental to creative collaborative practice. I argue that the daily activity of game development embodies skills fundamental to creative collaborative practice and that these capabilities represent fundamental aspects of critical thought. Simultaneously, numerous interests have begun to intervene in ways that endanger these foundations of creative collaborative practice.

[0.2] *Keywords*—Anthropology; Creative collaborative practice; Experimental system; Game studies; Games; Video game development; Work/play

O'Donnell, Casey. 2009. The everyday lives of video game developers: Experimentally understanding underlying systems/structures. *Transformative Works and Cultures*, no. 2. <http://dx.doi.org/10.3983/twc.2009.0073>.

1. Introduction

[1.1] When people talk about video games and the rules or games that reside within them, I frequently hear them mention game developers as almost malevolent gods: those who reign over the natural order of the system, a system that rightfully belongs to its players. But that wouldn't be the same game, would it? Aren't the artificiality of the constructed systems and the process of determining those rules what make it a game in the first place ([note 1](#))?

[1.2] This essay examines how tensions between work and play for video game developers shape the worlds they create. The worlds of game developers, whose daily activities are linked to larger systems of experimentation and technoscientific practice, provide insights that transcend video game development work ([note 2](#)). The essay

draws on ethnographic material from over 3 years of fieldwork with video game developers in the United States and India. I develop the notion of creative collaborative practice based on work in the fields of science and technology studies, game studies, and media studies. The importance of, the desire for, or the drive to understand underlying systems and structures has become fundamental to creative collaborative practice. The essay makes the argument that the daily activity of game development embodies skills fundamental to creative collaborative practice and that these capabilities represent fundamental aspects of critical thought. Simultaneously, numerous interests have begun to intervene in ways that endanger these very foundations of creative collaborative practice.

[1.3] This essay is structured to step into the everyday lives of video game developers and is organized around the consequences for creative collaborative practice. I begin by examining the category of work/play described as *instrumental work/play*, which is core to the process of understanding underlying systems and structures. The desire/drive/process to understand is ever present in the second part of the essay, which examines the tools of video game development. These are termed *experimental systems* that are able to plug into instrumental work/play in ways that enable our pursuit of underlying systems/structures, often in seductive ways. The third section of the essay examines the consequences and implications of how these systems and drives have assembled themselves and ultimately what the implications have been for creative collaborative practice.

[1.4] The subject of creative collaborative practice arose from my research in core categories. It became a means to address and talk about a system of interconnections between myself, my informants, and the structures that enable and constrain video game developers. It was a means by which to account for "how various kinds of systems (textual, social, technological, etc.) hang together," and how those systems "are continually being reconstituted through the interaction of many scales, variables, and forces" (Fortun 2006:296). Because of the complexity of this overall system, I focus here on the desire for or the drive to understand underlying systems and structures, which is one component or subsystem within the broader category of creative collaborative practice. The foundations of this system are shown in figure 1.



Figure 1. A cloud diagram depicting the numerous subsystems that plug into the larger system of creative collaborative practice. The image demonstrates how creative collaborative practice, as my particular object of concern, is connected with other areas of inquiry and investigation. [[View larger image.](#)]

[1.5] Ultimately, it is the entire system of creative collaborative practice that is important, and at times it can seem that parts of the story, such as subsystems or structural elements, are missing, in part because they are. To address the entire system, subsystems, and structures in one single essay would be a disservice to the complexity of the overarching system. Where necessary I have tried to help illuminate connections, though I do not explicitly focus on them (note 3). This allows me the opportunity to focus on particular elements of the systems that are likely the most relevant for a given community. For example, creative collaborative practice as an analytic category offers the most insight for communities interested in fan- and user-created content because of its crucial interest in understanding underlying systems and structures and the desire of would-be-developers to care about how games work and how they can be created. That desire is a particular leap necessary for creative collaborative practice and also a precursor to wanting to make games. Most game developers can recall one particular event that caused them to care about how games are made and encouraged them to attempt to understand what precisely was going on within these systems. Fans and users interested in creating video games have started this transition toward a desire to understand what lies beneath.

[1.6] At first, the conclusion appears too simple: that the very real worlds of game developers obviously shape the virtual worlds they create (note 4). They too are influenced by the play of games and the reading of other virtual worlds. Games dominate the language of both work and play for gamers and game developers alike. Most game developers have an extensive history of playing games and the line between the fan/user and the aspiring game developer is often not clear. While this is not always the case, it is predominantly so. Even game developers who, prior to working in the game industry, did not consider themselves gamers have one of the latest console systems. Many try to play, at least casually, the newest and most

popular game titles in order to keep their shop talk up to date. However, it would be folly to think that this is done simply as a mechanism to keep others at bay or explicitly exclude. This vernacular, which I have come to term *game talk*, does work for game developers. Game talk provides discursive resources for developers when trying to describe abstract concepts like game mechanics—that is, the underlying game that is then presented on a screen. Think of it as the rule that higher cards beat lower cards, and equal cards mean war in the card game war. Game talk is a way to talk about or understand those underlying systems and structures they are attempting to create.

[1.7] Game talk also serves as a bridge between the many different backgrounds of those who come to find themselves in video game companies. There are four primary disciplines that structure game studios: engineering, art, design, and management. Engineering tends to be made up of computer scientists, electrical engineers, and physicists. Artists tend to come from numerous artistic backgrounds, though all have at some point become proficient using digital art production tools like 3D Studio Max, Maya, and Photoshop. Designers tend to come from a variety of backgrounds, with the only commonality being their love of and interest in creating games. Managers tend to have risen through the ranks as engineers, artists, or designers. Occasionally they will have returned to school in order to learn management or business as a discipline, though this is quite rare. Two new subdisciplines or specializations have formed recently between engineering, design, and art, dubbed the *tools engineer* and *technical artist*. There are of course other aspects of game development, such as quality assurance (QA), information technology support, facilities management, marketing, human resources, and administration, which are all vital to the functioning of a studio. However, in an effort to focus on the work of game development practice, I tend to concentrate on the four disciplines and two subdisciplines that make up the majority of a studio's staff.

[1.8] Because there is no discipline of game design or game development, games become a kind of lingua franca ([note 5](#)). When you think and talk about games, they become aspects of the workplace. One particular experience while in the field demonstrated the importance of this vernacular for developers. The term *vertical slice* loomed large in the discourse of game development when I found myself among the developers of my primary fieldsite. Vertical slice is having one example of everything that would then go into the game; examples of every special effect, every game mechanic, and one feature-complete sample level; one of everything. In its crudest form, vertical slice can be a complete demo of a game. More often, it is more nuanced, where a slice demonstrates particular or single aspects of a game (such as a particular mechanic, visual style, or effect), which then can be iterated over with close attention to those specific details. For game developers the choice of which of these definitions

will be used is often dictated by the publisher or someone external to their company. The following is an excerpt from my fieldnotes during the lead up to a vertical slice. It demonstrates how game talk serves as a discursive tool for game developers.

[1.9] Vertical slice was looming. When deadlines loom, more meetings crop up, it seems inevitable. This meeting was about attempting to figure out the game mechanics of a possible, but not for sure, WiFi multiplayer aspect of the game. Of course everyone was already tired and the thought of having to define a new game mechanic and underlying technology to support it was low on everyone's list of priorities. Engineering was saying one thing, Design was saying another, Management was saying a third. At least from my seat, it seemed like they were all saying the same thing, but the meeting continued for nearly an hour before one designer looked at an engineer and asked, "So, do you mean it's like *Spy vs Spy*?" (figure 2)



Figure 2. A screenshot of the game *Spy vs Spy* (1986) for the Nintendo Entertainment System.

[1.10] After this suggestion, Engineering thought for a moment, and said, "Yes, like *Spy vs Spy*." Management and the other designers in the room also nodded their heads.

[1.11] *Spy vs Spy* is a game with a long history of its own, but the mechanic that interested my informants was the idea that rather than engaging in direct combat (that is, shooting or punching an opponent). the players would set traps for one another. Much as other researchers studying scientific fields have noted, game talk accomplishes numerous tasks for game developers. It "creates, defines, and maintains the boundaries of this...community; it is a device for establishing, expressing, and manipulating relationships in networks;...it articulates and affirms the shared moral code about the proper way to conduct [scientific] inquiry" (Traweek 1988:122).

[1.12] While game talk can be a productive tool for uniting disparate disciplines, it can also be used to exclude. Difficulty with game talk was encountered by many women working in the game industry, who may have been familiar with a smaller subset of games, but frequently found themselves outside the margins of knowing about particularly obscure or marginal games. Although the following quotation concerns the world of high-energy physics, many of the same things can be said of game development.

[1.13] Access to this world of oral communication is quite limited. In a community with easy access to widely disseminated written information, keeping crucial information accessible only in oral form is an impressively effective means of maintaining its boundaries...Protection of oral communication encourages the development of a closed community. In physics it is consistent with the group's image of itself as a meritocracy: only an informed, worthy member of the community will know what is to be said and what is to be written. (Traweek 1988:120)

[1.14] The focus on oral communication proved especially troublesome for many of my Indian informants. While some were avid gamers on personal computers, few had grown up with console game systems and they had not necessarily played the many games that were released on the Nintendo Entertainment System, Sega Genesis, Super Nintendo, or numerous other consoles produced during the 1980s and 1990s. This was frequently exacerbated by the idea that games were entertainment, and time was better spent by young people studying, rather than playing games. The Indian game industry is dominated by artists because most studios have only been contracted by U.S.-based studios to do art production work. There are a handful of companies that have attempted to overcome this limitation by using their contract work to fund internal game development. More recently, as large publishing companies like Electronic Arts have begun creating studios in India explicitly to house large numbers of digital artists, it has created unique difficulties for studios that are attempting to advance their companies by fostering talent internally. Rather than expecting their employees to already be versed in the language of games, Indian game studios have in many cases built rooms where employees can play games in their off hours or even as part of normal working hours. These rooms typically have consoles, computers, games, and couches, where game developers can increase their knowledge of those games that U.S. developers frequently reference in their day-to-day exchanges.

[1.15] This language set is the foundation that video game developers use to plug into broader work/play systems. Game talk becomes a means into or a means of understanding the underlying systems and structures of video games. Game talk serves as the working manual that instrumental play depends on where a formal

discipline, handbook, or set of standards does not exist. Game talk connects with the desire or drive to understand how these systems operate because it provides a resource by which to get inside them.

2. Instrumental work/play amid rigid systems

[2.1] For many game developers, if the process of creating games can be considered a game, then, at least as currently configured, it is a game for power gamers. I have come to call numerous aspects of video game development work practice *instrumental work/play*, a process essential to all aspects of game development. In many cases, it is impossible for artists, engineers, or designers to know precisely what they can manage to create without causing the underlying hardware to buckle under the pressure. Thus, instrumental work/play is the process/drive/desire to determine how to get the most out of the systems required to get the job done. The slash between "work" and "play" represents the ambiguity that often is present when people talk about these activities. Some classify it as work and others as play, and time and space affect that categorization.

[2.2] There is another piece that drives game developers and workers in the New Economy more generally—an aspect of the work that encourages workers to push further and harder than may be necessarily required. Perhaps on some level it is a realization of a Protestant work ethic based on the idea of a calling (Weber 2002), but corporations find that their workers are plugged into their work in new ways. As sociologists and ethnographers of virtual worlds and gamers more generally demonstrate, the reason for this goes beyond the simple answer "because it is fun," which the work may very well be some of the time. More than that, the extra work from workers gets at an underlying drive and dedication to "efficiency and instrumental orientation (particularly rational or goal-oriented), dynamic goal setting, a commitment to understanding the underlying game systems/structures, and technical and skill proficiency" (Taylor 2006a:72–73) on the part of game players. The instrumental or power gaming of the workplace is part of a desire to understand the structures workers move within—knowledge that will help workers do their jobs. In part, this is a product of a system that seems arbitrary or imposed, and hence the "broader ambivalence about what constitutes legitimate play[/work]" (2006a:72–73).

[2.3] It is this ambivalence that plugs into the mystique or desirability of video game development work. Game development work is not "real" or "ordinary" work and developers' presentation of game development work as hard but separate from other work provides a kind of cachet. As cultural historians and scholars of ludology (that is, the study of games) have noted in their examinations of play, all playgrounds are "marked off beforehand," providing the grounds for a new "absolute" order. "Into an

imperfect world and into the confusion of life [play] brings a temporary, a limited perfection" (Huizinga 1971:10).

[2.4] As the following quote by an engineer notes, up-front information does not always determine what can or cannot be done.

[2.5] Well...I guess the biggest thing is art, and they try to scope that at the beginning, with technology, the specs that are given about the PSP [Playstation Portable], how fast it is, how many triangles it can render, blah, blah, blah. So most of the, I guess a lot of the performance is on the art side. I mean, there is other performance issues, like physics, and stuff that are big, like our game code (Engineer, personal communication, 2007)

[2.6] But despite developers' best efforts to scope everything out at the beginning based on the specifications of a gaming system provided by manufacturers, developers frequently find themselves having to maintain a "commitment to understanding the underlying game systems." One engineer in particular, after doing specific optimizations for the Nintendo DS handheld console, noted that he had a feel for "every transistor and chip inside that thing" (Engineer, personal communication, 2007). Despite all of the documentation, specifications, and scoping, it required his investigation of and prying into the system to get it to do what they had assumed it would do in the first place. An engineering group manager, who had gone through this process many times, talked about the process of digging into a system at the software or hardware level, and its requisite attentiveness.

[2.7] **Manager:** The process of stuff not working like it's supposed to?

[2.8] **CO:** Yeah.

[2.9] **Manager:** So there are levels, right. And these are especially obvious to the new people. Where you cannot perceive of a reality where something doesn't work. Because it obviously should, right? So there is an emotional component of it where you just won't even believe it, you'll be like, "that can't be broken." You gotta get over that pretty quick. And I guess that's where the logical part comes in. You say, well, you know, it's broken. And if you take the time, the gruesome horrible time, you can always, if you have to, map out the transistors, and follow the flow of logic. It will all come out in the wash. So you have to be persistent. (Engineering group manager, personal communication, 2004)

[2.10] While some science and technology studies scholars have written about the process of debugging or feeling out systems as "being in a cave" (Turkle 1997:225), it is actually not so unsystematic. Just as much can be learned from negative knowledge,

from understanding the limits of what is known. As noted by sociologists, negative knowledge can provide focus, "things that interfere with our knowing, of what we are not interested in and do not really want to know" (Knorr-Cetina 1999:64). The ability to simultaneously think thoroughly/methodically and creatively/intuitively is a different way of thinking about how we come to understand how things work. To get at these underlying systems, as anthropologist of medicine Emily Martin notes, our tasks and relations with them begin to approximate "disorders" where our "exaggerated sense of urgency" and "exaggerated sense of boredom" contribute to our abilities to "stretch, cram, speed, warp, and loop poor old linear time and space." The capacity to "organize the chaotic mix of seemingly unrelated simplistic elements into a more integrated and comprehensive framework of understanding, approaching a clearer picture of complexity," begins to approximate the clinical definition of attention deficit disorder (1997:253–54). Although in both transcript excerpts these are engineers talking about software systems, the same ends up being true for artists and designers. They too depend on layers of software and hardware systems, each with idiosyncrasies. During one conversation, a technical artist talked about how willing artists were to assume that something that was broken was their fault. They would continue attempting to work with a model or other art asset, trying to get it into the game, despite repeated failures. In some cases they would even manage to make something fit into the game that should not have fit in the first place. Even working within the lines established by others, artists will continually modify and carefully check their work to see where changes can be made that improve the visual quality of their work without breaking the guidelines they have been given. Designers will frequently attempt to do things that they have been instructed will not work to see whether they can tweak them so they fall within guidelines.

[2.11] So it is not just technicians or engineers who, as historians and geographers of technology note, work at the "empirical interface between the material world and machine-generated representations of the world" (Downey 2001:229), but nearly every game developer. Each engages with instrumentalist tendencies in an effort to accomplish its goals. It has been said by engineering and design studies researchers that "designing is not simply a matter of trade-offs, of instrumental, rational weighing of interest against each other," that "nothing is sacred, not even performance specifications, for these too, are negotiated, changed, or even thrown out altogether" (Bucciarelli 1994:187). But the fact of the matter is, for game developers, the push-and-pull design process seeks to determine where the bottom is, where structures impose themselves on systems and subsystems. Despite the construction of rigid specifications, they are based on something that must be determined by the developers, and eventually they run into the physical limits of electrons and silicon. Specifications are made, but they are not made up. It is a negotiated process that is, time and again, the product of instrumental play.

[2.12] Studio heads and managers play with their organizations like artists play with textures and models. They must interface and play with game development studios and intellectual property holders that range from movie studios to comic book publishers, console video game system manufacturers, and video game publishing companies. Studio heads and managers also bump into the limits of their teams, their employees, their networks (both social and technological), and their access to secret, proprietary networks. Managers must do as much as possible with as little as possible. Teams will get shuffled around on the basis of the work available. If particular designers have proven themselves able to handle especially restrictive conditions, they may be moved from one project to another in the hopes of bringing new perspectives to existing teams. They must often instrumentally play within the structures of a project that they have been hired to complete. Managers must massage creative visions that differ, and changes will almost constantly be asked for by those who are not affected by the necessary reworking of systems to allow for these modifications. Other actors within corporate institutions will be instructed to use the video game that a studio is working on as a means of negotiating with other institutions. Managers must frequently engage with the management of this system, which in many cases began as play for them.

[2.13] Nearly every conversation with game developers begins with a disclaimer such as, "Not that we are representative of the industry more broadly," or "We do things a bit differently here from the other studios you have probably visited." The consequences of and symptoms of the secrecy that surrounds the daily worlds of game developers have dramatic consequences for those who choose to work in these worlds. Sociologists of New Economy and New Media workers have noted that while the "general lack of formality" may excite and entice young, driven workers, the "blurred lines between work and play pressure workers to participate in implicitly 'non-corporate' culture even if they do not enjoy it, or if they have to put in extra time" (Neff et al. 2005:321).

[2.14] It is not just a demand or coercion that everyone push harder and play/work in the same instrumental ways. Coercion happens; that is undeniable. However, the structures that we must work within are also crucial to why people feel compelled to play. It is not simply the systemization and regimentation of game/sport that causes the loss of gamelike innocence; the issue is larger than that. Simply bringing money in does not automatically ruin the game. The difficulty is that money brings those interested in playing other kinds of games into connection with what many had hoped might stay a game. It is the incorporation of a drive toward institutionalization that changes the game. As anthropologist John D. Kelly writes in the context of American baseball,

[2.15] But it wasn't commoditization that changed baseball so unmistakably. It was higher levels of capitalist organization. Above all, the leagues changed everything. What are they, and what is their relationship to commodities? Commoditization, yes, but we will need more tools than that: we will need to understand whole new layers of management. We will need a theory of the firm. Professional Sports leagues did more than commoditize the game. They incorporated it. What is that? (Kelly 2006:55–56)

[2.16] So what has changed the play of game development into the play/work of game development is the coming together of a willingness to play (or to be coerced into play) in particular ways, along with the systematic incorporation of the video game industry. That move to industry rather than to something else marks an event that begins to alter the space of play. Baseball is one example, but the connections with the video game industry are undeniably industrial. Baseball did not start as an industry, but it has moved to that. It is not simply that people good at playing a game begin to accept compensation to publicly perform their play. That might make it sport, but that, in and of itself, does not account for the work/play conflation. It is the connection with commercial profit-driven organizations that has so dramatically shaped game developers' worlds.

[2.17] At the same time, instrumental play requires feedback from those systems under investigation: surprising answers must be gleaned to provide further insight into how they function. This is the role of experimental systems within the daily activities of work/play.

3. Experimental interactive work/play systems

[3.1] Beyond the imaginary isolation and secrecy of the video game industry, beyond the rise of instrumental play and the rise of incorporated structures in the professionalized game industry, there is something else that gets more deeply at why we might call the work/play of game development "play." It has to do with all the daily activity. But that daily activity is linked to larger systems of experimentation and technoscientific practice: engineers continually bump into and play with the hardware and software of existing systems. Artists constantly play with 3-D models, attempting to balance creative desires with the demands of designers, managers, and the commentary of their peers. Designers tinker with games through custom tool kits and define the structure of play, in which others will participate. Managers experiment with personnel locations and team members active on any given project. Executives juggle product and project lineups. We begin to see why the conflation of work/play might be a useful way for us to consider what is happening during the daily experience of work.

[3.2] Our instrumental play is enabled by the desire for technological systems, data stored on hard drives across the network and checked into version control systems, and even people we can interactively respond to. Communication studies scholars examining digital games find that

[3.3] Digital games are interactive media par excellence because their entertainment value arises from the loop between the player and the game, as the human attempts by the movement of the joystick or keyboard or mouse to outperform the program against and within that, which he or she, with or without networked coplayers, competes. This interactive feedback cycle is often represented as a dramatic emancipatory improvement over traditional one-way media...But we insist these interactive potentialities are historically constrained and structured by the process of game design, technological innovation, and product marketing. (Kline et al. 2005:294–95)

[3.4] The desire for the complex system—game, workplace, peers—to respond interactively is seductive. When it responds like an interactive system, we feel connected or networked. When it does not, we become frustrated.

[3.5] But what about historical structuring aspects of game design and technological innovation? If we had known we were going down this road in the first place, might we have planned better, so they could be more interactive? I suspect not. It simply does not work that way. As historian of biology Hans-Jörg Rheinberger writes about experimental practice and experimental systems:

[3.6] An experimental system can readily be compared to a labyrinth, whose walls, in the course of being erected, in one and the same movement, blind and guide the experimenter. In the step-by-step construction of a labyrinth, the existing walls limit and orient the direction of the walls to be added. A labyrinth that deserves the name is not planned and thus cannot be conquered by following a plan. It forces us to move around by means and by virtue of checking out, of groping, of tâtonnement. He who enters a labyrinth and does not forget to carry a thread along with him, can always get back. (Rheinberger 1997:74–75)

[3.7] Not only have game developers been constructing the labyrinth of game development as they go, they have been doing so in such a hurry that they have not bothered carrying any thread along. Developers have made a headlong plunge in, with no way to get back or untangle their route. They occasionally make retrospective postmortem statements about where they think they have been and where they might have gone wrong. But why, then, do they continually take the same wrong turns? The situation becomes more dire when developers realize that they must maintain the

secret society of the office and game development generally. Suddenly they are not willing or able to talk about the labyrinth in any real detail. They talk about how pretty the vines look, or how they were able to grow them in a particular way, or how they sometimes take wrong turns and have to work late to find their way back. But meaningful collaboration? That has been rendered impossible.

[3.8] Postmortems are a genre of video game development writing frequently found in *Game Developer Magazine* or online at sites like Gamasutra (<http://www.gamasutra.com/>). These are often retrospective documents that look at what went right or wrong during the game development process. Because many companies think that they might be giving away secrets, or because the documents must be cleared by legal departments, they frequently contain few details. In most cases, the more detailed the postmortem, the more likely that it was written by an independent developer or company at the edges of the industry.

[3.9] The implications of these kinds of practices more broadly throughout New Economy business are troubling. If game development is an index into New Economy work, a space where experimental practice is crucial, then the ability to communicate and think about those experimental practices is even more important. However, demands for secrecy seem to have taken precedence over the maturation of game development practice. This pushes the complexity and uncertainty associated with game development practice to greater heights and increasing chanciness.

[3.10] There is a game in this as well. "Tension means uncertainty, chanciness; a striving to decide the issue and so end it. The player wants something to 'go,' or to 'come off'"; we want to succeed by our exertions (Huizinga 1971:10–11). I like the metaphors of the labyrinth and experimental systems because they emphasize the enjoyment of working within limits, but also acknowledge the reality of having those limits pushed. "Not anything goes. If there is construction, it is constrained"; game developers and scientists alike "meet with resistance, resilience, [and] recalcitrance" (Rheinberger 1997:225).

[3.11] What does this have to do with experimental systems and the interactivity of people? Experimental systems have become a useful way to think about game development, in particular the work of designers—those who end up interfacing with the work of engineers and artists. Their tools are created by the tools engineers, but frequently with a mind toward changes down the road. This is also why tools engineers and technical artists accompany these new systems, technologies, and practices. As sociologists of science have shown, "the more automatic and the blacker the box is, the more it has to be accompanied by people" (Latour 1987:137). This is partly because these "outcomes are often not consciously calculated, or even intended by any one of the parties involved" (Knorr-Cetina 1983:130). Because tools engineers are

embedded in a broader social context of practice, they must somehow retain those connections:

[3.12] Experimental systems are to be seen as the smallest integral working units of research. As such, they are systems of manipulation designed to give unknown answers to questions that the experimenters themselves are not yet able clearly to ask...They are not simply experimental devices that generate answers; experimental systems are vehicles for materializing questions. They inextricably cogenerate the phenomena or material entities and the concepts they come to embody. Practices and concepts thus "come packaged together." (Rheinberger 1997:28)

[3.13] Until the time of production, and frequently even after that, almost every aspect of the game development process must act like an experimental system: it must be open or capable of providing answers to currently unknown questions. Sometimes these unknown answers are frustrating, but often they become aspects of the game proper. But more than that, experimental systems must be interactive; they must respond in real time to other technical systems, data, and people. This is where the headlong rush of the game development process becomes readily apparent. As designers play with the art and code that were assembled with experimental tools, the possibility of an accurate reconstruction of the past becomes unimaginable, even for game developers. Rather, many developers—and engineers in particular—hope for the possibility of a kind of *Star Trek* "mind melding," because the reality is that it is impossible to understand fully where the game came from or even where it will finally be when it arrives at Golden Master, the point at which it is ready to be shipped to the manufacturing company for mass production.

[3.14] The daily worlds of game developer work are constantly shaped by work/play forces. Instrumental play and interactive experimentation encourage developers to understand their worlds as separate, distinct, and outside of work, that is, as play. The ability to play with systems providing interactive feedback encourages developers to get into and remain in their work. Instrumental work/play and experimental systems become a kind of double-edged sword, cutting in both constructive and destructive ways. Growth and sharing are promoted by a competitive spirit tempered by the knowledge that sharing and cooperation are not mutually exclusive from good, clean competition. Without that knowledge, secrecy and instrumental play reign supreme. This makes it advantageous to pay special attention to those things that do and do not work.

4. A postmortem on instrumental and experimental work/play

[4.1] In this section, I borrow the form of game developers' postmortems, from which I also take material. As noted above, such postmortems contain explicit examinations of what went right and what went wrong. In many respects, this format hearkens to "the dance of agency" theorized as "a dialectic of resistance and accommodation, where resistance denotes the failure to achieve an intended capture of agency in practice, and accommodation an active human strategy of response to resistance, which can include revisions to goals and intentions as well as to the material form of the machine in question and to the human frame of gestures and social relations that surround it" (Pickering 1995:22).

[4.2] The mechanisms that enable developers to interact with their systems, data, and one another are infrequently discussed or shared. Even when they are, they are simply referred to as tools. No explanation is given about what these technologies do or what they accomplish for game developers. Though they are cited as one of the most important components of the game development process, they are unknown outside of the game industry. The supervisor on the game *Final Fantasy XII* for the studio and publisher Square Enix notes the importance of tools that provide experimental or trial-and-error approaches to design without providing any specific information about these approaches: "Our various in-house authoring tools, coupled with commercial digital content creation tools,...created an environment in which we could use trial-and-error tactics with the new tools while also increasing productivity by using the ones we already knew well. It was especially helpful for us that the in-house tools enabled real-time previews using the game's rendering engine" (Murata 2007:24).

[4.3] This aspect of game development is largely unknown and unexplored by those looking to enter the video game industry. Even more disturbing, these resources are often kept from companies doing offshore outsourcing work for video game studios. Even though the global economy is widely purported to be flat (Friedman 2005), the reality is that development tools are often withheld, which makes work far more difficult. Real-time previews of a game's content inside of a game's rendering engine are frequently cited as essential by artists. Yet time and again in India, I encountered artists struggling to work within the confines of structures unknown and invisible to them because the experimental tools, which would enable them to understand where, how, and why aspects of their work were failing, were withheld by the contracting organization. It is a means of encouraging outsourcing companies to approach only the aspects of video game development that do not put pressure on the game industry more broadly. Outsourcing companies are encouraged to approach only those aspects of video game development that they are contracted to do. Rather than providing them with insight into how games are developed, they are left to rediscover how art assets move into games production pipelines.

[4.4] This is an old way of developing games. Everything is processed by hand in ways that require artists or designers to hassle someone, frequently an engineer, into seeing their work operating within the game. Most game developers use this approach early in their careers for lack of knowledge of any other method. The lead designer of *Diablo II*, which was created by Blizzard Entertainment, one of the largest and most respected game studios, noted the difficulties of not having these tools. *Diablo II* took more than 3 years to develop and required a 12-month crunch period. The producer for *Crackdown* on the Xbox 360, a project that took nearly 4 years, also noted the significant lag times created by poor experimental tools:

[4.5] We developed the original *Diablo* with almost no proprietary tools at all. We cut out all the background tiles by hand and used commercial software to process the character art...The greatest deficiency of our tools was that they did not operate within our game engine. We could not preview how monsters would look in the environments they would inhabit. We couldn't even watch them move around until a programmer took the time to implement an A.I. Even after that, an artist would have to hassle someone to get a current working build of the game to see his creation in action...Our lack of tools created long turnaround times, where artists would end up having to re-animate monsters or make missing background tiles months after the initial work was completed. (Schaefer 2003:88–89)

[4.6] The testing of a single asset could take upward of an hour, directly impacting productivity and indirectly impacting quality since it naturally discouraged regular testing. (Wilson 2007:30)

[4.7] This sort of time-consuming, inefficient process remains the industry standard. Aspiring game developers, including those in India and the United States, and newly created video game companies continue to function this way because this kind of experimental environment seems to be necessary for the creation of video games. No better way has been proposed; nor have new infrastructures been proposed, even though new mechanisms are clearly necessary for this kind of work (note 6).

[4.8] Despite the inefficiencies inherent in the overall construction of video games, flexible technologies have become crucial components in game development, offering artists and designers the ability to alter game characteristics. Although engineering teams must frequently create these tools, they are at the core of what makes game development practice work. The lead engineer on *Battle Engine Aquila*, a game developed for Playstation 2 and Xbox over the course of two and a half years, notes the importance of flexible and modifiable systems:

[4.9] Flexible core technologies. As much information as possible was read in from externally editable files, and several custom editors for different areas of the game were written to allow designers and artists to alter everything from level layouts and unit statistics to graphical effects, without needing code changes...This approach paid off both by reducing the knock-on effects of changes and potential bugs and by enabling a lot of experimentation during the game's development. (Carter 2003:51)

[4.10] At their best, flexible technologies allow members of a game development team to work independently enough that they don't have to constantly depend on the work of others to continue progressing with their own work. The ability to experiment with ideas concerning the constantly shifting set of components that make up a game is essential. In the end, it becomes good design practice to have the ability to expand or contract game components without the intervention of several people. Individuals can work simultaneously on a single component.

[4.11] These technologies also interface with the numerous disciplines that birth them in game development studios. Without engineers who interface well with designers and artists, the result is technologies that do not bridge these ways of understanding the world; they merely reinforce the old ways. One of the most critically acclaimed games of 2007, *BioShock*, which was released on the PC and Xbox 360 and took nearly 3 years and 4,000 files to create, experienced such difficulties, as noted by the project lead: "Many of the processes and tools we used to develop *BioShock* were inefficient or confusing in implementation, leading to slow iteration cycles and bugs" (Finley 2007:26).

[4.12] These tools are, however, frequently written in the engineers' spare time, when the engineers are not needed to provide basic game functionality. These tools, if poorly designed, become hazardous to the health of a project. Too much time is spent attempting to do something that does not work, and rather than approach an engineer to understand why a tool or process is not working, many designers and artists are convinced that it is their fault.

[4.13] But even when tools function properly, the ability to experiment, if taken too far, becomes hazardous. The limit of experimental systems becomes problematic for game developers. At some point, limits must be placed on projects, and experimentation must be given direction. The lead engineer on *Battle Engine Aquila* comments on the double-edged character of these tools:

[4.14] Some of the systems were so flexible that they were being used for things they were never designed to do. While in some cases such uses were perfectly reasonable and even quite clever, in others they posed a major

problem. Code was not optimized to work in the manner in which it was being employed, and hence was running very inefficiently. Sometimes further functionality had been based on this behavior, leading to even more trouble when trying to optimize it. (Carter 2003:58)

[4.15] Similarly, the project manager on the game *Resistance: Fall of Man*, one of the release titles for the Playstation 3, notes:

[4.16] The flipside of homegrown tools and technology is that our tools changed quickly and our ability to properly train people on all the changes proved impossible. Building assets while simultaneously building the tools needed to create them is akin to trying to build a house on quicksand. Artists would literally open their tools one day and discover new interface buttons and have no idea what they were or how to use them. Many assets needed to be rebuilt, re-lighted, and/or re-animated because of changes to our builder tools. (Smith 2007:35)

[4.17] Interactivity and experimentation taken as a goal disconnected from the broader aims of a project, or without a plan or trajectory, prove just as unmanageable as a process with no tools. Without careful attention and planning, the tools that provide the backbone of a game development process can destabilize entire projects. Without training, artists and designers will attempt to do things that perhaps they should not. Engineers will not explain why something should not be done in a particular way or where specific requirements were derived.

[4.18] Interactive work has limits. For example, people must have time to work on and think about their tasks. Although the ability to interactively and experimentally work on projects may work well in many cases, the same tactics can disrupt teams and prevent them from reflecting on the tasks at hand. Rather than critically approaching a problem, they attempt to interactively and experimentally solve the problem. The lead tools engineer on *Asheron's Call 2*, a massively multiplayer online role-playing game (MMORPG), comments on how the process of development, if not carried out carefully, can result in systems created without respect to their surroundings:

[4.19] Coding before design. All the old lessons drummed into my head during school still apply: design in any complex software system is crucial and cannot be skipped. In the early phases of tools development, I tended to jump right into the code pile and start hacking out a solution to the problem. This caused no small amount of headaches when a seemingly small task blossomed into a days- or week-long struggle. (Frost 2003:46)

[4.20] This is the core problem of interactive and experimental systems. On both large and small scales, they can supplant the importance of taking time to observe a situation, reflect on it, make more observations, and then act on the basis of those experiences. Instead, meetings are scheduled, instant messages pass back and forth, e-mails are sent by a build system, or design parameters are tweaked in an effort to correct a problem, which may or may not be solvable with those approaches. Especially as deadlines loom, the desire for an immediate fix, rather than one that takes time to implement, becomes particularly detrimental.

5. Boss fight

[5.1] The ability to get at underlying technical and social systems is the core mechanism of this process; it is central to the creative collaborative practice. Without this capacity, the system begins to collapse. Game developers have spent years wondering why they have yet to find a sustainable equilibrium. The answer lies in pursuing those underlying systems and structures that (dis/en)able the creative collaborative practice. With an understanding of the systems of game development, it's my hope that game developers and workers in the industry will consider more broadly when those structures work well and when they fail.

[5.2] This analytical attention to creative collaborative practice allows us to continue to talk about core categories like work/play, but with greater specificity. By examining the ways that instrumental work/play and experimental interactive work/play systems plug into daily practice, we begin to better understand why spaces of playfulness in the context of work are crucial. This is true for video game development and for other work spaces where disciplines come together, mediated by creole languages or trading zones (Galison 1997). The use of experimental systems and technologies that bridge the numerous fault lines between them is also important (Traweek 2000).

[5.3] Although my focus is on video game development, these tools are also important for the fan/user of video game systems. One example is the numerous systems that *World of Warcraft* players deploy in their attempts to gain insight into the game's underlying systems and structures (Taylor 2006b). Creative collaborative practice is crucial for fan/user interest in video games because it marks the transition from user/player to user/fan or user/creator.

[5.4] In particular, the importance of experimental systems or tools can readily be found among fans/users. Fan/user activities are enhanced or enabled by experimental systems that not only allow them to create within game systems, but also provide insight into how those stories, which prompted them to join fandom in the first place, were constructed. For those interested, these same tools can often be used to better

understand and instrumentally determine where the bottom is for a given game system. As figure 1 shows, there are many components as well as missing pieces from the system that I have termed *creative collaborative practice*. However, it is this experimental system that allows us to find answers to questions we never had in the first place; this is what makes it productive in a research space. It serves as a tool for cultural analysis.

[5.5] One of the foundational aspects of creative collaborative practice is the ability and desire to pursue underlying technical and social systems. This is a playful practice that is also a great deal of work; it requires spaces of play and spaces of playfulness to mature and grow. Its current adolescent state has more to do with the design of game developing firms than an inevitable result of its structure. Looking at work/play and understanding its tendency toward excess offers important insights into understanding creative collaborative practice; similarly, it's necessary to understand the importance and danger of interactivity. The ways that experimental systems plug into our ability to understand underlying systems and structures also structure us; these ways are simultaneously enabling and constraining. If they are too open, then it is difficult to determine how to precisely reach our goals. If they are too constrained, we might jettison the entire system out of frustration. An equilibrium must balance the desires of developers and users/players, and although this equilibrium has yet to be found, it is my hope that the systems of creative collaborative practice can help navigate a solution.

6. Acknowledgments

[6.1] This material is based on work supported by the National Science Foundation under grant 0620903, titled "Videogaming, Work, and the Play of the New Economy." Any opinions, findings, and conclusions or recommendations expressed in this material are mine and do not necessarily reflect the views of the National Science Foundation. I also draw on research conducted as part of a larger dissertation project at Rensselaer Polytechnic Institute in the Science and Technology Studies Department (O'Donnell 2008), which was approved by their institutional review board.

7. Notes

1. I intentionally use the term *system* ambiguously. This essay examines the process by which video game developers come to understand and work within underlying systems and structures. If I were to say that a system was one single thing—the circuits of a game console, for example—it would limit the utility of concepts like *instrumental play* or *experimental systems* to perhaps one discipline or one realm of work. But the reality is that systems are ambiguous in the information workplace, and

it is my argument that these conceptual tools span these ambiguous categories in productive ways. Instrumental play of an actual video game is a task or process intimately connected with that of instrumental work in the context of 3D Studio Max by an artist.

2. I tend to differentiate this approach from that of previous work that engages with the complex interconnections between work and play (Dyer-Witheford 1999; Terranova 2000; Postigo 2003; Dyer-Witheford and Sharman 2005; Dyer-Witheford and de Peuter 2006; Yee 2006; Deuze et al. 2007). My focus is on better understanding why that work/play notion continues to stick despite what other researchers have shown. Instead, I am attempting to pull apart in a theoretical way what precisely makes work/play tick. This is not an attempt to theorize on a grand scale what work/play is, as others have done (Wark 2007). Instead, I attempt to tease out specific aspects of work/play encountered in ethnographic fieldwork and understand why they maintain their hold. I do this through connections with the work of scholars of science and technology studies as a means of encouraging greater coverage of the studies of work practice among those interested in work/play. The category of work/play serves as an experimental system that must provide us with different kinds of results to remain productive. Thus I focus on aspects of work/play rather than the monolithic category.

3. For example, in a forthcoming essay, I examine the way particular video game technologies and practices have come to encourage the consolidation and acquisition that now dominates the industry. The construction of "inter/intranetworks" through a combination of nondisclosure agreements, licensing agreements, patents, and copyright has resulted in a conservative industry. The continued emphasis on console video game development has created situations that make it difficult to have any kind of institutional memory. Further, these relationships between developer and publisher/manufacture are heavily managed and controlled. As I argue in the forthcoming essay, while these networks of (in)access frequently delineate studios that have proven their ability to develop video games, it does not indicate that those who fall outside lack those capacities. There are other factors that influence this as well: organizational structures, systems of systems, software, technologies, and institutions, both legal and corporate. There is continually the sense that there is something else out there influencing these worlds. These too are components of the overall system that video game developers are a part of, and each necessarily shapes the micro level of practice. However, attempting to capture that entire system here is beyond the scope of one essay.

4. Because this work draws heavily on the field of science and technology studies—and the anthropology and history of technology more specifically—I began my research

with the assumption that video games are socially constructed (Bijker et al. 1989). Though I do not elaborate here about how I specifically position my work among that of other science studies scholars, noting that video games are socially constructed would be a weak argument. My research thus attempts to understand the resulting implications and connections.

5. Some might ask, "No discipline? Then what are all of these programs out there that claim to be game design programs?" The answer is complex. One only needs to pose the question on an e-mail list (such as the Gamesnetwork list, whose archives can be found at <https://listserv.uta.fi/archives/gamesnetwork.html>, available to list members) occupied by many of the faculty and professionals who run these programs to be convinced of the thorniness of the issue. The International Game Developers Association Education Special Interest Group has drafted a curriculum framework to begin thinking through what a discipline of game design/development/production might look like. However, because of the interdisciplinary character of the work, the job is difficult. Each program that offers game design programs typically favors one aspect of the process over another. In many ways, this fragmentation is productive, allowing and providing for innovations and specialization in different areas. At the same time, it can misrepresent the situation by masking the complexity of what it takes to make games. This is further complicated by the numerous offerings that deliberately mislead students (Hoffman 2007).

6. Several more publicly available sets of game development tools do exist, such as the Unreal Engine and Unreal Ed, Microsoft's XNA Express tool chain, the Torque game engine, or the multiplatform game development environment Unity. Each of these tools has varying degrees of freedom and licensing agreements that structure them. While these are significantly important tools for the video game fan/developer community, they represent different technological, design, and corporate interests. Some are programming environments, others are most often used as level editors, and others are entire development tool chains. Compatibility, interoperability, and portability are not things that are found in this space, however. Each represents different models or approaches to development. Moreover, many game developers consider this to not be "real" game development work. According to many developers, until you have worked in a space where version control, automated builds, Max/Maya exporters and optimizers, and numerous other "normal" systems are in use, you are simply an amateur. I do not make any statement about the validity of these beliefs, simply that they are commonplace.

8. Works cited

- Bijker, Wiebe, Thomas P. Hughes, and Trevor Pinch. 1989. *The social construction of technological systems: New directions in the sociology and history of technology*. Cambridge, MA: MIT Press.
- Bucciarelli, Larry. 1994. *Designing engineers*, ed. Wiebe Bijker, Bernard W. Carlson, and Trevor Pinch. Cambridge, MA: MIT Press.
- Carter, Ben. 2003. Postmortem: Lost Toys' *Battle Engine Aquila*. *Game Developer Magazine* 10(4):50–58.
- Deuze, Mark, Chase Bowen Martin, and Christian Allen. 2007. The professional identity of gameworkers. *Convergence* 13:335–53. [doi:10.1177/1354856507081947]
- Downey, Greg. 2001. Virtual webs, physical technologies, and hidden workers. *Technology and Culture* 42:209–35. [doi:10.1353/tech.2001.0058]
- Dyer-Witheford, Nick. 1999. The work in digital play: Video gaming's transnational and gendered division of labor. *Journal of International Communication* 6:69–93.
- Dyer-Witheford, Nick, and Greig de Peuter. 2006. "EA spouse" and the crisis of video game labour: Enjoyment, exclusion, exploitation, and exodus. *Canadian Journal of Communication* 31:599–618.
- Dyer-Witheford, Nick, and Zena Sharman. 2005. The political economy of Canada's video and computer game industry. *Canadian Journal of Communication* 30:187–210.
- Finley, Alyssa. 2007. Postmortem: 2K Games' *BioShock*. *Game Developer Magazine* 14(10):20–26.
- Fortun, Kim. 2006. Poststructuralism, technoscience, and the promise of public anthropology. *India Review* 5:294–317. [doi:10.1080/14736480600938993]
- Friedman, Thomas L. 2005. *The world is flat: A brief history of the twenty-first century*. New York: Farrar, Straus, and Giroux.
- Frost, Paul. 2003. Postmortem: The tools development of Turbine's *Asheron's Call 2*. *Game Developer Magazine* 10(9):42–48.
- Galison, Peter. 1997. *Image and logic: A material culture of microphysics*. Chicago: Univ. of Chicago Press.
- Hoffman, Erin. 2007. Tighten those graphics: A look at small college game dev. *Escapist Magazine*, May 29. <http://www.escapistmagazine.com/issue/99/20> (accessed July 29, 2007).

- Huizinga, Johan. 1971. *Homo ludens: A study of the play-element in culture*. Boston: Beacon.
- Kelly, John D. 2006. *The American game: Capitalism, decolonization, world domination, and baseball*. Chicago: Prickly Paradigm.
- Kline, Stephen, Nick Dyer-Witherford, and Greig de Peuter. 2005. *Digital play: The interaction of technology, culture, and marketing*. Quebec, Canada: McGill-Queen's Univ. Press.
- Knorr-Cetina, Karin D. 1983. The ethnographic study of scientific work: Towards a constructivist interpretation of science. In *Science observed: Perspectives on the social study of science*, ed. Karin D. Knorr-Cetina and Michael Mulkay, 115–40. Beverly Hills, CA: Sage.
- Knorr-Cetina, Karin D. 1999. *Epistemic cultures: How the sciences make knowledge*. Cambridge, MA: Harvard Univ. Press.
- Latour, Bruno. 1987. *Science in action: How to follow scientists and engineers through society*. Cambridge, MA: Harvard Univ. Press.
- Martin, Emily. 1997. Managing Americans: Policy, work, and the self. In *Anthropology of policy: Perspectives on governance and power*, ed. Cris Shore and Susan Wright, 239–60. New York: Routledge.
- Murata, Taku. 2007. Postmortem: *Final Fantasy XII*. *Game Developer Magazine* 14(7):22–27.
- Neff, Gina, Elizabeth Wissinger, and Sharon Zukin. 2005. Entrepreneurial labor among cultural producers: "Cool" jobs in "hot" industries. *Social Semiotics* 15:307–34.
[doi:10.1080/10350330500310111]
- O'Donnell, Casey. 2008. The work/play of the interactive new economy: Video game development in the United States and India. PhD diss., Rensselaer Polytechnic Univ.
- O'Donnell, Casey. Forthcoming. The Nintendo Entertainment System and the 10NES chip: Carving the videogame industry in silicon. *Games and Culture*.
- Pickering, Andrew. 1995. *The mangle of practice: Time, agency, and science*. Chicago: Univ. of Chicago Press.
- Postigo, Hector. 2003. From *Pong* to *Planet Quake*: Post-industrial transitions from leisure to work. *Information, Communication and Society* 6:593–607.
[doi:10.1080/1369118032000163277]

- Rheinberger, Hans-Jörg. 1997. *Toward a history of epistemic things: Synthesizing proteins in the test tube*. Stanford, CA: Stanford Univ. Press.
- Schaefer, Erich. 2003. Blizzard Entertainment's *Diablo II*. In *Postmortems from "Game Developer,"* ed. Austin Grossman, 79–90. New York: CMP.
- Smith, Marcus. 2007. Postmortem: *Resistance: Fall of Man*. *Game Developer Magazine* 14(2):28–36.
- Taylor, T. L. 2006a. *Play between worlds: Exploring online gaming culture*. Cambridge, MA: MIT Press.
- Taylor, T. L. 2006b. Does WoW change everything?: How a PvP server, multinational player base, and surveillance mod scene caused me pause. *Games and Culture* 1:318–37. [[doi:10.1177/1555412006292615](https://doi.org/10.1177/1555412006292615)]
- Terranova, Tiziana. 2000. Free labor: Producing culture for the digital economy. *Social Text* 18(2):33–58. [[doi:10.1215/01642472-18-2_63-33](https://doi.org/10.1215/01642472-18-2_63-33)]
- Traweek, Sharon. 1988. *Beamtimes and lifetimes: The world of high energy physicists*. Cambridge, MA: Harvard Univ. Press.
- Traweek, Sharon. 2000. Faultlines. In *Doing science + culture: How cultural and interdisciplinary studies are changing the way we look at science and medicine*, ed. Roddey Reid and Sharon Traweek, 21–48. New York: Routledge.
- Turkle, Sherry. 1997. *Life on the screen: Identity in the age of the Internet*. New York: Simon & Schuster.
- Wark, McKenzie. 2007. *Gamer theory*. Cambridge, MA: Harvard Univ. Press.
- Weber, Max. 2002. *The Protestant ethic and the "spirit" of capitalism*. New York: Penguin. [[doi:10.1522/cla.wem.eth](https://doi.org/10.1522/cla.wem.eth)]
- Wilson, Phil. 2007. Postmortem: Realtime Worlds' *Crackdown*. *Game Developer Magazine* 14(9):26–31.
- Yee, Nick. 2006. The labor of fun: How video games blur the boundaries of work and play. *Games and Culture* 1:68–71. [[doi:10.1177/1555412005281819](https://doi.org/10.1177/1555412005281819)]