THE WORK/PLAY OF THE INTERACTIVE NEW ECONOMY: VIDEO GAME DEVELOPMENT IN THE UNITED STATES AND INDIA

by

Casey O'Donnell

A Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the degree of
DOCTOR OF PHILOSOPHY
Major Subject: Science and Technology Studies

Approved by the
Examining Committee:

____

Kim Fortun, Thesis Advisor

____

Mike Fortun, Member

____

Nancy Campbell, Member

____

Atsushi Akera, Member

____

Christopher Kelty, Member

Rensselaer Polytechnic Institute
Troy, New York

March 2008
(For Graduation May 2008)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

It can be difficult to know where to begin when recognizing the vast number of people who had a hand in shaping this dissertation. I believe however, it would be important to first recognize the numerous video game developers who took the time to speak with me, participate in interviews, and discuss my research. Though I cannot thank you individually, you know who you are, and know that I could not have done this without you. Most importantly, I thank everyone at Vicarious Visions, Dhruva Interactive, and RedOctane India for the time, access, resources, and copious amounts of coffee they provided me throughout the research process. Without their willingness to participate in this research, it would have been impossible. Their contribution is a testament to precisely the kind of creative collaborative interdisciplinary work that this dissertation is about.

The analysis and conclusions contained in this document are based on significant empirical research and analysis, and subject to my particular position and context. Being one of a very small number of anthropologists allowed into this community, I recognize the precariousness of this kind of corporate research. All critiques are done with the sincere and earnest hope of bringing useful and productive change to the work of game development and the video game industry. More importantly for me, however, is to connect the lives of my informants with others. What you do can inform what so many others do, it is critically important to better understand, and I thank you for the opportunity for being able to do so. I only hope that I have produced something that, as my good friend and key informant has put it, "doesn't suck."

The illustrations used in the development of the "Game Industry Game" in Chapter 8 were created by Tyson Stecklein and I thank him. Not every image he has created is featured in the dissertation, many are waiting in reserve for the development of a "real"

game based on this work. Tyson has been a longtime friend and constantly reminds me of what is at stake in thinking about the worlds that I find myself writing about.

A document like this is the product of a long-term trajectory; a story and character arc of my own muddling. I owe a debt of gratitude to the Computer Scientists and Mathematicians of Drake University that introduced me to this community and gave me the technical skills and vocabulary to work within it. John Zelle, Ken Kopecky, David Oakland, and Michael Rieck were particularly willing to work within my peculiar understandings and desire to learn. Vibeke Petersen and Joseph Schneider also deserve significant recognition for beginning my "descent" into the domains of Feminist Science and Technology Studies. Without them, my arc would have been quite different.

I would like to thank every faculty member I had the opportunity to work with while a part of Rensselaer Polytechnic Institute's Science and Technology Studies Department. Particular thanks goes to those who have made us a part of their families, it means a great deal. I cannot thank the department enough for the opportunity and compelling environment to work within. A graduate department is a product of its graduate student community as well, and RPI is fortunate to have a group that pushed my project in new and interesting ways. My cohort in particular and Shira Chess deserve recognition as well. The new graduate students have in recent years reinvigorated the department and continued to engage with this garrulous graduate student, and I thank them.

This project benefited greatly from the assistance of a National Science Foundation (NSF) Dissertation Improvement Grant (#0620903), which funded my work in India and the presentation of preliminary results and data. The anonymous reviewers of this grant proposal gave me significant feedback, which advanced and developed my research project. I hope the resulting product is in the words of one reviewer, "more thoughtful and reflective" than suggested by my original proposal.

I am greatly indebted to all the members of my dissertation committee, each who has made this creative collaborative process complex, enjoyable, educational, and possible. They have each shaped and improved this project in different ways, as well as making my experience producing this as pleasurable as possible. Kim Fortun, Mike Fortun, Nancy Campbell, Atsushi Akera, and Chris Kelty, thank you. Special thanks goes to my advisor and committee chair, Kim, who pushed my thinking and my character. Through this process Kim, Mike, Kora, and Lena who sort have adopted Andrea and I, thank you.

I would also like to thank my mom, step-dad, and mother-in-law for all of their support throughout my pursuit of this dream. To my mom in particular for all of the random conversations about poststructuralism at odd hours, and for knowing me better than I knew myself based on your margin notes in my books. Thanks to my dad and step-mom for their support, I know my trajectory has not always been a simple one. Thank you to my sister, Kate, who is just about to start her own arc. I love you all.

Though it may sound silly, I feel compelled to recognize Rambo and Scoshi, our two Siamese cats. When you spend a year in a "dungeon" on your own writing a text, they begin to keep you company. Rambo spent numerous days "assisting" the writing process from my lap.

Lastly, but most importantly, I would like to thank my wife, Andrea. She has seen more of this document than even the committee. She has been with me through all the ups and downs of industry work and academic life. She came with me to Troy and has made quite a career for herself. She has taught me the value of finding games in new places. She reminded me the value of being a player, a coach, and a referee, but never trying to be all of them at once. She has helped me to become a better person, scholar, and husband. Perhaps most importantly, I know full well the jokes she will crack about just how sentimental this section has become, and look forward to it. I love you too.

ABSTRACT

What can the everyday worlds of video game developers teach us about the "new" economy? How do these worlds differ across national and cultural boundaries? This project demonstrates how the creative collaborative practice of game developers and game development sheds new light on our understandings of work, the organization of work, and the market forces that shape and are shaped by industries in the New Economy. Video game developers and video game development in the United States and India is used as a window into understanding these complex issues. The ways in which game companies work both literally and figuratively, organize work, and the market forces surrounding them offers an opportunity to rethink ideas about the New Economy. Three years of participant observation at a game studio in the United States, and several months of fieldwork with game studios in India form the foundational data this project is based upon. This is further supplemented by more than 45 interviews, internal documentation, practices, and protocols from each fieldsite. Patent documents, legal cases, SEC filings, and press releases serve to further illuminate the forces and activities of game developers. Trade press and "enthusiast" press material is also used as a means to validate and further contextualize ethnographic data. New Economy work, exemplified by game development practice, is dependent upon and producing new modes of creative collaborative work practice. The way these practices play out and the structural conditions they play out within, however, simultaneously undercut creative collaborative practice. The dissertation connects the diverse forces and activities – laws, technologies, and workplace cultures, for example – that make creative collaborative practice central to the way the New Economy works. These same forces and activities are also capable of undermining collaborative practices. At the core of creative collaborative practice is the ability and ne-

cessity of being able to play with and get at underlying systems – technical, conceptual, and social. When access to underlying systems is undermined, so too is creative collaborative practice. By making collaborative practice the central concern, this project demonstrates how diverse systems across multiple scales come together in the context of New Economy work.

INTRODUCTION: VIDEO GAMES, WORK, PLAY, AND THE NEW INTER/INTRANETWORKED ECONOMY

0.1 Developers in the Mist

I never had any intention of studying the game industry. Then I stepped into the offices of Vicarious Visions (VV) in September of 2004. For the most part, I was open to studying any software company. I had come expecting to study the lived reality of work practice in the context of the "New Economy." VV was a medium sized independent game development studio, employing roughly 75 employees, a mixture largely of artists, engineers, designers, and various managers and support staff. At the time of my arrival, development had just begun on a game for Sony's, as of yet unreleased handheld console video gaming system, the Playstation Portable (PSP). The game was based on a forthcoming movie from a major movie studio in partnership with one of the largest comic book companies (among other companies) and VV had been contracted for the project by one of the larger game publishing companies. It was presumed that the title would be developed and released simultaneously with the June of 2005 release of the movie and shortly after the March release of the new game system. The project, code-named "*Asylum*," was tasked with producing a series of prototypes and levels which would then be used to determine if the rest of the development work would be entrusted to VV.

Asylum was behind schedule even when it was first being scheduled. It was behind schedule before there even was a schedule. Work had long since begun by another developer creating a version of the game for Microsoft's Xbox, Nintendo's GameCube, and Sony's Playstation 2 game consoles.[1] The PSP was an afterthought, a realization that

1. Later VV was contracted to produce a version of the game for Nintendo's Game Boy Advance (GBA) system.

a possible market for consumption might be missed. I watched as a team of talented engineers, artists, and designers toiled to create a prototype suitable to justify the remainder of the project. For some it was a chance to play with hardware that was only available to a handful of game developers. For others it was a chance to work on a "real" console title rather than on systems with greater hardware limitations. Others were excited about the opportunity to work on a game title linked to a blockbuster movie. Some developers felt a passion for the comic book characters contained in the title. Whatever the reasons, the team toiled day and night for four months creating the foundations for a new game for a new game system. This coalescing of corporate interests and developers desires creates the foundation for work/play practices shrouded in secrecy with a propensity for collapse under the extreme time pressures demanded by intellectual property (IP) holders.

By December of 2004 things were tense. *Asylum*'s pre-production had come a long way, but there were many moving parts that had to behave well for the game elements to come together. Code from engineers had to be in place to display special effects overlays created by artists. Animations from artists could not be displayed until the requisite data from designers were added to configuration files. Engineers were waiting for software development kit (SDK) updates from Sony to fix bugs found in the hardware or firmware of the PSP. By the end frustrated outbursts were common, with one element of the game breaking as other components were added. The automated build and "smoke test" system almost always seemed to be "broken" and everyone seemed to constantly be in a hurry up and wait mode of action. A few beers late at night while a daily build was executing sometimes took the edge off, though only for a few short hours before the process began again the next day. In late December a build was delivered to the publisher for evaluation and I quietly left the field from my pilot research for a winter break. I could not resist the feeling that this story was more complex than I had first thought. The numerous complex software systems, which were expected to be nearly real time and interactive, strained under the pressure of technologies in transition and the demands of developers. The experimental practices and instrumentality of work/play was quickly transitioning into the realm of "crunch" or mandatory overtime.

My unease, and that of my informants, was complicated by the November 2004 publication of a blog on the LiveJournal site by an anonymous poster "ea_spouse". Some of the concerns voiced by ea_spouse echoed those of the developers working on *Asylum* and for others it was a blip on the radar screen. It was something that people knew held relevance but certainly did not affect what they needed to get done to hit the next milestone deadline. The blog, written by the "significant other" of a game developer, voiced frustrations over work practices in the Los Angeles studios of Electronic Arts (EA).

> Our adventures with Electronic Arts began less than a year ago. The small game studio that my partner worked for collapsed as a result of foul play on the part of a big publisher -- another common story. Electronic Arts offered a job, the salary was right and the benefits were good, so my SO took it. I remember that they asked him in one of the interviews: "how do you feel about working long hours?" It's just a part of the game industry -- few studios can avoid a crunch as deadlines loom, so we thought nothing of it. When asked for specifics about what "working long hours" meant, the interviewers coughed and glossed on to the next question; now we know why.
>
> Within weeks production had accelerated into a 'mild' crunch: eight hours six days a week. Not bad. Months remained until any real crunch would start, and the team was told that this "pre-crunch" was to prevent a big crunch toward the end; at this point any other need for a crunch seemed unlikely, as the project was dead on schedule. I don't know how many of

the developers bought EA's explanation for the extended hours; we were new and naive so we did. The producers even set a deadline; they gave a specific date for the end of the crunch, which was still months away from the title's shipping date, so it seemed safe. That date came and went. And went, and went. When the next news came it was not about a reprieve; it was another acceleration: twelve hours six days a week, 9am to 10pm.

Weeks passed. Again the producers had given a termination date on this crunch that again they failed. Throughout this period the project remained on schedule. The long hours started to take its toll on the team; people grew irritable and some started to get ill. People dropped out in droves for a couple of days at a time, but then the team seemed to reach equilibrium again and they plowed ahead. The managers stopped even talking about a day when the hours would go back to normal.

Now, it seems, is the "real" crunch, the one that the producers of this title so wisely prepared their team for by running them into the ground ahead of time. The current mandatory hours are 9am to 10pm -- seven days a week -- with the occasional Saturday evening off for good behavior (at 6:30pm). This averages out to an eighty-five hour work week. Complaints that these once more extended hours combined with the team's existing fatigue would result in a greater number of mistakes made and an even greater amount of wasted energy were ignored.

...

EA's attitude toward this -- which is actually a part of company policy, it now appears -- has been (in an anonymous quotation that I've heard repeated by multiple managers), "If they don't like it, they can work someplace else." Put up or shut up and leave: this is the core of EA's Human Resources policy. The concept of ethics or compassion or even intelligence with regard to getting the most out of one's workforce never enters the equation: if they don't want to sacrifice their lives and their health and their talent so that a multibillion dollar corporation can continue its Godzilla-stomp through the game industry, they can work someplace else.

...

I look at our situation and I ask 'us': why do you stay? And the answer is that in all likelihood we won't; and in all likelihood if we had known that

> this would be the result of working for EA, we would have stayed far away in the first place. But all along the way there were deceptions, there were promises, there were assurances -- there was a big fancy office building with an expensive fish tank -- all of which in the end look like an elaborate scheme to keep a crop of employees on the project just long enough to get it shipped. And then if they need to, they hire in a new batch, fresh and ready to hear more promises that will not be kept; EA's turnover rate in engineering is approximately 50%. This is how EA works. So now we know, now we can move on, right? That seems to be what happens to everyone else. But it's not enough. Because in the end, regardless of what happens with our particular situation, this kind of "business" isn't right, and people need to know about it, which is why I write this today.
>
> If I could get EA CEO Larry Probst on the phone, there are a few things I would ask him. "What's your salary?" would be merely a point of curiosity. The main thing I want to know is, Larry: you do realize what you're doing to your people, right? And you do realize that they ARE people, with physical limits, emotional lives, and families, right? Voices and talents and senses of humor and all that? That when you keep our husbands and wives and children in the office for ninety hours a week, sending them home exhausted and numb and frustrated with their lives, it's not just them you're hurting, but everyone around them, everyone who loves them? When you make your profit calculations and your cost analyses, you know that a great measure of that cost is being paid in raw human dignity, right? (ea_spouse 2004)

The words of ea_spouse caused a ripple in the video game industry, one that is still being felt, though in different ways. The International Game Developers Association (IGDA) took up the call and encouraged developers everywhere to think hard about issues of Quality of Life (QoL) (Bates et al. 2004). Recently GameDeveloper magazine published a follow up article examining the success and failure of QoL efforts in the game industry (Hyman 2007). This appeal has since metamorphosed into several different IGDA sponsored projects, several of which I began participating in, viewing it as a demand made in vitro during my research project. Most of these projects are still uncertain of even where to begin; top down (management lead) or bottom up (instigated by the rank-and-file), neither direction has gained significant traction. The roots of the problem I argue are in the industry's emphasis on secrecy, closed networks of access, and the use of the State[2] to discipline those networks. The creative collaborative practice of the video game industry is simultaneously enabled and constrained by these conditions.

In January of 2005, a few months after I had begun my pilot research, Activision, Inc. (ATVI), one of the largest video game publishers, bought VV. ATVI was a competitor to the publisher that had contracted the development of *Asylum*. The ensuing press release was how I discovered the acquisition was taking place. Most of my informants found out hours prior to the press release. I emailed the lead designer who aided me in getting access to VV, asking, "What was happening with *Asylum* and what's happening with all of you?" The answers came back quickly: the acquisition by ATVI and the looming release of the PSP had convinced someone somewhere that *Asylum* would likely be unsuccessful, and the project had been shut down. Everyone was being transitioned to a new project assigned by ATVI. It was at this moment that I realized simply talking about "work practice" in the New Economy was a woefully inadequate way to look at the project I had undertaken. This was the video game industry. There was no away

2. While I draw out conceptually more precisely what I mean by the State in Chapter 4, the foundation is located in a Deleuzian notion of the State apparatus based on the "abstract machine of overcoding" which it "tends increasingly to identify with the abstract machine it effectuates" (Deleuze and Guattari 1987, p. 223). So one can, for the time being, substitute "the State apparatus" for "the State" in my writing. While I largely conceptualize of it in a Gramscian sense of "coercion" and "consent," (Simon 2001, pp. 24-32) I am particularly interested in the moment where the "perogative" power of the State is mobilized, the "'legitimate' arbitrary aspect" or "extralegal, adventurous, violent" aspects of the state (Brown 1995, p. 186). The moment when consent transitions to coercion.

around this, no matter how uncomfortable I was at first in saying it, I was studying the video game industry. Many of my emerging core categories and distinguishing characteristics, such as work/play, interactivity, Inter/Intranetworks, experimentation, and collaborative practice, were being further entrenched through corporate acquisition and consolidation.

In April of 2005 the book The World is Flat by Thomas Friedman was released. At the same moment as game development work was spreading to other parts of the globe, the two brothers who started VV traveled to China and India to speak with aspiring game development entrepreneurs. It was not lost on my informants or me that the issue of offshore outsourcing was soon to follow. To better understand this New "Flat" Global Economy I assumed it would only make sense to travel to one of these emerging game development sites. With my choices split between India, China, Korea, and a lagging Vietnam, I made the decision, to travel to the place that spoke English most widely. Also crucial was an introduction by the Studio Head of VV with the Studio Head of Dhruva Interactive, the self titled "premier game company in India," situated in Bangalore. The rapid growth of the IT sector in India and Dhruva's willingness to grant me site access made the choice easy. Also appealing was India's exploding outsourcing developments, and the U.S.'s dramatization of Indian workers.[3] It took 18 months to secure funding, access, and time to travel to India. During that time I was able to contact other development studios, some willing to provide me site access, others willing only to speak with me upon my arrival. RedOctane India in Chennai also offered their studio as a site which I could perform fieldwork at.

3. This particular aspect of globalization is one which I presented and published on early in my graduate academic career, looking at Wired Magazine's construction of Free/Libre and Open Source Software (FLOSS) and Indian New Economy laborers (O'Donnell 2004).

Finally, with the assistance of a National Science Foundation (NSF) Dissertation Improvement Grant, I traveled to India in November 2006. For two months I scrambled from field site to field site speaking to employees and Studio Heads of game development companies and aspiring developers in Bangalore, Hyderabad, and Chennai. I performed many hours of participant observation, ran numerous interviews, and had after work conversations with any game developer willing to share their perspectives of the global game industry. Dhruva and RedOctane were two studios willing to have me onsite for extended periods of time. FXLabs in Hyderabad, GameLoft Hyderabad, and Microsoft's Casual Games Group in Hyderabad were also accommodating, though reluctant to allow me to perform fieldwork over an extended period of time. When I had moments of free time I found myself being introduced to and conversing with more individuals interested in working in this emerging area of India's booming IT sector. Many simply wanted to know more about U.S. development practices. They pointed to the small number of resources available detailing how to make games at a level beyond that of a hobbyist. Sure they could find snippets of information here and there, but very little that they could use to inform themselves and their projects. I shared my Firefox game development bookmarks and news feeds with more people than I can recall. My time with Indian developers taught me numerous things, but the most important it seems is that all of the issues that face developers in the United States are frequently the same issues that Indian developers face, though frequently exacerbated by temporal and physical distance from the numerous networks and secrets which structure the industry. In some cases this distance protected Indian developers from the secret society syndrome rampant amongst U.S. developers.

The biggest commonality was a shared interest in learning more about what game developers do and how they can get better at doing it. Time and again however I found myself wondering, both in India and the U.S., "How in the world can the terrain of game development be thought of as flat?" It has been integral to this project to keep the lives and worlds of Indian and U.S. based developers connected as the story is presented, for at its core, that is what the New Global Economy is about. It is about the connections, often times so numerous and interwoven that we fail to recognize them. It is no longer about "our jobs" and "their jobs," it is about learning enough about how work gets done that we do not fail to grasp the importance of what has transpired. It has become a question about structure and how people play the game amidst these rules, which frequently we are never taught to question or seek to understand. The importance of being able to get at or seek out the systems and structures that underly systems has become an overarching concern of mine, in part exacerbated by what I understand as a decline in our right to pursue underlying social and technical systems.

0.1.1 Carrying out the Research: Methods

Throughout my fieldwork, my informants did not know how to define my position at VV. The inability to place me within existing understandings of what and who counted as legitimate members of the video game development community was problematic. For most this manifested in humorous ways, discussions of tribes, gorillas (silver-backs in particular), pith helmets, and mating rituals were common. For others I represented someone placed to determine just how much time they were wasting, or if they were expendable. For those people I was a threat, and kept at arms length. A few simply could not comprehend what value might be found in observing their world. Despite my best attempts to explain, they felt speaking with me would simply not be useful. To the remaining people, frequently those who became key informants, I represented a break or schism. I was a person outside of a system they felt they could not critique. Somehow I had been authorized to ask the questions that they could not. The inability to fit into the system made it possible to ask questions and make statements that would not otherwise have been vocalized. It was an opportunity to explicitly reconnect work experience to the political economy within which it is nestled.

I recorded most of my field notes digitally, in text files on my laptop. The majority of these files were created with TextWrangler, a freely available text editor for Mac OS X. I hand coded my notes, and inserted them into the bibliographic software Bookends which I used as both a means of data storage and analysis in grouping coded entries together. Interviews were also recorded digitally on my iPod via a Griffin Microphone adapter. I transcribed my interviews using ExpressScribe for Mac OS X and a USB transcription foot-pedal. Interviews were then coded using TextWrangler and entered into Bookends similarly to my field notes. All of these files were consolidated on my computer and encrypted using Apple's Mac OS X FileVault functionality, which encrypted the data using the AES-128bit encryption algorithm.

Four primary codes emerged from these activities, "work/play," "interactivity," "networks of access," and "corporatization of the state." Each of these primary categories was also broken down into sub-codes. I coded and re-coded literature which I was reading and had already read in Bookends. Using the "Smartgroup" categories in Bookends, I was able to create categories that automatically updated themselves with elements with particular codes - "work/play AND secrecy" for example. By selecting a Smartgroup I was able to see all the material I had associated with my codes. This was used to organize the material used in the text.

My interest was in the people who toil to produce the black boxes on the shelves. I excused my initial myopia of games simply as software as the view of an ex-engineer in the game industry. It had been the viewpoint of someone that had long since left the world of video game development for a more stable job, working with a company that produced one of the most widely used 3D modeling software packages by artists in the game industry. A year in graduate school, immersed in the literature of Science and Technology Studies (STS), had taken me far from the world of game development. Remembering the language, practices, and history gave me the ability to speak with my informants on their terms.

By February of 2005 a new project for the PSP was already underway. For the most part everyone was used to having me spend time at VV. I continued to show up and my biggest difficulty seemed to be finding available horizontal desk space at which I could sit, watch, and listen. Sometimes I would partake in conversations, asking questions, and attempting to better understand those aspects that I was not familiar with. I would attend meetings, sometimes with introductions to those who were not familiar with "their resident anthropologist," and sometimes not. I drank copious amounts of their coffee and partook in bagel/donut Fridays. I interviewed them and sometimes simply talked to them about life and work. I became friends, attended parties, and had evening beers with some of them, which as Chapter 1 demonstrates has become an important aspect of work. With a couple of my informants I submitted talk proposals to the Game Developers Conference (GDC). I answered their questions as best I could, though frequently unsure of what questions to ask, they simply were curious about my findings. They often talked about themselves and VV, as being distinctly different from other studios, though repeatedly the concerns they voiced were the same as other game developers. Some even looked to me to help find answers to their questions of QoL, a summons that I can only hope to contribute.

In the mean time, I continued my fieldwork at VV. I saw the development and release of numerous game titles by VV. I managed to see three projects started then canceled or transferred elsewhere. Game console systems came and went. Microsoft's Xbox 360 came and the Xbox left. Sony's Playstation 3 (PS3) came and the Playstation 2 (PS2) still hasn't left. Nintendo's GameCube (GC) never really arrived and the Wii has managed to take the world by storm. The Nintendo DS (DS) came and has planted its two little LCD screens firmly into the worlds of game developers. I gathered data from video game related news sites, blogs, web comics, and corporate websites. I saved every press release and Security and Exchange Commission (SEC) document I could find that might better contextualize the worlds of game development in a broader political-economic context. I first searched the U.S. Patent Office's online system and later turned to Google's Patent Search service once it became available.

In late April of 2007 after returning to the U.S. from India, I watched as VV released their most ambitious project to date, an endeavor that I had seen progress from concept to completion during my time there. VV had swelled to more than 175 employees and various contractors. At this point, I decided it was time to leave the field. For nearly three years I had sat with the developers: the engineers, designers, artists, and managers that produce the products we simply call "video games." These developers often disappear behind a single name: Activision, Sony, Microsoft, Nintendo, Miyamoto, John Carmack, Will Wright, Spiderman, World of Warcraft, Xbox, Wii, or Playstation. I began to see in the worlds of my informants a reflection of broader worlds of work practice. I realized that I had been attempting to ask questions about this world from upside down. The questions I needed to be asking should be simpler.

0.2 Taking Things Apart and Reassembling: Analysis and Arguments

What can the everyday worlds of game developers teach us about globalization and the New Economy? How do these worlds differ across national and cultural boundaries? What does the New Economy mean for what work looks like? The first two chapters in particular examine these issues, looking at the conflation between work and play leads to particular characteristics of creative collaborative practice. Chapters three and four investigate what structures those worlds and how creative collaborative practice is structured and disciplined, or enabled or disabled, in the context of the New Economy.

Ultimately the dissertation is about creative collaborative practice, and video game development serves as an exemplary example of what this practice looks like, situated in the New Economy. Creative collaborative work practice is dependent upon and producing new modes of work. The way that these practices play out and the structural conditions they play out within, however, simultaneously undercuts creative collaborative practice. Figure 0.1 represents how to think about the dissertation system with collaborative practice located at the center of of focus.

Each mode of work practice is both necessary, but capable of being pushed to extremes, where they can dissolve into "crunch" or destructive modes of work practice. Certain forms of work imaginaries, work practices, and ways of organizing work dominate. Work/play both enables new forms of work practice and conceptions of what work is, but it also creates possibilities for new forms of disengagement and exploitative practices. These are complicated by national and cultural boundaries. Desires and drives differ from one location to another. Secrecy dominates in one, while a desire for greater sharing and communication prevails in others. Tight control and access to work differs from place to place. The speed and "interactive" character of these workplaces becomes integral as well. Interactivity also points to the feedback loops within networks. These emergent structures are "played" by those within it. Interactivity cuts both ways; it gains workers and organizations freedom in new directions, as well as places demands on them for new forms of engagement. Interactivity can be useful for producing new ideas, concepts, and products. Yet these feedback loops also have the ability to fail dramatically resulting in the much publicized "crunch" mode of work within the video game industry where time at work and the intensity of interaction in the workplace rapidly increases. As other professions move towards work and organizational practices that now dominate the game industry, will both the positive and negative aspects be transferred as well?

Figure 0.1: The Structure of the Object of Concern

Central to all aspects of creative collaborative practice became am emergent category, "access to underlying systems and structures." The ability to be able to access underlying systems and structures, be they cultural or technical, is one aspect of this. Perhaps more importantly, however, is the desire to understand not just how they work, but how you can leverage those systems and structures to do work for you or how to make them function in ways which they were not originally intended. Creative collaborative practice depends on this ability to play with underlying systems and structures. The capacity and desire to pursue underlying systems and structures emerged as centrally important to the creative collaborative practice, exemplified by game development. With this in mind I began to recontextualize my arguments about those systems that (dis/en)able our ability and drive to pursue underlying social and technical systems/structures.

What kinds of forces - laws, technologies, collaborations, and workplace cultures for example - shape video game development and make it possible? Networks - corporate, individual social, internal computer, external computer, and collaborative networks, for example - seem to permeate the video game industry. It is so pervasive that frequently people talk about, perhaps overstatedly so, the game industry being an industry based more on social networks than others. All incredibly consequential for those hoping to get into the networks, or those who do not even realize they exist. Corporate networks of access enable certain forms of collaborative practice and disable others. These networks provide the foundation for the channels by which information and collaborative practices flow. While a "network" approach to examining sites of scientific or technological production is nothing new, what marks this project as different is that it examines how networks are structured and restructured over time. It examines how networks are extended, contracted, and actively maintained. It examines how networks are actively and passively opened or closed off. The concept of the Inter/Intranetwork is presented as a tool for understanding these networks, structures, and associated issues of access. Consolidation and acquisition has had dramatic effects on these networks, preventing new players, modes, and configurations from gaining access. The network is often talked about in vague terms. In many cases it is used simply as a means of pointing to the interconnections of heterogeneous elements. In this analysis I am more explicit, addressing which networks are connected and the effects of connections and disconnections.

The network's ability to force and reinforce certain kinds of relationships, activities, or simply rules, begs for better understanding how force is mobilized. The State is being mobilized to ensure control over production and distribution of the products of New Economy work. These control mechanisms alter the players, modes, and configurations present in New Economy work. In this respect the analysis is extended to the State, which has reemerged as an important aspect of analysis in the New Economy with its accompanying neoliberal ideology. In many cases this has lead to people talking about the "hollowing out" or decline of the State. Observing the game industry prompts me to gravitate the other direction, pointing to the ways in which corporations, via specific projects, are actually mobilizing the State. The State has capabilities that are unavailable to corporations. The analysis is extended to these projects, their outcomes, and the consequences for the kinds of work practice formations. It is the idea that the "Corporatized State" is being mobilized to meet the needs of corporations rather than simply dismantled. While the word "corporatize" points to corporate and business involvement in the processes of the State, the same processes can also be mobilized by individuals and organizations to meet their needs as well, though frequently resource availability or position limits these projects.

For example, many of the corporatized state projects designed to address changing characteristics of copyright in the digital era simultaneously disable career trajectories that many of my female informants have cited as integral to their involvement in the game industry. At the same time, these instruments can in and of themselves create instability in the system. While litigation is frequently aimed at large established organizations, they are more potentially detrimental to new players entering the system. For example, while the US Patent and Copyright system, designed to protect the intellectual property rights of corporations and individuals and encourage innovation, are being used as a means to discourage innovation and the entry of new global players into this arena.

Can thinking about the New (Global) Economy as a game offer us insight into its operation or provide us with new ways of engaging or questioning it? This entire system of networks, structures, relations, and play are synthesized and critiqued through understanding the video game industry as one game among many in the New Economy. New questions can be asked in this context. Is it fun to play? Is it a balanced game? How might it be redesign? It also provides a different way for people to understand and grapple with the analysis presented. Does it reflect the experience of work in the video game industry? What are similarities or differences between this game and those of other industries? What are the consequences of playing this kind of game?

These questions have provided the foundation for my analysis of work practice in the context of a rapidly globalizing video game industry. It is also an opportunity to present a complex system in a way that people are familiar. Games are often simultaneously complicated yet easily understandable. The format is useful, because it is done in a way that encourages players to understand the connections between the local and the global. The overall system matters, not simply the local or the global level. That they are inextricable and the best understanding comes from experimentation within that system. The ability to feel out those underlying systems is precisely what is so crucial to the future of creative collaborative practice. Game development and game developers as such became a window into a world of New Economy work, which is often talked about in vague and ungrounded language that frequently fails to engage with the lived reality of work practice. As well as being exemplary of the New Economy endeavor, the video game industry provides an affectual departure, which I have found fruitful for thinking about work practice. Video game developers are a humorous group. Most love all sorts of games. Their outlook and ways of understanding the worlds they live in provide a different kind of language that we can use to examine the worlds they occur within. Throughout my analysis of the material I have attempted to preserve this difference.

0.3 Gaming the System: Learning from Game Developers

My focus continually returns to creative collaborative work practice. Linking work practice up to the structures within which it is situated is the best strategy for understanding why work looks the way it does. The components of work/play and interactivity that emerge from my field site do so in relation with inter/intranetworks and rapidly corporatizing State. I do this; because the day-to-day realities of work practice leaves my informants with little time or opportunity to better understand their context. The focus is continually at the level of the local, fighting fires to keep work moving forward. While SEC and patent documents may seem distant from the everyday experience of work for developers, they are actually far closer than most realize. The connections between the local and the broader system is crucial to understanding the entire system, or game. As

Dorothy Smith writes about "ruling relations," it is the "heterogenous extra-local that organizes the local" (Smith 1999, p. 73). It is too easy to get lost in those everyday realities and not connecting them to broader structures, often leaving workers frustrated and disenfranchised.

Many organizations have also begun to encourage more playful and experimental workplaces. One indicator of this is Google's appointment to *Fortune Magazine*'s cover for the number one position amongst the "100 Best Companies to Work For." It is presented as a place where workers "can climb, play beach volleyball, lift weights, and go for a dip - without ever leaving work" (Lashinsky 2007). This Googlefication of the workplace has long been seen as a strategy of startups, technology companies, and more importantly video game companies. The distinction between what work and what play is becomes difficult to pull apart, and frequently displaces many other aspects of our lives. Work practice becomes inevitably intertwined with another core category, "work/play." This conflation plugs into a different set of drives, which enables and encourages workers to push harder and longer than they would otherwise. It also encourages them to forge new connections and think creatively. These new modes of work practice are simultaneously crucial, yet capable of being pushed to too far, dissolving into destructive work practices.

Rising levels of interactivity go hand in hand with the decreasingly hierarchical or "flat" organization that has been touted as a distinguishing aspect of work in the New Economy. Interactivity allows workers to experiment with the systems they both work within and create. Interactivity goes hand in hand with the connections between disciplines and cuts to the heart of what makes workers able to produce. This interactivity can also push too far, resulting in "infinite" meetings, emails, instant messages, and other forms of feedback and response. Interactivity can become the work, rather than whatever was originally intended. Real work only gets done after hours.

Policy makers and lawyers have not shied away from attempting to capitalize on the explosive growth of the video game industry[4] attempting to both simultaneously entice game development companies to their cities while simultaneously attempting to censor or penalize companies for producing game content which is deemed nebulously too violent or extreme. Lawyers have taken up the calling and scent of money to be made both from legal cases against companies for doing "harm" but also to encourage expanded use of patenting and litigious action against one another and their users. This project demonstrating the lack of foresight or ability to adjust to new contexts that policy has had, especially in highly technological industries like the video game industry. It also points to specific locations where changes could be made to encourage mutually beneficial outcomes.

This document hacks many of the disciplines that birthed it. Numerous disciplines have begun to stake out video games as their new territories. Often a single-minded approach on game worlds and content prevail. Games are seen simply as virtual environments to study within or new media to study. The newly emerging discipline of Game Studies suffers from this myopia most explicitly, not stopping to wonder about the broader networks, which its newfound demand is being produced within. This project proceeds with the assumption that video games are both media and technology, both of which are constructed. This construction occurs within extensive networks that have

4. The game industry had over $15.1 billion dollars in sales in 2006 and is projected to beat that number again in 2007, with most projections placing it at $15.9 billion in sales (Weber 2007). These figures do not include numbers for video game rentals, the sale of used video games, or money made from the licensing of video game intellectual properties (IP) to movie companies.

largely been ignored. The secondary task of the project is to place work practice in a contextualized setting, such that it comes into connection with the structures that affect and shape it so dramatically.

0.4 This Text's Software Development Kit: Situating the Text

Broadly considered, this text draws on three primary bodies of literature. I draw connections to relevant literature throughout the text where they are most productive in connection with my empirical material. I provide a basic situating in the introduction to provide scaffolding.

The first body of scholarship that situates this text is Media Studies and the emerging discipline of Game Studies. Though this study does not explicitly engage with the images or games produced by game developers, the insight and research in these areas provide a wealth of resources which to draw. Studies of online worlds and gamers provide insight into the world of developers, because most are avid gamers. They provide a foundation for understanding how or why a developer might make some of the decisions or pursue particular interests over others. Some look explicitly at the politics and economics of online worlds created by developers (Castronova 2005; Taylor 2006a). Others examine the different ways people play online or offline and how other aspects affect these, like gender (Cassell and Jenkins 2000). Some have made the turn more explicitly toward production and how user created Game Modifications (MODs) shape play spaces (Taylor 2006b). Some texts examine the issue of play and games, central to the theoretical foundation of this text. How and why people play, or the human or animal propensity for games and play offer extensive resources from which to draw upon (Huizinga 1971; Sutton-Smith 1998; Burghardt 2005). These texts are put into conversation with post-Marxist, Cultural Studies, and critical race theoretical frameworks that provide new resources for understanding hegemonic structures and hegemonic projects (Omi and Winant 1994; Hall 1996). Finally, other texts actually come into contact with game developers from time to time, though primarily in small doses, corporate approved doses, or based on fictional situations (Chaplin and Ruby 2005; Coupland 2006; Wark 2007). They point to some of the ambiguity and difficulty of working in the game industry, but stop short of offering new empirical perspectives. Most never make it beyond the big names and highly publicized meltdowns. Each of these texts provides framing and resources from which to draw upon.

The second set of literature that activates this text are studies of work and work practice. I include in this sphere studies of technological and scientific production, as they provide a wealth of theoretical and methodological resources for making sense of the worlds of video game development (Latour and Woolgar 1986; Pickering 1995; Forsythe 2001). These texts demonstrate the often-neglected social and technological aspects that disappear behind completed science and technology. They demonstrate the contingent and constructed character of these endeavors. Texts included in this category also point to the influence of political and economic aspects on the lives and approaches used by practitioners. The way gender and social networks structure labs and workplaces offers significant resources for examining the worlds of game developers. Also supplementing this category are studies of work and work practice (Suchman 1995; Orr 1996; Barley and Orr 1997a). These texts demonstrate both the importance of studying the everyday lives of people working as well as how little we understand about what work has become in recent times. In many cases work is much more complex and nuanced than acknowledged by management or those external to those professions. What is typically portrayed as simple or straightforward is often quite the opposite. Even "obvious" problems require much more skill and ability than is recognized.

Anthropological and sociological inquiries into work and work organization offer a wealth of theoretical and empirical resources from which to draw on. In particular, the examination of technical, engineering, and (new) media work indicate a significant disconnect between how work is conceptualized, these new breeds of workers "violate our concepts for making social sense of work" (Barley 1996, p. 412). It is with this in mind that researches have attempted to better understand the relationship workers have with their work, wondering, "what are we to make of someone who says they love their work and cannot imagine doing anything they enjoy more, yet earns so little that they can never take a holiday, let alone afford insurance or a pension" (Gill 2007, p. 9)? While it might seem at first glance that these jobs are different, "hot," "cool," or unpredictable (Neff et al. 2005) in ways that make them less indicative of work more generally, these types of jobs, work, and organizations are often cited as exemplars of our "Brave New World of Work" (Castells 1998; Beck 2000).

Rather than signifying these workplaces as distinct or different, all indications seem to be that this form of work and the organizations which support them "may become the modal form of work for the twenty-first century" (Barley and Orr 1997b, p. 3). So while these forms of work seem to be dramatically important in the context of the New Economy, they also prove significantly problematic for existing forms of organization. Initially it was assumed that a natural transition towards "horizontally" organized work would result more horizontal forms of organization (Whalley and Barley 1997). Rather, it seems that it has proven more contested, that despite "horizontal work processes, collaboration, rather than command, is the key to getting work done" (Zabusky 1997,

p. 130) most organizations balk at the necessary autonomy and trust that must be placed in the individual (Barley 1996). Some organizations attempted a kind of "industrialization of bohemia," that while at a surface level plugged into the self-image of workers interested in these emerging industries actually tended to be quite detrimental to workers lives outside the workplace (Ross 2003).

What many of these investigations agree upon is that more research in the corporate world are needed, precisely when they are becoming more and more difficult to perform (Smith 2001). More needs to be learned about how work gets done in contexts where frequently the work of individuals becomes lost or invisible and that in these new horizontal technology organizations it becomes much more frequent an occurrence (Downey 2001). Many researchers have attempted to reconstruct work practice in ways that encourage greater attention to the collaborative social aspects of the workplace (Suchman 1995; Suchman et al. 1999). These fieldwork centered inquiries indicate that it is this collaborative and social aspect of work in the New Economy which makes simplistic approaches to globalization, offshoring, and management particularly difficult (Hakken 2000b). It is in this same vein that this research attempts to resocialize and use the creative collaborative efforts of game developers as a means to rethink work in the New Economy.

The final set of literature that is drawn on are studies and accounts of the New Economy and globalization. These texts vary greatly in their empirical engagement with the New Economy and as such it is most useful to explicate their connection within the text. Some texts, which I place in this category, focus on modernity, postmodernity, and the changing position of the State (Lyotard 1984; Harvey 1990; Appadurai 1996). These texts often include insight into the New Economy, as it is one aspect of, or perhaps a result of (post)modernity ushered in by new communications technologies. Some point to the decline of the State and the consequences, like the rise of neoliberalism and the commercialization of formerly government run institutions. Others look at the new means and mechanisms by which organizations discipline workers and one another or make use of global differentials in monetary systems. Some texts look explicitly at the process of the information or New Economy and globalization (Tsing 2005; Kelly 2006; Varma 2006). These texts exemplify the importance of understanding these new global processes, and provide readers with an appreciation for the complexity of the New Economy by examining: the interweaving of corporate interests in historical and social processes, the global movement and training of new generations of workers for the New Economy and the different ways in which global workers are viewed and encouraged or discouraged from working together, and how globalization is experienced on the ground where conflict is experienced.

This document differs from most other media that has covered the video game industry. A small number of publications and online websites cater to game developers, offering new methods or reviews of development tools. Occasionally magazines like Wired, Newsweek, or Time will engage with the game industry, but infrequently with game studios. Journalists will swarm the most well known executives or game designers, but never rank and file developers. Entire magazines are devoted to the latest video games in development or recently released, and perhaps interview the games producer. The online enthusiast press observes all the meanderings of video game corporations but offer very little analysis. Each of these perspectives is useful, and can frequently access information and people that I cannot. In this respect they have been invaluable resources. However, this text is different in that the focal point remains on "normal" developers and work practice; the people who devote the majority of their time bringing these things into reality. It is also executed with an eye to better understand why they work the ways they do. It is about observing these worlds, which have disappeared, and better comprehending why things go right or wrong.

0.5 Learning to Use the Debugger: The Structure of the Text

Most of the developers I spoke with in this project came after or grew up as a part of the Nintendo Generation. This sense of a shared history and experience provides foundations for how video game developers talk about their occupations. Of course this is not really any different from other disciplines or environments of production where experience and language become entangled in ways that prevent broader accessibility. To make the work experiences of developers decipherable, the dissertation is structured in a way that provides readers with the tools necessary to disentangle the unfolding narrative of game development worlds as it progresses. I have begun to compare this approach to using a debugger, a software programmer's tool that allows them to observe the execution of a program's source code as it progresses. It also allows the reader to "step-into" functions, moving from higher levels of abstraction to lower and lower levels until reaching the assembly code, which is fed to the processor. The setup gives the reader the ability to "step-out" or move up a level of abstraction after moving lower. For us this means starting at the level of work practice and stepping into those other functions that run in the background.

Each chapter or "World" is broken into four Levels. One must "beat" each Level before moving forward to the next World in the text. These Levels are denoted by something like, "World 1-3." Each World or chapter ends in a "Boss Fight," much like video games often mark the transition from one segment to another with particularly difficult levels or enemies. Frequently, when facing a "Boss" at the end of a World (in a Boss Fight), it will be necessary to bring those tools you've gained along the way to bear. Each Boss is progressively more difficult to battle, because it requires bringing all of your acquired resources and skills to bear. So, like a boss battle, these concluding remarks re-emphasize the argument of the chapter and the Levels the precede it. The Boss Fight also denotes your progression from one aspect of the story to the next. This practice is explained further in Chapter 1.

At Chapter 1, the dissertation begins at the level of work practice, examining the worlds of work and play of game developers. It looks at four aspects of work/play that plug into work practice: secrecy, instrumental play, experimental play, and the zone. These aspects come together ultimately in a system, which almost naturally drives towards "crunch" modes of production. Closed access and secrecy lend an air of mystery to the game industry, which demands that one "make games before you make games." "Titles" matter more than the work itself, creating a situation that is problematic given the politics associated with game crediting. The demand that game development in many ways be an "instrumental" game, where its players must assume a particular style of play, has significant impacts on who enters the game industry and how the game eventually gets played. All of these combine in the "experimental systems" which developers create to enable the development of video games. The desire to tinker and have highly responsive feedback loops between one's work and the final game, result in a tightening relationship between worker and product. And the delicate balance between "the zone" and spiraling forms of disengagement in the context of work/play practice creates a potential for collapse. It is this balance between the necessary and productive aspects of work/play and the drive toward excess, which taken too far crumble in on themselves.

Chapter 2 examines more closely how work practice is organized and technologized. The chapter focuses at the four primary areas of expertise in game studios: engineer, artist, designer, and manager. How these disciplines communicate and interact with one another and their product shapes work practice. The means by which they attempt to bring numerous technologies together into one technological system must be made to work cleanly and efficiently if they are to produce games on time and within budget. The interconnection between disciplines is integral, though it can also be unpredictable and complex for those attempting to manage or work within it. The numerous forms of feedback and communication that forms the foundation of this system can become goals in and of themselves, quickly becoming counterproductive. The ways in which these organizations and technologies plug into the work/play practice is also explicated.

Chapter 3 takes another step into the debugger, asking questions about why work has been organized in particular ways. What other institutions and activities have influenced the daily lives of game developers? It places game developers and work organization and practice in the broader context and structure of the game industry. Corporate networks enable and disable certain forms of collaborative practice. They also provide the foundation for channels by which information and collaborative networks form. Those same channels can be disabled or acquired. These networks can facilitate or impede the arrival of new players, modes, and configurations. The chapter traces the connections between organizations and how those ultimately come back to influence the work/play of game development.

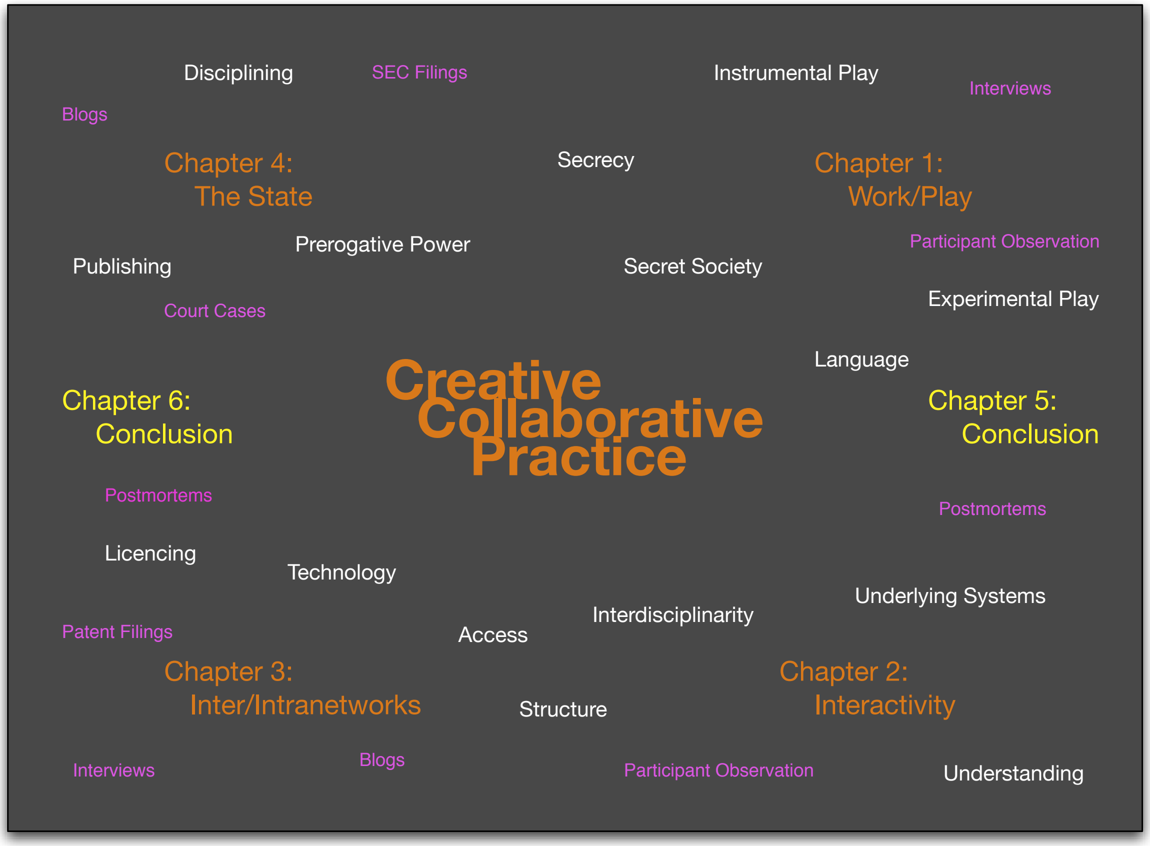


Figure 0.2: Diagram - Structure of the Program

Chapter 4 connects the institutions and activities more broadly with the State, and the ways in which companies within and outside the game industry have mobilized the State apparatus. The nearly 30 years of interaction that the game industry has had with State institutions and apparatuses have resulted in structures, which developers must navigate. I examine the ways in which the same companies that have mobilized the State apparatus to their benefit have systematically attempted to privatize those operations, which have previously been under its administration. The mobilization of the State on the part of corporations to ensure control over the production and distribution of the products of New Economy work ultimately affects who becomes involved in the creative collaborative networks of the game industry. The resulting State powered systems both compel and constrain the possible futures of creative collaborative practice, and are just as deserving of scrutiny in our analysis of work. Even these seemingly distant structures come back to influence and discipline work organization and practice for game developers.

The concluding and "conclusion" Chapters 5, 6, and 7 represent the numerous systems presented in the first chapters "at play" or at work. They also demonstrate how these systems are open to change and modification. These chapters demonstrate how industries in the New Economy, and the game industry in particular, are dependent upon new modes of creative collaborative practice. Throughout the dissertation I connect the diverse forces and activities - laws, technologies, and workplace cultures, for example - that make creative collaborative practice central to the way the New Economy works. These same forces and activities are capable of undermining collaborative practices. At the core of creative collaborative practice is the ability and necessity of being able to play with and get at underlying systems - technical, conceptual, and social. When access to underlying systems is undermined, so too is creative collaborative practice. By making collaborative practice the center of my focus, I am able to demonstrate how diverse systems across multiple scales come together in the context of New Economy work.

Chapters 5 and 6 take the format of a "postmortem," a format familiar to many game developers; they contain explicit examinations of "what went right" and "what went wrong" in each of the first four chapters. They re-articulate the arguments of the first four chapters by connecting them up to the experiences of other game developers as written about in the trade press. It is done as a means to help developers understand how their local experiences influence and are influence by those systems they tend not to pay attention too. It takes quotes from Game Developer Magazine Postmortem articles and connects them to the local experiences of my informants and the arguments of the first four chapters. What aspects of work/play serve productive purposes? Which ones spiraled out of control? What does interactivity buy us, and at what cost? Our networks do work for us, but is it all desirable? The State is being mobilized by corporations to meet their needs, but have the outcomes been valuable? Each Chapter takes these lessons and crafts specific recommendations for academics, practitioners, and policy makers. These chapters make the argument that particular interventions can be made at four different levels - secret societies, standards/tools, copyright/patent, and content - that I believe would alter the trajectory of less desirable aspects of New Economy work. It returns to the overarching argument, that the ability to get at underlying systems is integral to the future of creative collaborative practice at the core of New Economy work. However, as currently configured, this ability is being eroded and ultimately will have massively detrimental effects, both at the local and global levels.

Chapter 7 is the analytic conclusion of the dissertation. It returns to each of the theoretical concepts introduced in Chapters 1 through 4 and unites them as part of the broader system under examination throughout the text, that of creative collaborative practice. Each concept: work/play, interactivity, the inter/intranetwork, and the corporatization of the prerogative power of the State is returned to in turn. The way they come together in the context of the creative collaborative practice of the video game industry and the consequences of this system are brought to the foreground. It is a higher level articulation about this broad system, which is at play and played in the epilogue chapter.

The epilogue chapter (Chapter 8) synthesizes much of the dissertation, consolidating and presenting the material in the format of a "vertical slice" of the game, in the form of a game-play narrative followed by the game design document. These provide the reader with a feel for the game play, description of the design, rules, and play of the video game industry game. This is one of the chapters of the dissertation I hope to elaborate on in the future. Both the language and perhaps artifact can contribute to an improved understanding of the ways in which work practice becomes entangled with broader forces. Games are a useful means to understanding complex systems, because the overall system must remain in focus, not simply the local or the global level. There is a feedback loop between the system and the local conditions, they are inextricable. It also provides the opportunity for players to feel out those systems for themselves, perhaps determining their own interventions or conclusions. Those are ultimately the skills integral to New Economy work.

For those interested in the nitty gritty details of research design and methodology, I have placed the majority of this material in the appendices. They are provided for the reader and future researchers.

PART I. ANALYZING GAME DEVELOPMENT WORLDS

CHAPTER 1
WORK/PLAY: DRIVING TOWARDS "CRUNCH"

This chapter marks the entry point into the world of video game developers. Much like a player just starting a game, a foundation must be provided. A tutorial or introduction must occur, or the player (that is you) is left with no idea of what they can do or are expected to do. World One is a collection of four "Levels," much like a game. Each Level provides experiences or tools which will serve the player in the "Boss Fight" at the end of each World. The Boss Fight in a game often requires a player to bring the lessons learned, or tools gained, in order to progress to the next world. Each Boss Fight also marks the transition from one stage of the game to the next. This process is usually introduced in the first World and Levels of a game, such that the player can become familiar with the flow of gameplay. In a similar fashion this chapter introduces you to the worlds of game developers, a process necessary for using the game industry as a lens onto broader social and political issues. It can be thought about as a collection of four Levels of the first World. Subsequent Worlds retain that structure. In the meantime you might just learn something new about a world that you have yet to encounter.

The chapter steps-into work and play, work as play, or work/play, a recent and popular conception which I seem to be encountering ever more frequently. There is purported to be (new)found conflation between work and play by social analysts, frequently in connection with video games and the work of creating them or working within them. But then, most of the existing accounts of work/play do not do any real analysis or unpacking of the situation. Rather they often point to a couple of instances of what can abstractly be called work/play and hope that the vagaries of the mark work/play will do the labor of actually tugging apart what that means and how it operates (Terranova 2000; Kline et al. 2005; Yee 2006; Deuze et al. 2007; Wark 2007). Of course the move is somewhat excusable given the complexity of acquiring access to field sites, a point that is also rarely examined or unpacked. In many cases source material is limited to relatively few informants, or material, which has been filtered through a process of people, time, and access restrictions; points, which again are ignored in favor of striking on the hot anvil of work/play.

Each section of the chapter lays out the empirical and theoretical foundations of aspects of work/play that emerged from the research site. In some cases work/play is not so new, other aspects seem to be plugged into our system in new or at least freshly configured ways. Certainly there are other components that plug into work/play, though only four are extrapolated here, a pattern that the attentive reader interested in tugging apart the underlying game of this system (those "power gamers" out there) will soon recognize. In a similar fashion, each aspect of work/play also progresses down through our debugger starting at the top and working down through our system of systems. The first Level, "World 1-1" focuses on the "networks" of work/play access. Level two, or "World 1-2" moves into and examines the fun of work/play amongst rigid structures, an enjoyment of instrumental play. The third section of this chapter, "World 1-3," is our first insight into notions of interactivity and feedback, which plugs into work/play. The final section, "World 1-4," leads us to desire, and the consequences of the "zone," which if taken too far, collapses work into "crunch" or "autoplay." The first "Boss Fight" is always simultaneously the most difficult and easiest. The Boss Fight for World One returns to the lessons learned at each Level and reminds the player that these tools will be crucial to understanding the remainder of the game. The Boss Fight marks our transition to the next World, or set of four Levels and a Boss Fight.

Crunch really starts with one dilatory night, which then turns into a dilatory morning, which then turns into another late night, and eventually you still have to come early, but you are at the office until late at night. Then you come in on Saturday, because what the team had been working on Friday night (which was really Saturday morning) didn't quite solve the dilemma. At some point, you just resolve to keep at it until the current crisis has been averted.

There is a cot in one of the conference rooms, so you sleep at the office occasionally to avoid driving to and from work, or you take a cat-nap, because you are dead tired from the lack of sleep. But, the upside is that to keep people focused, and because everyone is working so hard, food starts getting catered in for lunch and dinner. There is always a plentiful supply of caffeine, though there is nothing new about that. By this point, the office has begun to take on a peculiar look. Food cartons, beverage cups, and cans litter desks, except for a couple of the clean guys. They always talk about how nasty everyone else's desks are. A couple more cots appear here and there. It begins to resemble a college dorm room, or squatters who have come to occupy a computer lab. It starts off fun.

Sometimes, especially late at night, when you have already put in 12 hours, but know you have a few more left, but you need a break; some beer will show up. People may take a break and play some games, because you really haven't done that in a while, besides playing the game you're working on, and it isn't fun anymore. That might mean getting home a bit later, but at least you have had some enjoyment in the day. No one really said that you have to be here tonight, but you stay, because so does everyone else. The trouble begins when the deadline is getting dangerously close. People start getting on edge. Their fuses get shorter. Sometimes someone breaks something which leaves you unable to test your work for a couple of hours. Eventually that someone will be you, so you don't get too upset about it, but you can't help feeling a bit peeved.

Just before vertical slice the art team was trying to get several newly created full screen effects placed into the game. Something was going wrong somewhere in the progression from its export and appearance in the game. Time and again the technical artist walked from his room to engineering attempting to find the solution. Engineering would make one change, commit it to version control, then the technical artist would sync his system to version control, then run the game again. Time and again the same result occurred, the effect did not appear as it was generated in the artist's tools.

As early evening turned to night, the process became more frustrating. Engineering would only respond with a brain dump of their understanding of the effects system. Emotions began to run high as the technical artist, who had been hoping to get home in time to have dinner with his wife, began to realize that it would be impossible given the current progression. It almost made sense when a shouting match between the tech artist, his computer, and eventually an engineer ensued.

Table 1.1: Crunch Stories: Crunch Begins...[5]

1.1 World 1-1: Work/Playing the Imaginative Secrecy Networks

Secrecy surrounds numerous aspects of the video game industry. In some respects this attempt to hold itself apart, as distinct from other industries, lends it a mystique or desirability. This is not work like other people's work, this is not "real" work or "ordinary" work; this is game development. The presentation of game development work as hard, but separated, an intermezzo or interlude, provides a kind of cachet. As cultural historians and scholars of "ludology" 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, p. 10).

Nearly every conversation with game developers begins with a disclaimer, "not that I (we) are representative of the industry more broadly," or "we do things a bit differently here from the other studios you have probably visited." While not always intentional, the consequences of and symptoms of the secrecy, which surrounds the daily worlds of game developers, has dramatic consequences for those who choose to work in these

5. Each of the three "Crunch Stories" featured in World One are fictional, though they are rooted in reality. Each is based upon field notes as well as the stories and experiences of developers I worked with and spoke with throughout my research. Any similarity to those living, dead, or imaginary, however are purely coincidental.

worlds. Sociologists of New Economy work and New Media workers have noted, 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, p. 321).

Secrecy plays out in different ways amongst individuals, disciplinary divisions, corporations, and inter/intra-corporate entities. The focus on the individual in this World is not a statement about where secrecy pervades, but rather that it is foundational in a way more fundamental than the disciplinary boundaries examined in World Two or the inter/intranetworks of World Three. The importance of knowing those unknowable aspects, the "Konami Code" for example, pervades inter/intra-disciplinary or professional/hobbyist/independent distinctions. Secrets matter between engineers, engineers and artists, managers and leads, studios and publishers, even manufacturers and publishers. Secrecy crosses scales as well, the secrets of console manufacturers matter for artists and engineers everyday activities. While I detail more specifically how secrecy connects up in other Worlds, it is necessary to examine it in a context that precedes the distinctions made in later Levels.

I have cut what I term "imaginative secrecy networks" in four different dimensions. While they might be cut other ways, or other dimensions, these best reflect my ethnographic experiences in connection with prior research and literature. Each dimension has potential benefits, though when taken to the extreme can become detrimental. Secret societies provide a foundation for communities of practice, yet taken too far can lead to isolationism. This is the difficulty with work/play, each aspect is both necessary, but capable of excess in a way that renders the player less capable.

One of the most fundamental dimensions of imaginative secrecy networks is the importance of "speaking the language" of the industry, video games, and video game development. This is particularly integral for "Networking," (the second dimension) which has become one of the most important professional activities for game developers. It is instrumental in forging social connections with people already working in the industry. Once adequate "initial capital outlay" (Neff et al. 2005, p. 318) has been invested, only then can aspiring developers "break into" (the third aspect) the game industry. Finally, the importance of "game titles" (the final dimension) as source of credibility or social capital in the game industry is examined. Each dimension provides insight into how the "imaginative secrecy networks" are formed and managed in the game industry.

1.1.1 A Shared Konami Code: Speaking the Language

Language is a precursor to many of the other barriers or secrets of the game industry. If you cannot access and understand the language of those who work in the game industry, you certainly cannot play the game.[6] You may or may not have noticed the name of this section begins "World 1-1." This carries with it, a reference to a video game, Super Mario Bros. released in October of 1985 for the Nintendo Entertainment System (NES) (Nintendo 2003); a staple in the gamer vocabulary, if not a temporal reference much like "A.D." or "B.C." for non-gamers. Most game developers have an extensive history of playing games. 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 to be

6. And so too is it important for researchers interested in studying the industry to be acquainted with these languages. Without this knowledge it would have been impossible to gain access to my initial field site. It was actually a conversation about game development and project planning at a party that began the friendship, which provided me the opportunity to do pilot research at a game studio.

gamers have at least one of the latest console systems. They try to play, at least casually, the latest and most popular game titles.

Figure 1.1: Super Mario Bros. World 1-1

World 1-1 is the first level of the game. Rather than hold the game player by the hand, explaining to them the functionality of the world in which they have been thrust, Super Mario Bros. asks the user to intuitively test out their new world. The cross-shaped controller in your hand when pressed causes a blob of color to move left and right. A button causes it to jump. A box which you might want to examine or interact with is labeled with a "?". It is an invitation to play in this world, to interact with the objects it contains. Other entities in the world move around and scowl at you with angled eyebrows, an indicator of ideally less than honorable intentions. Others chomp their pixelated mandibles at head or abdomen level, giving you more than enough idea of their pointy intent. A clock ticks down towards zero, an abstraction of life we are all too familiar with. These are the kinds of abstractions game developers are frequently forced to engage with; too much to investigate, learn, and do in too short a time.

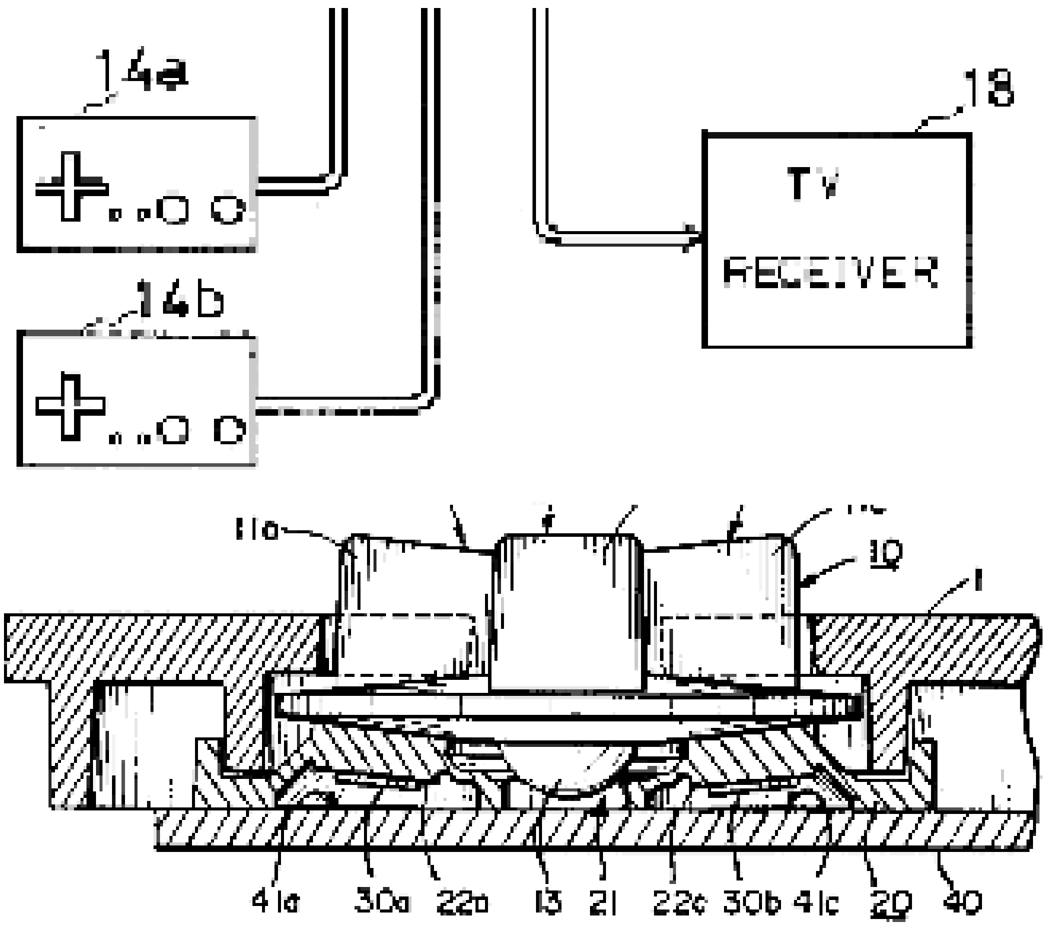


Figure 1.2: The NES's Iconic Controller (Nakagawa 1985; Shirai 1985)

While the NES was not the first video game console introduced in the U.S., it is the console which many game developers began their gaming lives playing with. Nearly every developer talks about the impression that the NES made on them. The following quote comes from an online email discussion with an informant about the "Konami Code," which was first found in the game *Contra* on the NES. It was a series of movements on the game controller which provided the player with extra "lives." Many gamers are quick to wax nostalgic about these kinds of experiences with early video game console systems.

> Oh! Those were *the* days.
>
> Circa 1988-92, at my hometown in India, we used to get the game console and "2 free" game cartridges for Rs. 20 per hour (approx $0.50) and then the prices came down to Rs. 5 per hour and Rs. 25 for the whole night (evening to next morning). And then those "5 games in 1" cartridges were available at extra cost.
>
> *Super Mario Bros.*, *Contra*, *Donkey-kong*, *Popeye*, *Road Rash*?
>
> *sigh*
> (Informant and O'Donnell 2006)

Games dominate the language of both work and play for gamers and game developers alike. It would be inadequate to think that this is done as a mechanism to keep others at bay or explicitly exclude. This vernacular also does work for game developers. Games provide discursive resources for developers trying to describe abstract concepts, like game mechanics. Because there is no "discipline" of game design or game development, games themselves have become a kind of lingua franca. When you think and talk through/with 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 amongst the developers at VV. Like many of my readers, I had never heard the term before. Vertical Slice for game developers 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. When deadlines loom, more meetings crop up; the contradiction seems inevitable. In one particular meeting the project leads and upper management were attempting to determine the mechanics of a potential WiFi multiplayer aspect of the game. A game mechanic is the underlying "game" which 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*. Of course meeting participants were already exhausted from recent long nights and the thought of having to define a new game mechanic and underlying technology to support it was low on everyone's priority list. Engineering was saying one thing, design was saying something else, and management yet another. 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?"[7]

7. "Spy vs. Spy" was originally a Mad Magazine comic strip which was later inspiration for a video game on several different console game platforms. It is a game with a long history of its own, but the mechanic, which interested my informants, was the idea that rather than direct combat (ie. shooting or punching your opponent) the mechanic they referred to was that of setting traps for one another.

Figure 1.3: Spy vs. Spy for the NES

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. So, much like other anthropologists examining the worlds of knowledge work have noted, game talk and game development 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, p. 122).

While game talk can be a productive tool for uniting disparate disciplines (a topic covered in World Two in more depth), it can also be used to exclude. Anthropological studies of high-energy physics have shown how oral communication enables collaborative communities, but also simultaneously can be used to close communities off. Many of the same things can be said of game development as well.

> 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, p. 120)

The closure of this community, based on access to oral communication, proved especially troublesome for many of my Indian informants, many of whom had not grown up with console video game systems. However, this quote seems to favor the notion that oral communication is primarily being used to maintain boundaries. It is also used as a means to convey information for which my informants had no other language for. It is reductive to consider it only a means of exclusion. The use of game-talk serves productive capacities crucial to the collaborative capacities of developers. It is when it is taken too far that it becomes problematic, the idea that if you cannot speak our language, you are not worth talking to or listening to.

While some of my Indian informants were avid gamers on personal computers, few had grown up and had experience with console game systems and had not necessarily played the many games which were released on the NES, Sega Genesis, Super Nintendo or numerous other consoles introduced through the 1980s and 1990s. Lack of experience with console games was a difference frequently exacerbated by the idea that games were "mere" entertainment, and time was better spent by young people studying, rather than playing games. Similar difficulty was encountered by 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. Indian game studios, rather than expecting their employees to already be versed in the language of games, have in many cases built rooms where employees can play games in 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 which U.S. developers frequently reference in their day-to-day exchanges.

Though examined more closely in World 5, it is also incredibly consequential that access to oral communication is closely controlled. Anthropological studies of technical work practice have demonstrated that the closed access to oral communication inhibits our understanding of what goes on in the workplace. This has consequences both at the level of the corporation, but also more broadly in how game development work is perceived. The frequent assumption that game development is "merely playing games" is, in part, a consequence of this.

> "One of the interesting results of the ethnographic investigation of work practice is that one discovers that what is done on the job is often rather more than and different from the job as described by the corporation." (Orr 1991, p. 12)

Because game development work practice is quite different from what it is thought to be, and developers do not communicate actual practices, it becomes difficult to learn from and understand the experiences of others. While companies continue to invest in ways of attempting to capture this information, because it is often rooted in social relations in addition to technological ones (Hakken 2000b), that may remain impossible to access or store.

Most aspiring U.S. developers have become accustomed to needing to learn the language of game development on their own. For example, being able to understand someone when they say, "Like in God of War's context sensitive button-pressing minigames," was a reference to a game for the Playstation 2 console, with a particular game mechanics that require the player to time precisely certain button presses that resulted in unique or context sensitive actions. Websites like *GameDev.Net*, which provide information for developers, hosts sites aimed at helping would-be engineering developers learn the tools of the trade, like *Nehe Productions*, *Code on the Cob*, and *AngelCode*. The information on these sites is heavy on technical issues and the concerns of engineers in learning game development. This engineering-heavy focus and lack of available information about how design, art, engineering, and management come together to produce games is an aspect of the game industry that is not widely discussed in explicit terms. My India based informants and inexperienced U.S. based developers were most inquisitive about this component of game development. Engineers understood that they needed to be able to work with models, textures, sounds, and physics. The most difficult aspect was the construction of "pipelines," or means and methods of simply and quickly bringing artwork from artists and data from designers into the underlying code of the game.

1.1.2 Becoming a Professional Networker

Like other sociological and anthropological studies of new media work, the social work required by game developers to maintain connections within a community are "rooted in similar avocational" or "professional" affiliations (English-Lueck and Saveri 2001, p. 8). The "tightly interlinked social settings, the network economies of production that fuel creative industries" (Neff 2005, p. 150) have emerged as intricately important to both working in new media production, as well as to gaining access to these industries. After-hours events, "cocktail parties, seminars, and informal gatherings" have been said to be integral to the "rapid dissemination of information" (Neff 2005, p. 135). Game developers have less interest in the rapid dissemination of information, which is frequently tightly controlled either by convention or through Non Disclosure Agreements (NDAs). The technological and political control of information access is covered in more detail in World 4. Instead the focus is on social networking, getting to know people and generally what they are working on.

The Game Developers Conference (GDC) serves as an excellent example. Students interested in careers in the game industry are given "scholarships" to the event, which is one part electronics/entertainment expo and one part professional conference. For many developers, the sessions are found wanting in detail, since despite claims to the contrary, much of the actual information about how things are done is withheld from fellow developers. Frequently, it becomes a chance to hear well-known voices in the game industry talk about past and future directions, or studios doing new work to present abstract information about the challenges and promises of the technologies which they are attempting to put to work in their games. Anyone who has attended the event regularly brings hundreds of business cards and spends the majority of their time chatting with other game developers. Many studio heads no longer purchase passes to the actual conference sessions, opting to instead focus their efforts on networking with other developers.

Game developers network at conferences like GDC, "expos" like the recently retired Electronic Entertainment Expo (E3), and local events like International Game Developers Association (IGDA) chapter meetings. Each event functions in different ways, networking different people. Local networking resources are the most important for those interested in getting jobs within the game industry. Larger events serve studio heads and new developers for the purpose of pitching existing or prototype games to video game publishers. Numerous private black box booths are available at GDC specifically for these activities. While some new media industries are limited only to local events where "noise" about the industry can be heard and engaged with (Neff 2005, p. 150), in its thirty years of activity, the video game industry has managed to produce more regular and institutionalized networking events. However, it does not strike me that these more institutionalized "organizational and industrial support" networks are any more accessible or equitable than the purely social networks of less mature new media industries (Neff 2005, p. 138).

For many developers, the word "diversity" is incredibly divisive. For most diversity is simply a question of numbers, do you have some women and non-white people? It is also closely linked in people's mind, no matter how rightly or wrongly, to favoring a candidate simply based on gender or race. It is important to keep in mind that diversity of all kinds is hard to find in game development studios. While diversity along numerous metrics (race, class, gender, sexuality) is a concern of some members of the community, the goal of improving gender demographics has become the most prevalent concern. The factors of why game work is dominated by men are of course numerous, and the most obvious conclusion that many make is that because men are gamers more so than women.[8] Despite the dubious and perhaps flawed nature of this assumption, the fact remains that there are very few women working as game developers. The below graphic and excerpts from qualitative responses to a demographic survey of game developers demonstrates how the restrictive access becomes implicated in lack of diversity. This lack of diversity is not only along gender lines; similar divisions remain along all lines of diversity. Ironically, the graphic resembles the early video game *Pac-Man*.

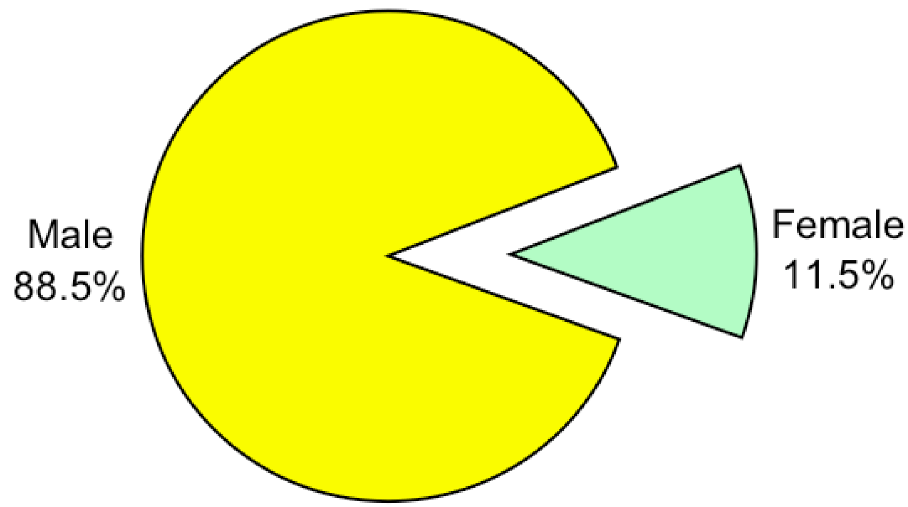


Figure 1.4: The Gender Dynamics of Game Studio Life, Pac-Man Style

> #17 «I don't think workforce diversity has anything to do with making great games. Hiring should be based soley on skills, work ethic and personality. Race, gender, sexual orientation and ethnic background have NO bearing on hiring policy.»
> - M, 35, White, Uni, USA

8. This assumption is also false. While many maintain that video games are produced by white men for white men. The statistics are actually more equivalent, with most numbers that executives talk about hovering around 57% male and 43% female for the gaming public. The number of women playing consoles is significantly lower, into the 35% range. Oddly, most of those same statistics report that women actually spend more money on games than men, though the question remains if this is money spent for games for themselves or others (ESA 2005).

#13 «I'm tired of being the one girl designer in the company. Please make more girl designers. Caveat: if they're not any good, don't bother. I hate having to swim upstream against the current of expectations the not good ones create.»
- F(bi), 32, White, disabled, Uni, USA (Gourdin 2005)

Now, in part, this reflects the assumption by many that game development can be equated with writing source code for a living. The popular imagination of game development as an extension of the software industry means that the already small numbers of women in computer science programs are the only ones that consider game work. While the number of women game developers is certainly small, the number of women game engineers is even smaller. For the most part, for women game developers, gender is something that is not talked about, at least not in the trenches. One experience in particular painfully reminded me of this. Since VV moved into the new offices, summer cookouts became not only possible, but also a staple of Fridays when the weather permitted. One of the young design interns sat down amongst the circle of artists, engineers, and designers, most of who were also working on *SM3* at the time. After several minutes of casual conversation, he posed the question to a female tools engineer, "So, what's it like working in a company full of men?" Silence descended upon the circle of developers, and she spent a moment or two plainly staring at the young man. At which point another developer changed the topic and the conversation continued.

Read through the lens of this engineer's previous experiences coming up through a computer science program, the organization of a campus game development group, and numerous other experiences in which she was often the only woman in the room, it was pointing out something that she had grown used to. Almost as if someone had asked, "So, what's it like being in a wheelchair?" For the most part, women in the game industry have taken career paths and trajectories in which they have gotten used to being "one of the boys." This is not unlike what other anthropologists of scientific workplaces have found, that the abundantly male character of the Lab workplace tends to produce an assumption of lack of gender or mono-gender. This works well until gender reemerges as something which must be paid attention to.

> Instead, I suggest that Susan's visibly pregnant body presented him with a problem. In the male territory of the Lab, Susan gets along well as one of the boys, but this reframing is difficult to maintain in the face of her protruding belly - dramatic testimony that she is not in fact a boy at all. Unsure how to deal with that dangerous and unpredictable belly, John simply deletes it. Having made Susan's anomalous female body disappear, he carries on working with her all day as 'one of the boys,' worried only by a slight suspicion that perhaps he should have said something after all. (Forsythe 2001, p. 178)[9]

The fact that it took an intern to point to the difference merely showed his ignorance of the system of relations which attempt to not mark the difference, as it calls out those already in marginal positions. Indian game development companies, though still male dominated, tended to exhibit a greater degree of gender diversity. One studio in particular had more women engineers than I had encountered in all of my studio visits in the US. The number of women working as part of the art team at this studio was also significantly greater than I had encountered at most studios in the US. When I asked the

9. This story also exemplifies what other researchers have seen in scientific labs where gender differentials are quite dramatic. "Mono-gender is the sylistic rendering of everyone as a physicist, regardless of their gender in everyday life. The mono-gendered physicist looks more like a male than a female. Females, too, drive home that they are, above all, physicists, not women" (Knorr-Cetina 1999, p. 232). Game developers, like physicists become mono-gendered buddies, a remedy that frequently only lasts until inter-office dating occurs, or the boyfriend or girlfriend of a female worker shows up at a social gathering.

women why they had chosen to pursue careers in game development, they often mentioned having played games with their brothers while growing up. Indian women also did not comment on the distinction between themselves and other women as my US based informants did.

In one-on-one conversation, US women developers frequently comment that the environment is one which they have grown used to, or that they prefer the company of men to that of other women, that they are not "girly girls."[10] On the other hand, that same rejection of "girly girls" can be difficult for other women. It can create the assumption that "beautiful" or "girly" women simply do not belong in the game industry.

> It is dangerous to be a beautiful woman in the games industry. Oh, it's difficult to be a woman, period. But if you also happen to be attractive, you are doubly cursed. On the one hand, yes, when you're at a conference where you are among a handful of women, you are remembered, and that is advantageous. But for every break you may get for being female and attractive you get a chorus of voices telling you that you don't deserve it because, well, you are attractive, and obviously you can't possibly have gotten where you are without seducing men along the way. And I am devastated to say that sometimes joining in those voices are other women.
>
> And then on the other hand you have groups that want to use you because you are beautiful - whether its the marketing machine, PR, the press - it's all a form of exploitation, honestly. And while men like Cliffy B and Will Wright are also pimped out to move product, they don't suffer from quite the same sexually tinged commentary that comes with being the female spokesperson for a product.

10. The very particular construction of femininity (as well as masculinity) amongst game developers is particularly interesting when thought of in relation to "sociohisorical process by which racial [and gendered] categories are created, inhabited, transformed, and destroyed" (Omi and Winant 1994, p. 55). Racial/gender formation and the "projects" that accompany them, "the building blocks not just of racial [and gender] formation, but of hegemony in general (Omi and Winant 1994, p. 68).

It's disgusting. And distressing. And depressing. (Pinckard 2007)

Diversity does have its strongholds. Some studios make it an explicit goal to have a diverse community of developers housed within their walls. Unfortunately even these studios must deal with the fact that there are very few women engineers going into the game industry. Though also difficult to find, women artists and designers exist in greater numbers and studios looking towards gender diversity use this to bring greater numbers of women into the game development workplace. The minorities of the game industry almost unequivocally believe that the industry needs greater diversity, but they check their comments frequently by saying that diversity is no substitute for those with the talent and skills necessary to work well in the environment. The defensive stance which many game developers take with regard to the hiring of "less qualified" candidates seems to have more to do with an identification with those people trying to "break in" as well as with an acute awareness of the repercussions of workplace connectivity. If someone is unwilling or unable to get work done, it is likely to impact coworkers before it impacts a project, a studio, or a games publisher. This is the particular aspect of "projects" and "hegemony" that I find useful in my analysis. "'[H]egemony' is a very particular, historically specific, and temporary 'moment' in the life of a society. It is rare for this degree of unity to be achieved. ... Such periods of 'settlement' are unlikely to persist forever. ... They have to be actively constructed and positively maintained" (Hall 1996, p. 424). As you progress through the dissertation, there are numerous critical points where less gendered/raced/classed projects could be mobilized, or projects that affirm the current settlement are mobilized. These projects intersect and mingle with other political economic and cultural projects. "Perhaps this is the real lesson" for those interested in "the relations between women [and other minorities] and [game development] scientific activity - not the discovery of an 'other' reason, but the exploration of what reason is capable of when it is liberated from the disciplinary models that normalize it." There must be resistance at some level to the "irrationality" of the normalization for real change to the worlds of video game development to occur (Stengers 1997, p. 130).

1.1.3 "Breaking In"

Talk to game developers for even a short time, and you will quickly hear that the game industry is "more social" than other industries, and it is frequently said that networking is one of the most important aspects of "breaking in." Gaining access to the industry has become a game in and of itself. Nearly every game developer has a story to tell about how they managed to break into the industry and is more than happy to share it. I often joked with new developers that, "You've got to make games before you can make games," which usually elicits a laugh, or a story about the demo reel, engine, or portfolio that they were certain had something to do with their hiring.

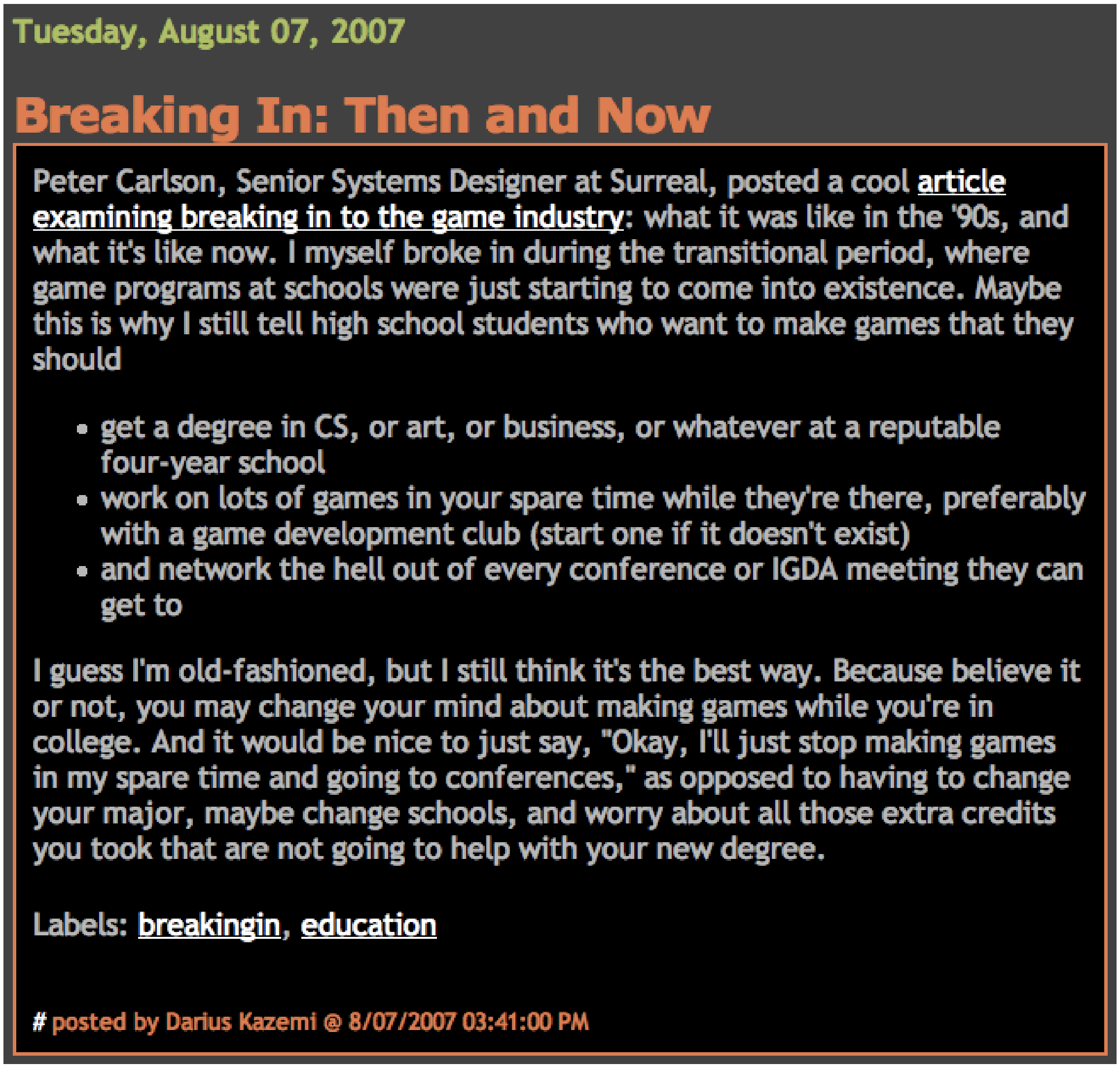
Nearly every U.S. game developer who has "broken into" the game industry has done so by making games, or something as close as they can get, on their own time. While this means that game development companies benefit from new hires being moderately more knowledgeable of what it takes to create video games, it also means that it imports some of the work habits of college students into their studios. Procrastination and an "it will come together in the final hours," mentality frequently prevails. Some developers even bring college sleeping and eating habits into game companies. Late to rise and late to leave can frequently become the modus operandi of a studio if young developers are left to their own devices. For many, game development is the interlude to a host of other work or school related activities. It is quite literally play for those looking to break into the industry.

A: Uh, in '98 I worked for a company in [CITY-1] called [COMPANY NAME-1], that was my big break job. And I got hired as a U.I. [User Interface] person, because I had a strong reel that was done like in Director [A Macromedia Multimedia Authoring Application]. And it had a lot of design-ie kind of things in it, but I had 3D in there too. And then once I got there, I did do some UI stuff, this game called [GAME NAME-1] and I did some U.I. stuff on [GAME NAME-2] ... but then it was kind of a weird place, they didn't have a lot of work after the [GAME NAME-1], and I think we went almost a whole year without a real project. They had floated on the money that they made on the [GAME NAME-2]. So I didn't really do a lot there, I kinda just worked on my own stuff.

A: While I was there, they were still working on [GAME NAME-3], a port of it for [COMPANY NAME-1]. I was working on that at the time. I had a web site, and a guy from [PUBLISHER] saw it, and we went through the whole thing, and then they decided to hire me. (Informant and O'Donnell 2004)

GNB: But I did my own stuff on the side... Just making random, not even full games, ya know? Trying to make different kinds of "engines" or whatever. That's the big thing when you don't have a paying job, and you're making game stuff. My interview was the worst interview. Like I interviewed with five, four or five people. This was the worst interview, that I ever had. It was like nothing, a two minute interview...literally, like two minutes. I couldn't, I didn't ask any question, cause this was like my third interview that day, and I just couldn't think anymore, and I just wanted it to be over.

GNB: They said to email them demo stuff, so I did that as soon as I got home. And, I'm pretty sure that's what got it, because the interview was horrible. But, I don't think I'd have gotten in here, if I hadn't sent in my stuff. (Informant and O'Donnell 2007)



Tuesday, August 07, 2007

Breaking In: Then and Now

Peter Carlson, Senior Systems Designer at Surreal, posted a cool article examining breaking in to the game industry: what it was like in the '90s, and what it's like now. I myself broke in during the transitional period, where game programs at schools were just starting to come into existence. Maybe this is why I still tell high school students who want to make games that they should

- get a degree in CS, or art, or business, or whatever at a reputable four-year school
- work on lots of games in your spare time while they're there, preferably with a game development club (start one if it doesn't exist)
- and network the hell out of every conference or IGDA meeting they can get to

I guess I'm old-fashioned, but I still think it's the best way. Because believe it or not, you may change your mind about making games while you're in college. And it would be nice to just say, "Okay, I'll just stop making games in my spare time and going to conferences," as opposed to having to change your major, maybe change schools, and worry about all those extra credits you took that are not going to help with your new degree.

Labels: breakingin, education

posted by Darius Kazemi @ 8/07/2007 03:41:00 PM

Figure 1.5: Screen Shot of Tiny Subversion Blog Post (Kazemi 2007)

Breaking into the game industry has a secondary effect related to the culture of secrecy already mentioned. Since the learning process is frequently divorced from insight by those with experience working in the game industry, either because they are unwilling or unable to share information gathered through experience, many developers view their own solutions to common development problems as the only possible ones. Because U.S. developers have not had an opportunity to make games in the context of work, the ability to discipline or improve their practices is rarely taken. Developers who suddenly find themselves at studios that have taken the time to engage with these tasks frequently rail against their newfound restrictions and parameters. These developers do not last long at established studios and instead strike out on their own. Learning how to make large scale games, or games that can be easily modified to meet the demands of numerous actors is not done until one reaches the workplace. These practices are rarely (if ever) documented or discussed. This is not unlike the work of engineers or technicians more broadly demonstrated by ethnographic fieldwork of these organizations (Orr 1996). Each studio becomes responsible for developing sets of practices, ones usually based on those who started the studio, and their subsequent experiences.

Many Indian game developers go straight from their undergraduate education to game development companies often without a portfolio or game development experience. While these developers must learn the ropes of game development on the job, they do so in a context of work. This frequently makes it easier for them to understand game development as work, rather than only as a personal passion. It is desirable for developers to have experience making games, but most companies assume that a significant amount of training will be necessary. The resulting benefit is that breaking in is not in operation in emerging game industries as it is in the U.S.. Even Western European developers seem to feel that breaking in is an experience of American developers.

The disadvantage of a less rigid barrier, combined with game developer's secrecy over everyday work practices that speed and enable development, is that many young game development companies are on uneven footing when it comes to development operations. This tension is greatest when it comes to the full game development process; the development of a game from start to finish, or the completion of a "title."

1.1.4 "What Titles Have You Worked On?"

Amongst game developers, the "titles" or games that you have appeared in the credits, tends to be the form of cultural capital that gets you hired when changing studios. For developers in emerging industries, like India, this can be problematic since the same networks that govern studios end up governing individuals. Studios heads may have willingly traded away the ability to publicize their involvement in the production of a game for greater monetary compensation. This works at the level of the studio, but individual developers in these contexts rarely find their names in the credits of produced games. While many of my Indian informants joke that the Indian legal system makes it difficult to enforce some of these limitations rendered by NDAs, most of my informants did not consider the possibility of breaking the agreement. The issue is of crucial concern for game developers, causing it to become one of the many issues in which the IGDA invests time.

> Crediting in the game industry has become a hot topic in recent years. As development teams grow bigger and outsourcing becomes more prevalent, the informal crediting procedures used become increasingly insufficient to describe each developer's exact role within the development process. Additionally, the non-standard naming procedures for job titles that have thus far characterized the free spirit of the gaming industry have now become a liability for those who wish to prove their skills when moving from one company to another. A movement to standardize crediting procedures and titles has never been more needed.
>
> The IGDA Credit Standards Committee is a group of volunteers who have come together to study, document and propose voluntary game industry crediting practices that properly recognize those responsible for the creation of games. To do this, we are creating two documents: one which details the current methodology of credit assignment as well as catalogs a set of the most accepted job titles found within the industry;

> and another report to propose a set of "generally accepted practices" which can be adopted voluntarily by developers within the industry to resolve difficult crediting dilemmas. (Feil and Weinstein 2006, p. 3)

Depending on the credits or titles as the measure of a developers worth, especially given the controversy and lack of standards around the practice, seems a weak metric for measuring the vast amount of work which goes into the production of a game. Which ultimately leads the question, what is it precisely which is desired of those working within these secret worlds?

1.1.5 The Importance of "Skill," "Passion," and "Talent"

The imagination or imaginative capacities of play have a great deal to do with our linking of work/play. As play theorists note, "[i]magination, flexibility, and creativity" of the "play worlds" link to narratives of innovation and progress (Sutton-Smith 1998, p. 11); but the ability to be creative, flexible, and imaginative on the job is not all that new. Many in game development, and New Economy work more generally, have attempted to create a hierarchy of work, one that separates the imaginative capacities of one set of laborers over another. While it is true that there is an important difference between creativity and imagination for personal amusement or resistance and it being an aspect of the work produced, the linking up of work and play in this way as something particularly unique to New Economy, or game development work in particular, seems premature.

The necessary "skill" of workers in game development is difficult to place.[11] The work of creating games is highly complex, but also it can be highly repetitive. I was con-

11. Latour writes about "technical skill" as applying to those "with a unique ability, a knack, a gift, and also to the ability to make themselves indispensable, to occupy privileged though inferior positions, which might be called, ... obligatory passage points" (Latour 1999, p. 191). Though I do not think it is entirely applicable to game development, it encourages critical thinking about the work, which concepts like "passion," "skill," "talent," and other terms, seem to imply innate capacities.

tinually impressed by what my informants could do, and surprised by how much repetitive work they put up with. Game development is varied in its capacity, a fact that is frequently ignored in favor of returning to the focus on engineering tasks of game development. Most important, at least in hooking our work/play system up to those systems of playful relation that actually drive it, is the closure and secrecy of social networks in the game industry. "Skill" serves as another means by which access is managed, rather than more concrete metrics. The presumption of, "This is for us, not for the 'others.' What the 'others' do 'outside' is of no concern of ours at the moment." Event more importantly is that, "Inside the circle of the game the laws and customs of ordinary life no longer count. We are different and do things differently." "The exceptional and special position of play is most tellingly illustrated by the fact that it loves to surround itself with an air of secrecy," is of critical importance to our understanding of work/play (Huizinga 1971, p. 12). Those skills, which are necessary, are wrapped up in a world of secrecy. For you to find out how to make games, you must make games. This circular logic elicits laughs in developers, precisely because of its contradictory character. To prove yourself as capable of working in the game industry, you must work in the game industry. We do things differently here; so do not expect your ways to be our ways.

1.1.6 Secrecy all the Way Down

These cyclical restrictions hearken back to the "breaking into the industry" narratives of informants, and the entire sub-genre of game development community writings about how one can gain entry to the restrictive networks of access. While it is true that work/play is imaginative, interesting, and desirable, this is not what seems to elevate it nearly as much as the way it wraps itself in secrecy and closes off networks of access. These levels of secrecy and networks of access pervade numerous aspects of the work/play of the video game industry.

The secrets of manufacturers, if even revealed to engineers, place restrictions on what may be freely shared outside of an individual studio. Even if two studios are covered by the same contracts and non-disclosure agreements, they will be reluctant to speak in specificities for fear of transferring some specific piece of "proprietary" information. Artists that move from one company to another will encounter entirely new, though perhaps quite similar technical systems which enable their work/play. How or why those systems have been constructed in that fashion will remain unknown, an aspect of this historical past of the secret society of that studio. What is covered by copyright, non-disclosure agreement, or corporate contract encourages the fall-back position that everything is secret. Names of characters, model requirements, engineering standards, organizational structure, clients, publishers, and hardware being worked with all begin to fall into the category of "secret."

There is a distinction between crunching for a week to hit one deadline and the dreaded death-march crunch. One is temporary, a surge to ensure the victory, or so you hope. The other is a quagmire that just does not get any better. The trouble becomes that crises continue to mount and multiply as the projects drives onward.

That is when things start to go downhill. Management and leads have started to get nervous about the deadlines, and they start scheduling meetings. The more work you seem to have, the more time you seem to spend in meetings. The more crises, the more you try to avoid them through coordination, but you have the feeling it has more to do with bleary-eyed mistakes. It is usually about this time you hear something about a demo which needs to be created for an upcoming trade conference, or that marketing would like to start circulating to the enthusiast press magazines. A new deadline gets created that suddenly requires the recalculation of many other deadlines. More meetings get scheduled.

None of the tools are ready yet. Everyone is trying to use them to get their pieces of the puzzle into the game, but they keep breaking. But, the tools are polished compared to the status of the game engine in places. Some of the engineers have ascertained that one of the platforms doesn't have support for a crucial special effect that was being used everywhere. Well, it supports it, but in a bizarre format that is going to require rework by probably 30% of the team. One of the technical artists stayed late last night and made a tool which helps with the rework process, so that is good. Unfortunately that means that he could not get much done on fixing a couple of bugs that the art group found in their tool pipeline. Hopefully that will be next. To ramp up the production schedule, especially because the new demo will require components to be done earlier than they were originally scheduled, several new people have been hired. They got a crash course in the new tools and software used to get their work into the game, but they still kept messing up and checking things into the version control system that cause the build to break. The engineers are treating them like children now, but the tech artist is trying to help them out. Life starts to feel like a shampoo bottle: work, rinse, and repeat.

Table 1.2: Crunch Stories: Crunch Keeps Going...

1.2 World 1-2: Instrumental Work/Play Amidst Rigid Systems

There is something else that drives game developers and workers in the New Economy more generally. It is an aspect of the work, which encourages workers to push further and harder than necessarily required. This of course has become the upshot for corporations, that their workers are plugged into their work in ways they have not been previously. Perhaps on some level it is a realization of a Protestant Work Ethic based on the idea of a "calling." As sociologists and ethnographers of virtual worlds and gamers more generally demonstrate, it goes beyond the simple answer, "because it is fun," which it may very well be some of the time, but more than that, it gets at an underlying drive, 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," on the part of game players. The instrumental or "power gaming" of the workplace, gets at a desire to understand the hows and whys of the structures they move within; knowledge that will enhance the ability of the knower in their work/play quests. Amongst many game developers, the desire to know the structure of the system within which they work has led to an almost instrumental or "power gamer" approach to work/play. In part, this is a product of a system, which seems arbitrary, or imposed, hence the "broader ambivalence about what constitutes legitimate play[/work]" (Taylor 2006a, pp. 72-73).

1.2.1 Bumping into Hardware

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. Instrumental gaming is essential to game development. In many cases it is impossible for artists, engineers, or designers to know what precisely they can manage to create without causing the underlying hardware to buckle under the pressure they put on it.

> GNB: Ah. 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 [SYSTEM NAME-2], 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... (Informant and O'Donnell 2007)

But despite the best efforts to "scope that out at the beginning" based on the "specs that are given" about a system, developers frequently find themselves having to maintain a "commitment to understanding the underlying game systems." One engineer in particular talked about after doing particular optimizations for the Nintendo DS handheld console having a feel for "every transistor and chip inside that thing." 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 numerous times, talked about the process of digging into a system, at the level of software or hardware systems, and its requisite attentiveness.

> MG: The process of stuff not working like its supposed to?
>
> C: Yeah.
>
> MG: So there are levels right. And these are especially obvious to the new people. Umm…where you cannot perceive of a reality where something doesn't work. Because it obviously should right? Umm…so there is an emotional component of it where you just won't even, 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 then there's a component of sort of you know, being afraid of, or just thinking its impossible to fix, or something. Really to, ya know, you're not expecting a system call to do what it should, or hardware to do what it should, and I think, good programmers, a key realization is, well, ya know. This was written by someone else, just like you, and umm…ya know, 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. Umm…I think, specifically about debugging, if you come into and you just accept the situation, and then you start attacking it, and you attack it in a thorough and methodical manner, and ya know, there are probably about ten or twenty different strategies to go about it in a thorough methodical way, and simultaneously, you have to be thinking wildly creatively, and taking big guesses. So you're going through this plan, that you can almost do by rote, and you know, step B, step C, finally results in a bug, three days later. But at the same time, you're sort of drawing information off that and just thinking wildly about stuff, ya know, wait a minute, it's probably this,

> and you just skip to the end and try you're little thing, and if that works, great. Ya know, more often times than not, it doesn't, but for especially hard stuff, I think that's just what you gotta do. (Informant and O'Donnell 2004)

While some STS scholars have written about the process of debugging or feeling out systems as "being in a cave" (Turkle 1997), it is actually not so un-systematic. You learn just as much from "negative knowledge," knowing the limits of what you know. 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, p. 64). The ability to think simultaneously thoroughly/methodically creatively/intuitively is a different way of thinking about how we work out how things work. To get at these underlying systems, as anthropologist of medicine Emily Martin has noted, 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, pp. 253-254). And while in both transcript excerpts, these are engineers talking about software systems, the same ends up being true for artists and designers. They too end up depending upon layers of software and hardware systems each with their idiosyncrasies. During one conversation, a technical artist talked about how willing artists were to assume that something, which 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, which should not have fit in the first place. Even working within the lines established by others, artist will continually modify and carefully check their work to see where changes can be made which improve the visual quality of their work without breaking the guidelines they have been given. Designers will frequently attempt to do things, which they have been instructed will not work, to see if there are ways in which to tweak their work so it falls within guidelines.

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, p. 229), but nearly every game developer. Each engages with instrumentalist tendencies in an effort to accomplish their 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, p. 187). But, the fact of the matter is, the design process for game developers, that push and pull, is being done to determine where the bottom is. Despite the construction of rigid specifications, they are based on something, which must be felt out and determined by the players. Eventually developers run into the limits of electrons and silicon. Specifications are made, but they are not made up, a negotiated process, which is frequently the product of instrumental play.

The trouble is, not everyone is interested in playing this way. Instrumental play too has it limits. Certainly in some cases it makes you wonder about the possibility of not playing the instrumental game that has begun to unfold. "This sense that somehow these players are just too dedicated, indeed almost bordering on the psychologically pathological, is a popular theme. What I found in conversation with power gamers, however, is that they consider their own play style quite reasonable, rational, and pleasurable" (Taylor 2006a, pp. 72-73). If "these functions are mainstreaming the focus on quantification in that play style," then likely this work/play style will become more widespread. "But if the adoption of these tools and with it the play styles it brings becomes mandatory, must we start to deal more concretely with notions of emergent coercive systems" (Taylor 2006b, p. 332)? Which seems to get more directly at the heart of what we are discussing.

Instrumental play should be distinguished from a kind of "instrumental rationality" or "instrumental reason" as it might be defined by critical theorists of the Frankfurt School (Adorno and Horkheimer 1976). Instrumental play is distinguished in that it has no claim to the irreducible or absolute, if anything it would continue to press against what is considered irreducible. As my informants might put it, "You've got to get over that [a commitment to the absolute] real quick." Instrumental play is about searching out associations, analogies, and relationships, much like "Enlightened" scientific inquiry, but it makes no assumptions about the absolute character of those suppositions. This is where the "play" component of instrumental play is crucial. There is always the assumption that what you are working on or working with will swerve and send you in new directions.

In many cases developers do not even have the opportunity to form specifications or play in instrumental ways, methods that are crucial to pushing the tools, expertise, and products of game studios. Developers doing outsourcing work frequently are given rigid guidelines for their work. Unlike developers working inside studios, outsourced work is outright rejected if it does not meet specifications. There is no link between the specifications and the process from which they were derived. The ability to get back at the underlying system which demanded changes has been compromised. The ability for an artist or designer to see their work within a game is obscured. Only the contracting studio has the ability to view the results. Anthropologists of work have shown that in the "absence of informal working knowledge from technologically sophisticated production processes," "technical workers ... develop complex cognitive models that represent the ... messy world of work" (Baba 2003, p. 19). While these models provide the frameworks that workers can use to get things done, developers remain hamstrung by their inability to access the underlying systems from which these demands derive.

1.2.2 Bumping into Organizational Structures

Studio heads and managers play with their organizations like artists play with textures and models. They must interface and play with game development studios, intellectual property holders ranging 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 secrecy networks. Managers must do as much as possible with as little as possible. Teams will get shuffled around based on the work available. If particular designers have proven themselves able to handle particularly 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. There are creative visions that may differ, and changes will almost constantly be asked for by those who are not impacted by the necessary reworking of systems to allow for these modifications. Other actors within corporate institutions will be instructed to use the game a studio is working on as a means of negotiating with other institutions involved. Managers must frequently engage with the management of a system, which in many cases began as play.

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 key constituents of why people feel compelled to play in instrumental ways. It is not simply the systemization and regimentation of game/sport that causes the loss of game-like innocence. The issue is larger than that. Simply bringing money into the whirlwind does not automatically cannibalize your 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 towards institutionalization that changes the game. As anthropologist, John Kelly has written about in the context of American Baseball:

> 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, pp. 55-56)

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 be coerced into play in particular ways, along with the systematic incorporation of the video game industry. That move to industry rather than something else marks an event, which begins to alter the space of play. While baseball is one example, the connections with the video game industry are undeniably industrial. It does 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.

> Competent professional players began to make the game a means to profit, starting with the prowess of the Cincinnati Red Stockings. The line of movement from clubs to leagues to Organized Baseball remade baseball into an increasingly interconnected congeries of commercial institutions, made baseball into a branch of what Americans like to call 'free enterprise.' Baseball reorganized from independent clubs, originally player-oriented leisure groups, into profit seeking corporations in a legally powerful cartel. We can use this history to discuss dynamics of capital, profit, and finance in the actual capitalist world. What are the key genres of capitalism, its defining institutional structures, drive belts of its history? What is baseball, when it is not only a genre of game, but also, a genre of capitalist enterprise? What then constitutes its best interest, and how? (Kelly 2006, p. 59)

And now we are getting somewhere. Baseball and game development are each one genre of game/play. It is the connection with commercial profit driving organizations that have so dramatically shaped game developers worlds. It also provides us a normative point of entry. What drives this car, how does it go? More importantly, "what constitutes its best interest, and how?" But the normative is a question we cannot fully answer until we become familiar with later Worlds.

1.3 World 1-3: Experimental Interactive Work/Play Systems

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 of the daily activity. But that daily activity is linked up to larger systems of experimentation and technoscientific practice.

Engineers are continually bumping into and playing with the hardware and software of existing systems. Artists constantly play with 3D models, attempting to balance creative desires with the demands of designers, managers, and the commentary of their peers. Designers tinker with games through custom toolkits 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, and with other executives in the industry. 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.

The desire for technological systems, data stored on hard disk drives across the network and checked into version control systems, and even people to respond in an interactive way pushes us forward. They enable our instrumental play. Communication studies scholars examining digital games find:

> 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, pp. 294-295)

The desire for the complex system, the game, the workplace, your peers, to respond in an interactive fashion is seductive. When it responds like an interactive system, we feel connected or networked. When it does not, we become frustrated.

But what about historical structuring aspects of game design and technological innovation? If we had known we were going down the road in the first place we might 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:

> 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, pp. 74-75)

Not only have 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 even untangle where they have been. They occasionally make retrospective statements, "postmortems," 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 situations 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 that sometimes you take wrong turns and have to work late to find your way back. But collaborate in any meaningful fashion? That has been rendered impossible.

The implications of these kinds of practices more broadly throughout New Economy business is 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 are even more important. However, demands for secrecy seem to have taken precedence over the maturation of game development practice.

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, pp. 10-11). I like the metaphor of the labyrinth and experimental system, because it connects up to our enjoyment of working within limits, but also 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, p. 225).

But 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 people who end up interfacing with the work of engineers and artists. Their tools are created by the tools engineers, but frequently with a mind towards changes down the road. This is also why you have tools engineers and technical artists accompanying our new systems, technologies, and practices. As sociologists of science have shown, "[t]he more automatic and the blacker the box is, the more it has to be accompanied by people" (Latour 1987, p. 137). In part it is because these "outcomes are often not consciously calculated, or even intended by any one of the parties involved" (Knorr-Cetina 1983, p. 130). Because they are embedded in a broader social context of practice, they must somehow retain those connections.

> 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, p. 28)

Until you hit "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 unknown answers. Sometimes these unknown answers are frustrating, but often they become aspects of the game proper. But more than that, your experimental systems must be interactive. They must respond in real time to other technical systems, data, and people. This is where the headlong rush begins to become readily apparent. As designers play with all of the art and code assembled with experimental tools, the remotest possibility of an accurate reconstruction of the past becomes next to unimaginable, even for game developers. Rather, many developers, and engineers in particular hope for the possibility of "mind melding," because the reality is that it is impossible to understand where it came from or even where it will be when it arrives at Golden Master.[12]

Development practice is constrained by their histories and technological systems. Because of this, many developers decided that being able to manipulate things on the fly,

12. "Golden Master" refers to the state of a game when it is is ready to be shipped to the manufacturing company for mass production.

interactively, has become a necessary component. Developers bump into their histories and technologies, and if it continually required going back to the drawing board, nothing would ever get accomplished.

> The end is always evil. Everyone is around continually verifying that everything was set to go. Even if they were not responsible for whatever was going wrong now, they all stuck around anyway. They said it just seems wrong to leave one person pulling their hair out. It was also entirely possible that about the time they got home something they were responsible for would break. But, everyone had spent almost nine months in a continual mode of crisis management, attempting to keep things together, so this was not so extraordinary. Everyone is exhausted though, and there are rumblings of people splitting from the company, moving to other places. Even a rumor of this studio's sister site closure after completion.
>
> The upside was that for those who did not find themselves with a crisis task, finally got the chance to play the game. What they had produced was impressive. They never really got to see all of it when it was coming together, as they were constantly focused on the numerous other pieces. A shipping party is quick to follow. Management distributed prizes to people who busted their butts on this project, but almost everyone came out with at least a t-shirt. The drinks and food kept rolling out. Unfortunately, it was a bit bittersweet.
>
> The game apparently benefited from massive pre-orders on Amazon.com, but the reviews in the enthusiast press have been brutal. The meta-score rating of the game (an aggregate of the most respected reviewers) didn't break the 80% mark that was necessary for anyone to get their profit sharing bonuses. It is not going to stop the publishing company from making bank.

Many people are off on vacation now, resting up after everything has ended. Some people have been working on new project ideas in the mean time, waiting for the next project to hit. Others have turned in their resignations. Some are going other places, other studios, other states, other industries. They did close the sister site. The internally developed game which so many of them were excited about has been shelved indefinitely, since the publishing company is not sure how to market it. They wonder if any of these new project ideas can come to fruition and others think about how to incorporate the new ideas into those projects that the publishing company plans to move forward. The next project will hit in a couple of weeks. Again: work, rinse, and repeat.

Table 1.3: Crunch Stories: Crunch "Ends"

1.4 World 1-4: The Crunch - Autoplay, The Zone, and Desire

Why do academics always wind up at ea_spouse? And they do, it has become work/play pornography, in part because of its accessibility. I suppose academics have not had much opportunity to observe these occurrences in the flesh, because field site access is so limited, and a LiveJournal site is so much more readily accessible to the social scientist (Dyer-Witheford and Sharman 2005; Dyer-Witheford and de Peuter 2006; Deuze et al. 2007; Wark 2007). Just like porn on the Internet. That is not to say that it has not been an important index, or an important galvanizing point for game industry workers. It most certainly is that. But the fixation on it draws analytic attention away from the broader issue, why and how does work/play have such a propensity for damage to Quality of Life?

> A: I think it's a lot of things. Some places it's like the pure love of what you're making. You know? Like, I would imagine at say like Bungie, they're making *Halo 2*, and they know they are making like something millions of people are going to buy, and it's going to be incredible. So, I mean in that case, I probably wouldn't mind as much. I've been at places where working tons of overtime was purely to make up for management miscalculations, or bad scheduling, or um demos for higher ups that just got sprung on people that you know, our bosses could have said no to, but

> they didn't because its not them working really hard. So, I think there is just a lot of immaturity in sort of the business.
>
> C: What separates sort of management doing a good job versus, is there, do you think there is a defining characteristic that helps management do a good job? You know, not make those mis-calculations.
>
> A: I think it's…well, I mean, I could make up a bunch of anecdotes. I think it depends. With certain people, I think it's just uh, expecting the same level of buy-in from the entire team as like people at the top have. Not understanding that people that work for you, are working, and like getting paid, and they're not, that this might not be their soul. It's not that I don't feel that way, but I know not everyone in the game industry is like hard core about it, and they want to have a job and do a job.
>
> C: Do you think that will change in the next couple of years?
>
> A: I don't know. It's so competitive to get into, that I feel like if you don't have the drive, ya know, to go above and beyond on your own, then you will get weeded out. But I think it's when it comes down to management saying, you need to be here 12 hours a day for the sake of it, or because we're falling behind, so you just need to be here for longer. That's when it becomes a problem. So that I think when you have driven people that are really efficient then, you don't run into that as much. I mean, I can't put it all on the managers, it's like the whole culture. There is a lot of screwing off that goes on in games companies too. Much less here. At [PUBLISHER], we would in down time, just play Battlefield all day. There is a lot of just bad decisions. (Informant and O'Donnell 2004)

The academic fixation remains on ea_spouse, rather than on the broader context of the video game industry or new media production broadly. The damage is not merely a product of bad management. It has to do with all sorts of different levels, though unfortunately that nuance does not come out in many analyses. Nor has anyone recognized that EA and ea_spouse's spouse are positioned in a broad system, which encourages practices that enable poor Quality of Life. Cultural studies analysts have commented on ea_spouse, and the EA context, but not more broadly:

> You could be forgiven for thinking this is just a game, but it is somebody's life — as reported in a widely circulated text written by EA Spouse. EA, or Electronic Arts, is a game company best known for its Madden sports games, but which also which owns Maxis, which makes The Sims. EA's slogan: Challenge Everything — everything except EA, of course — or the gap between game and gamespace. In the gamespace of contemporary labor, things are not like the measured progression up the ranks of The Sims. In The Sims, Benjamin could work his way from Game Designer to Information Overlord much the same way as he had worked up the levels below. At Electronic Arts, things are different. Being an Information Overlord like EA's Larry Probst requires an army of Benjamins with nothing to work with but their skills as game designers and nowhere to go than to another firm which may or may not crunch its workers just as hard. As the military entertainment complex consolidates into a handful of big firms, it squeezes out all but a few niche players. Gamespace is here a poor imitation of its own game. (Wark 2007, p. 044)

It is easy to fixate on ea_spouse. Erin[13] is a superb writer. However, the fixation on ea_spouse, even in positive ways, draws our attention from the reality that the situation is much more difficult than we first thought. The desire game, "I want, and will pursue" strikes so clearly at the issues faced by video game developers. What is it precisely that drives them to do what they do, and to do it so intensely in many cases? Most troubling in all of this is the collapse of desire, work, and play into AutoPlay. AutoPlay, a term coined by an anthropologist studying casino game players, "marks the point at

13. Nearly a year and a half after the publication of the ea_spouse blog, Erin Hoffman, now a designer in the video game industry, made her identity public in an online interview (Wong 2006). During the course of my research she began working with a game studio that spun-off from my primary field site. She continues to write for online game development and game related publications.

which the varied, complex forms of interactivity and productivity that have become the trademark of the 'digital age' loop into recursive forms of disengagement … Players cease to be desiring subjects" (Schüll 2005, p. 78). In other words, AutoPlay marks the line at which aspects of work/play that encourage our involvement or enjoyment ("fun") in work practice transitions to collapse, disengagement, or crunch.

> A: I've seen it…yeah. I think it's a huge strain on people's relationships. There is a pretty standard industry idea of just people burning out after X number of years, and just getting out of the industry, because it is so demanding on your time and on your life.
>
> C: So where do people go when they're done with games?
>
> A: Well, I mean you might go into other areas of software development, if you're an artist, you might go work for a graphics shop. Who knows? The thing about games is that it's so multi-disciplinary, that you can usually branch off and do something else. I don't want to at all, but, I've been on message boards amongst developers, with that topic. How much longer do you think you can go, and stuff like that?
>
> C: A lot of that discussion is a part of what prompted me to do what I'm doing now, so I can identify. So, I guess…
>
> A: Yeah, I mean I've seen a lot of guys just crack and quit and then they come back like to another company a few months later. But a lot of times, there is that getting pushed to the brink, is, I see it in a lot of places.
> ...
> A: How is that all gonna work? And when I'm 45 or 50, am I going to be valued, or that much better because of my experience? Because I know that now, somebody that got started in games in 1982, that means nothing now. It means absolutely nothing about their relevance now. So, if that cycle continues, of the industry re-inventing itself every 15 years, and all the tech completely getting turned upside down, and what you knew then means nothing now, then, that's pretty frightening. I'd like to think we're

> on a course right now, where I will be able to keep up with things, and it always takes work, you have to be conscious of the nature of the industry being fast moving and stuff, but I like to think that if I just stay on top of things, as they come out, that I will have long and nice career in the game industry. Instead of something like, like when the 16 bit era ended, and everybody was pixel artitsts, if you like, say you didn't even have an aptitude for 3D, you'd be screwed from that point. And then you had all these kids jumpin in saying, I know 3D studio and I'm 17, and I happen to learn software well, and I'm not a good artist, but you get into the industry. And then there is this big roll-over thing going on. I guess its mostly there is no history to look at how it happens. (Informant and O'Donnell 2004)

> What is equally curious is to see how the word wage, originally identical with gage in the sense of a symbol of challenge, moves in the reverse direction of pretium - i.e. from the play-sphere to the economic sphere and becomes a synonym for 'salary' or 'earnings'. We do not play for wages, we work for them. Finally, 'gains' or 'winnings' has nothing to do with any of these words etymologically, though semantically it pertains to both play and economics: the player receives his winnings, the merchant makes them. (Huizinga 1971, p. 51)

And some might assume that this is no fun. Right? It certainly is true that 50% of game developers can barely stand to work in the video game industry longer than ten years. Nearly a third drop out before their fifth year (Bonds et al. 2004, pp. 30-31). Certainly this is one assumption that we could jump to, and many communication scholars do.

> All this casts doubts on the myth that game making is 'fun.' Such labour does not live up to rose-coloured post-industrial visions of knowledge work. But nor does it match the straightforward picture of deskilling and degradation painted by the neo-Luddite left. What emerges is more contradictory. The creation of a new creative high-technology industry has required management to recruit a post-Fordist workforce whose

> control requires the use of techniques that are very different from the rigid routinization and top-down discipline of Fordism. They involve a high degree of soft coercion, cool cooption, and mystified exploitation. (Kline et al. 2005, p. 201)

But, if that were the case then why do we find similar characteristics amongst people working for free? Sure there is soft coercion and co-optation, these are corporations. But it doesn't make any sense, because despite these conditions, and beyond simply the "cool" factor of it all, people are driven by their jobs. There is something about the intellectual, visual, collaborative aspect of it that hits at something deeper, as anthropological studies of Free/Libre and Open Source Software workers demonstrates:

> Hackers describe this mode of labor as "deep-hack mode," [or "zone"] a cavernous state of mental and often physical isolation in which one reaches such a pure state of concentration that basic biological drives like sleeping and eating are put on hold during the hours or days that pass. (Coleman 2005, p. 233)

Deep-hack mode does not hit just hackers. It hits artists, designers, graduate students, and many others. Work/play has tapped into something that when allowed to drive to its own beat, does become "mystified" exploitation.

> Just as the military industrial complex once forced the free rhythms of labor into the measured beat of work, so now its successors oblige the free rhythms of play to become equally productive. Alan Liu: "Increasingly, knowledge work has no true recreational outside."* The time and space of the topological world is organized around the maintenance of boredom, nurturing it yet distracting it just enough to prevent its implosion in on itself, from which alone might arise the counter power to the game. (Wark 2007, p. 172)

This is a different set of desires, the "phenomenology of the zone." I wonder, however, if all of these games are worth playing? What is the payout? What is being pursued and often at such risks? Why then do my, as some would say, "mystically exploited" informants keep doing it?

> 'So far so good, but what actually is the fun of playing? Why does the baby crow with pleasure? Why does the gambler lose himself in his passion? Why is a huge crowd roused to frenzy by a football match?' This intensity of, and absorption in, play finds no explanation in biological analysis. Yet in this intensity, this absorption, this power of maddening, lies the very essence, the primordial quality of play. (Huizinga 1971, pp. 2-3)

In an environment flush with both legitimate and illegitimate distractions, developers often find themselves much like Schüll's gaming machine players, searching out isolated locations where the pursuit of creation can take place in an uninterrupted fashion. They too desire the deep hack mode of where lines of code, level design, mission scripts, animation frames, texture art, or model geometry can be produced. Like all of their desirous compatriots, the drive to pursue these desires often pushes towards excess. Why are we continually testing ourselves with these games? The question becomes, how can we hack, or "exploit, refigure, and thrive off those social contradictions related to technology, contradictions that emerge more palpably in the tense intersection between liberal values and a neoliberal knowledge economy" (Coleman 2005, p. 46)? Perhaps we can turn, at least in part, to hacker practice as a site for inspiration.

> "This means a hacker will at times dutifully respect a system of logic while, in other instances, he will blatantly and with succulent pleasure disrespect it, either for the sake of play, exploration, making a political statement, or to accomplish the immediate task at hand. As often as one

can find hackers distorting language, laying bare its contingent nature, one can also find hackers who dutifully respect the formal rules of grammar and praise its internal or deep logic with the same incisive precision they treat object variables while programming. Based on elements like individual stylistic preferences, one's ability to manipulate form, and especially the social context of activity, the hacker attitude toward form is relational, oscillating between respectful awe and playful irreverence. " (Coleman 2005, pp. 213-214)

Coleman uses hacker practice as a site for her own theoretical innovations, using the term "irony" (in close proximity to historicity) to, "simultaneously accentuate the existence of powerful systems of coercion or hegemonic institutions, as well as the ways in that they are intentionally and accidentally evaded." This is intricately linked to the practice of simultaneously working dutifully within structures of constraint while also disrespecting them, frequently in humorous ways. This play is considered "ironic precisely because it is still possible to tease out the sensible or expected elements within the shell of the unexpected" (Coleman 2005, p. 34).

But what does this ironic play have to do with breaking out of the recursive infinite-loops of AutoPlay? It can best be described as a process of debugging coercive or hegemonic structures, which requires an attention to detail and awareness that cannot be described in any ways as disengagement. Debugging requires an attention to detail and "disrespect" to a system that enables us to examine the inner workings of a system. Often times debugging involves "stepping into" systems that were previously closed or considered outside of the frame of interest. This is complicated when we are on the "bleeding edge." "Technical complexity" is compounded by "urgency" (Pentland 1997, p. 118).

In the case of gaming machine players, they can still work in their "zones," "deep-hack," or "crunch" conditions, but like Ronnell suggests, they must break away prior to reaching a state of AutoPlay. The jamming of toothpicks into a button to watch the credits go up and down would constitute AutoPlay. The random number generator (RNG) is left to make all subsequent determinations. The moment of stepping into, or engagement would be to ask questions about the RNG. What is it? How does it operate? Why would mathematicians prompt us to place a "P" for "pseudo" at the beginning of that abbreviation? Even Schüll does not debug the "really new god" of the gaming industry. An entire sub discipline of mathematics is left closed, un-interrogated. This stepping into the system's structure is what separates the hacker from other system (structure) inhabitants. This is how we can escape the mechanism of AutoPlay, rather than being absorbed into the recursive flow of the system. We can step into the operational mechanics and disrespect them. Perhaps this insolence will manifest itself in humorous ways. Perhaps we debuggers will be labeled "black-hat malicious saboteurs" (Coleman 2005, p. 45), but it is precisely that ability and desire to disrespect the systematic nature of structures that drives us to make changes. This is precisely what it means to transition from being a gamer to a game developer, asking questions of the underlying system.

1.5 Boss Fight: Meeting Bowser for the First Time

You have now reached your first Boss Fight. It is time to bring the lessons learned at each Level to bear. This also marks our transition to another World, another collection of Levels. The daily worlds of game developer work are constantly shaped by work/play forces. Secrecy, instrumental play, interactive experimentation, and the phenomenology of "the zone" all encourage developers to understand their worlds as separate, distinct, and outside of work. A competitive spirit untempered by a knowledge that sharing and cooperation are not mutually exclusive with good clean competition prevents growth and sharing. Instead secrecy and instrumental play reign supreme. The ability to play with systems providing interactive feedback encourages developers to get into and remain in the zone.

These secret production societies are "hegemonic" rather than "despotic." Managers strike a balance between coercion and consent, rather than ruling with iron fists (Ong 1991, p. 286). When workers consent to their conditions, there is no need for coercion. The disciplining of workers is an "intricate, long-drawn-out process involving a mixture of repression, habituation, co-option, and cooperation" (Harvey 1990, p. 123). At the same time however, ambiguity begins to operate again, "other activities" (than those dictated by managers) "may, in fact, be equally necessary; but, since management has not defined them, their status as work is, at best, arguable" (Orr 1991, p. 12). But again, functionality must be preserved, and if allowing "non-work" or "ambiguous" work (English-Lueck and Saveri 2001, p. 8) to occur preserves the operational success of a unit, then it may be later reassessed, but likely it will be left to perform.

These modes of work practice are simultaneously necessary, yet capable of collapsing in on themselves. This is the Boss or the big bad-guy at the end of the World. Work/play continues to function, in part because it produces. Despite how the mentality of secret societies, instrumental play, and the need for interactive experimental systems, continually pushes towards excess, it continues to produce. The system depends on individuals willing to work within these conditions without asking questions about how or why those systems break down, or where they break down. What encourages or demands secrecy and why do developers acquiesce? Why the commitment to instrumental play, when pushed to extremes locks out the entry of other kinds of players? The ability to experiment and get at underlying systems is crucial to game development, but hindered by many of other other aspects of work/play. Ultimately, as currently configured, this system tends toward excess, rather than sustainability. Why is this the case?

But, this is not simply based on the actions and desires of developers. This is the trouble with Boss Fights, they frequently also hint at trouble ahead. There are other factors, which influence this as well; the constantly mentioned organizational structures, systems of systems, software, and technologies. Not to mention institutions both legal and corporate. You can get the feeling that there is something else out there influencing these worlds. But what are they? That my friends are answers you will find in other castles.

Figure 1.6: A Message to Mario at the End of World 1-4

CHAPTER 2
QUESTING/RAIDING THE LAND OF INTERACTIVITY

World Two focuses on the organization of work practice in game development studios. It revolves around the concept of "interactivity" and the consequences of the desire for it. Interactivity is intimately rooted in the daily work/play of game development, the instrumental play amongst rigid structures, and the necessity of "experimental" tools, which allow developers to produce the resulting complex systems examined in World One. World Two is organized around the four primary disciplines that structure game studios: engineering, art, design, and management. There are of course other aspects of game development, like quality assurance (Q/A), 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 concentrate on the four predominant disciplines that make up a studio's staff. I look at the formation of two new sub-disciplines or specializations amongst engineering and art, those of the "tools engineer" and "technical artist." Each is examined more closely in the section from which it derived.

Interactivity is conceptualized in relation to media and digital games, more specifically. In part because "digital games are interactive media par excellence because their entertainment value arises from the cybernetic loop between the player and the game, as the human attempts, by the movement of the joystick, to outperform the program against and within which he or she competes." This "feedback cycle" is seen arguably as more active than mass media (Kline et al. 2005, p. 14). Interactivity is also structured as a "matter of tactical choices and issues that arise within scenarios whose strategic parameters are preset by a design practice: an invitation to comply or collude in the construction of a particular universe rather than in the deconstruction of its boundaries. ... Different games are designed with varying degrees of openness or closure, option and limitation" (Kline et al. 2005, pp. 54-55). While I am not making an argument about games per se, I am making an argument that work practice has become rooted in interactive "cybernetic" (political, technological, racial, gendered, classed, discursive, performative, and embodied) loops between player and "game" (work place, work product, and broader structure). Rather than a "joystick," work practice movement is generated in numerous forms, also cybernetic. The action/reaction loop between player and game becomes an important aspect to how the game gets played and how people understand the game.[14] Interactivity, however, can be taken too far. Too many feedback loops can cause just as many problems as too few. Interactivity can transition from being a means to out-performance, to an end in its own right. This transition, which I argue, has complicated the lives of game developers.

World 2-1 examines the interactive loops that engineers inhabit. The work practice of engineers though widely imagined is poorly understood. Yet it still dominates the imagination of what all game developers practice. Because video games are historically rooted in engineering or software development work, this conflation of game work with software engineering in some cases make sense. However, it was the coming together of artists and engineers that took video games from simplistic graphics to the colorful immersive ones that most people are now familiar with. World 2-2 looks at the interactive knots of artists. Though the engineer dominates the imagination of who does game development, artists are far more prevalent in the game industry, and massively more prevalent in emerging game industries like India's. While art and engineering are the

14. While I primarily look at interactivity at a more contextualized local level, many would claim that it extends further, "It takes only the merest acquaintance with the facts of the modern world to note that it is now an interactive system in a sense that is strikingly new" (Appadurai 1996, p. 27).

most "visible" aspects of a game, the images and the code that animates them; it is the design of a game, which makes it fun. World 2-3 focuses on the work practice of designers, perhaps the most convoluted of interactive loops. A discipline without a discipline, or a profession without a standard set of tools or practices, game designers frequently draw on numerous intellectual frameworks in an effort to tame their spiraling connections. This finally brings us to the management of a system of interactive loops. World 2-4 examines how managers attempt to deal with the complexity of this emergent organizational system. The interactive systems of managers is massively complicated by their interface with broader interactive "networked" structures, which are examined more closely in World Three. Despite this, it is important to understand how managers make sense of and work within their own internal interactive loops. The Boss Fight for World Two examines the consequences of these interactive systems, how they come together for better or worse in the context of the game development workplace.

2.1 World 2-1: Engineering Interactivity

In my conversations with would-be game developers, two images dominate their imaginations. One is the gamer, someone who plays games all day. The second is the "engineer" or computer programmer. Most quickly discover that one is entirely incorrect, so students interested in making video games frequently enroll as Computer Science or Software Engineering majors when going to college or attempting to learn the tools of the trade. In this respect engineering is the most visible aspect of game development. On the other hand, frequently much of what an engineer puts into a game are not any of those strictly visible components, which players of games associate as making up a game. Most engineers graduate from computer science, computer engineering, software engineering, or mathematics programs. Some either through experience, additional education, or work on the side have developed certain proficiencies in computer science sub-disciplines. Artificial intelligence, computer graphics, networking, sound, computational physics, and encryption are all possible avenues, which engineers may specialize. In addition to being proficient in computer science, most engineers are adept mathematicians, and in particular have developed proficiency with "discrete mathematics," sometimes referred to generally as "computer math." This encompasses numerous aspects of what makes game development difficult. Linear algebra, computational theory, probability, and algorithm analysis, to name a few, are courses that game developers frequently find themselves referencing as having been significantly important in preparing them for game development work, no matter how much they may have hated taking them.

Many engineers have spent significant amounts of time working with numerous "IDEs" or integrated development environments. These are software applications that integrate different parts of the software development process, such as writing source code, organizing file structures, compiling source code into machine or byte code, and most importantly debugging. They will have likely interacted with graphics interfaces like DirectX or OpenGL (often called APIs or Application Programming Interfaces) at some point, though smaller numbers will be particularly adept with the finer nuances of these systems. Some will have done application development specifically for certain personal computing operating systems like Window, MacOS, and Linux/Unix based systems. Those with time spend working in the game industry may have familiarity with the APIs that accompany console systems, for example, XNA on the Xbox 360 or the Nitro SDK on the Nintendo DS. As a developer acquires an expertise, their familiarity with APIs will also expand and become more specific. Engineers interested in physics will become familiar with packages like Havok, Ageia, or the Open Dynamics System (ODE). Each area of expertise typically has an accompanying set of competing technologies.

When an engineer is working, and not in meetings, they are frequently in front of their computers, in front of a white board with other game developers, talking in person with other developers, or reviewing code written by other engineers. Time spent in front of the computer screen can be spent "working" or writing game code or documentation for game code, or it can be spent managing massive numbers of emails from across the company, personal emails, instant messages from co-workers, some work related, some not, and instant messages from family and friends. Web browsing looking for documentation, reading up on industry news, or pursuing personal interests all occur simultaneously. Meeting reminders pop up on the screen courtesy of Microsoft Outlook or Mozilla Thunderbird.

However, most of this time is spent in front of their IDE, the software tool designed for writing software. If they are writing source code, then typically documentation will dominate one computer screen and their code will fill the other. Engineers pour over documentation determining how to take abstract concepts or solutions to problems and make them function in whatever environment or on whatever system they are developing for. While similarities will frequently exist between systems, oftentimes specific code must be written or rewritten if being developed for different systems. Code for ODE is different from code for Havok (both are physics APIs), DirectX is different from OpenGL (both are graphics APIs), and Windows is different from MacOS (both are operating systems or "platforms"), despite frequently similar conventions and goals driving each system.

Most importantly, at the core of engineering for games is the idea of managing a game's state. This includes both the running simulation of the game's rules as defined by designers, and the maintenance and putting into motion the art assets created by artists. Because this process is frequently unknown or at least experimentally defined, engineers spend a great deal of time using the debugger, a component of their IDE, observing what a piece of code is doing, attempting to figure out why a piece of code is not functioning as they intended or believe it ought to, or correcting mistakes made in the process of translating their own ideas into the code, which runs the underlying hardware of a system.[15] The process can be time consuming and tedious, but most importantly, it exemplifies how interactive systems begin to inundate the game development process. Below is an excerpt from one conversation amongst my informants.

> I look up from my computer screen as Eric begins yelling, "fuck you" at his computer monitor. He's in the debugger looking at some of the data moving through the game's engine. There's nothing pretty about this, it is numbers, strings, and source code. From across the desk David looks up and asks:
>
> Did that function fix it?
> No.
> Did that function help?
> No.
> I assume you figured it out?
> No.
> Do you want to talk about it?

15. There are two ways, which are productive for thinking about this process, of determining the functionality of a system through experimentation. The first is the "mangle of practice," there is a "temporal structuring of practice as a dialectic of resistance and accommodation" (Pickering 1995, p. xi). But the mangle is rooted in practice, the second, is getting better at understanding resistances. "Listening to noise and transforming it into a signal depends as much on acquired intuition." The experience of a developer "organize[s] the experimental gropings" (Rheinberger 1997, p. 134).

At which point a math discussion ensues at the marker board.
(Informant and O'Donnell 2005a)

IDEs allow engineers to quickly and easily make changes to their code, recompile the game and see changes either in the debugger, or by observing the behavior of the game.[16] This is the first step down our path of fetishizing interactivity. In some cases changes are made specifically as an engineer experimentally progresses towards a solution to a problem. In other cases changes will be made with little forethought, for example, the addition of a "-" sign before a number or the adjustment of a constant number in the source code, all towards the end of a desired outcome, but with little analytical foundation. This practice is much more common amongst younger developers, many of whom have not had the experience of it back-firing on them; too many minute changes ends up in an unsalvageable or unreturnable past, death by 1,000 cuts. Because of this constantly changing set of what represents a game's underlying code, most engineers have adopted some sort of version control system (VCS).

The VCS can be as simple and inexpensive as a "shared folder" on the network that contains the latest working version of files used to build a game, or as complex as proprietary systems such as Perforce produced by a company with the same name. One key aspect of a version control system is that it tracks changes over time. While a shared

16. It is this kind of experimental feeling out of functionality and behavior, which has lead some to refer to "successful practice is shown to depend on trial and error or on local and contextual knowledge, then that too has generally been acceptable to most engineers" (Whalley and Barley 1997, p. 30). Or the "view of skill as having an improvisational quality is in stark contrast to lab managers' and administrators' conceptions of technician skill," (Scarselletta 1997, p. 207) is interesting, but also problematic. The approaches of engineers have been thrust on numerous other actors who may not work well within these systems or approaches. Again, this seems to reference the "gate-keeping" aspects of engineering work in game development.

folder cannot do this on its own, it can be approximated by compressing and saving the shared folder periodically along with a date. Other systems such as the open source CVS or SVN (concurrent versions system and subversion respectively) are free, though their integration with tools such as Visual Studio and Max are minimal.[17] An engineer using the VCS will "check out" a file of the version control system when they are working on it. This is visible to other developers who may chose to not work on the file at the same time, and in some cases the system may prevent others from working on it at all. When a file is "checked in" after work has completed, an engineer will typically make note of what was changed in the file.

"The Build" is closely tied to the VCS and more than any other technological system in game development is the one that everyone seems constantly aware. So much so that it becomes a kind of obsession, something that everyone must constantly have his or her finger on the pulse of. Traffic lights or strands of Christmas lights are connected to the build machine, providing instant visual feedback, green is good, yellow not so good, and red means "broken." Sometimes the breaking of the build is accompanied by an audio alert, the sound of an explosion or the screeching of tires. A Windows task-bar popup generated from Figure 2.1 delivers warnings or errors to the user. The magenta area of the image is used to make those areas transparent.

---

17. This complicates the matter significantly for Indian studios, who frequently cannot afford the combined price tag of Max, Visual Studio, and Perforce. This frequently leaves them using less expensive, though more difficult methods such as shared folders or SVN. It has also provided entrepreneurial opportunities for Indian development firms. Some studios have developed their own systems for solutions with an eye towards commoditizing them with an awareness of what the Indian market can bear. US based software companies have assumed that emerging economies would automatically adopt the same tools as their predecessors, no matter the cost. Given the extremely high cost of software combined with global differentials in money markets, this has driven foreign companies to develop their own technologies, ones that may eventually compete with those of US companies.

Figure 2.1: The Image Used to Deliver Warnings for Users of the Build System

When combined, the build and the version control system allow for automated builds and testing systems used to keep track of the relative health of a game's progress. These automated builds also dramatically cut down on the amount of time necessary to "build" the game.

Figure 2.2 was generated from the massive number of files that it took to create the "AAA" ("triple A") video game title, *SM3* for the PS2, Wii, and PSP. Because of the large number of files, the build can begin to take a very long time. Automated build systems simplify this by having the latest version of processed files pre-generated, allowing an engineer, artist, or designer to only generate a small number of files to see the latest version of the game along with their most recent changes. Once changes have been tested, they are checked into the VCS, at which point the build system will incorporate them into the next build cycle, at least in an ideal world.

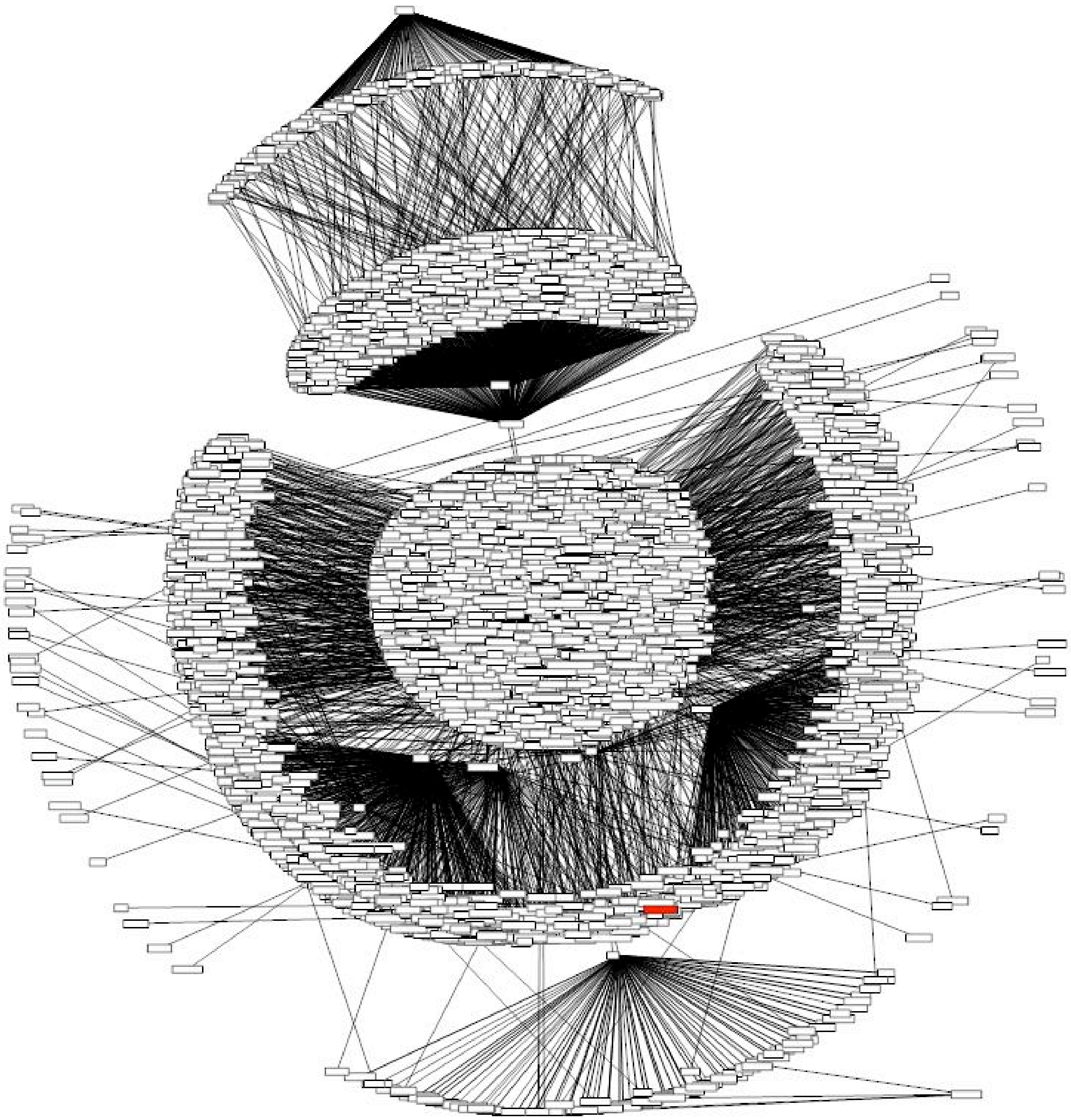


Figure 2.2: A Graph Generated from File Reference Statistics for *SM3*

One informant, a long time engineer and the lead of the project I spent a great deal of time with opined that the problems that engineers face on a daily basis could be broken down into two categories. The first category of problem was string parsing and the second number crunching. Having been an engineer at one time and having experience with software system design, I asked how the design of software "objects" or the breakdown of "functions" into reusable components fit into that equation. He responded that even when designing object-oriented software, what you were really doing was a combination of those two things in an abstract way. Your goal was to produce a system to effectively crunch numbers or parse data; the design is an abstraction, which allows you to better understand how that is being done. As he saw it, games are simply an exemplar of this situation. The upside is that your output, rather than being some web page, database, or spreadsheet, was a graphical and interactive system. But as far as engineers are concerned, the activity of developing software for games was a matter of parsing or arithmetic.

Engineers and specifically "tools" engineers are responsible for developing the software that combines the efforts of artists, designers, and everyone else into a playable game. These software systems read in art assets produced by artists and scripts, levels, or other pieces of "data" produced by designers. Even input read from controllers, keyboards, or mice is passed through the data produced by designers. The knowledge of when to play a particular sound file or display a particular model must come from somewhere, and frequently this is not "hard-coded" into the game code, it too must come from design data. Parsing data or files requires a detailed understanding of their format. Sometimes they are as simple as text files that can easily be parsed by reading the characters from the file and interpreting the information. In other cases the format is more complex. Binary, image, or sound data is for all intensive purposes unreadable by humans, though we may recognize these kinds of data when we see them. These file formats are often more multiplex, sometimes containing compressed data that must be uncompressed prior to its usage in game. This is the process of "parsing" as my informant saw it. Reading data and placing it within an internal representation of the software system. But simply loading the data is not enough to make a game. Nothing has even been displayed on the screen at this point. The loaded data must then be combined, interpreted, and displayed on the screen and in response to the actions of the user.

Engineers are responsible for the tools and workflows, which parse and assemble game assets from artists, designers and even other engineers. The creation of these tools is a process, which demands that an engineer is capable of working closely and collaborating with both artists and designers. These also frequently require engineers to understand the foundations of rendering, effects, and cinematic principles that will be underlying the understandings of artists and designers.

The next difficulty lies in the fact that most video games do not use a common data format, this is further complicated on consoles, because of proprietary file systems, data formats, and application programming interfaces through which they are accessed, all covered by NDA. Each game "engine," or the number crunching and parsing heart of a game, frequently makes use of different capabilities from one another. In an effort to maximize use of game media or processing power, engineers will frequently strip unused data from files, resulting in data parsing systems specific to each game. While things like extensible markup language (XML) or other attempts at industry standard formats have had some traction, frequently these systems are so generalized that they also require a great deal of engine support and specification for them to be of any use.

Because of this, engineers are frequently seen as the gatekeepers to game functionality, as they must implement the systems that expose functionality to artists and designers. Engineers become sentinels because they ultimately have to answer to the hardware resources available to them. "No, that will take too much memory." "Well, if we do that we'll have to rework the way in which those files are being read." "That will definitely break our CPU budget." One engineer told me that his customer is not the gamer, but rather the artists and designers. He was there to make what they wanted to create functional, to the extend possible considering time and hardware.

This view of engineers as gatekeepers can cause friction amongst developers. Artists perceive the unwillingness of an engineer to expose functionality as a lack of interest in the overall visual appeal of a game. Engineers can frequently perceive the demands of artists as superfluous or distracting, preventing the completion of other aspects of game functionality.

2.2 World 2-2: Interactive Artistry

Artists must frequently be the "Swiss Army Knives" of game development, balancing between fine arts skills and technical knowledge. The majority of a video game's CPU usage actually goes to the rendering or drawing of images to the screen. Artists produce the majority of what game players see on screen. Examples include visual interface elements like "health meters" or user interfaces (typically called "HUDs" or heads up displays), or actual game content like 3D models and 2D textures. Animators put these models into motion and texture artists give them "skins." Concept artists sketch foundational pieces that define the look of a game. Full motion video artists assemble game cutscenes. Lighting artists, with an understanding of the dramatic effects of illumination, place light sources in maps to set a game's mood or tone. Most games dramatically redefine the user interface and rarely make use of standard user interface elements. Each of these artistic elements must be created by an artist and made available to the game "engine" in some format.

Generally speaking, artists are trained either in fine-arts programs or more professionally minded institutions, which claim to balance artistic training with up-to-date software package training. Occasionally self-taught artists will make their way into game studios, though in my experience this was the exception rather than the rule. Much like their engineering counterparts, artists develop particular areas of expertise over time. While each might be an "artist at heart," they must be familiar with sophisticated technological tools that are used to create their artistic visions. Even if an artist is familiar with the software package 3D Studio Max ("Max") or Maya, both products created by AutoDesk, they may not be familiar with the particular add-ons or additional applications, which are made to run within these programs in an effort to speed or simplify particular tasks. Artists must be able to quickly grasp and work with new tools, in many cases custom technologies that may not have been designed by artists. When what they have spent so much time creating does not appear in game or does not look as expected, artists must frequently engage with the knowledge of engineers and designers. This provides the foundation for an artist's interactivity.

Much like engineers, when artists are not in meetings, they are seated in front of their computers, in front of white boards with other developers, talking in person with developers, or reviewing one another's work. These review sessions can frequently result in tense situations where artistic style comes into conflict with the overall aesthetics of a game. "Would he really move like that?" "Have you seen the animation that the other team made?" They also come into conflict with technological limitations. "Of course it would look better like that, but I've only got seven bones to work with." "If I do that, I'm going to break our budgets." Entire conversations can be dominated by the examination of several discrete frames of animation or the tweaking of several bones or vertices on a model. In some cases, those animations or models may never make their way into the finished game product. Time spent in front of the screen can be spent working on creating models based off of drawings, taking existing models and modifying them for new purposes, changing textures or the ways in which a texture is distributed over a model, or the way a model has been animated. Computer time is also spent dealing with IMs, email, web browsing, all work related and unrelated.

Many artists feel like work horses, with their inclusion in a project typically the sign of having reached "production." The very distinctions: art, code, and data are telling for artists. Art is often referred to as simply "assets" to be deployed in the game. The number of artists rapidly ramps upward as a game reaches "production." Very few artists participate prior to this point. Those artists who were part of "pre-production" often look forward to entering production, as it is a time in which they can, "simply go in their hole and make some stuff." It is this distinction; art "assets," rather than code or design, have led many game development companies to seek art production work overseas. It was logically differentiable at least at the level of management. But this actually neglects the interactive character of artistic game work.

Most artists on teams in the U.S. have the opportunity to see their artwork "in game" on a daily if not more frequent basis. They understand why they are limited or "working within budgets" for polygon counts, vertex counts, texture size and shape, and the numerous other expectations, which go along with producing artwork for games. If they do not understand the limitations or if they are curious about those limitations, which are harder or softer than others, they can ask their lead, or an artist who was part of the team during pre-production when those limits were decided upon, or they can walk over to an engineer and ask. Some will even experiment, simply determining which limits were hard or soft based upon if those changes break the game or do not.

Many of the developers in India whom I spent time with were working on some aspect of artistic production for games being developed by studios in the U.S.. In most cases the games they were producing artwork for were established game franchises, sequels in many cases. These games were already in full production mode, with the limitations and requirements for artists already established. Almost none of the artists working with these games saw what they were creating within the game itself until after it had been released. Comments from U.S. based artists and designers would come back to them annotated for changes, but there was no clear understanding of why or how the limitations that they were working within had been established. The interactive and experimental aspects were removed and, worse yet, made completely unavailable. In particular this proved difficult for an Indian team of artists and engineers working on a prototype for the Nintendo DS. Having not had an experience of working with any of the libraries for a console like the DS, the engineers were largely left to learn the conventions of the system from Nintendo-supplied documentation and private messaging boards. These resources did not make it clear how to actually make games in practice, either for the engineer or for the artist. Instead the artists and engineers found themselves walking from desk to desk, transferring files, converting files, asking one another questions, making changes, looking at the debugger together, and struggling with the lack of interactivity. This process of making the transition from the tools of the artist into the game is frequently called the "art pipeline." Pipelines are the least talked about, least documented, and frequently most identified as critical points in the game development process. They are also the places that when changed or re-defined can result in massive amounts of rework for artists, engineers, and designers. Because they are frequently defined as a game is being created, they must often go through several iterations based on their use. Unfortunately, this process typically occurs much later in a games development cycle than most would like.

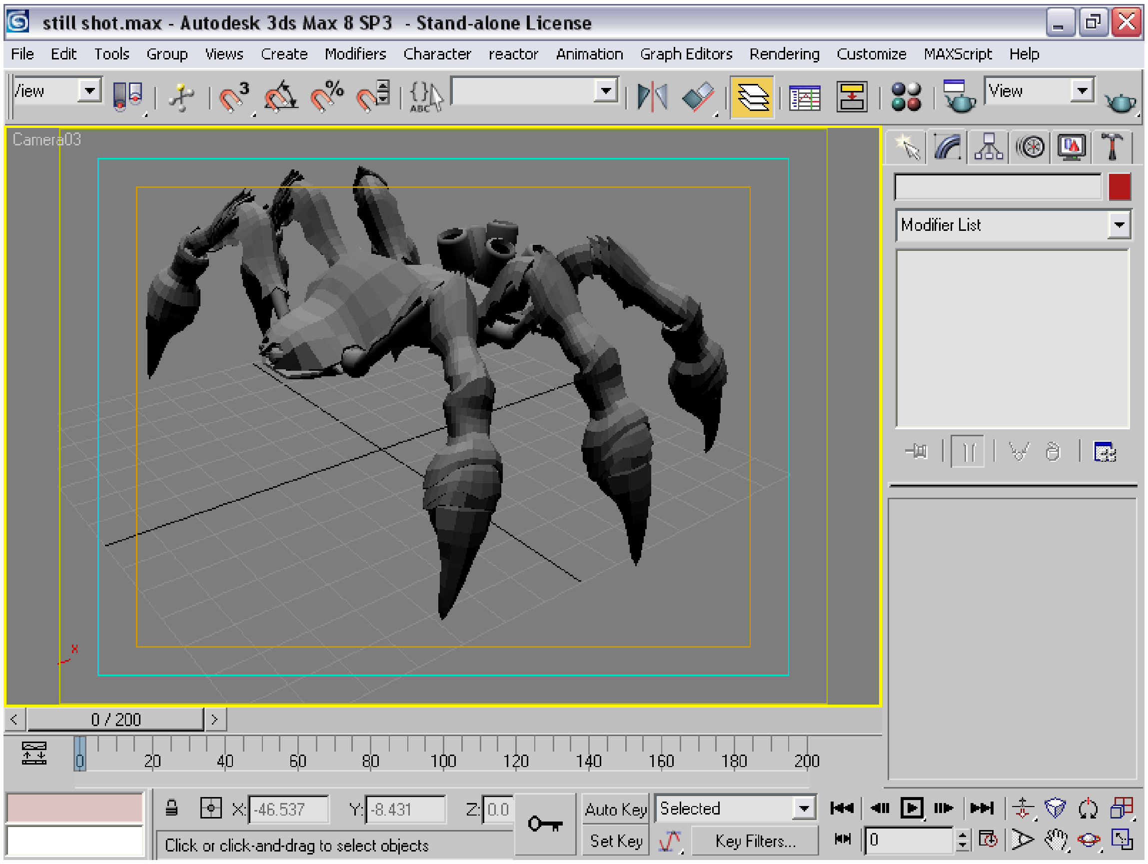


Figure 2.3: A Modelers View of the World in 3D Studio Max

Modelers and animators work primarily within 3D Studio Max (Max), the screenshot above is show as an example. Texture artists primarily work within Adobe Photoshop. All of the data, which they create for these models, are stored within a single Max file or texture file. Much like an engineer's IDE, these tools are the experimental apparatuses, with which artists work within. So too, like engineers, artists typically use the VCS to store a historical record of their work. Unlike engineers however, there is no choice in being able to work on a file that has already been checked out. Because of the data formats of these files, artists are unable to simultaneously work on the same files, unlike engineers who when necessary can work simultaneously and "merge" their efforts later.

Pipelines typically begin in Max. It has an extensive set of scripting and exporting features, which technical artists and tools engineers use to extract the art data for placement into the game. For most artists in the U.S., this means by the time full blown production on a game has begun, they will have a "make art" button within Max that will export the necessary data from their work in a format such that it can be seen within game. This allows artists to more quickly tweak and view their work. Artists are limited in that there is no "debugger" for this process. If something does not work or does not appear correctly, there is often no obvious way to determine how or why. Artists typically then proceed to make changes to their models, textures, or animations in the hope of feeling out the reasons. This is often a point of conflict for artists and engineers, who continue to negotiate pipelines throughout the development of a game.

This is where the experience of the "gate-keeping" is felt explicitly, though broadly labeled as a "communication issue" in studios. Because technical artists and tools engineers are responsible for the creation of the tools that power these pipelines, they are also often responsible for other things, and time is always at a premium, causing conflict to frequently arise. At one level, artists and engineers speak different languages. The same words have different meanings. A technical artist and I put together a Game Developers Conference submission, which was subsequently rejected, that attempts to get at this disconnect.

Programmer: "I need this model in under 300k."

Artist: "Ok."

> Artist: *Spends a week and makes the model in under 300,000 polygons.*
>
> Programmer: *Head explodes.*

The "joke" is that the engineer meant one thing and the artist heard something entirely different. The engineer was thinking about the size of a file on disk. The modeler interpreted the statement as being the number of polygons that make up the model. The session description continues.

> Artists and Programmers have worked together in games ever since the first game programmers said to themselves, "My art sucks." From that day forward, we have tried to integrate Artists and their craft into this highly technical field. Here in 2006, we should consider this a work in progress which all the disciplines of game development can endeavor to improve upon. This session dissects common issues and provides solutions in the Artist/Programmer relationship that development teams of all sizes face.
>
> A few months ago, the two presenters spent time speaking with both programmers and artists at Vicarious Visions. They conducted a one-hour roundtable session for artists only, where they could talk about what they did and did not understand about programmers. Then they ran the same session with only the engineers. The one thing that amazed both of the presenters was how professional and genuine both sides were. They both wanted only the best for the game and their team. How then, could they end up at each other's throats in the middle of development? What is the problem?
>
> The presenters will begin by having the audience ask themselves the following questions:
>
> Artists:

- Have you ever tried to suggest a feature to a programmer, only to walk away frustrated and upset?
- Are the in-house tools that you use bug ridden and overly complex?
- When you have a problem that programmers can solve, do you hesitate to ask for fear of the response?
- Do you find yourself overwhelmed with techno-babble?
- Do you ever feel cut out of the loop in designing your own workflow?
- Do you ever operate outside of your team structure and go to a programmer on another team for advice?

Programmers

- Do artists break your tools and game code with stunning regularity?
- Have you ever given an artist a checklist of steps to follow, only to have them fail to do so repeatedly?
- Have you found artists performing mind-numbingly repetitive tasks that you could have fixed with code had you only known about them?
- Have you ever needed a simple art fix, only to have an artist tell you that you are asking for the impossible?

(Informant and O'Donnell 2005b)

Video game artistry, in many respects, is interactive based upon the tools that define it. Yet that interactivity can break down, resulting in a different kind of interactivity, one in which artists and engineers can "end up at each other's throats."[18] In part it is rooted in disciplinary ways of understanding the world and what is being created. However,

18. In many respects I see this breakdown related to "standards" and "classifications" that "may become more visible, especially when they break down or become objects of contention" (Bowker and Star 1999, pp. 2-3). But it goes beyond just social relations or disciplinary differences, or differentials of power in the setting of standards and classifications, which are typically developed in already compacted inconvenient timeframes. It has just as much to do with "institutional" deficiencies that prevent broader discourse about "standards" or "classifications." The demands for secrecy prevent that. These dictates are examined more closely in Worlds Three and Four.

it is also rooted in lack of conventions, standards, documentation, public discourse, and other institutions. All can be felt even more explicitly in the worlds of game designers.

2.3 World 2-3: Designing Interactivity Interactively

The "designer" is a relatively new discipline amongst game developers, but it has quickly become the professional aspiration for young game developers, likely because designers are the front line for constructing what is finally viewed as the game. While the imagination of engineering as the work of game development, it is frequently the game designers, which occupy the privileged position as "author." The Will Wrights, Shigeru Miyamotos, and Richard Garfields, famous designers of the game development world, typically lead an army of game developers, artists, engineers, and other designers, which construct the products that are then credited to their generals. While design is something that has long been a practice amongst those making games, the specialization has been relatively recent. Every designer I met seemed to come from a different background: physics, computer science, media studies, film studies, graphic arts, writing, or journalism just to name a few. More designers were "self taught" than artists or engineers. Though they seemed to come from every disciplinary background imaginable, the common theme was: designers are gamers more than any other discipline within game development. Designers frequently had skills that seemed to transcend disciplinary boundaries. Designers must possess analytic skills, which allows them to deconstruct games, examine their core elements and mechanics, and determine the underlying rules and structure of a game.

Because designers must play a significant number of games to make this possible, they frequently speak in the language of games, rather than in any single disciplinary language, "Like Spy vs. Spy," this language becomes a bridge between an imaginary concept and an actual game mechanic. Designers were fluent in the language of games broadly defined, including tabletop games, role-playing games, board games, or video games. The mechanics of games is what drove their interest in game development. Many came up through the organization as quality assurance testers; others had transitioned from engineering or art to the design teams. Some designers have taken existing video game engines and customized them, building MODs or levels to demonstrate their abilities. Those coming out of software heavy backgrounds may have created small stand-alone games. Those that were hired directly as designers were coming from other game studios where they had followed similar tracks through the organization. Though "game design" programs have existed for some time, my informants say that thus far, there was no indication that these students were any better at design than someone who had come out of a physics program.

Because the "science" of game design (and perhaps design more generally[19]) is in its infancy and most game developers have had a difficult time deploying many of the ideas developed in the academy, or simply do not have enough time to implement them prior to the next deadline, most designers expressed the urgent need for better tools or new ways to talk about and do game design.[20] Nebulous ideas like "play," "fun,"

---

19. The idea that, "Design is thus best seen as a process of communication, negotiation, and consensus-building," (Bucciarelli and Kuhn 1997, p. 214) has been a useful one for me to think with, especially in the context of the video game industry, because so much communication and negotiation is involved. It also dramatically complicates the idea that the designers are solely responsible for the final game, which may have been dramatically shaped by numerous other forces.
20. Here, the idea that game development companies, like many new media companies are continually in "permanent beta" is a useful conceptual category to think with. "The influence of design - where the design of products, technology, or services - and organizational form on each other emerges partly due to the process of continual technological change, in which the cycle of testing, feedback, and innovation facilitates ongoing negotiations around what is made and to organize making it. We call the organizational state of flux that emerges from this negotiation 'permanently beta'" (Neff and Stark 2004, p. 175). I also see this as a product not only of the process and organization, but also

"verisimilitude," and others abound. Each propels and constricts new designs in different ways. In the vacuum left by no formal methods, designers speak in terms of games, all sorts of games. The critical gap for designers is making the leap from talking about games to constructing games. They must be able to translate that gamer vocabulary into the intermediary languages of engineers and artists. Designers must be able write and choreograph the experience of playing a game, balancing their personal desires with what other players will find fun. This means that designers frequently cross between the worlds of artists and engineers, and experimentally construct ways through which they bridge code and art through the nebulous mechanism "data."

For the most part, generating data consists of creating files, which are combined with artistic "assets," and interpreted by the underlying source code written by engineers. This can be an XML file or files containing scripting languages like Ruby or LUA, which direct game-code how to behave. Scripting is similar to programming, though scripts are interpreted during the execution of a game rather than pre-compiled into the native machine code. Sometimes these activities are enabled and assisted by custom tools built by tools engineers and technical artists at their company, other times external software packages are purchased, and other times they may only be able to work with a text editor. This combination of art and data passed through the underlying code defines the structure of the game. This of course can intensify the feedback loops amongst engineers, artists, sound engineers, and designers. If changes must be made in different places to accommodate new concepts or approaches as a game is developed, then changes may be required throughout the different components that make up a game. This

out of relative youth and lack of institutional memory, which these industries seem to exhibit. There is nothing inevitable about the permanent beta state.

is not the "fault" of designers, merely a product of their position in the creation process. This means, that much like their fellow game developers, designers when not in meetings, also split their time working in front of a computer screen, working with designers, engineers, and artists in front of marker boards or chatting in person. Their time in front of the screen is dominated by work with custom software tools created by engineers, simple text files processed by the build system, and the now standard assembly of inter-corporate and extra-corporate IMs, email and web browsing. Below is an example of a text file, which is interpreted by the games underlying code to generate a particle system that looks like fire. Other text files indicate where this particle system is placed in the context of the game.

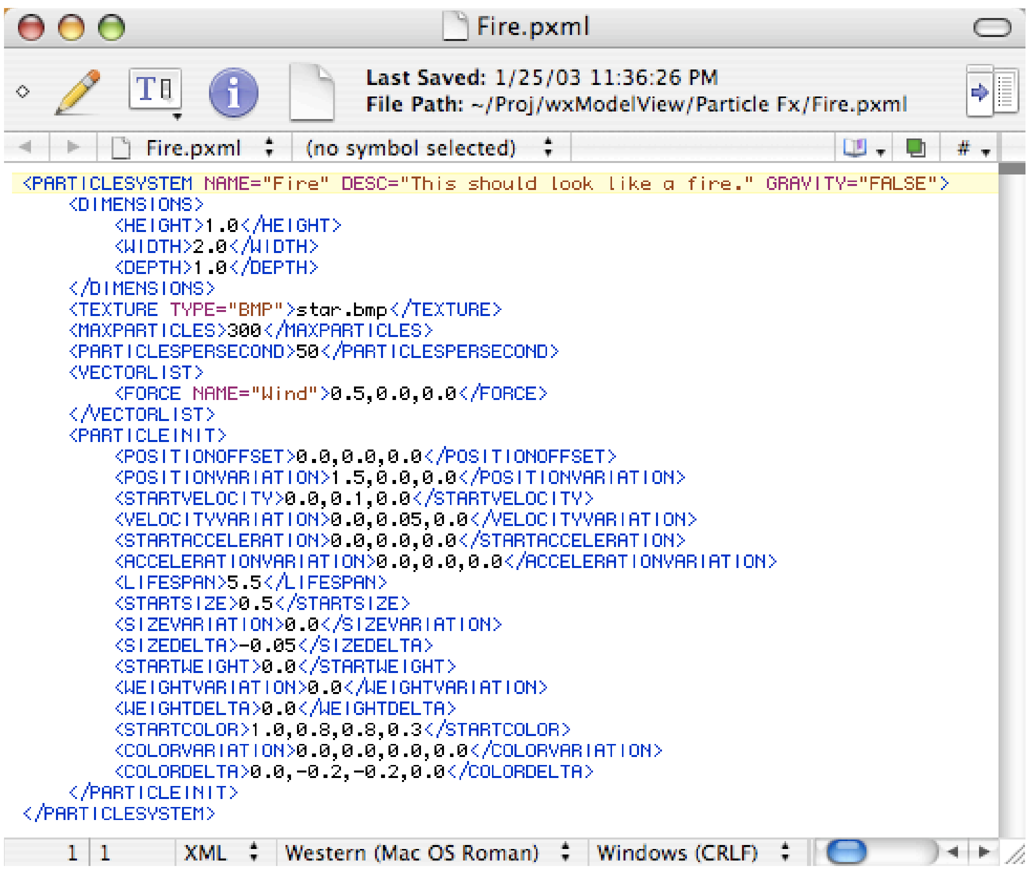


Figure 2.4: A Text Editor and XML Describing a Particle or FX System

Designers occasionally mix up this standard amalgamation of computer time by playing games that have preceded the one they are currently working on, with a critical eye towards what is enjoyable, and what is not. Other games are played to work out new game mechanics, which might become useful in the development of their own components. Perhaps second only to tools engineers and technical artists, designers must have excellent communications skills. They must be able to collaborate and work well with one another, as well as the engineers and artists, which they connect through their rather abstract job of generating data for a game's engine.

Designers were one of the most difficult game developers to find in India, in part because gamers can be difficult to find. Beyond the general derision that parents level at video games, the idea that game development is a profession has not yet caught hold. A son or daughter interested in making games faces the assumption by parents that they, "are going to be playing games all day." Video games are still viewed as a diversion from those educational tasks, which students ought to be preparing for. For this reason, design has been a difficult leap in the Indian industry. This is made more problematic by the general lack of professionalization of game design or game development more broadly. The technical, social, and procedural connecting of engineers, artists, and designers in ways that enable collaboration has also been difficult. Because these practices have been developed experimentally over time, through experience, and are entirely undocumented, means that they are rarely communicated outside studios that develop them.

As previously mentioned, there are no commercially available "standard" tools for designers to use in the process of making games. This has become the job of technical artists and tools engineers at each company, who create new software systems for designers, artists, and engineers. While there are numerous "middleware" companies creating tools, frequently these fill "engine" gaps rather than tools gaps. Meaning, they are designed to be used in the underlying "tech" or code of a game, and while they have tools of their own, they cannot link together all of the other pieces of design, which must be done to create a game. The "tech" is one of the most important pieces of code in game companies. They are the foundational pieces that form the core of every engine or game, which a company creates. This can include aspects of the pipelines as well. My primary field site in particular has spent years and massive amounts of money and mind power to create their internal tech. This system was originally purchased from a company called "Alchemy," which retains the name. More recently, the goal has become grander, creating a foundational layer of tech and tools, which can support not only a standard art pipeline, but also the data pipeline of designers.[21] It was with this in mind that "Peaches" was created. The name was based on a historical practice of the tools team that stipulated that all tools would be named after some sort of food, which could later be justified by an accompanying acronym.

Peaches was created in part to assist in dealing with the complexity of a project like *SM3*, and which Figure 2.2 illustrates. The sheer number of files and references between files requires a new set of tools for developers. While level editors and other systems have been quite common in the past amongst game development companies, the goal of Peaches was much broader. It was designed to be a system that could be expand-

---

21. The "mangle" of game development is in part a product of the dance with hardware systems that may or may not work as advertised. This is further complicated by "the world, ..., continually doing things," and these things, such as electrons moving through circuitry, media devices spinning up, power flowing out of a batter, are frequently mediated or "threaded" through technological devices (Pickering 1995, pp. 7-8). The situation is even more complicated if your devices are highly unpredictable, and not necessarily documented.

ed on, as new kinds of design data were necessary in future projects. The same tool could be used by designers to create special effects, levels, missions, or the scripting of cut scenes.

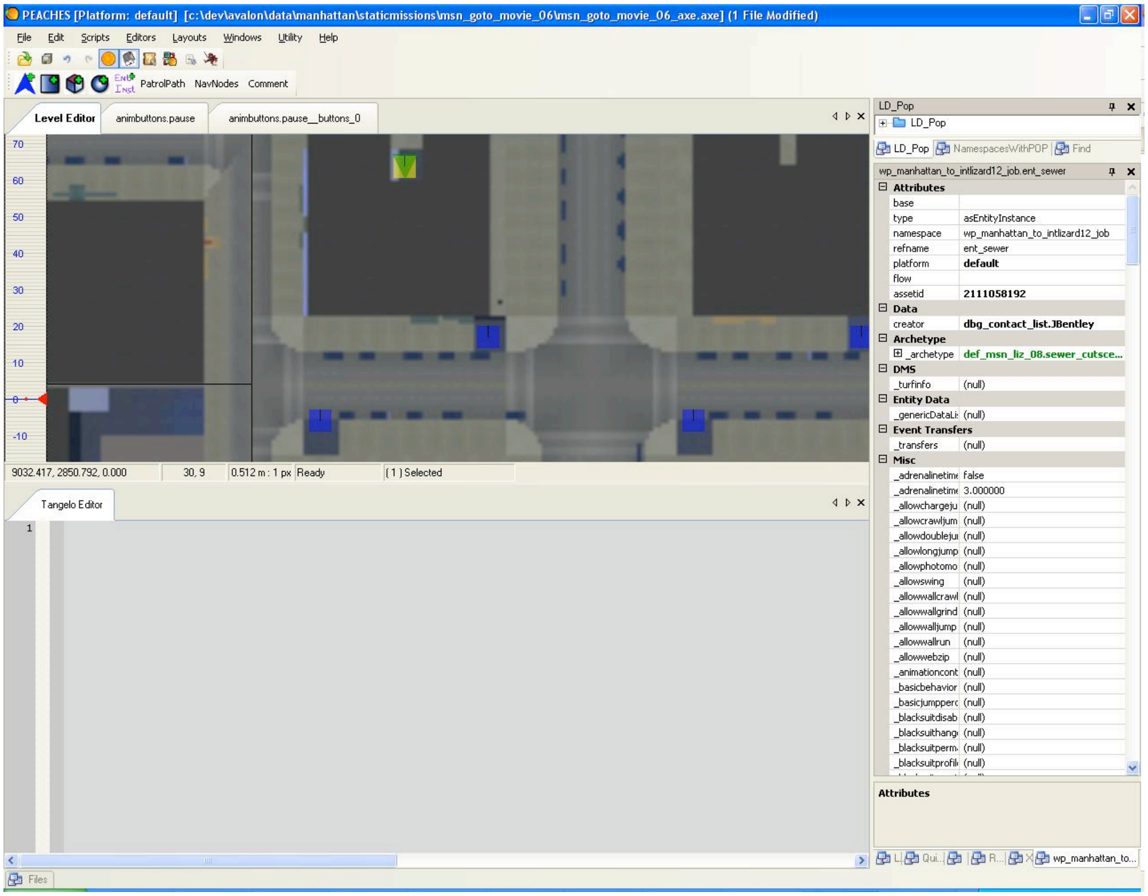


Figure 2.5: "PEACHES" in Action Editing a Game Level

The worlds of game designers are intertwined with those of engineers and artists. Each depends on the other for the successful completion of a game. In the interim, with no standards and no standardized tools, designers must constantly work with and without artists and engineers in the construction of virtual worlds, story-lines, and characters for, which there is no agreed upon language. Designers, much like artists are constrained by their tools, the game's tech as defined by engineers, and the ability to translate abstract concepts into forms that work with each of these. Designers do not locate their ire at engineers, however; instead they take the tools they have managed to cobble together and run with them, attempting to make them do things they were never intended in the hopes that what results is indeed "fun." The interactivity of game development practice only increases.

2.4 World 2-4: Managing Interactivity and Managing Interactively

As an engineer, artist, or designer proves their abilities and gains experience, they will more than likely begin to move into either a "lead," "manager," or "producer" role in a game company. This is assuming that the studio has grown large enough to warrant these roles. For the most part these distinctions were quite similar between U.S. and India based studios. In smaller studios, every employee fulfills part of these roles out of necessity. Though similar, leads, managers, and producers[22] are different and the distinction is important.

A lead is frequently an artist, engineer, or designer who has proven their ability to produce quality work and exhibits some leadership characteristics. They typically come up through the ranks of an organization. Their responsibility is to represent the group's interests in meetings and planning sessions. They tend to work closely with their teams to ensure that they have the information they need to adequately report back to producers and management on the status of a project. That is the key difference between managers and leads; leads tend to be responsible for a single project.

Managers on the other hand "manage" across projects. For the most part managers do not take part in the production of games. They may be involved in the production of internal resources for their teams, but their overall goal begins to shift to ensuring

22. The term "management" is used to reference these as a collective game development endeavor.

the long-term success of their "groups" as they are typically called. Most managers, like leads have risen through the ranks of the organization. These are frequently artists, engineers, or designers who have been with the company for a significant amount of time and have been willing to move into managerial positions. Some developers choose to remain leads rather than moving into management positions. Ultimately this means that very few members of a game studios "management team" have any management training. Some organizations will work to train in these areas, such as communication and leadership.

The role of the producer is the management of a project or, for executive level producer, several projects. In the end, they are responsible for the overall quality, profitability, schedule, and effective production practices of a given project. Producers must also understand the scheduling and staffing needs of a project and ensure that milestones are met. The producer is the person who ultimately, at least organizationally, is considered responsible for the relative success or failure of a game. Producer roles generally seem similar from the U.S. and India, though the number of producers in India was diminutive, as the number of "end to end" or complete game development projects was much smaller.

In the description of the roles of engineers, artists, and designers, the phrase, "when not in meetings" was used as a recursive feedback placeholder. The interactive organization loops in on itself. Management schedules meetings to better understand what is transpiring at each level. Leads will have team meetings, producers will have meetings with leads, and managers will have meetings with their disciplinary groups. Studio heads will have meetings with managers and producers. Entire teams will participate in company wide meetings. Information flows in both directions, but primarily it flows up the chain of management. Management schedules meetings, but they also participate in them, their time is dominated by meetings and ensuring that things are functioning properly. But the importance of face-to-face meetings, which U.S. or Indian developers will arrive early at work or stay late for "manifests a delicious contradiction; work becomes more dependent upon workers' abilities to create close social relations at the same time as globalization inhibits their construction" (Hakken 2000a, p. 771).[23]

Much like designers, leads, managers, and producers of video games have had to assemble their tools in an ad-hoc fashion, a small few borrow from more formal project management techniques. They make use of data gathered from the VCS, build logs from the build system, or even implement other technological systems for developers to report their work or time spent on projects. Others seem to have extended the role of lead artist, engineer, or designer into that of a project lead. In some cases this is effective. In other cases, management who take this particular approach will often find themselves micromanaging certain tasks of the development process, those that they likely had been responsible for in their previous role as lead. Much like leads and group managers, producers also suffer and benefit from having been active game developers. It gives them an intimate understanding of how games are created and developed, but it frequently results in lack of management training and sometimes with an actual disconnect from develop-

23. This seems to contradict the idea that work though, "increasingly individualized, labor is disaggregated in its performance, and reintegrated in its outcome through a multiplicity of interconnected tasks in different sites, ushering in a new division of labor based on the attributes/capacities of each worker rather than on the organization of the task" (Castells 1998, p. 502). Furthermore, based on my experiences, the unfortunate side effect of this has been increased time at work, to make up for the re-socialized workplace. This further contradicts the idea that, "skilled labor is required to manage its own time in a flexible manner, sometimes adding more work time, at other times adjusting to flexible schedules, in some instances reducing working hours, and thus pay" (Castells 1998, p. 468).

ers because their attention may be focused more particularly on those areas with they are most familiar.

This configuration is exacerbated as corporations globalize and become a more "complex tangle of remotely related parts," "both tightly coupled and dispersed," a recipe which as discussed in the Boss Fight for this Level proves particularly problematic (Fortun 2001, p. 93). The process of relating the parts to the whole must eventually occur during the development of a video game, it cannot remain a complex tangle, it must come together. This process frequently gets out of hand because feedback loops are too plentiful in some locations and lacking in others. The fetish becomes to provide more feedback loops that again result in an aggregate of unparsable information.

When developers talk about what is missing, or more than anything the aspect of game development that prevents them from being able to work well, they frequently settle on the highly problematic term, "vision." Vision, as they conceptualize it, is a clear idea of what you want at the end of a project. Vision is assumed to come (or not come) from somewhere above, delivered to help developers understand how to direct their experimental efforts. When a vision is combined with a plan for how that vision can be brought to life, the work can then be scheduled. Unfortunately because of the constantly changing technological landscape, frequently "how" is also an unanswered question. "It ignores the fact that outcomes are socially accomplished in context rather than individually calculated ... it ignores the fact that outcomes are often not consciously calculated, or even intended by any one of the parties involved" (Knorr-Cetina 1983, p. 130). If both, what and how, are unanswered, then your ability to plan becomes significantly compromised. The conversation below indicates how many developers have come to see vision as central to their undertakings.

> Umm...I would have to, well, my opinion, which is wrong, its umm...lack of vision. Umm...if you know what you want, then you can get it sooner, but most of the time, you don't know what you want, so you have to see it. And, umm...so, you have the deadline, lets say you have the deadline that's 100 days away...At the beginning, you don't really know what you want, so you just kind of like, like trying things out, and like, you get one thing, and you can't really tell how that works, because you need these other things in place, but you didn't know quite how the other, what the other things would be...because there isn't like an overall...theme. And then, the later that you go, the more concrete things get, and its more apparent with the pieces that you have, what you have to do, and so then you end up with, its kinda like how each stage, at pre-production, everything is very free, and idea flowing, and then when you go into production, and try to execute those, and then you're at the end of production, and you see how everything turned out, which is probably very different from where you started. But if you had a clear vision of what you wanted at the start, then you would have known...what differences you needed to make into the system. (Informant and O'Donnell 2007)

Having spent three years in game studios in the U.S. and India, I tend to agree. that games, which begin without a clear vision of what they are supposed to be have trouble being implemented prior to their deadlines. Furthermore, games without a clear vision are also ones that fall prey to the multifarious desires of those companies, which fund its creation. Without a clear vision, there is no reason to protest "feature creep," or the additions of aspects that simply have no place in the game being developed. Developers often run after half formed or vague ideas of how to go about realizing them, only to find themselves miles down a road, which may not have been the best one to first traverse.[24]

24. This "creative ambiguous process" or the necessity of "intellectual flexibility" is both empowering for many, but also places the onus of production on the individual. If unable to produce, then they

Perhaps it is too idealized to expect to know precisely what you want and how you are going to get it. The "waterfall" method of design is a concept from the dark ages of game or software development. On the contrary, all I believe is that like many of my informants point to, you need a clear vision of what you want, with the knowledge that it will likely shift in new directions throughout the life of a project. The distinction is that the game will swerve in directions that fit the overall vision, not in random directions that allow the overall structure and rules to take control of the development process.

This vision problem becomes particularly problematic for Indian developers. While they frequently have more detailed contracts that govern their relationships with developers in the U.S., the constant flux of project needs in one location impacts the other site. Rework of art assets is common when changes to the art pipeline in one place require changes in another. Managers must either negotiate change orders with the studio that has contracted them, or have employees make the changes without adjustment to the contract. Because U.S. developers are so accustomed to rework, Indian studios risk amiable social relations when they ask for compensation for rework activities, which U.S. developers assume, are "natural."

2.5 Boss Fight: Bowser Bites Back - Differentiating Between People and Systems

The pulling away from "interactive" systems as goals in and of themselves is the "Boss Fight" for World Two. It argues that the emphasis needs to be on understanding where interactivity is productive and where it is destructive. When and where is it right for less

---

simply are not skilled or smart enough. Even more troubling is that while "desired results or functions are what are demanded of workers, the contextual mechanisms, by which output is reached, while often dictated, frequently has little to do with the actual means by which things occur," (English-Lueck and Saveri 2001, p. 8) yet workers are often judged based upon dictated demands rather than on the contextualized mechanisms necessary to actually do the work.

interactive systems and practices? This interconnectedness of game development work cuts to the heart of what makes it both able to produce the technologies it does, as well as why it can be so unpredictable and complex for those working in it and those attempting to manage the globalization of this industry. Game developers stress not only the sheer number of disciplines, which go into the creation of games, but the numerous forms of communication that ultimately becomes a goal in its own right. Despite this, developers continue to talk past one another. The relationship between interdisciplinary work and communicative collaborative practices are constantly on the minds of developers.

> Between…I mean…practically every discipline of everything you could think of goes into making a game. Even more so than movies. Every form of art practically ends up getting piled into there. Writing, visuals, engineering, I mean all kinds of different people kind of piling into it. So, I don't think there are too many things that are like that.
>
> ...
>
> I spend more time communicating than I do doing anything. And that is kind of the nature of where I'm at in the project that I'm in. I'm in pre-production, and that is all about making decisions about how things are going to go. So, it requires a lot of communication. I mean, I would like to see, just…[LAUGHS]…I don't even know how. You spend so much time writing emails, and posting on forums, and having meetings and everything, if there was some way we could just mind meld [LAUGHS] and just get it done in a more streamlined way, but I don't know what that would be. (Informant and O'Donnell 2004)

At some point during the pre-production phase of the title I watched from start to finish, the lead technical artist came to me looking for advice. He had been advised that "communication was not so good" between the art group and engineering group. Being located at the site of turmoil, he turned to me asking for guidance. We spent most of a morning talking about the situation, a massive amount of time for a game developer at work. We collaborated, discussing both of our observations, and came to decide that the real fault-line was based on disciplinary difference of understanding what makes the project tick. We identified four different ways that the project was being viewed, illustrated by Google Maps and Google Earth. Figures 2.6 and 2.7 represent different perspectives of the same problem and its subcomponents. They were separated by scale (level of detail) and in content (art or code). We found that artists were typically interested in understanding the game in a way that favored artistic aspects (represented by the satellite images). Engineers were primarily concerned with implementation (represented by the road maps).

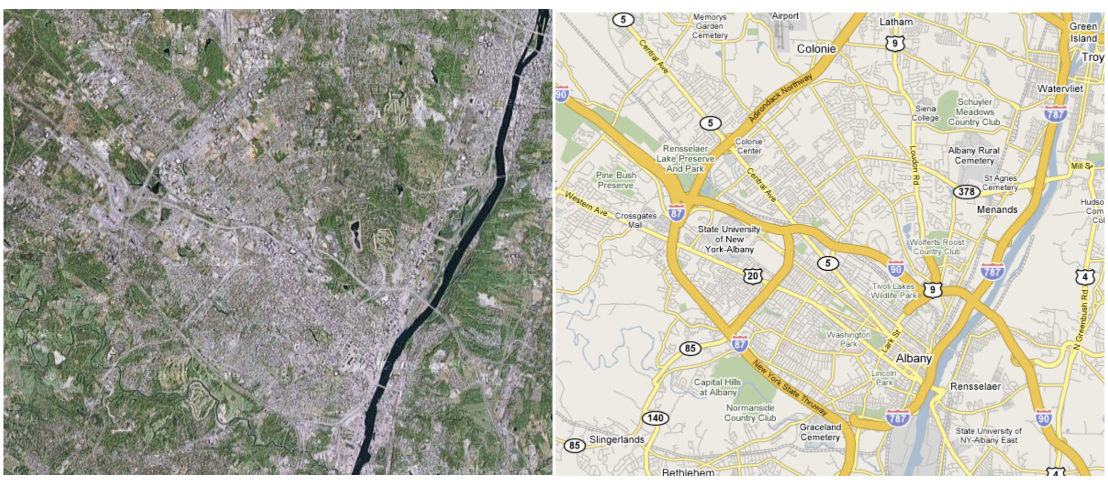


Figure 2.6: High Scale Images of Art (left) and Code (right) Conceptions

While one possible resolution would be to simply lay the maps on top of one another, making a hybrid, a popular solution amongst many developers. Unfortunately, it was not be a helpful solution. Attempting to teach your artists about all of the engineering aspects and vice versa, would be both cumbersome and likely impossible. The utility of specialization is that they should not need to know everything the other knows. Different scales and content is useful. Homogeneity was not the goal.

The higher scale "engineering" or "art" map (Figure 2.6) illustrates the viewpoint of the engineering or art lead. Based on the map, it is obvious that this person's greatest knowledge will be the overall functionality of a system. They will likely have less knowledge of the system's lowest level of functionality (represented by Figure 2.7). Nor will a lead artist have the details of lower scales. But these acknowledged differences are necessary for the project to come to completion.

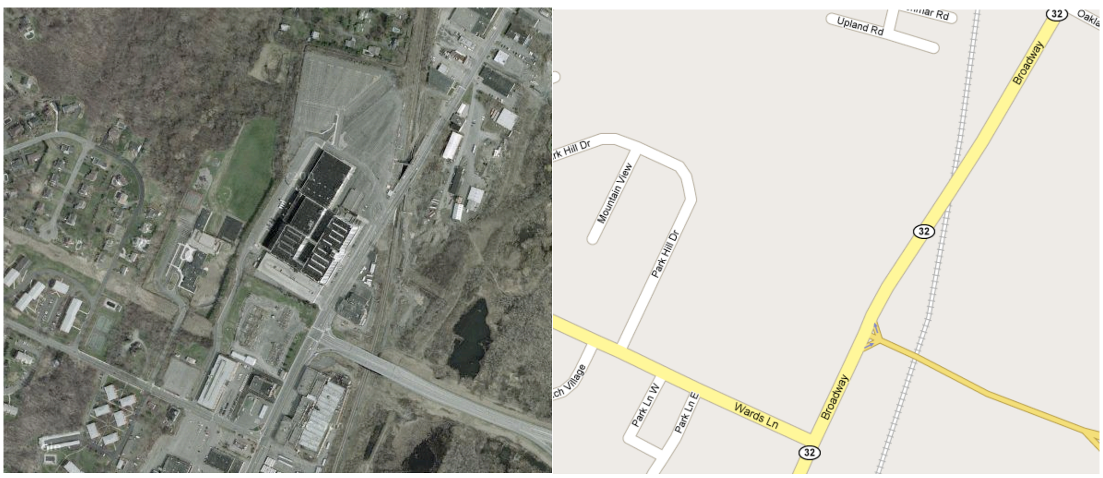


Figure 2.7: Low Scale Images of Art (left) and Code (right) Conceptions

The solution was to encourage all parties to understand the utility and "correctness" of each interpretation. While engineers might control the flow of art assets into the game, artists wanted some information about why an engineer was saying "no." Engineers also needed to understand why artists were attempting to create certain effects or models. We encouraged both groups to understand the differences in their viewpoints and scales people were working at, providing developers with a new language for discussing collaborations helped them work together.

Anthropologists of science and technology have demonstrated that it is frequently at these "faultlines" or the intersections of disciplines are where the most interesting and critically important outcomes occur (Traweek 2000). Historians of science have similar findings, noting that it is at these sites or "trading zones" where "creole languages" emerge and local coordination and cooperation can be worked out in practice (Galison 1997). Intimately important to this process, however, is the ability to get at the underlying systems that historically situate our object of concern. This in particular is significantly limited in the video game industry. For the collaborative process to really function, it is important for "open system analysis" to be possible, a process dependent upon the historicization of the object of concern (Fortun 2006). Because of the emphasis placed on closed systems and closed collaboration, it becomes difficult to historicize or situate the object of concern.

Of course, the question remains if this "solution" changes anything, does providing new tools to think with matter when it comes to the production schedule of a game? While the hopeful answer is that over time it will change, the short-term answer has been that it has not yet altered how people work.[25] In one recent case, the producer of a project was meeting with the executive producers from the publishing company for their upcoming game. The publisher commented that the buildings in the Bronx, despite being to scale, did not look tall enough in SM3. The producer wishing to please the publisher went to an artist and asked that he scale all of the buildings in that block so that they

25. In part I see the lack of change in how disciplines work together as connected with the continued separation of disciplines amongst game developers. It matters which group you are working with. Are you an engineer, designer, artist, or manager? The setting of groups off from other groups creates a "sort of Mafia," or "inbred group of buddies," who "do things the way the want" (Fortun 2001, p. 116). This exacerbates the differences and separations between groups, which matters when they are constantly interacting and working within and among those elements which each other makes.

would be taller, a task that was relatively easy for the artist. Unfortunately, since the producer had bypassed the art lead, no one thought to check that all of the collision data (the data used to determine what objects in the game a game character "collides" or impedes their motion, such as the ground, a rooftop, or a wall[26]) would also need to be altered. Rather than impressing the publisher, the resulting demonstration, in which characters passed through walls and fell through the ground, caused the publisher to doubt the capabilities of the development team.

2.5.1 "Leeeeroooy Jeeennkins" - So Much for Best Laid Plans

In the spring of 2005 a video clip released on the newly introduced Google Video service began making the rounds of VV's offices. Even though eventually everyone loaded up the video on their own machine, you would frequently see small groups of developers crowded around a machine watching the clip together, bursting into laughter at the very moment "Leeroy" charged into battle to the dismay of his clan brothers, who were discussing the tactics they would so delicately deploy to ensure victory.

The two and a half minute clip was made from within the game *World of Warcraft (WoW)*. It features what appears to be a WoW guild, "PALS FOR LIFE," preparing for a "raid" of a difficult sections of the game. The team members are busy audio chatting with one another about the tactics and actions which they will deploy upon entry into the room. The conversation occupies the first nearly minute and a half of the clip. Their careful preparations are suddenly interrupted when one of the guild member, "Leeroy" returns from being "AFK" or Away from Keyboard only to yell, "Alright,

26. "Collision data" being stored separately from the model geometry of a level is one of the many contextual game development practices that are never shared more broadly. Storing collision data separately allows for faster or "cheaper" computation of collision detection. Rather than using the level geometry, simpler objects, such as a sphere or box can represent more complex objects.

time's up, let's do this! Leeeerooooy Jeeennkins!" There is a moment of stunned silence before the raid leader says, "Oh, my god, he just ran in." The clan then attempts to run in and complete the mission only to be "wiped." The video ends with the death of the guild members and much audio chatter about how stupid Leeroy is, to which he replies, "At least I have chicken," which can only be assumed to be his reason for being AFK.

Figure 2.8: Screenshot of Leeroy Jenkins WoW Movie Clip [27]

Though I had forgotten Leeroy for a while, when I least expected it, it resurfaced. I began to think of it as an effective analogy for how, despite all attempts to otherwise stave off defeat, when your "chances of survival" are only "32.33 percent repeating,"[28]

27. The full video can be viewed at: http://video.google.com/videoplay?docid=-7714643693602998196
28. The "32.33 percent repeating" chance of survival actually comes from the dialog of the Jenkins video clip. Because the clip was actually staged and meant to be humorous, I can only surmise that

things often do not go quite as planned. Complex in different ways than making games, the humorous WoW guild raid video clip bears many similarities to the world of game developers. Frequently, despite all of the planning and attempts to manage the process of creating games, results in a final melee bear little resemblance to what many hoped the final battle would look like.

2.5.2 The Importance of Process

"Process" is the general term used to refer to the activity of producing a video game. Most games are believed to go through an idealized waterfall process from "pre-production" to "production" to "testing" or "Q/A" finally to "golden master" or when the final version of a game is sent to the publisher. This idealized process of course scratches the surface of how games actually get developed, but the widely held belief that this is how games are developed means that the notion persists.[29] Many game companies have attempted to get better at the "process" by having process managers or people who make it their job to better understand how to more effectively make games.

More recently, many game development studios have begun playing with software development processes established more broadly. "Agile" development or one of its incarnations, "Scrum," have been widely touted as making significant improvements on the game development process. The steadily growing number of sessions at the Game Developers Conference, which feature the words "process" or "management" in their ti-

the ".33 percent repeating" component was meant as a joke to poke fun at WoW players or gamer "nerds" more generally.

29. Game developers certainly "muddle through" the "socially complicated as well as intellectually complex" process of creating their technological systems (Fortun and Bernstein 1998, pp. x-xi). What I think differentiates the game industry is that it as of yet has no systematic system for reporting, publicizing, or thinking more broadly about that process of muddling. While tension remains between talking about and documenting the process of scientific production, the difference in the game industry is that there is no broader discussion, not even an opportunity for tension.

tle will give you an idea of the growing popularity of this new area (GDC 2007a; GDC 2007b). Indian game companies in particular have made extensive attempts to bring proven software development best practices into the world of game development. In many cases the upper management of these companies come from other areas of software development where similar methods have improved the management of software production. In the U.S. and Western Europe on the other hand, there is a widely held belief that "game development is just different," unmanageable, or teams actively combat management techniques that attempt to discipline the methods, by which games are developed. This is of course due in part to many U.S. game developers getting their start while in college, working out of dorm rooms, garages, or basements. Much has been done to innovate in the area of process; even still these practices rarely sufficiently prepare developers for the melee that frequently ensues when these highly coupled, complex systems interact.

2.5.3 Who Broke the Build? Who's Got a DevKit? or Highly Coupled Systems Break

It is important to point out that not only do highly coupled systems break, they break spectacularly, and more frequently when they have no documentation.[30] As can be seen in the sheer number of disciplines, technologies, and practices, which make up the practice of game development work, there are a large number of interconnected and dependent pieces.

---

30. Though perhaps far removed from a chemical plant in Bhopal, the continued systematic failure of breakdown of game development practices seems to me to indicate a problem more systemic than user error (Fortun 2001, pp. 123-131). The "modifications" and ad-hoc modifications of complex technical systems can have unforeseen results, something game developers can certainly understand. Furthermore, as the complexity or coupling of a system increases, the opportunity for "catastrophe" or "system accident" increase rapidly (Perrow 1999, pp. 62-100).

An artist or designer cannot see the results of their work until the underlying code or tech to support those features has been created by an engineer. An artist may be unable to see their work in game without data defined by a designer. A designer may not be able to see the results of their work without the associated art assets of an artist. At the same time numerous complex software systems are mediating the interaction between these individuals. The build system itself is a system that may fail or break, regardless of the health of the underlying game. In other cases a game may function on one system, but not another. The connection to Development Kits or DevKits, explained in more detail in World Four, is an important one. These complex technological systems are supported by complex software systems and custom processes must couple together to ensure the overall health of the build.

While "incidents are overwhelmingly the most common untoward system events," I suspect that given the frenetic pace of game development, lack of broader discussion of best practices, or any practices for that matter, the commonality of full-fledged "accidents" becomes much more common. As other industries privatize and place more emphasis on secrecy rather than open discussion, the implications outside of the game industry are troubling. While a full-fledged "system accident" for a group of game developers results in long hours and a stressful work environment, the implications in other aspects of work worlds, whose developments are not so perfectly contained, becomes particularly troubling (Perrow 1999, pp. 70-71).

This general kind of unpredictability and instability has led many to see the answer as more real time feedback throughout the systems, including the human component. The trend has been to move towards what I have termed an "interactive" model of game development, where changes and modifications to the overall complex system can be viewed in real time and instantly.[31] While in some respects, the goal of instant feedback and response can indeed be a boon for developers, as in the ability for an artist to know precisely how and why things are not working as they had thought they might, the same goal does not necessarily extend itself to the realm of human work or work organization. At the same time, these goals can be overextended, resulting in what I term generically "churn," or the inability for workers to find a reasonable space of time to sit and work on their assignments. Feedback and information for the sake of feedback and information results in situations where the system comes to a standstill.

Creative, collaborative, and interdisciplinary work is difficult already. There is a reason that terms like "faultlines," "sedimentation," and "volatility" are used in these contexts (Traweek 2000; Fortun 2006). The process is fraught with the continual (re)formation of creole languages and the experimental process. Tools break and complex systems fail all around the development process. In an effort to increase efficiency, interactive systems are deployed, but can distract us and become goals in and of themselves. Most importantly, it is this process in conversation with broader system that is so important. Game developers by and large have been lost in their inability to really reflect, document, or talk about those experiences that would historicize their activities. It is in the spirit of this we step into our next set of subroutines, the networks of (in)access of game development. World Three awaits.

31. What game developers need just as much as "interactive" systems is better processes for pursuing their assemblages. Much like the sciences, game development is "such a dense, intricate, and volatile assemblage of practices, metaphors, articulations, and other kludged-together elements of nature, culture, and power, they have to be muddled through." But more importantly, this process must remain, "cautious, nimble, and respectful, since they deal with explosive matter" (Fortun and Bernstein 1998, pp. 147-148).

CHAPTER 3
THE VIDEO GAME INDUSTRIES INTER/INTRANETWORKS

World Three examines the structures that shape the worlds of video game developers. It is rooted in conversations with and the online activities of game developers discussing, "The Industry". The industry is an object of concern for nearly every game developer. It is an object of scrutiny that both constrains and compels them; it is the system that they play within. One afternoon, an artist took some time to speak with me. During the process of the conversation he began doodling and sketching supplements to my queries. When I asked him about his place in the process of creating a game, he drew a picture of the industry, which illustrated numerous aspects of his perspective of the world around him.

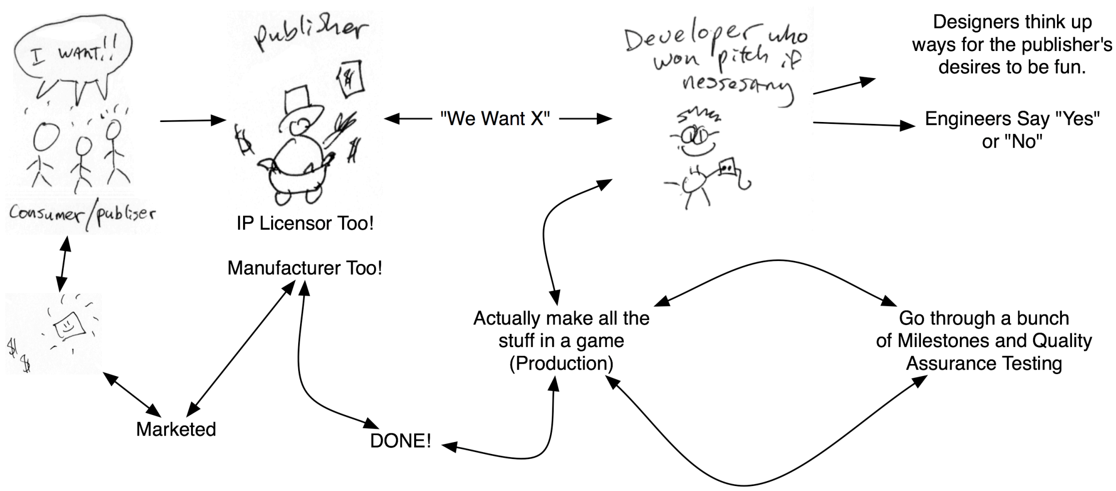


Figure 3.1: An Informant's Rendition of "The Industry" (Informant and O'Donnell 2005c)

What the image nicely demonstrates is how he and many developers feel they fit into the networks of video game industry. Though each developer is quite close to the creation of the product, they end up feeling quite distant from the consumer, the teams of marketing specialists who decide how to sell the game, and the money, which ultimately funds and fuels the video game industry and their jobs. Because of their position in a network of companies, many studio heads said that they consider themselves to be in a "service" industry rather than "product" industry. Relationships within the networks of game production end up mattering as much or more than the individual projects that are developed. This in many respects counters the belief that the product focus of the game industry is one of the features that differentiate it from other New Economy workplaces.

I use the Figure 3.1 as a foundation for the structuring of World Three, which examines the networks of (in)access of the video game industry. World 3-1 begins by examining the ways that developers structure their companies, the kinds of development studios and the ensuing relationships. World 3-2 focuses in on the networks of publishers, manufacturers, and intellectual property (IP) rights holders. There are networks of (in)access and (in)operability between these different networks, and there are consequences. World 3-3 focuses on the perspectives of developers and their experience of these networks. World 3-4 looks at recent consolidation and acquisition trends in the video game industry and the consequences for work practice. The overarching argument for World Three is that, as the industry has "matured," the networks have become less accessible and less interoperable. Consequentially, this trajectory is one that limits developers more than they might like to believe and in ways many are unwilling to criticize.

The structuring effect of the network is particularly interesting. Frequently "network" approaches to understanding work, the economy, production, or society fail to actively engage with the structuring effects, or more generally "power" in a very un-theorized sense. The "flows" of knowledge, which are then networked, are also structured in ways that is not addressed in current research (Castells 1998). "Access" is such a key aspect of the game industry and game development work more generally, yet it is frequently glossed over in research, which attempts to examine the networks of game production. These studies rarely look closely at what is necessary for a developer to gain access to these networks (Johns 2006). The concept of the Inter/Intranetwork is a useful tool for thinking about the structuring effects of networks. The structure that has emerged is "networked" but, more explicitly, networked in a fashion that I have termed "Intranetworked," or closed off. Much like a corporation's private internal network, or "Intranet," it is tightly controlled and connections to the broader world or "Internet(works)" are highly monitored. While it would be folly to assume that "everyone else" is on the Internet in a literal sense (most Internet Service Providers like Comcast, Verizon, or America Online, operate as Intranetworks), figuratively that is how it is imagined.

I use this concept in conversation with prior work in the anthropology and sociology of science that uses "actor-networks" as a means to analytically understand how science and scientific practice unfolds. While actor-networks provide some insight, I have become critically interested in why particular nodes become obligatory passage points, or why entire networks become closed off from other nodes in the network.

> If technoscience may be described as being so powerful and yet so small, so concentrated and so dilute, it means it has the characteristics of a network. The word network indicates that resources are concentrated in a few places - the knots and the nodes - which are connected with one another - the links and the messages - these connections transform the scattered resources into a net that may seem to extend everywhere. (Latour 1987, p. 180)

This game has become highly structured with very little forethought. The networks of the video game industry are structured in a inter/intranetworked fashion. Networks may seem to extend everywhere, but accessible to only specific individuals and organizations. Publishers consolidate their interests by acquiring smaller (moderately) successful game development studios. Console manufacturers (who are also frequently publishers) do this as well. Frequently, just as often, new connections end up closing off networks. This often results in islands disconnected from the mainland. Even independent studios tend to operate only in concert with a small number of other studios if any at all.

The inter/intranetwork stands in stark contrast to how networks are frequently talked about, particularly in the context of the New Economy. Sociological inquiries into New Economy work has drawn heavily upon the network metaphor, emphasizing the limitlessness of them over their structuring effects.

> Networks are open structures, able to expand without limits, integrating new nodes as long as they are able to communicate within the network, namely as long as they share the same communication codes. (Castells 1998, p. 501)

But as already established, the communication codes in the video game industry are largely closed, and must be re-discovered by many aspiring developers looking to enter the video game industry. This lack of openness and collaboration is fostered by the highly restrictive legal agreements and sense of secrecy, which dominates the video game industry. It is this kind of complex corporate, social, technical, and legal network that the game industry so (in)effectively played.

3.1 World 3-1: A Development Studio Eye View of the World

Video game development companies, or "studios" as they are frequently called, are where games are created. It is within these companies that code is written, art is generated, and designs are made. The worlds of game developers are not that much removed from those of other artistic networks of production. "Information circulates through networks: networks between companies, networks within companies, personal networks, and computer networks" (Castells 1998, p. 177). Sociological analysis of artistic networks demonstrate compelling parallels the the networks of the video game industry. What seems crucially important, however, is that artistic networks, despite their scale, rely on other networks of production.

> Some networks are large, complicated, and specifically devoted to the production of works of the kind we are investigating as their main activity. Smaller ones may have only a few of the specialized personnel characteristics of the larger, more elaborate ones. In the limiting case, the world consists only of the person making the work, who relies on materials and other resources provided by others who neither intend to cooperate in the production of that work nor know they are doing so. (Becker 1984, p. 37)

In much the same way, there are several different "kinds" of game development studios. The most basic and most nebulous is the "independent," or self-funded studio. They range in size from several developers, artists, and designers working together to create a video game to large companies without an exclusive relationship with a particular publishing company or console manufacturer. These companies or loose affiliation of individuals create games of their own design. Most independent developers eventually enter into some kind of relationship with a publisher to release their first game on a PC. Then they must begin responding to the desires of the publishing company to ensure the release and distribution of their game. If the game is successful or garners critical acclaim, then the relationship with a publisher continues, and only then does a game development studio gain access to the resources necessary to develop console games. In some cases independent developers instead opt to distribute their games online as a downloadable game for the PC or via Adobe Inc's Flash Web-based technologies. Either way, they take on the task of distributing and supporting their game. However, this is not the most lucrative path in the video game industry, nor is it the most prestigious. Often game developers consider this path "amateurish" or as something that one does before you have actually "made it" in the industry.

Once an independent developer has proven their ability to successfully create and release a game, they frequently become what are called "third party" development companies. In some cases companies will enter this category as a means to make money during the development of their first "independent" title. Other companies enter this phase immediately, if they are created by developers with connections to publishing companies or console makers who have diverged from other game companies. A third party developer is similar to an outsourcer for a publisher and they are instructed to make a particular game. In some cases a previous game title can be used as a reference. This is most common in the case of established franchises. For example, a company may be contracted to develop "Shrek 3" or "Batman Forever." Developers will frequently play or examine the titles that came before it. In most cases, the publisher already owns, or has acquired the rights to these "IPs." The developer is then authorized to make a game to the publishing company's specifications. There is often not a clear delineation between a third party developer and an independent developer. Most third party developers have internal, independent projects, which are funded by the revenue based on "non independent" work for a publisher.

While it may sound as if third party developers are simply "outsourcing" houses for publishing companies, this would betray the dynamic and complex relationship between developers and their publishers. Developers are not given a precise description of the game, which is often desired. Instead developers must frequently base new designs on older ones, which are then vetted by the publisher. The oversight of publishing companies over third parties is varied and complex. In some cases very little direction will be given except at milestones or intermediate steps along the development process. In other situations, publishers may place a producer in the offices of the third party developer to provide constant feedback. Frequently publishers have a conceptual foundation that third parties can begin working from, but the actual game which winds up on the shelf is a product of both companies working in concert. This means that publishing companies in many cases do little to zero actual game development. The design "document" which becomes a part of the "contract" is actually developed by the game studio. The effort of creating games is reserved for those working for game studios.

The distinction between outsourcing and third party development is important to make, considering that there are true outsourcing companies in the video game industry. These may also be individual freelancers who create art assets, localize text, port code to new platforms, or test games for defects. These relationships are governed with relatively precise specifications and contractual obligations on both ends. Many game companies in India have chosen to use outsourcing as a means to fund internal development projects. While most game companies in the U.S. begin as a mixture of independent and third party studios, most Indian companies begin as a mixture of independent and outsourcing studios. In part this has been due to the readily available manpower with experience using and creating media arts. Very little code outsourcing occurs, as it is frequently highly protected by game studios. Contractual obligations in many cases governs lines of credit, which is one of the most important resources for aspiring game companies, as it begins to establish their credibility as a game development studio. However, given the climate surrounding outsourcing in the United States, companies will often pay more to prevent outsourcing studios from speaking about or placing their logo in games bound for the U.S., for fear of consumer retribution or bad press. This restricts employees at these studios from being able to claim having worked on a "title," which as previously discussed limits developers from gaining access to social networks within the game industry.

There is a typology amongst those companies doing offshore outsourcing work within the video game industry. This differentiation is primarily based on the markets that a company is most interested in venturing into. For some the drive is developing games for an internal market, closely targeted to users in their home countries. For others it is the eventual creation of games targeted at a global market, much like their counterparts in already established markets. Other companies seem to be interested primarily in acquisition by large multinational publishing companies, which is not to say that they do not have aspirations of those other companies, simply that they have set their current goals in a different direction. More recently, publishing companies have begun acquiring outsourcing studios in India, China, Vietnam, and in other countries, hoping to use these acquired studios as sites for Intracorporate off-shoring. In these cases, publishing companies have established production pipelines for particular games, which only need content created for them. These studios are then used to create content in a highly controlled, but cheaper environment.

Closely related to the outsourcing companies are "middle-ware" companies, which provide software that enables more rapid development of game systems. As the complexity of games has risen, this new class of company in the video game industry has exploded, and in many cases these companies have sprung from countries that otherwise do not have a large established video game industry. Rather than outsourcing, these companies have developed extensive libraries of source code, software tools, and process management systems, which they can sell to developers in Japan, the U.S., and Western Europe.

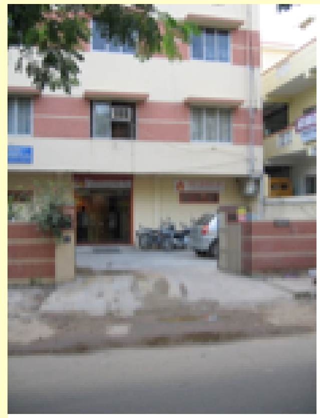
The final kind of company in this typology is the "in house" development studio, which is wholly owned by either a publishing company or console manufacturer (who are often publishers as well). These studios frequently act like independent developers, third party developers, or a mixture of the two, not unlike studios not owned by publishing companies. While there is some element of collaboration between studios under the same publisher, it is often minimal. Different studios have different practices, systems, technologies, processes, and "cultures." In rare cases collaboration occurs, though in most cases the extent of their interaction is through the interface of studio heads and employees of the parent publishing company.

For many developers in India, the industry is very distant. In some locations developers have mobilize through informal meetings, or through local International Game Developers Association (IGDA) chapters. In other areas companies have banded together with larger organizations, like India's National Association of Software and Service Companies (NASSCOM) to encourage new growth in game development. Of course, for some, like those working for Microsoft's Casual Game Group in Hyderabad, India, whose games are being placed onto the online distribution network of Microsoft's Xbox 360, the feeling of distance is minimal. The networks have already been established. For companies that have yet to gain access, the distance is palpable.

Dhruva - Bangalore, India

Dhruva Interactive was featured in the book by Thomas Friedman, The World is Flat, and seemed an excellent example of a successful startup game company in India. Dhruva has resisted the temptation to be acquired by a U.S. based publishing company. It has widely been acknowledged as the first "real" game company in India. Most of the company is focused on the production of artwork for video games published in the U.S., doing "outsourcing" service work for the latest generation of console video games. Dhruva also has a significant number of employees working on Mobile (cell phone based) Games for both the Indian and Global market. They are currently expanding their engineering capabilities on the personal computer, hoping to learn more about the Indian game market and thereby producing games specifically for India.

Table 3.1: Featuring Dhruva Interactive - Bangalore



RedOctane - Chennai, India

RedOctane is the India based branch of a U.S. video game publishing company that was recently acquired by Activision, another video game publishing company. RedOctane is fairly uniquely positioned because of this relationship. RedOctane is the only studio in India currently authorized to work on creating console video games, and in particular are working towards creating games for Nintendo's DS (dual-screen) system. This also makes RedOctane unique in their ability to work on full-fledged games from start to finish, rather than other studios doing "service" work, which typically allows them to only see one piece of the game development lifecycle. RedOctane is also using this relationship with Activision to send a handful of employees to the U.S. for several months to learn more about the game development process.

Table 3.2: Featuring Red Octane - Chennai

Microsoft - Hyderabad, India

Microsoft's Hyderabad office, often called Cyberadad or Cyber City. While this campus serves numerous purposes for Microsoft, it also houses a group of developers working as part of Microsoft's Casual Games Group. Casual Games are primarily played online via web browsers. They are frequently based on board games puzzle games, and card games. Casual games are thought to have a broader appeal than standard video games. While the name "casual" games might indicate that players spend less time gaming than other genres, though not yet supported by evidence from those people who study gaming habits, most casual gamers clock just as many hours playing as the more standard "hard core" gamer. More recently these games are making their way onto the home console game systems via online distribution, such as the Xbox 360's Live Arcade functionality.

Table 3.3: Featuring Microsoft Mobile - Hyderabad

More so than any other question I received while in India was, "How do we differ from what developers in the U.S. do?" This relatively simple question frequently lead to conversations about a disconnect between what Indian developers are allowed to contribute to game development projects, and those tasks, which are necessary to produce a video game from start to finish. Some companies do create games from start to finish, though at a different scale. They create games for mobile (cell phone based) game platforms. Although at a different scale, these networks are also difficult to access. These studios tend to fund these development efforts by also offering art asset production outsourcing services. Because of this, the companies become more specialized in one aspect of game development and not others. This disconnect cuts deepest at these companies, who become solely identified as locations for outsourcing by U.S. companies. As the majority of these studios resources become focused on that singular aspect of the development process, it becomes disconnected game development more broadly. Science and Technology Studies scholars have demonstrated how Indian scientists and engineers have remained in conversation with American institutions through electronic means.

> [India's] scientists and engineers are highly connected with their peers in American institutions. This is partly because scientists and engineers in India overwhelmingly enjoy access to the World Wide Web, but institutional linkages are even more important. With the IT revolution, Indian S&E educational institutions have been increasingly connected with the United states, as well as with the rest of the world, essentially comprising one global system. (Varma 2006, pp. 40-41)

However, this has largely not been the case for game developers. Scientists and engineers, unlike game developers, have avenues or venues in which collective knowledge is shared more broadly. The norms of secrecy strike again, preventing the formation of a community of practice more broadly.

For those developers interested in developing for any platform outside of the Web, personal computer, or mobile (and even this platform is notoriously difficult to work with due to the domineering attitude of carrier companies), the opportunities are extremely limited. It is frequently only when the concerns of a U.S. company are involved that some sort of agreement can be reached, which provides the requisite hardware, software, and documentation to be given to Indian developers. However, this information is often provided without connection to the tacit knowledge of what it takes to create a game for these platforms. The immense body of knowledge, which has become codified only in the practices and conversations of developers, is not transferred along with the capabilities to produce games for these systems. When Indian developers go through the same learning process, which other established developers went through only several years earlier, they are confronted with questions that amount to, "Don't you know anything?"

This network disconnect is not simply limited to those in distant countries. Numerous independent developers, and even those simply struggling to bring developers together outside of the United States' east and west coast face similar barriers. If you are not part of the game development world, the only way you can get in is to create a game, but it can be difficult to develop a game without connection to those existing networks. Instead, you must fumble your way until you have learned enough on your own to prove your worth, at which time interested parties would rather move to where the networks have already been established. As already mentioned, there are secret social networks as well. "An industry's cocktail parties, seminars, and informal gatherings form its social backbone and are especially important to innovative industries that rely on the rapid dissemination of information" (Neff 2005, p. 135). In the case of the game industry it is less about the dissemination of rapidly changing information, which I have already addressed that this is rarely addressed. Instead it is about social networks. These closed intranetworked social structures "increase the experience of labor market inequality" and "workers unable to access or maintain these networks may be at a disadvantage" (Neff 2005, p. 138).

Our social inter/intranetworks are not without differential power relations. In the move from despotic power relationships to hegemonic relations, a new kind of measure seems to be ever more predominant. Anthropological analysis of networks amongst high-energy physicists demonstrate similar boundary marking and maintenance by practitioners.

> Networks of exchange link otherwise autonomous units at every level of social organization. The primary commodities exchanged are students, postdoctoral research associates, and 'gossip' (oral information about detectors, proposals, data, organization of groups and labs, and the location and professional genealogies of individuals). The boundaries of the networks as a whole are closed, marking off the outsiders. ... The

boundaries of the community as a whole are negotiated with great circumspection. (Traweek 1988, p. 123)

The use of reputation networks as a mechanism for structuring numerous resources within a section of networks is consequential for any social network. The same is true for labor and knowledge production networks. If the practices of a given subsection of the inter/intranetwork are not meeting the expectations of other components of the network, their reputation, and subsequently their income will begin to fall. While these networks are social, they are also technological, corporate and intricately connected to complex legal and legislative systems. In effect, the network structure has systematically blocked out those mechanisms by which access for developers both foreign and domestic can be granted. More and more work is only being done from within the networks. Those hoping to "break-into" the network must battle numerous difficulties in what is largely being touted as a "flat" economic system. However, this is simply not the case.

3.2 World 3-2: The Manufacturing and Publishing Game

As far as developers are concerned, publishers and manufacturers control the video game industry. Development studios structure themselves around their relationships with these companies. As already noted, developers can create games for the PC and the Web without publishers and console manufacturers. Why is it then that so much emphasis is placed on the ability to work with publishing companies on console video game systems, beside the "prestige" of being allowed to do so? What is it about consoles that separate them from other gaming systems like the common PC? Console gaming systems did not mark the beginning of computer-based gaming. That point has been made by numerous studies of the "birth" of the video game (Kent 2001; Kline et al. 2005; Malliet and

Zimmerman 2005). But consoles did several things differently from the (personal) computer, with the first being the way that people interacted with the system. Typically this was through a simplified manner of data entry, perhaps by a rotating knob and button, a joystick with one or more buttons, or with something like the controller from World 1-1 (Figure 1.2) with its directional pad, (visually similar to an equilateral cross) and one or more buttons. Consoles were also connected to televisions rather than to separate and costly monitors or "dumb" terminals connected to mainframe computers. The console was simply another component of the growing "entertainment center" in the home.

More importantly, consoles are significantly less expensive than the computer configurations necessary to play games. Several hundred dollars for a console rather than several thousand for a computer was far easier for average purchaser to justify. As time went on and the price of personal computer began to fall, more people acquired them. Many believed that console gaming was something that would vanish, a remnant of history (Carless 2007; Edery 2007; Snow 2007). On the contrary, console gaming continues to be most lucrative sector of the video game industry. Simply by looking at the differential scores between PC and console game sales, you get an idea of the magnitude of difference between the sectors. This would explain publishers' interest in the console systems.

Year	**Console Game Sales**	**PC Game Sales**
2006:	$11.2 Billion	$3.9 Billion
2007:	$12.2 Billion	$3.7 Billion
2008:		

Table 3.4: Screen Digest Sales Data 2006-2008 (Weber 2007)

This differential is only partially explained by the simplicity and lower cost of gaming consoles. The difference is both in the capabilities and limitations of a gaming system. Consoles frequently have certain technological capabilities beyond those found in affordable PCs. In the beginning it was graphics processing power that marked the difference. In many cases the graphics processing power of console gaming systems when initially released have been beyond those of the average computer. While throughout the lifetime of a console similar technologies frequently become incorporated into general computer systems; consoles begin their lives more capable than average computer systems. Consoles were also frequently priced at rates that bring a loss to the console manufacturer. The logic of this practice I will return to later, but until recently this was standard operation for console makers. The system itself was historically a loss leader. The technologies were so much at the cutting edge they were providing it to buyers at prices below cost. This also brought about the downfall of more than one console manufacturer.

Limitations are also important in producing games for consoles. Because they are not required to do "typical" personal computing tasks they can be built with very precise specifications and limitations to their possible uses. Unlike a standard personal computer, there are no differences to be taken into account. No differences in amount of memory. No differences in graphics systems. No differences in operating systems. No differences in processing power. There are no warnings that you do not have the proper driver installed. Each console is created with specific specifications, and when creating games for it, game developers know these capabilities. The case is very different on the PC. Differences in software/hardware configuration, the possibility of other processes running, which might interfere with the functionality of a running game, or simply the unknown availability of options frequently provides great difficulty for companies making games for the PC. Some argue, and I tend to agree that it is these "non-standardized advances in home computers," that have encouraged the growth of the video game industry and massive improvements in the graphics processing capacity of video games. These developments also create one of the biggest worries and limitations for developers hoping to create games for personal computers (Williams 2002).

In particular, 3DFx, ATI, and NVIDIA managed to significantly alter the creation of video games and the expectations of users beginning in 1996. 3DFx was the first company to release their consumer targeted graphics card, called the "Voodoo" and accompanying software development libraries called "Glide." These tools made it significantly easier for game developers to render complex and rich 3D scenes. While NVIDEA later acquired 3DFx, they had a significant impact on the video game industry. From the time of the Nintendo64 and Playstation forward, all game consoles created had some involvement from one of two companies, ATI and NVIDIA. These companies are the two major competitors for consumer and gaming 3D accelerator cards. These graphics processing units (GPU) manufacturers have been in a constant race for pushing the limits of how many, how quickly, and how richly detailed polygons can be drawn to the screen of a PC monitor or television screen.

As can be seen with the latest consoles released by Nintendo, Sony, and Microsoft,[32] each features a GPU developed in cooperation with ATI or NVIDIA. These de-

32. The appendix contains several tables for each of the major console makers since the release of the NES in 1985. In many respects you can see at a miniature level the "MHz arms race" which was soon to grip the computer and software industry more broadly. However, rather than talking about the speed of the processors, the short hand typically was in the number of "bits" a system was; from the "8-bit" era of the NES and Sega Master System through the "16-bit" times of the Super NES and Sega Genesis. Of course in many respects this arms race can even be seen in the names of systems, the 32X by Sega and Nintendo's N64. For the most part this was the expression of a

vices have begun to incorporate more and more processing power of their own, and the development of "shaders," which actually allow developers to create miniature programs that describe the way graphics primitives should be drawn to the screen. The concept of a shader and the ability to perform them in real-time revolutionized the graphics expectations of users. This new level of visual fidelity is a basic component of modern game development, and is now expected by players of video games. These new techniques frequently go by short hand names like, "bump mapping," "normal mapping," or "environment mapping," and each is enabled by these new specialized processors. This "graphics and glitz," or focus primarily on visual fidelity over all other matters, has only begun to change in recent generations of video games, but its consequences are still being felt. This is examined more closely in World 3-3.

The focus on console games by developers and the perception that it is the only market worth developing for, or is the most prestigious market to develop for, has meant that it remains a focus for most developers. However, unlike personal computers, you cannot simply begin developing games for a console without first establishing a relationship with that console manufacturer. Because of this, these companies continue to have the single largest impact on the video game industry. The rules of this game are experienced in three primary ways, technologically, legally, and monetarily. While most developers avoid the pitfalls of legal prosecution, some do come under legal scrutiny if they attempt to circumvent the technological limitations of their position. More than any other aspect of the game industry, it is this particular area that developers could conceivably have the most impact in their desires to both shape and care for the industry. At the same

broader phenomenon of growing computer power available at significantly lower costs. Each company struggled to out-pace the other while still providing enough margin on the retail sales to not cause bankruptcy for themselves.

time however, it is the part of the game industry, which disappears from their perception. Once a company has gained access to DevKits and SDKs, they recede into the background, despite the fact that they were once one of the major gatekeepers of access to industry networks. Far too quickly developers allow themselves to forget just how difficult it can be to work amongst the structures of the industry.

The situation is even more complex though, because frequently before a request for a DevKit is granted, developers must demonstrate a proof of concept or playable demo of a game. This is frequently done using freely available resources and tools, targeting the personal computer or the Web. All of this work must typically be re-done once a DevKit is acquired, as those systems are completely different from those freely available. This is a particularly sticky aspect when talking about a business rather than a hobby or something set up in your garage in the hopes of making it big. Many informants noted a distinct difference between between game development as work and game development as play, especially when livelihoods are on the line.

> Money. I mean, in one you're doing it not to support yourself, and the other you are. And like one you're making a game because you love to make games, and the other is, you're making a game for your living, and not only your living, but everybody else's living too. And in order to do that, you need someone to give you money. (Informant and O'Donnell 2007)

Even if a game development team is successful creating a game for these systems, for distribution on the proprietary Nintendo, Microsoft, or Sony networks will require approval of the self-governing ratings system of the game industry, which will cost another $2,000-$3,000. If you receive a rating from the Entertainment Software Rating Board (ESRB) that limits the audience of your game too significantly and requires massive rework, then so be it. This is the logic that leads many developers looking to publishers to fund the massive effort required to make a game, though that investment comes at the cost of relinquishing some control of your project, and frequently the rights to the IP of a project to the publisher. This way if the risk does pay of for the publisher, they can continue to capitalize on that risk for years to come.

Despite all of this, it remains a largely invisible aspect of the game industry. While developers will frequently lament the way the game is being played, they continually neglect to pay attention to the rules, by which it is being played, something you would not expect from people (self) trained to pay attention to these structures. While Nintendo first emerged as a force to be reckoned with during the time of Atari, the NES release was particularly important because it brought about massive change to the way video games were developed, sold, marketed, and in the very technological core of game consoles. For the most part, these changes were non-obvious to the user. The only visible difference was the emergence of the Nintendo Seal of Quality on the boxes of games. It was placed on "official" game titles released for the NES. These were the games licensed by Nintendo.

Figure 3.2: The Nintendo Seal of Quality for NES Games in North America

Nintendo's logic was clear. They believed that the low quality of games released for the Atari were partially to blame for the "crash" that came shortly after the release of several games that were massive economic failures. By offering their "guarantee" of quality, gamers should feel safe investing their money into the games. If it were simply a quality control issue, you would have likely had two groups of games for the NES, those "guaranteed" and those that were not. But that is not what happened. "Unlicensed" games were few and far between. The only games available for the most part bore the Nintendo Seal of Quality. It is what lies behind the Seal of Quality, which endures even today on all games released for Nintendo's current generation of consoles, which makes a world of difference. The very term "licensing" suddenly entered the minds and vocabulary of game developers. It is in this web of connections, this network, which we find the real power of the seal. But why focus on manufacturers like Nintendo? It is because they structure the network more so than other actors. This is where concepts like those proposed by sociologists examining the "Network Society" or "Networked/New Economy," rightly index the importance of switches or points of control within the network. Unfortunately these important aspects are frequently not examined closely enough, despite their ability to dramatically shape resulting networks.

> Switches connecting the networks are the privileged instruments of power. Thus, the switchers are the power-holders. Since networks are multiple, the inter-operating codes and switches between networks become the fundamental sources in shaping, guiding, and misguiding societies. (Castells 1998, p. 502)

But it is not about connecting the seal up with these other components, it is about the ways that these networks structure the very work of producing games, which is so dramatic.

3.2.1 License Agreements

By reading into several court cases, we can learn more about Nintendo's licensing practices, which were otherwise typically invisible. Up until the introduction of the NES, companies were largely able to create games without licensing. The Seal of Quality changed all that. But what kind of deal was necessary to play this game? It is at this point that you begin to understand the frustration companies were having playing this new game. So much trouble in fact, that Atari was simultaneously seeking an antitrust suit against Nintendo (Atari et al. 1992).

> "In December 1987, Atari became a Nintendo licensee. Atari paid Nintendo to gain access to the NES for its video games. The license terms, however, strictly controlled Atari's access to Nintendo's technology, including the 10NES program. Under the license, Nintendo would take Atari's games, place them in cartridges containing the 10NES program, and resell them to Atari. Atari could then market the games to NES owners. Nintendo limited all licensees, including Atari, to five new

> NES games per year. The Nintendo license also prohibited Atari from licensing NES games to other home video game systems for two years from Atari's first sale of the game." (Atari et al. 1992)

Five games per year, and all costs must be paid for at manufacture. This is the game that so many people were having no fun playing. A company's entire earnings were limited to five games per year, and those companies bore all of the risk associated with the costs of production. Nintendo was the only company unhindered by these limitations on production. If you attempted to change the rules, you were met with not only the ire of Nintendo, but also the force of the state apparatus. The legal ramifications of copyright and patent systems were leveraged by Nintendo to alter the entire playing field of the video game industry. But what did it take to play on this field? Perhaps it wasn't all that bad? Of course Nintendo needed to cover their expenses in the manufacturing of games for their console.

We do know that things have changed, and that Nintendo no longer places such severe restrictions on the number of games a publisher can create in a year. We can quite simply demonstrate that by looking at the number of games released for consoles each year by different publishers. But, if we examine the top publishers of video games, an interesting trend emerges.

3.2.2 Publishers

Top Video Game Publishers		
2004	**2005**	**2006**
Electronic Arts	Electronic Arts	Electronic Arts
Microsoft	Activision	Nintendo
Sony Computer Entertainment	Microsoft	Activision
THQ	Nintendo	Sony Computer Entertainment
Ubisoft	Sony Computer Entertainment	Take Two

SCi/Eidos	Ubisoft	Microsoft
Activision	Konami	THQ
Take Two	THQ	Ubisoft
Atari	Sega Sammy Holdings	Konami
Nintendo	Take Two	Sega Sammy Holdings

Table 3.5: Top Ten Video Game Publishers in 2004, 2005, and 2006 (Donovan 2006)

No matter the year, every single console manufacturer is in the top ten. The ability to control what makes it into the content stream obviously has effects on who is making money. A particularly interesting outlier is Electronic Arts, who year after year manages to displace even console manufacturers. The small amount of motion you see in charts like this is that only very large and very established publishing companies are managing to get their games into the console stream. The simplest answer is that this is an expensive game to play. Because the entire manufacturing run must be paid for in advance, and all marketing for a game must be covered by the publisher, creating games for consoles, while lucrative, is also extremely risky and requires a high initial investment.

Examples of the cost of manufacturing console games are hard to come by, but thankfully not impossible. The source of Figure 3.3 is a 2001 SEC filing; one of the accompanying examples of material contracts demonstrates some of the different deals available to a company interested in manufacturing a Nintendo 64 console game. Companies are forced to up front decide on the technological limits of what they can place onto a cartridge. Do we risk less and limit ourselves to a smaller game? How much data do we need to save a game's current state?

<PAGE> 14

SCHEDULE 1

NINTENDO OF AMERICA INC.

PRICE SHEET

N64 LICENSED GAME PAKS

<TABLE>
<CAPTION>

Memory Capacity	NOA Price
<S>	<C>
32 Megabit	$ [*]
32 Megabit + 4K bit E. ROM	$ [*]
32 Megabit + 16K bit E. ROM	$ [*]
32 Megabit + 256K SRAM + Battery	$ [*]
64 Megabit	$ [*]
64 Megabit + 4K bit E. ROM	$ [*]
64 Megabit + 16K bit E. ROM	$ [*]
64 Megabit + 256K SRAM + Battery	$ [*]
96 Megabit	$ [*]
96 Megabit + 4K bit E. ROM	$ [*]
96 Megabit + 16K bit E. ROM	$ [*]
96 Megabit + 256K SRAM + Battery	$ [*]
128 Megabit	$ [*]
128 Megabit + 4K bit E. ROM	$ [*]
128 Megabit + 16K bit E. ROM	$ [*]
128 Megabit + 256K SRAM + Battery	$ [*]
256 Megabit	$ [*]

</TABLE>

PAGE 13

* Confidential Portions Omitted and Filed Separately with the Commission.

<PAGE> 15

256 Megabit + 4K bit E. ROM	$[*]
256 Megabit + 16K bit E. ROM	$[*]
256 Megabit + 256K SRAM + Battery	$[*]

Price includes an instruction manual up to 40 pages. There will be an extra charge for manuals larger than 40 pages (including the front and back cover).

EXTRA PACKAGING (Must be ordered with product on a separate PO)

Game Pak Box	$[*]	
Instruction Manual	$[*]	(under 40 pages)
Instruction Manual	$[*]	(over 40 pages)
Game Pak Label	$[*]	
Game Pak Poster	$[*]	
Warranty Card	$[*]	
Inner Carton	$[*]	
Master Carton	$[*]	

ALL PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE

Revised 7/15/98

Figure 3.3: N64 Manufacturing Price Sheet (Bam Entertainment 2001)

This becomes crucially important when attempting to understand the relatively conservative behavior of publishers and the kinds of games that they are willing to fund. Publishers bear the majority of the risk associated with physically creating a game. This does not mean that publishers bear all of the monetary risk associated with game development, though many developers I spoke with believe this to be true. Frequently publishers only bear the full risk of a game development project if it is one that they have entirely sponsored the development of. In many cases these games already have proven "franchises, " "brands," or more generically, "IP" or "Intellectual Property," though a fairly restricted sense of that term. Some have called this a move towards a "hit driven" industry, much like the movie industry, where for the major publishers to be successful every title must not simply be profitable, but massively so to recuperate the development costs.

This is compounded in the current industry situation, where the massive growth of available storage space on game media has caused many companies to place an emphasis on rapid expansion of game art assets. Many of my informants have pointed to modern games being "asset limited" rather than "engineering limited." Simply stated, the greatest percentage of a games cost has become creating content for the game, not the underlying code, which would put a game into action. The release of "next generation" or "next gen" console systems has led many developers to focus on the creation of highly detailed art assets for games that frequently require more production time to create content. As can be seen in Figure 3.4, in the early days of video games art assets may have only amounted to roughly half of the size of game, with executed machine code taking the rest of the space; however, this is no longer the case. This image nicely illustrates the complex "mash" of engineering, art, and design, which make up video games.

This entire picture is further complicated by the "unpredictability" of video game development, which is frequently touted as being more difficult to mange than traditional software development, though some argue otherwise. Regardless, simply looking at the archives of Game Developer Magazine's Postmortems, you begin to see a pattern of unknowns coming back to bite game developers late in the production of their games. Frequently this results in rework for every aspect of the development team, where changes ripple across engineering, art, and design. This of course has serious repercussions when most publishing companies want their premier titles on retailers' shelves during rush buying seasons; in the U.S. this is the Christmas sales season. In other cases publishing companies have partnered with the movie industry using their established franchises to encourage sales of video game titles. In each case this leads to a rigidity of release dates, and missing these dates can be disastrous for both publisher and developer.

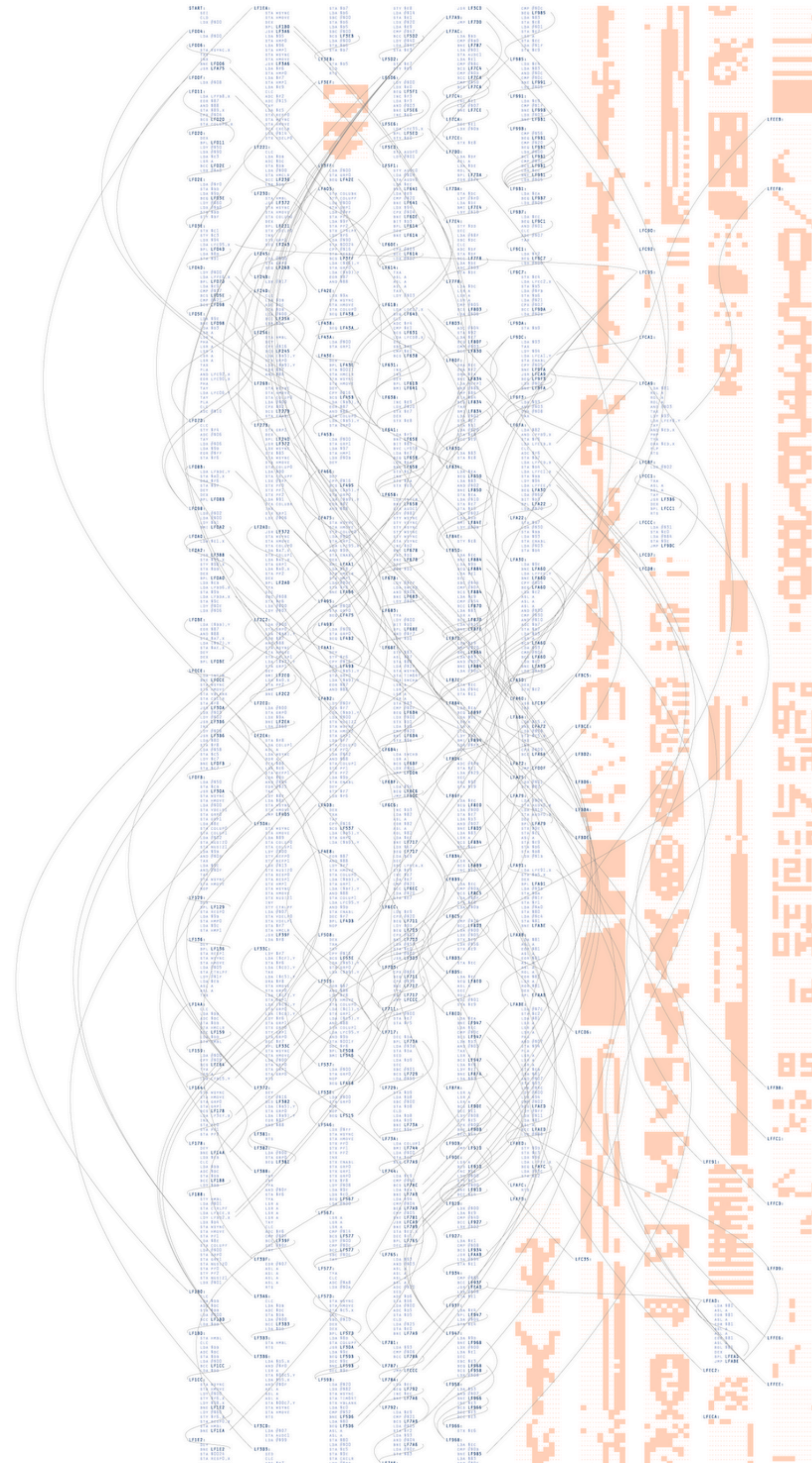


Figure 3.4: An Image Generated from the Activision Game "Pitfall" (Atari 2600)

To reduce these risks, publishing companies have begun offloading the risk of developing new IP or franchises to independent developers. While this is not possible for a publishing company hoping to make a game in concert with a movie studio, it is frequently the case for new games to come from otherwise unknown studios. Often the publisher becomes involved in the development of these games only after an independent developer has already developed large portions of a game concept. Once this is complete, publishers will frequently "milk" these new franchises with or without the original development team, depending upon the contract agreements between publisher and developer. In many cases if a developer retained the rights to their new "IP," the publisher instead purchases them so they can "milk" the new franchise regardless.

This same mentality leads publishers to have a certain conception of the market and consumers. In particular, because of the emphasis on "Hits," most games attempt to capture the core or "hardcore" gamer market. Franchises like "Barbie," "Bratz," or "Batman" are viewed as less risky than titles without an established market. "Madden NFL" or other sports franchises oscillate between being seen as a kind of bread and butter and being derided by game developers. While it is the job of publishers to identify and market games to consumers, many have become even more conservative in their approaches, stonewalling development efforts that do not have an obvious brand or market associated with them. Despite this, many publishers, once a game has proven its appeal as an online "Flash" game or downloadable game, only then will they "risk" further development and distribute the game via more traditional channels on more restricted hardware like console systems.

3.3 World 3-3: What's that Box Doing Over There? Structuring the Industry

More so than any other relationship, developers perceive their relationship with publishers as the one that is the most problematic. Even developers working for studios wholly owned by publishing companies view these as troubled marriages. In part it is because the publisher is their interface with other networks in the game industry. It is also because the publishing companies do exercise their position of power within industry networks frequently at the expense of developer's time and effort. The relationship between publishing companies and developers frequently took the foreground in conversations about developer frustrations.

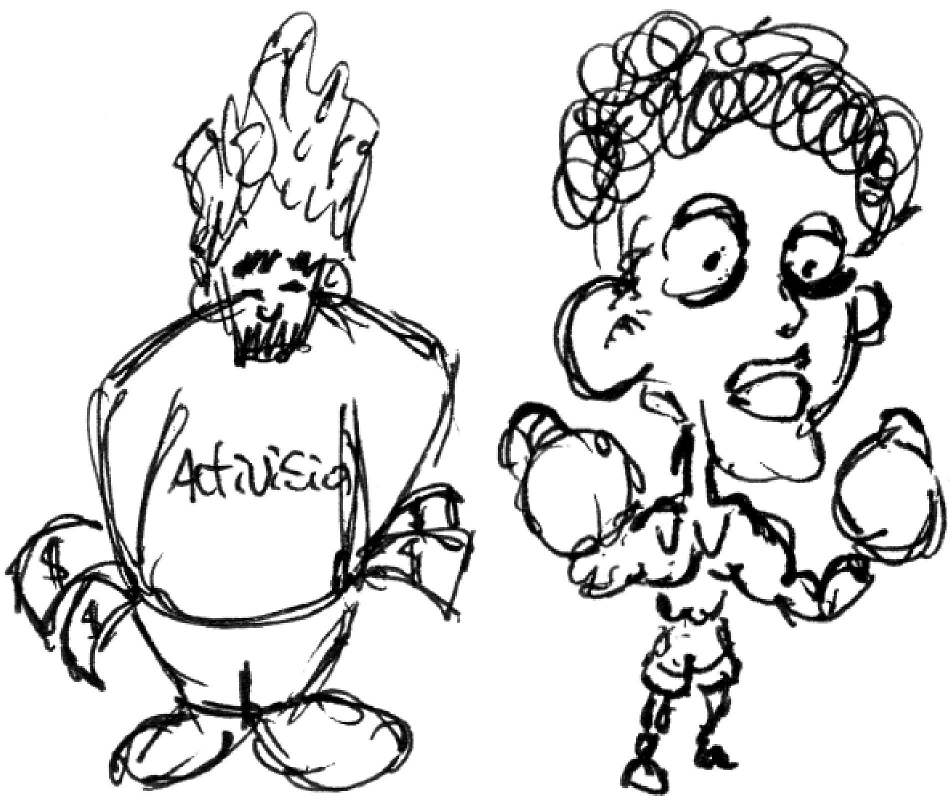


Figure 3.5: An Artist's Interpretation of the Publisher/Developer Relationship

> Like with the whole, developer, publisher, manufacturer relationship. The whole way that that works. Like the way that certain components of that make decisions that are completely silly, that affect the rest of everything

> else, and how there isn't really any like, like they mentioned how there isn't any standards, there's not much sharing between companies, and I understand why that happens. (Informant and O'Donnell 2007)
>
> In order to stay ahead, the people making decisions need to push their employees. Because it is much easier to say, "you work… you employee work a lot harder, and do a lot more," than to spend money in some area, where you might have to hire another person or whatever.
> ...
> I've been at places where working tons of overtime was purely to make up for management miscalculations, or bad scheduling, or um demos for higher ups that just got sprung on people that you know, our bosses could have said no to, but they didn't because it's not them working really hard. (Informant and O'Donnell 2004)

In March 2005, at the Game Developers Conference (GDC) in San Francisco, California, hundreds of game developers crowded into session sponsored by the IGDA titled, "Burning Down the House: Game Developer's Rant." This is a perennial event at the GDC, it is an (in)famous gathering of developers who, "cut the shit and speak truth to power." This year was different. Something interesting was about to happen. It certainly takes something special to shake game developers such that a rant later becomes known by its radical similarity to events leading up to the American Revolution. When Greg Costikyan[33] took the stage, few knew what was about to transpire, but few have forgotten, it has since been called, "The Rant Heard 'Round the World."

> "As recently as 1982, the average budget for a PC game was \$200,000. 1992. Today a typical budget for an A level title is \$5 million, and with

33. Greg Costikyan is an interesting figure in the video game industry, who began as a board game developer and has worked on numerous video game projects. He has also been a caustic and amusing commentator on the video game industry. His website can be found at: http://www.costik.com.

the next generation it'll be more like $20 million. As the costs ratchet up, publishers become increasingly conservative, and decreasingly willing to take a chance on anything other than the tired and true. So we get Driver 69, Grand Theft Auto: San Infinitum. And license drivel after license drivel. Today you cannot get an innovative title published unless your last name is Wright or Miyamoto.

"How many of you were at the Microsoft keynote? The HD era, bigger, louder, more photo-realistic 3-D, teams of hundreds, and big bucks to be made. Not by you and me of course. Not by the developers--developers never see a dime beyond dev funding--by the publishers (and Microsoft, presumably). Those budgets--those teams ensure the death of innovation. This is not why I got into games. Was your allegiance bought at the price of a television?

"Then there's the Nintendo keynote. Nintendo is the company that brought us to this precipice. Nintendo established the business model under which we are crucified today. Nintendo said, 'pay us a royalty not on sales, but on manufacturing.' Nintendo said, 'we will decide what games we'll allow you to publish,' ostensibly to prevent another crash like that of 1983, but in reality to quash any innovation but their own. Iwata-san said he has the heart of the gamer, and my question is what poor bastard's chest did he carve it from?

"My friends, we are fucked! We are well and truly fucked. The bar in terms of graphics and glitz has been raised and raised and raised, until no one can any longer afford to risk anything at all. The sheer labor involved in creating a game has increased exponentially until our only choice is permanent crunch and mandatory 80-hour weeks, at least until all our jobs are outsourced to Asia." (Davis 2006)

Obviously developers seem to not be having the best time, and cultural analysts typically have only the insight of "ea_spouse" as their guide to what work is really like in the video game industry. At the time the blog was published I was sitting with a group of developers working on a video game based on an upcoming movie title for an unreleased handheld console. They too were in "crunch" mode, working to beat timelines, which had arbitrarily been set to meet the demands of movie executives, game publishers, and console manufacturers. The game was later canceled, but those hours late at work fighting against pre-release hardware with pre-release software development kits (SDKs), a new engine, an in-development build system, and no proven "pipeline" for art assets or design data were not forgotten. That is not to say that ea_spouse was wrong, but rather the situation is even more difficult and complex than analysts had previously envisioned. Developers have not just traded their allegiance for televisions, but for a slew of new and interesting technologies and access to private networks.

3.3.1 Productive Myths? Perpetuating the Startup Cycle

The primary consequence of this structure is what has been called the "perpetual startup cycle" of the video game industry.[34] A situation where most of the risk associated with expanding markets and developing new IP is borne by small startup companies, those frequently with the most to lose. Large companies with enough capital to afford risks frequently do not take them, in favor of ensuring good quarterly reports. When a startup is able to prove itself capable of producing value, it is acquired and then strapped down with milking the value out of those resources. Those that tire of this leave and frequently start new companies, taking risks, and again pushing the industry in new directions. Again the risk and cost associated with risk is borne by those least able to. Startup companies frequently have demanding and tiring work environments for anyone with a significant other at home or significant social life outside of the office. These conversations

34. This is in direct contradiction of what many refer to as a "mature" industry. While I would agree with the "structured" assessment, maturity or stability has not been reached. What I think can be mistaken for "maturity" is rather what the same author identifies as the "entry barrier created by an existing dominant network" (Williams 2002, p. 51).

tend to focus on "sustainability" or the ability for the industry to exit its state of perpetual startup.

> Well, I mean, I have...a lot of, I mean, like kind of general direction kind of things that's a question of mine, where it's going. And we're not really in a sustainable state right now. It will be interesting to see where we are in ten years. Umm...yeah. I mean, a lot of, to me right now, the industry is pretty stupid. So, I just wonder where we're going. (Informant and O'Donnell 2004)

The sustainability question, which is what drives much of the anxiety about the industry, has very little to do with monetary stability. The massive amounts of money swirling around the video game industry at the moment, though staggering, and likely unsustainable are not the elements, which concern developers. They understand that there will likely be a correction in this regard as well, but their unease is more about the long-term viability of people making games.

> Not of myself, but more of like, more how is that all gonna work? And when I'm 45 or 50, am I going to be valued, or that much better because of my experience? Because I know that now, somebody that got started in games in 1982, that means nothing now. It means absolutely nothing about their relevance now. So, if that cycle continues, of the industry reinventing itself every 15 years, and all the tech completely getting turned upside down, and what you knew then means nothing now, then, that's pretty frightening. I'd like to think we're on a course right now, where I will be able to keep up with things, and it always takes work, you have to be conscious of the nature of the industry being fast moving and stuff, but I like to think that if I just stay on top of things, as they come out, that I will have long and nice career in the game industry. Instead of something like, like when the 16-bit era ended, and everybody was pixel artists, if you like, say you didn't even have an aptitude for 3D, you'd be screwed from that point. And then you had all these kids jumpin in saying, I know 3D studio and I'm 17, and I happen to learn software well, and I'm not a

> good artist, but you get into the industry. And then there is this big roll-over thing going on [the transition to "NextGen" or next generation game consoles]. I guess its mostly there is no history to look at how it happens. (Informant and O'Donnell 2004)

In part, the sustainability question is linked to a widespread lack of understanding of the myths and realities of the video game industry. The complicated matter is that in many respects the industry as it has currently constructed itself, depends on widespread belief of the myths noted in Table 3.6, which encourage or hinder new employees and new investment.

Myths
1. You get to play games all day. 2. You get to make the games you want. 3. You get infinite time and resources to make a game. 4. Every game makes millions of dollars, and so do game developers. 5. Games are for kids.

Table 3.6: Common Industry "Myths"

For the most part the first myth has begun to wane in popular U.S. conceptions of game development. Though the particular imagination of what a game developer does all day still fall back on this myth, as can be demonstrated in a movie titled, "Grandma's Boy." The story line's basic premise is a weed smoking, unmotivated, yet brilliant, Q/A developer who plays games all day with his coworkers. In his free time of course he is able to develop a game for the Microsoft Xbox completely on his own and without a DevKit, SDK, Debugger, a computer, or any actual resources of any kind. This "development" process also appears to be his playing of a video game, which his grandmother plays on several occasions, offering her feedback. Even game development programs at schools and universities will fall back on this myth in an effort to recruit new students.[35] The actual work of making games involves tools, which engineers, artists, and managers have already used for years, with new custom tools filling in the holes when necessary.

The second myth is perpetuated far and wide, even amongst some current game developers; only those who have worked in the game industry for years and developed a jaded or realistic attitude towards the reality of their projects. The majority of the time, the games studios work on are those crafted by others. The game developers assignment is to make it a reality or make that idea as fun as possible. This myth however is critically linked to the myth that most games make millions, and easily recuperate the associated costs. Frequently, the most successful games are breakout hits, which are different and new, or those that capitalize on an already established market and franchise. Most of those established games are developed on contract with third party development companies. This of course leads to the myth of unlimited time and resources. While from an external perspective this may seem true of certain companies such as Bungie, Blizzard, or Id ("legends" in the game development community), the reality is that even these companies who frequently respond to questions about schedules with, "It will be done when it is done," are internally quite aware of the limitations on time and budget.

Though no game developers I spoke to ever imagined making millions of dollars based on their work in the video game industry, the myth remains. "[T]he computer industry's garage-to-riches myth fuels the hope of instant success despite evidence to the contrary." "Indeed, younger workers are well on the way to believing that taking entrepreneurial risks is necessary to building careers. This is the legacy of 1980s-era enterprise culture and corporate restructuring." The glamorization of risk encourages develop-

35. An infamous example on YouTube: http://www.youtube.com/watch?v=spSGNMJhWV0

ers and other startup companies in the New Economy a "disincentive to exit during difficult economic times." This same mentality, which is deeply rooted in notions of "meritocracy," also enables "continued attacks on unionized work" (Rogers and Larsen 1984, p. 154; Neff et al. 2005, pp. 317-330). Work performance and compensation has a direct relationship with an individual's skills. If you are unsuccessful or are not paid well enough the problem is a personal one, there is no possibility that the structural might enter into the equation. Put in the context of the video game industry, if you crunch it is the result of a personal lacking.

The final myth is in many respects the most complicated, especially given the contrast between India and the U.S. For many Indian developers, working in the game industry is not considered to be a reasonable career choice. For U.S. developers, many of whom do not take their family or parents advice when considering a career path after college, recognize their family's lack of understanding and support to pursue a game development job regardless. For Indian developers, the choice is not so simple. Many game developers take their game job against the will of their parents, and eventually take positions at companies like Google, Microsoft, or Infosys, much more "reputable" companies that families can feel proud to have their sons and daughters employed with. The perception of games as both a waste of time or simply for kids significantly impacts the ability for students to move into these positions. Given the choice, many pursue work with established companies over local burgeoning game studios.

Given the fact that many foreign video game publishing companies are establishing development studios overseas, the conception that video game development career paths are not practical may decline. However, it will also likely negatively impact the viability of locally started game development studios who will have to compete with more well known names, such as Sony Computer Entertainment, Microsoft, Electronic Arts, and Activision. The very ability of these new industries to push themselves onto the global scene is hindered by the perception of illegitimacy, or lack of rigor of video games.

Regardless, this emerging system of relations between corporations and game development studios plugs directly into a system of labor relations found in many "cool" industries. Sociologists studying new media companies frequently found these same kinds of connections.

> [T]he labor relations within cultural production provide global capital with a model for destabilizing work and denigrating workers' quality of life. The cultural workers in fashion modeling and new media work long hours, networking even while they are schmoozing and boozing, constantly try to improve their skills, and live with a high degree of insecurity about their income and employment. These workers now directly bear entrepreneurial risks previously mediated by the firm, such as business cycle fluctuations and market failures. Popularized in media images of cool jobs and internalized in subjective perceptions, this work creates a model of labor discipline for other industries to follow. Moreover, given the ethnic and gender characteristics that have been associated with entrepreneurial culture, the effect of these changes will exacerbate persistent social inequalities. (Neff et al. 2005, p. 330)

The risk associated with developing new ideas has been offloaded to workers rather than corporations. The image of game developers as pushing the envelope has led them to willingly trade a sustainable industry for the negligible possibility of making it really big. Corporations capitalize on the "venture capital" of game development workers by cherry-picking the best companies for consolidation and acquisition.

3.4 World 3-4: The Consolidation and Acquisition Game

If you watch the video game industry (or any "industry," for that matter), you are bound to notice that currently there are many acquisitions occurring. The video game and other new media industries seem particularly exemplary of this kind of activity. "'In this industry, because it's changing so fast, you're lucky if you're in the same job for a year,' says a producer for a corporate, online retailer" (Neff et al. 2005, p. 326). Even the once famously independent game studio "Blizzard Entertainment, Inc." has long since been acquired by Vivendi/Universal, itself a consolidation of two large media organizations. Figure 3.6 is just a taste of what is a frequent activity amongst game companies.

My primary field site in the U.S. transitioned from independent/third party developer to in-house developer in the winter of 2005. While some have written about the "crisis of the large corporation" and the "resilience of small and medium firms as agents of innovation and sources of job creation," the fact remains that "small businesses are less technologically advanced, and less able to innovate technologically in process and in product than larger firms" (Castells 1998, p. 167). The large publishing companies which employ the vast majority of experienced game developers in one way or another (either through direct ownership of studios or through licensing deals) end up doing less innovative work, despite their higher levels of technological and experiential capacity. Instead "inter-firm" linkages (I would call them intra-firm in most cases, because they are the firms already capable of doing much of the work) are "the multidirectional network model enacted by small and medium businesses and the licensing-subcontracting model of production under an umbrella corporation" (Castells 1998, p. 172). Anthropologists of scientific and technology production note the increasingly complex interconnections between organizations.

ACTIVISION

Print page | Email page |

Activision Acquires Game Developer Vicarious Visions

Acquisition Further Strengthens Activision's Next-Generation Development Capabilities

SANTA MONICA, Calif., Jan 20, 2005 /PRNewswire-FirstCall via COMTEX/ -- Strengthening its next-generation development c
Inc. (Nasdaq: ATVI) today announced that it has acquired game developer Vicarious Visions, the creative studio behind the #
Nintendo® DS title, Spider-Man 2™. Vicarious Visions' proven proprietary Alchemy™ middleware technology and tools will be
Activision's next-generation tools and libraries to further enhance the company's overall development efforts.

The acquisition underscores Activision's strategy to continue building its technical infrastructure in order to take full advantage
potential of the next-generation hardware platforms by acquiring technical and design talent with a history of high-quality pro
over time enhancing the company's financial operating model.

ACTIVISION

Print page | Email page

Activision Set to Acquire Video Game Publisher RedOctane an Guitar Hero Franchise

Activision Establishes Leadership Position in Music-Based Gaming

SANTA MONICA, Calif., May 9, 2006 /PRNewswire-FirstCall via COMTEX News Network/ -- Underscoring its commitment to br
portfolio and deliver compelling entertainment experiences to consumers worldwide, Activision, Inc. (Nasdaq: ATVI) announce
entered into an agreement to acquire video game publisher RedOctane, Inc. the publisher of the popular Guitar Hero franchis
provide Activision with an early leadership position in music-based gaming, which the company expects will be one of the fas
the coming years.

Guitar Hero is the fifth highest teen-rated game ever released on the PlayStation(R) 2 computer entertainment system with
challenges players to be a lead guitarist of their own rock and roll band. The game is sold with a guitar-shaped controller that
simulate playing the guitar for more than 30 popular songs that were made famous by such artists as the Red Hot Chili Pepp
Ozzy Osbourne. Guitar Hero features six venues, eight unique characters, many guitars, and a range of difficulty levels, as w
mode that lets gamers challenge friends to see who rocks harder.

THE TIMES OF INDIA The Times of India -Breaking news, views. reviews, cricket from across India

UTV Software buys Indiagames, Ignition Entertainment

8 Dec 2006, 0016 hrs IST, T Surendar, TNN

SMS NEWS to 8888 for latest updates

MUMBAI: Mumbai-based UTV Software Communication â€" which is currently producing Mira Nair's latest film The Namesake
€" has staked company's claim in the virtual gaming business.

In one go, the company has acquired two gaming companies, Mumbai-based Indiagames and UK-based Ignition Entertainme
for a total consideration of Rs 128 crore. The acquisitions will make UTV Software India's largest gaming and animation comp

Figure 3.6: Screen Shots of Several Acquisition Press Releases

Production increasingly takes place within larger organizations, each of which is more likely to include multiple locations, many of which in turn are in different regional, national, and cultural locations. Moreover, more permeable organizational boundaries mean production occurs within technical and social networks which cross company cultures. (Hakken 2000a, p. 770)

This structure doubly complicates the situation for game development companies in countries other than those with already established networks. While in India the acquisition behavior is in full swing, networks of access limit it. Indiagames was of particular interest because they had inserted themselves into the networks of mobile phone networks. They had developed connections that UTV Software could use to its benefit, thus the acquisition. RedOctane - India, while part of the U.S. company, which had published Harmonix's wildly popular game, Guitar Hero, is also structured by networks of access. Prior to its acquisition by Activision they had no ability to develop games for console systems. After the acquisition they were able to use the networks within Activision to begin working with Nintendo's DS systems.

Of course this does not begin to address the complexity of dealing with publishing companies that in many cases have no knowledge of an emerging industry's internal market. These companies only know that they are supposed to be moving their "activities of production, consumption, and circulation, as well as their components (capital, labor, raw materials, management, information, technology, markets)," towards being "organized on a global scale, either directly or through a network of linkages between economic agents" (Castells 1998, p. 77). For companies looking to develop games targeted internally, they must frequently self publish, which limits them to distributing online for the personal computer or Web. If publishing companies have become "conservative" in the U.S., they are doubly so in emerging markets, which they often understand as only capable of consuming First Person Shooters, Drivers, and Massively Multiplayer Online games.

Even possibly lucrative franchises derived from local markets (Bollywood or Hindu legends, for exmple) are such that U.S., European, or Japanese Producers have little or no understanding of and are unwilling to risk entry. In this context rather than taking a risk to extend themselves into new markets, publishers and manufacturers instead ship only existing titles and refuse to authorize the development of titles specifically for those markets made by people who know the market. In many cases manufacturers simply do not market their games or consoles in these regions. Microsoft is a particular outlier in the Indian context, marketing the Xbox 360 throughout India and even partnering with local banks to provide financing to encourage broader adoption amongst the affluent middle class. Even given this daring move on the part of Microsoft, no authorized licenses for developing games on the 360 have been given to developers in India, the games continue to be developed at U.S. and Western European studios.

While Sony maintains retails stores in India that have begun selling the PSP and PS3, they have not aggressively marketed it in other regions, and the particularly expensive character of the PS3 makes it a difficult sell in emerging economies. Nintendo on the other hand seems to refuse to acknowledge India; even in a market with massive support of mobile devices most gamers have no knowledge of the Nintendo DS system, despite the prevailing "mobile" market, which the DS caters to. The only Nintendo systems in India are those that have been imported or brought back by the few gamers who travel between other countries and India. The importance of being "part of the club" or on the right sides of network switches is dramatically important, though frequently under-examined in the context of emerging networks.

> The more countries join the club, the more difficult it is for those outside the liberal economic regime to go their own way. So, in the last resort, locked-in trajectories of integration in the global economy, with its homogeneous rules, amplify the network, and the networking possibilities for its members, while increasing the cost of being outside the network. This self-expanding logic, induced and enacted by governments and

> international finance and trade institutions, ended up linking the dynamic segments of most countries in the world in an open, global economy. (Castells 1998, p. 142)

And in the case of the video game industry, there truly is "only one game in town," a game that rather than being controlled by political elites working within massive publishing and manufacturing conglomerates. "This is because the global economy is now a network of interconnected segments of economies, which play, together, a decisive role in the economy of each country - and of many people. Once such a network is constituted, any node that disconnects itself is simply bypassed, and resources (capital, information, technology, goods, services, skilled labor) continue to flow in the rest of the network" (Castells 1998, pp. 146-147). The local, the social, and the distinctness of network segments is reemerging rather than disappearing.

> In sum, the more the process of economic globalization deepens, the more the interpenetration of networks of production and management expands across borders, and the closer the links become between the conditions of the labor force in different countries, place at different levels of wages and social protection, but decreasingly distinct in terms of skills and technology. (Castells 1998, p. 254)

Corporate consolidation of video game studios under umbrella publishing companies has severely limited the ability of developers to gain access to the networks necessary for creating games for consoles or distributing them more broadly. Publishers and manufacturers have both participated in the disciplining the particular function of this system, but more in an interest of playing the game to the advantage of their bottom lines rather than out of malice.

3.5 Boss Fight: Bowser is Online and "Open" Source - Digital Distribution

The mechanism of online distribution has recently overtaken the console market. All three consoles have built in networking functionality, connecting them to private networks that the manufacturers control, whether that is Wii Channels, Xbox Live, or Playstation Online. This is exemplary of the ways the video game industry has fought long and hard to ensure exclusive access to networks of production and distribution.

One of the more recent and significant developments has been Microsoft's introduction of XNA Game Studio Express, targeted at hobbyists and independent development studios. This system allows new forms of participation in the creation of video game technologies. Microsoft still maintains control over the more "interactive" Dev-Kits, but provides a means by which curious developers can begin tinkering with console game development.

> "In the 30 years of video game development, the art of making console games has been reserved for those with big projects, big budgets and the backing of big game labels. Now Microsoft Corp. is bringing this art to the masses with a revolutionary new set of tools, called XNA Game Studio Express, based on the XNA™ platform. XNA Game Studio Express will democratize game development by delivering the necessary tools to hobbyists, students, indie developers and studios alike to help them bring their creative game ideas to life while nurturing game development talent, collaboration and sharing that will benefit the entire industry." (Microsoft 2006)

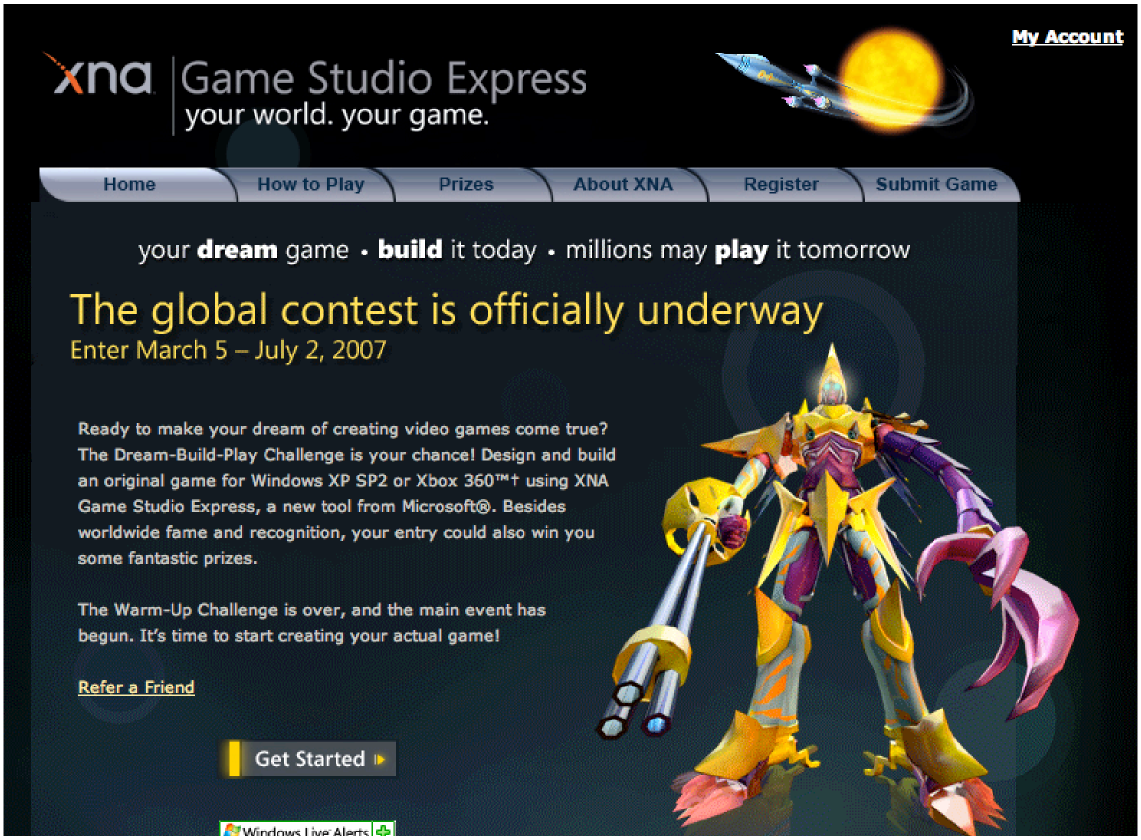


Figure 3.7: Screen Shot from the XNA Express "Dream, Build, Play" Website

Microsoft's motivation is clear, they hope that by dramatically shifting the structures of access in the industry, they can destabilize Sony and Nintendo who have previously depended upon this structure. This structure is one that has kept Microsoft from attaining the top position. While Microsoft continues to control the distribution and availability of these games to consumers, they have managed an opening of the production process that was previously unthinkable. Thus far, however, XNA Express does not go far enough. Instead, it locks developers into the use of proprietary Microsoft technologies.[36]

36. C#, a Microsoft invented programming language, is currently the only supported language for XNA Express, and when developing games, developers are limited only to Microsoft's proprietary

The technological limitations on production and distribution of video games are a vital aspect to how game developers live and work. Troubled relationships with publishing companies are a product of this structure, though rather than looking at the root causes of those relationships, developers frequently focus their ire on the symptom, their parental publishers. The entire system is accepted as a necessary aspect to running a business in the context of the global game industry. The costs associated with developing games can be astronomical even when they are purportedly designed for those with "small budgets and big ideas."

> SANTA MONICA, Calif., June 27, 2007 – The search for the next ingeniously ground-breaking video game has begun. At a private developers conference this week, Nintendo announced the introduction of WiiWare™, a game-creation service that will allow developers large and small to create new downloadable video game content for sale by Nintendo through the Wii Shop Channel of the hot Wii™ home video game system. WiiWare paves the way for smaller, more creative games to make their way to the public at lower prices, without any inventory risk to developers. The first WiiWare content will launch in early 2008.
>
> "Independent developers armed with small budgets and big ideas will be able to get their original games into the marketplace to see if we can find the next smash hit," says Nintendo of America President Reggie Fils-Aime. "WiiWare brings new levels of creativity and value to the ever-growing population of Wii owners." (Nintendo 2007)
>
> Would this mean a price cut for development kits, we inquired? Or would there be a new set of tools and libraries--easier to use, but less fully-featured--aimed at the indie and hobbyist game developer? No. "First, the development tools and SDKs [software development kits] that enable developers to participate are already available," he replied, referring to

DirectX 10 set of video technologies. For users to play many of these games requires upgrading to the latest version of Windows, Windows Vista.

> the standard tools that Nintendo sells to its licensees. "We enable the marketplace where consumers can buy these games using Wii Points. Developers and publishers bring their ideas for games and marketing to entertain and entice consumers." As for a price cut, Fils-Aime insisted that Wii dev kits are already plenty cheap. "All our SDKs and dev tools are already--I don't want to call them inexpensive--they're darn near free to developers. This is unlike our competitors, where you have to spend a lot of money building high-res assets to be competitive. So in that sense, there's almost no cost to developers; the tools are already available at rock-bottom prices. We're providing the venue and light of day for games that might not have gotten attention otherwise." (Croal 2007)

While gamers get excited about Nintendo's WiiWare at a conceptual level, and Microsoft's XNA predecessor, the fact that "darn near free" in reality is minimally $2,500.00 or more for a single DevKit, this appears particularly misleading. Developing games for Microsoft's XNA is free, although to run the game on an Xbox 360, requires an Xbox 360, costing $280 to $600 depending upon configuration and a yearly fee of $90.00. Moreover, what you purchase under a WiiWare agreement is a leased product. The $2,500.00 or more that is spent is for an agreement. The DevKit provided with the agreement will need to be returned once your game has been completed.

It is important to consider that some networks are more influential than others. The inter/intranetworks are not without power relations. However, in a move from despotic power relationships to hegemonic relations, a new kind of measure seems to be more predominant. Reputation networks become a mechanism for structuring the numerous resources within a section of networks. The same is true for both labor and knowledge production. If the practices of a given subsection of the inter/intranetwork are not meeting the expectations of other components of the network, their reputation, and subsequently their income will begin to fall.

If the conditions are so dire why has the situation remained so static? Why has there been so little change? What forces are mobilized to impose the existing structure of the network? For, "[e]lements in the network prove difficult to tame or difficult to hold in place." Value is developed contextually amongst the subsections of inter/intranet-works. Because of this, organizations or individuals must often "display to others, since so much of the work is internal and invisible, leading people to create elaborate accounts of who they are and what they do" (English-Lueck and Saveri 2001, p. 11). These kinds of maneuverings are visible amongst companies both internally and externally. Even as global corporations aquire up new start-ups, they are often maintained such that they can compete with other intranetworks of an organization. This is not limited to local or national contexts. Globalization means precisely that, "a corporation must find ways to connect and integrate its diverse functional operations in different parts of the world so that they can function smoothly together as a global network or system, not as separate, distributed pieces" (Baba 2003, p. 20). "Vigilance and surveillance have to be maintained, or else the elements will fall out of line and the network will start to crumble" (Law 1989). What kinds of vigilance and surveillance? That my friend is an answer you will find in the next castle.

CHAPTER 4
THE GAME INDUSTRY AND THE STATE

The first World introduced you to the domains of game developers, forms of work/play, which plug into World Two's (dis)organizational practices and World Three's networks structured by (in)access and (in)operability. World Four focuses on the State apparatus and its constructed structures that enforce the rules that are experienced in Worlds One through Three. This World seeks to understand some of the rules and structures that underlie the video game industry game. We are gaming instrumentally now; we must be power gamers for the moment. We need to desire to know what makes this game tick in the fashion it does. We must desire and pursue those underlying systems and structures. This World points to some of the technological mechanisms that legislate this game space. More than that, it considers how technological components are networked with interested parties and political and legal structures of regulation to ensure enforcement in ways more effective than the simplest or most elaborate technological fixes. The force of these rules is truly felt when they come in concert.

World Four also reconnects "users" with "producers" of video games. Those technologies that structure users also shape the working worlds of producers. This is done partially by drawing historical connections between technologies that began as a means of controlling production and have transitioned to technologies to control distribution. However, at a fundamental level, these become the same thing in practice. Many of the mechanisms, which structure users, structure developers. What is cast as simply the "right" to manage copyright becomes a means to control speech.[37] World 4-1 exam-

37. I see this argument as intimately connected with, "Our experiential relationship to our technology has already shifted dramatically over the last two centuries, from one where we were primarily the makers of our own tools to one where our tools are largely mass-produced for us elsewhere, but we at least in principle retain access to their inner workings as potential tinkerers. With the robustness

ines the different sets of interested parties that have a stake in the regulation of game content or game hardware. World 4-2 focuses on the objects in/of regulation concentrating on the hardware and technologies of video game systems, with particular attention to console game systems. World 4-3 takes the objects in/of regulation with specific attention to the content of video games. Content is largely conceptualized as something to be consumed, and I instead use the term "producer" throughout to signal the productive capacity of the individual who uses, plays, and co-constructs the narratives of video games. The role of consumer seems too one-sided, especially in the video game industry. The user is not without power over video game companies, though access and capacity may be seriously constrained. The framework of "ruling relations" as developed by sociologist Dorothy Smith serves as a means to conceptualize how the local is regulated by the extra-local (Smith 1999). This theoretical construct guides the analysis of this chapter, but with a critical eye towards the productive potential of users and game developers. The ruling relations structure the local activities of "users" and game developers alike, certainly in different ways, but they are not disconnected. The Boss Fight for World Four and the final Boss Fight of the substantive sections of this text analyzes the rising deployment of the "prerogative dimension of the State" (1995, p. 186) on the behest of corporate institutions. It is my contention that this coercive moment, the extralegal and violent capacity of the State ought to signal an important moment when thinking about productive capacities in the New Economy. Ready? ... Fight![38]

---

rules built into DRM systems, this relationship may shift again, to one in which we are discouraged from investigating our own tools and from using them in ways other than intended and authorized" (Gillespie 2006, p. 662).

38. Frequently in fighting video games the start of a game round is signaled by a phrase along these lines. Given the particular attention to the violent and extralegal aspects of the State in this World, this beginning seems apt.

4.1 World 4-1: The Interested Parties - Agencies and Groups

There are numerous interested parties intent on regulating the video game industry. This interest ranges from ideological to economic. Video game console manufacturers, intellectual property (IP) holders, video game studios, and software development firms make up the largest contingent of "non-political" entities acting in this arena. These groups then organize strictly political organizations like the Entertainment Software Association (ESA) that lobby on their behalf. Other organizations, like the International Game Developers Association (IGDA) organize to represent the workers of the video game industry. Less organized groups mobilize around parent groups and other interests that vary across the political spectrum.

Most of the regulation has occurred along the "capitalist dimension" of the State, "securing through private property rights, capitalist relations of production in the first place; buttressing and mediating - through production subsidies, contracts, and fiscal regulation - the relations of production." These efforts have been spearheaded by corporations looking to ensure that existing structures exert force over users and producers. Simultaneously, workers within the game industry have mobilized around the "liberal dimensions" of the State, stressing their "right" to unregulated content production (Brown 1995, pp. 184-186). While companies are happy to employ state structures to exert force over users and producers, they have simultaneously made the double movement insisting that government oversight of what they are producing need not be a piece of the puzzle. Companies and trade organizations have argued that ratings and monitoring of the video game industry is something that should be done outside of the State apparatus.

4.1.1 The Formation of the ESRB

The Entertainment Software Rating Board, or ESRB is a non-profit self-regulatory agency created in 1994 by the ESA. This organization is responsible for the assignment of ratings to games. They are also responsible for enforcing advertising guidelines and privacy practices amongst game development companies. While it is "voluntary" to participate with the ESRB, most retailers will not place games on shelves without an ESRB rating. More recently the ESRB has come under fire as several games, which had received one rating, were discovered to have more explicit content than originally disclosed. The particular emphasis which is placed on sex and nudity and lack of attention to violence has also lead consumers and game developers to question the ratings system altogether. Many developers also felt that parents were not paying any attention to the rating system. What sense does it make to take so much time and effort to go through the rating process if it was simply ignored? The ESRB's response was a public awareness campaign, an example of which can be seen in Figure 4.1.

The ESRB is also an obligatory passage point for any "legitimate" game developer looking to distribute their game. None of the major retail chains will sell a game unless it has been rated by the ESRB. This cost is roughly $2,500, above and beyond any costs incurred while trying to gain access to the already mentioned social and technological networks of access, and you must ensure that your content abides by the ratings standards of the organization, whose standards have been frequently questioned by developers and users alike.

> The Entertainment Software Rating Board still has a role to play. "The developer is responsible for getting an ESRB rating for their game, just as with our current publisher agreements. We don't allow AO-rated (Adult Only) games on our systems. (Croal 2007)

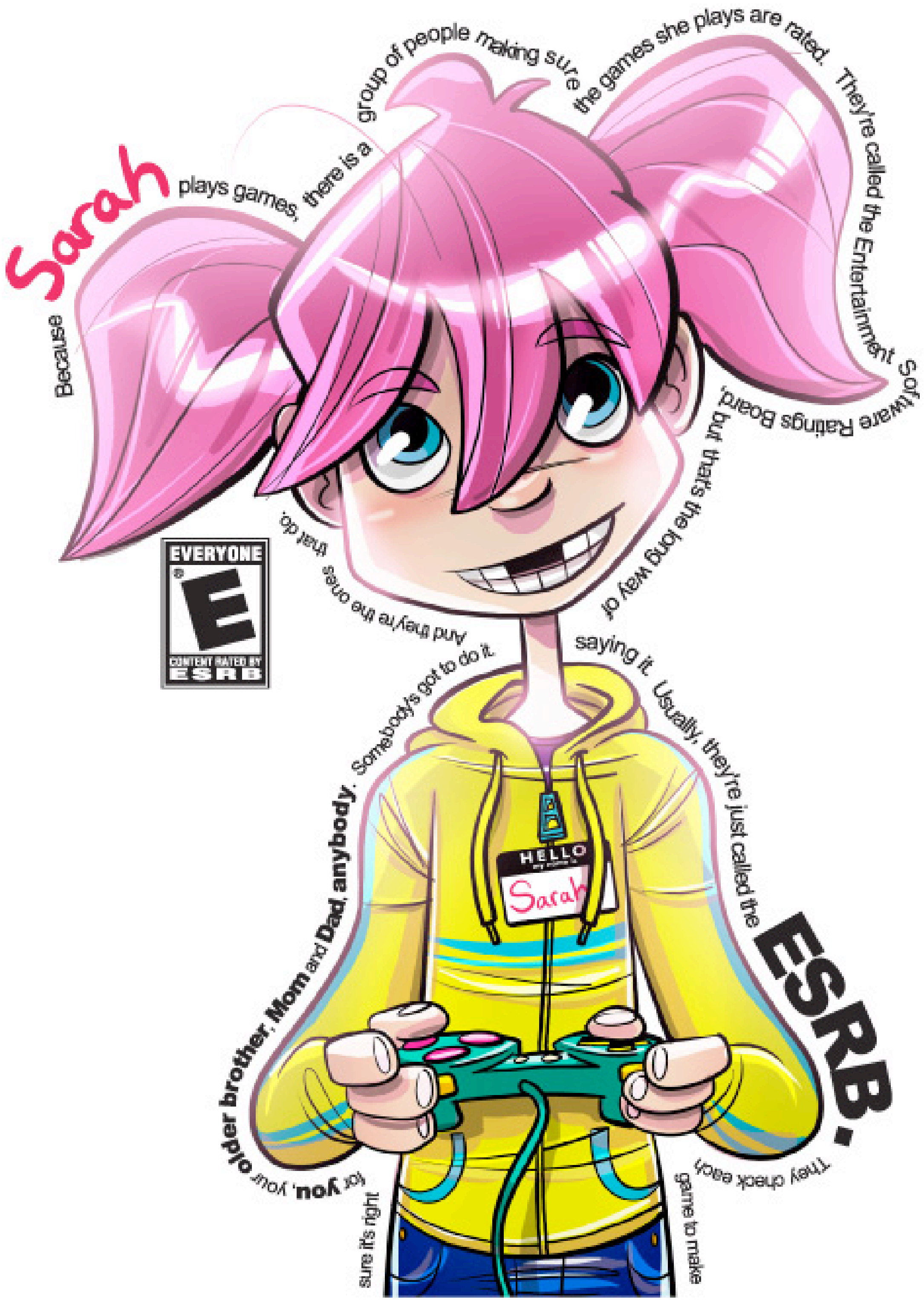


Figure 4.1: An ESRB Public Awareness Ad (Mizrachi and Khoo 2006)

Until recently, games were played very little by the rating organization, the ESRB, which was one of the more interesting aspects of the process. They employed people who played games, but it was not an emphasis of the ratings process. Rather video clips of "representative" content were observed to understand what a game would typically look like. Unlike movies and movie ratings, you cannot simply sit down and watch a game for two hours and assume that experience you would have had will be the same as the next "player." Indeed, the difference between someone who watches a movie and a person who plays a game is indicative of the difference. The fact that games can be played in different ways and with different purposes indicates the complicated nature of rating a game. Because of this the ESRB has begun to bring more people into their organization to spend more time playing the games that are being rated. Many of their raters are not gamers, which is both useful and problematic. It is of course useful because it brings a broader perspective into the interpretation of the images, what a gamer might not find offensive or problematic, someone else may. However, gamers will typically play games differently from non-gamers and if the two worked together to rate games, it is possible that a more thoughtful and in-depth understanding of a game would be gleaned.

In locales where rating standards have not been implemented, the ESRB reigns instead. Developers in India interested in publishing a game in India on the Xbox 360 must navigate Microsoft's networks as well as those of the ESRB. Even once this has been completed, they are not guaranteed the ability to publish their work. The same holds true for independent U.S. developers, though the ability to tap into the social resources for determining this information is more readily available.

4.1.2 Political Interest in Legislation

More recently other governmental institutions have stepped into the fray, hoping to modify and adjust some of the ways in which games are sold and marketed. This debate rapidly polarizes into the "freedom of speech" and "protect our children" camps, neither of which attempts to learn from the other. In the freedom of speech camp you have primarily gamers, game developers, producers, and users of game technologies. In the other camp resides primarily politicians, lawyers, and in some cases parents groups. Unfortunately only one camp contains the majority of producers and consumers of games, and the other group contains few individuals with knowledge of games, gamers, or the historical foundations of what now exists. This is not a conducive dialog, because in many respects there is no real solution being sought by either side. Gamers and game developers are happy to continue doing what they have done, they perceive no problem in the process. Politicians and lawyers view the debate as a campaign strategy or path to litigation. Obviously this is not a foundation conducive to productive dialog. The following quote, from a letter to the editor written by the studio heads of my primary field site, traces the divide between legislatures and game developers.

> "Just like movies, books, photographs, music and other forms of art and entertainment, video games are fully protected speech under the U.S. Constitution. In fact, nine federal courts in the last six years have ruled that legislation in other states substantially similar to what is being proposed in New York violates free speech protections. States have wasted hundreds of thousands of taxpayer dollars to defend these statutes. Several states and municipalities have been ordered to pay more than $1.7 million to the video game industry for legal fees. Given New York's pressing economic needs, it can ill afford to spend money enacting and then having to defend this proposal.
>
> ...

> "First, the vast majority of New York retailers only sell video games rated by an independent organization called the Entertainment Software Rating Board. It evaluates computer and video games for age appropriateness and content, and provides a rating system for parents.
>
> "That system works. A recent Federal Trade Commission report calls the rating board 'a useful and important tool that parents increasingly use to help them make informed decision about games for their children.' The FTC said the board leads in 'providing clear and prominent disclosure of rating information.'
>
> New York retailers have voluntarily adopted policies not to sell mature and adult video games to children -- just like movie theaters voluntarily prevent children's access to R-rated movies." (Bala and Bala 2007)

And while the New York Legislation does risk further waste of taxpayer dollars to defend statutes, which are likely to be defeated as unconstitutional, it too pays little attention to the ways in which games are actually produced. The focus on criminalization of "M" rated game sales to minors looks at a very small portion of the game development process. The retail solution is trivial and easily circumvented. If governmental interest in games is going to be limited simply to controlling the retail channel, then their endeavors will either fail in the legal fight, or fail at the cash register. In many ways the concerns of politicians simply echo those of previous generations concerned about the Internet, television, and radio. Oddly, rather than learning from the previous generations that distribution point attention is only a small piece of the puzzle for success, the choice has been made to trod heavily on an already beaten path. The question for new producers of games makes these activities particularly problematic. Is making a game available online a sale? What about unrated games? Must games be rated to avoid being prosecuted for distributing "M" caliber games to under 17 minors? The already confusing set of networks becomes rapidly more convoluted, and ultimately unavoidable despite your geographic location.

4.2 World 4-2: Objects of Regulation - Hardware

As previously noted, the NES marks a kind of pivotal moment in the world of video game development, and in the realm of rule making and rule enforcement, this also holds true. In December of 1985, shortly after the release of the NES in the U.S., Nintendo filed for a patent, the only public record of the technological device that had been developed.[39] This microprocessor, the piece of code carved in silicon, was about to change the world of video game development forever; code can be legislation simply of another kind.[40]

Without this filing, the innovation would be largely undocumented, yet another one of the pieces of the "Black Art" of video game development. This deceptively simple description belies a much more complicated device, a semiconductor lock and key. A silicon lock and key to ensure the "authenticity" of an external memory device. This is what added force to the seal. The key and seal worked together. To get the key you needed a seal. To get the seal you needed Nintendo. The following patent document excerpt demonstrate the "invention" of this silicon/digital lock and key.

39. Several texts have mentioned the existence of the 10NES chip, but none of these texts ever offered any proof of the existence of the device (Clapes 1993; Sheff 1993). Despite this, others have been willing to us these reports as facts, without inquiry into the validity of these claims or the functionality of the device (Kline et al. 2005).
40. While I think the situation is more complicated, as this World and much of the dissertation would seem to indicate, I do believe that viewing code/technology as "legislation" as well as "speech" is a productive tool for thinking about technology (Lessig 1999). Taken to its extreme however, I think it assumes to much about the deterministic character of technology.

U.S. Patent Jan. 24, 1989 Sheet 2 of 10 4,799,635

FIG. 2

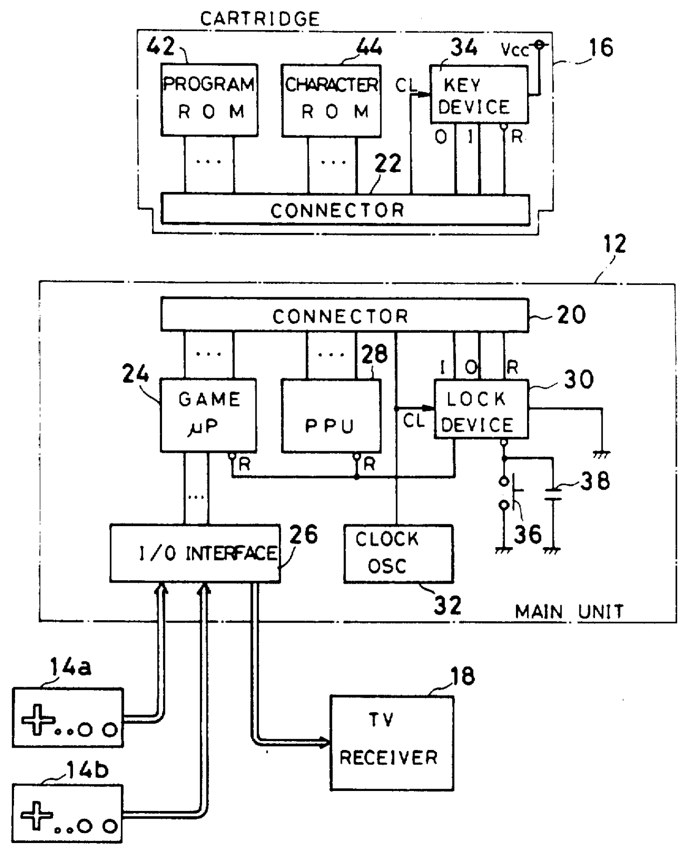


Figure 4.2: The Lock and Key of the 10NES Patent

"United States Patent Number 4,799,635 - System for Determining Authenticity of an External Memory used in an Information Processing Apparatus.

"To verify that the external memory is authentic, duplicate semiconductor devices, for example microprocessors, are separately mounted with the external memory and in the main unit, respectively. The semiconductor associated with the external memory device acts as a key device and the duplicate device mounted in the main unit acts as a lock device." (Nakagawa 1985)

If you have any doubts about Nintendo's intentionality with regard to why this method was developed, one need only to look as far as the court cases that quickly followed for those companies who did not care to work through the system they designed. Their own testimony indicated that the 10NES chip, as it came to be known, was designed specifically as a means to enforce licensing agreements. It is interesting to note, that at this moment the interest was in protecting Nintendo's ability to say who could make and release games for the NES, rather than on copy protection, and while I do not address those concerns in depth in this World, I will return to them in World 4-2 and 4-3. The intentionality of what the device was designed to do is directly indexed in court cases.

> "Nintendo designed a program -- the 10NES -- to prevent the NES from accepting unauthorized game cartridges. Both the NES console and authorized game cartridges contain microprocessors or chips programmed with the 10NES. The console contains a "master chip" or "lock." Authorized game cartridges contain a "slave chip" or "key." When a user inserts an authorized cartridge into a console, the slave chip in effect unlocks the console; the console detects a coded message and accepts the game cartridge. When a user inserts an unauthorized cartridge, the console detects no unlocking message and refuses to operate the cartridge. Nintendo's 10NES program thus controls access to the NES." (Atari et al. 1992)

While this technological legislation encouraging developers to work directly with Nintendo was dramatically different from how things had been managed for previous consoles, it was in many ways only the beginning. This leads us to two questions, "Why not pick the lock?" which can in many ways be answered by the second, "Why the patent?" Nintendo had learned from Atari's experiences and knew that the kinds of people interested in making games were very resourceful. A simple technological device while capable of influencing the way in which games were developed for the new NES, was not enough to ensure control over the rights of production. This power is simply out of the hands of most organizations. To exert that kind of control requires the mobilization of government intervention, which is precisely why we have a patent to point to in the first place.

Interestingly, NES's Japanese counterpart, the Nintendo Famicom did not contain the 10NES lockout chip, and while this did result in some levels of piracy, which Nintendo combated with the Seal of Quality without technological or legal networks, it also resulted in a longer life cycle for the console. Well after Nintendo had released the Super Nintendo Entertainment System, games were still being released for the Famicom. Many of them were unlicensed, but gamers continued to buy them, keeping the console in the living room well beyond Nintendo's expectations.

To exert control over the networks that Nintendo hoped would form around the NES, code was not a sufficient form of legislation. The patent office provided one legal means, by which force could be mobilized against those wishing to get around Nintendo's seal of quality. They were also careful in their copyrighting of the code that composed the 10NES chip. This provided them a second means of mobilizing the state to enforce compliance with their rules, which they could not. Those wishing to pick the lock were now subject to litigation, and while it was still possible to reverse engineer the patented technologies, those doing so must be careful so that when they were sued, they could properly defend themselves in court.

While several companies did manage to circumvent the lockout capabilities of the NES, it was largely not the case. In Nintendo's most publicized legal loss, Galoob demonstrated that their Game Genie product for the NES made no use of copyrighted

Nintendo technologies. Rather the Game Genie merely altered the code being transmitted from cartridge to console. The Game Genie did not circumvent the 10NES lockout chip, instead using the key device in the cartridge to allow normal booting of the NES. However, the far more common case was that companies attempting to market games outside of Nintendo's new rule system paid the price.[41] The most famous of these cases involved both patent and copyright infringement. Nintendo versus Atari and Tengen became the precedent for many of Nintendo's future legal claims. The case was foundational for all subsequent litigation to control the means of production. The case is also particularly important, because without it, the details of licensing arrangements mentioned in World 3-2 would have remained invisible. Unfortunately the case was exceptionally poorly played by Atari and Tengen, as is demonstrated in a brief excerpt from the court report. Needless to say, Atari went about "reverse engineering" the 10NES in a most harebrained[42] fashion.

> "Atari first attempted to analyze and replicate the NES security system in 1986. Atari could not break the 10NES program code by monitoring the communication between the master and slave chips. Atari next tried to break the code by analyzing the chips themselves. Atari analysts chemically peeled layers from the NES chips to allow microscopic examination of the object code. Nonetheless, Atari still could not decipher the code sufficiently to replicate the NES security system.
> ...

41. There are of course examples of companies that managed to circumvent Nintendo's lockout mechanisms. As near as I can tell based on unscientific searches, 87 unlicensed titles have been released, compared to the 670 licensed titles (Nintendo 2003).

42. As the following court case selection indicates, Atari's resulting silicon lock pick was called the "rabbit program." Though I use the term "harebrained," I do so with tongue in cheek, the kind of ironic word play that would not be uncommon in game development worlds. I do not intend it simply as "stupid."

In early 1988, Atari's attorney applied to the Copyright Office for a reproduction of the 10NES program. The application stated that Atari was a defendant in an infringement action and needed a copy of the program for that litigation. Atari falsely alleged that it was a present defendant in a case in the Northern District of California. Atari assured the "Library of Congress that the requested copy [would] be used only in connection with the specified litigation." In fact, no suit existed between the parties until December 1988, when Atari sued Nintendo for antitrust violations and unfair competition. Nintendo filed no infringement action against Atari until November 1989.

After obtaining the 10NES source code from the Copyright Office, Atari again tried to read the object code from peeled chips. Through microscopic examination, Atari's analysts transcribed the 10NES object code into a handwritten representation of zeros and ones. Atari used the information from the Copyright Office to correct errors in this transcription. The Copyright Office copy facilitated Atari's replication of the 10NES object code.

After deciphering the 10NES program, Atari developed its own program -- the Rabbit program -- to unlock the NES. Atari's Rabbit program generates signals indistinguishable from the 10NES program. The Rabbit uses a different microprocessor. The Rabbit chip, for instance, operates faster. Thus, to generate signals recognizable by the 10NES master chip, the Rabbit program must include pauses. Atari also programmed the Rabbit in a different language. Because Atari chose a different microprocessor and programming language, the line-by-line instructions of the 10NES and Rabbit programs vary. Nonetheless, as the district court found, the [F.2d 837] Rabbit program generates signals functionally indistinguishable from the 10NES program. The Rabbit gave Atari access to NES owners without Nintendo's strict license conditions. (Atari et al. 1992)

This case begs the question, why was Atari trying to get around the limitation? Why not just talk to Nintendo? There is of course the possibility that Atari was simply bitter, having gone from the leader of the video game industry to a player forced to work within the rules of another company. It is possible that Atari simply wanted more of the money coming to them rather than going to Nintendo. What neither answer gives us is any insight into the world, in which Atari, Tengen, and Nintendo were operating. We have no insight into why companies were feeling compelled to work around Nintendo rather than with them.

4.2.1 Taking the PPU Bait

One aspect of the answer is that the NES was so chock full of new and interesting technologies that developers were happy to buy into licensing schemes as a means of inserting themselves into the networks, which were emerging, allowing them a chance to play with new technologies. One particular innovation that separated the NES from previous generations of consoles was its use of a new technology, a precursor to the now ubiquitous GPU. Nintendo modified the way graphics were processed and delivered to the television screen on the NES. This innovation improved and simplified the way graphics were stored and delivered to the screen. It also allowed the CPU of the console to spend more time doing game related operations and less time doing graphics related operations. This Picture Processing Unit or PPU was a major design innovation for the NES.

Developers were willing to trade their rights for new and shiny technologies. This process can be understood as a kind of hegemonic project designed to encourage developers to broadly trade freedoms, like the freedom to talk about their work, for privileged access to new technologies and markets. One can think of hegemonic projects[43]

43. I see the hegemonic project or counter-hegemonic project as also answering some of the calls to defetish the State, or understand the State as an actual human process as being more like a "nervous Nervous system." An "'organization, i.e. a collection of individual human beings connected to a complex set of relations'" (Taussig 1992, pp. 112-113). This gets at the distinction between the

"as occurring through a linkage between structure and representation." Hegemonic "projects do the ideological 'work' of making these links." A hegemonic "project is simultaneously an interpretation, representation, or explanation of [differential] dynamics, and an effort to reorganize and redistribute resources along particular ... lines" (Omi and Winant 1994, p. 56). Hegemony as defined by Antonio Gramsci operates through process of coercion and consent (Simon 2001, pp. 24-32). So too does the imbuing of networks with power relations. Without our acquiescence these systems of control will fail. While it may seem that I am leveling all blame at the console manufacturers, in fact I am not. In the video game industry, there is no boogieman; rather the boogieman is constructed through the co-construction of these systems and networks. These networks are ones, which ultimately, we allow to be constructed. The following sampling from patent documents index one of the numerous technological innovations which encourage game developers to trade their rights to share and collaborate for new and interesting technologies, like the NES's PPU.

State apparatus and the State as we know it, or as an assemblage approximating the apparatus. The notion of (counter) hegemonic projects gets at some of the underlying agency of human beings connected in a set of relations. Those relations become increasingly important in our understanding of how the State assemblage operates.

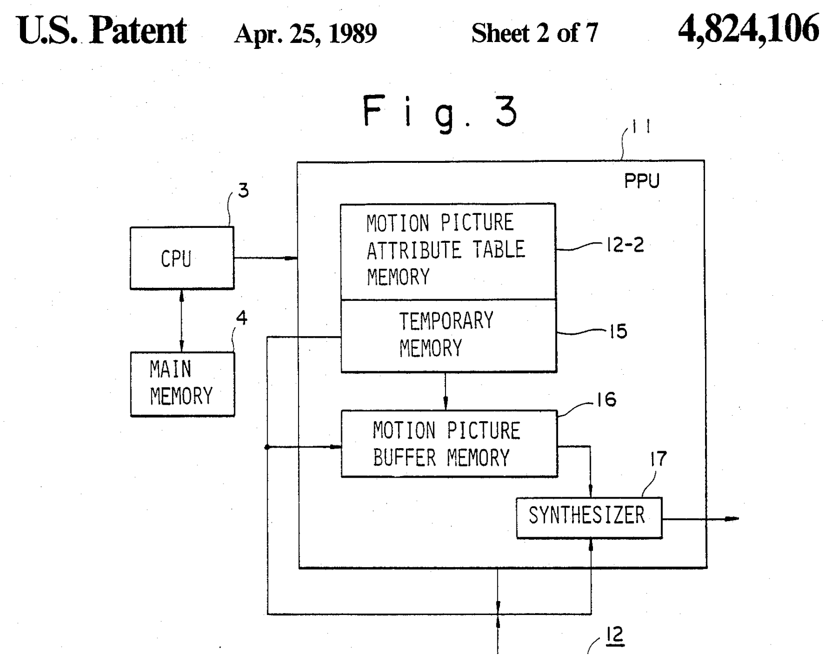


Figure 4.3: The Picture Processing Unit for the NES

> "...a central processing unit for controlling the overall operation of the system under the control of the operator, and a picture processing unit for combining motion and still picture patterns to form a video signal to be supplied to the T.V. set..." (Ueda and Yagi 1987)

The PPU changed the way that resources were allocated for games. Pixel memory differed from system memory. Sprites, or small images, could be loaded into memory in a different location from that of running game code memory. The visualization below depicts quite well how this system functioned in operation.[44] The sprites on the cartridge are loaded into the temporary memory of the PPU. These can then be moved around with the motion picture buffer and synthesizer without the constant intervention of the console's CPU. Figure 4.4 demonstrates how the PPU provides both power and new lim-

---

44. This visualization would not be possible without some of the "Righteous Hacking" referred to later in this World.

itations on developers working with the hardware of the NES, with sprite memory of the cartridge pictured on the left, and the PPU's currently loaded sprites on the right, with the resulting game code piecing together the data running in the center of the image.

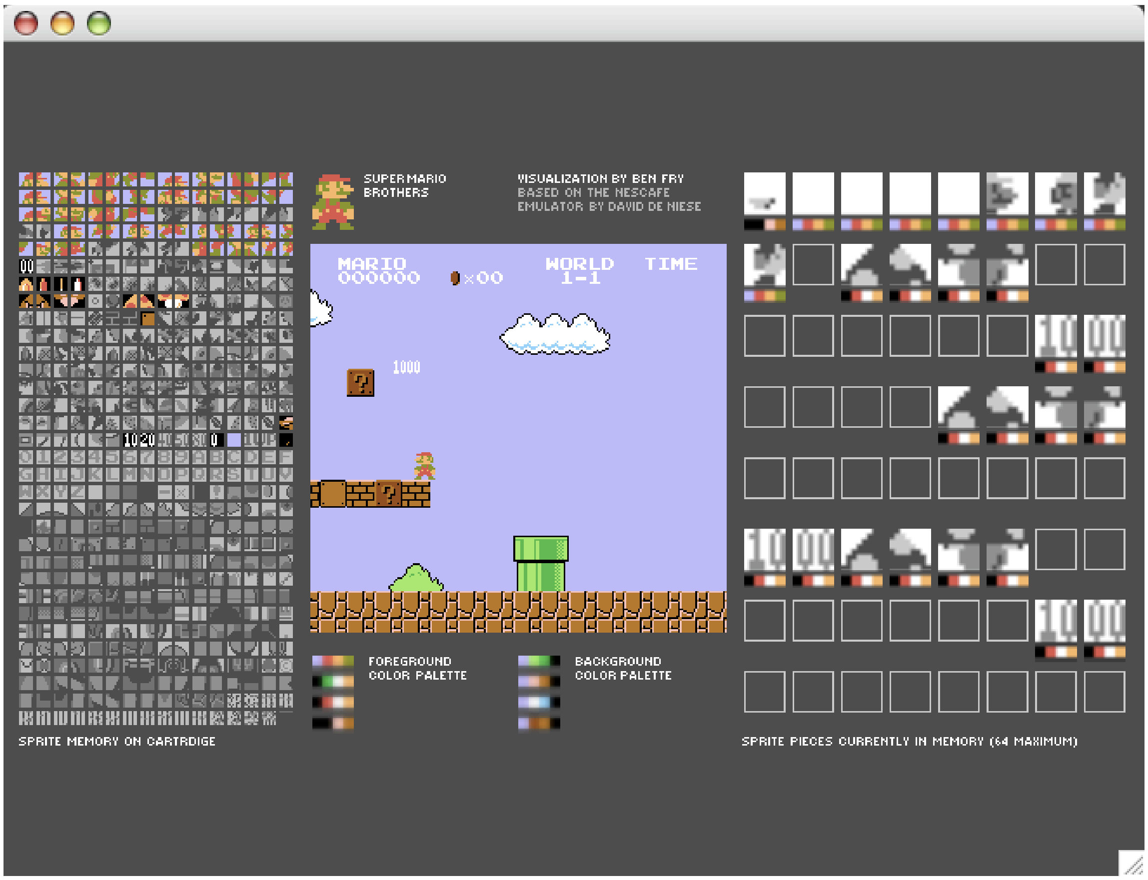


Figure 4.4: A Visualization of the NES's Art Hardware[45] (Fry 2003)

Of course government institutions, patent, copyright, and regulatory bodies are not simply in the service of or opposed to corporate, user, or producer interests. There are always swerves within the system. Unfortunately this slipperiness tends to favor

45. When I showed this to several engineers who had worked on the NES, many commented on how they only wished such a visualization had been possible when they were working on the system. Paradoxically, this visualization and the software emulator, which is uses to generate the images, violates Nintendo's patents on the NES and copyright of Super Mario Bros.

those who already have ready access to the system. Those looking to just enter into the networks of video game production stand much to lose. Existing corporations with large bank accounts can afford to either buy their way out of difficult situations or keep a case in court until its opponents can no longer afford to keep up the battle.

While my characterization of the patent system becoming a tool of corporations to control production and distribution, rather than an institution protecting the rights of producers of intellectual property, the system is not without its swerves in other directions. Throughout this research I have encountered numerous occasions where patents become a particular problem for companies, or corporations patenting devices and not leveraging their right to sue corporations who they likely could have.

One of the most well known cases of inter-corporate patent litigation amongst game companies was fought first outside of court between Immersion and Microsoft. Microsoft settled the case rather than fighting it. Sony on the other hand fought the case. The case appeared to revolve around the "Dualshock" controllers for Sony's PS1 and PS2.

One of the more complicated aspects of this case is that Sony's patent appears well ahead of that by Immersion. Even Nintendo did not patent their force feedback controller until nearly the same time as Immersion (Nishiumi et al. 2001). Nintendo is conspicuously absent from the litigative story. What seems to be argued rather than simply the use of force-feedback, or the controller is the relationship that feedback has with the underlying software. While Sony had patented the placement of actuators in their controllers, they failed to establish the relationship of the "System of Exploiting" with the virtual content of the game. Nintendo did this in their patent, which prevented them from coming under fire from Immersion.

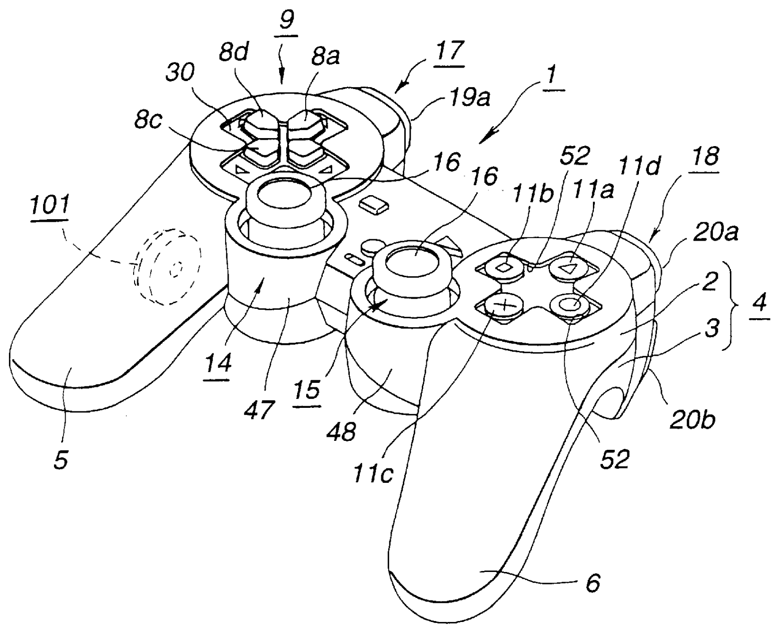


Figure 4.5: The Sony "Dualshock" Controller (Ogata et al. 1998)

> In brief, Immersion prevailed at trial on certain of its claims that Sony's Playstation consoles and Dualshock controllers, in conjunction with forty-four accused games, infringed U.S. Patent Nos. 6,275,213 and 6,424,333, owned by Immersion. The jury found that the asserted claims of Immersion's patents were not invalid due to anticipation, obviousness or inadequate written description. The jury awarded Immersion eighty-two million dollars as a reasonable royalty for Sony's infringement. The Court later found in favor of Immersion on Sony's inequitable conduct defense and denied Sony's motions for judgment as a matter of law. Sony has appealed the judgment against it. (Corporation and Sony Computer Entertainment 2007)

This troublesome ground of what precisely must be placed in patent documents to adequately protect from litigation is particularly troublesome. The tendency for patent documents to be as general as possible with far reaching application, while also being specific enough to be deemed non-obvious is exceptionally problematic. This battleground has exploded in recent years since the Immersion case, with companies attempting to patent game related technologies ahead of the major manufacturers. In many cases however, these companies fail to actually develop a product based on their patent, which is a stipulation of ensuring the continued validity of a patent. Despite this, many companies are more likely to settle a patent dispute out of court rather than go to trial. This has resulted in numerous companies patenting technologies that have questionable merit.

More interestingly throughout this story is the patenting of game related technologies which are certainly deployed in products, but have likely also been independently developed by others in the video game industry without litigation. Nintendo in particular seems to be content to patent technologies which have become commonplace in the design and development of video games, and then not pursue litigation. One can only guess at the motivation for this behavior, though it is commendable, because enforcement of these patents could have extremely negative repercussions on the game industry more broadly.

One such example of this is a patent by Nintendo, which relates the angle and degree of pressure placed on an analog joystick, to the direction and speed of movement of an object in game space.

Anyone who has played a 3D game on a modern console utilizing analog joysticks will recognize that this patent actually applies to nearly every single one of these games. One wonders if this kind of patenting is done to protect the industry or simply themselves from the kind of litigation that has risen dramatically in recent years. Regardless of the motivation, it seems fortuitous for the game industry more broadly to be protected in such ways.

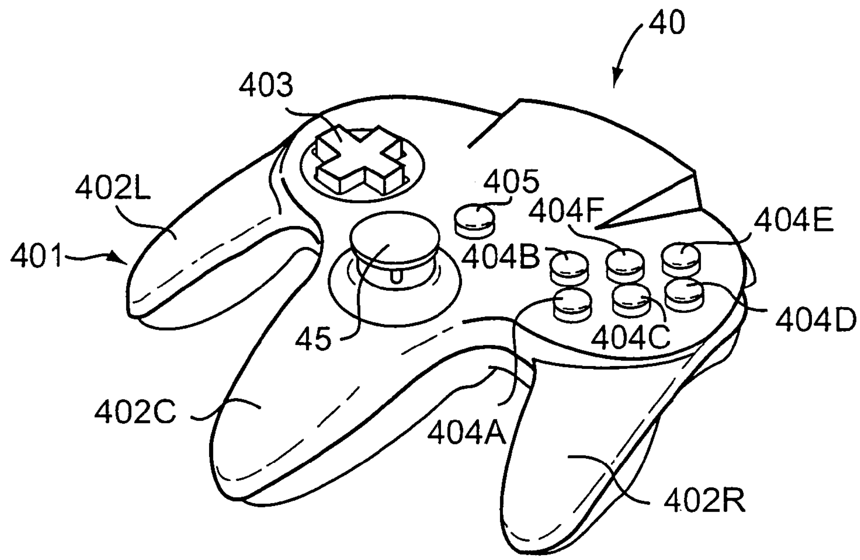


Figure 4.6: The N64's Analog Joystick Controller

> The system further includes direction determining circuitry operable to determine a direction that corresponds to an inclination direction of the operating member based on the inclination amount data, and moving object direction determining circuitry which determines a moving direction of the object in three-dimensional space based upon the direction determined by the direction determining circuitry and a point of view angle at which the object is being viewed by the operator in three-dimensional space. (Nishiumi et al. 2005)

Unfortunately the existing condition of patent, copyright, and legal structures tends to favor those already in positions of power. Those looking to enter these worlds must navigate amongst companies hoping to patent poach, ratings boards, social and technological networks of access and secrecy, and hope that in the end they are allowed to produce a game which can enter the not-so-flat global New Economy.

4.3 World 4-3: Objects of Regulation - Content

Placing limitations on the distribution and use of media, writing, and technology has long been a concern and interest of those that create it. Video games are no different, being an assemblage of all of those and more. The methods of limitation have changed however, and it is important to recognize that these changes have come to more and more leverage the powers of the state.[46] These limitations are also inherently limitations on the ability to produce for technologies covered under these methods. The more tightly controlled distribution, the more tightly controlled production will be.[47]

4.3.1 Leveraging the High Cost of New Technology

At its simplest level, the cost of technology can do a great deal to ensure that users and distributors of video games follow the rules set by video game manufacturers. If you look at the technologies used in each console system (a table of console manufacturers and their products can be found in the Appendix) and look for the cost of a given distribution medium it becomes apparent that when first released each was prohibitively expensive for an average consumer. In 2000 when the PS2 was introduced, the cost of a DVD burner was more than $4000 and disks for those burners started at $40, nearly the cost of

46. I actually construct this in contradiction to accounts of neoliberalism as a hollowing out of the state, or the "subjugation of political and social life to a set of processes termed, 'market forces,'" or the advocacy of "a competition-driven market model," (Farmer 2005, p. 5). Rather, it seems a mobilization of the state to perform duties that corporations operating in a market cannot do. In many respects it is making the move for both de-regulation and increased regulation of political and social life.

47. Or in a Deleuzian way, "this whole chain and web of power is immersed in a world of mutant flows that eludes them. It is precisely its impotence that makes power so dangerous. The man of power will always want to stop the lines of flight, and to this end to trap and stabilize the mutation machine in the overcoding machine" (Deleuze and Guattari 1987, p. 229). This is precisely why control must be maintained over the realm of the producer. Left to its own devices it will move in directions which may make SEC filings more difficult, but perhaps more beneficial to new markets and new producers; competitors.

a video game to the average consumer. By the end of the PS2's life span however, DVD burners were a common add-on for new PCs.

FIG. 2

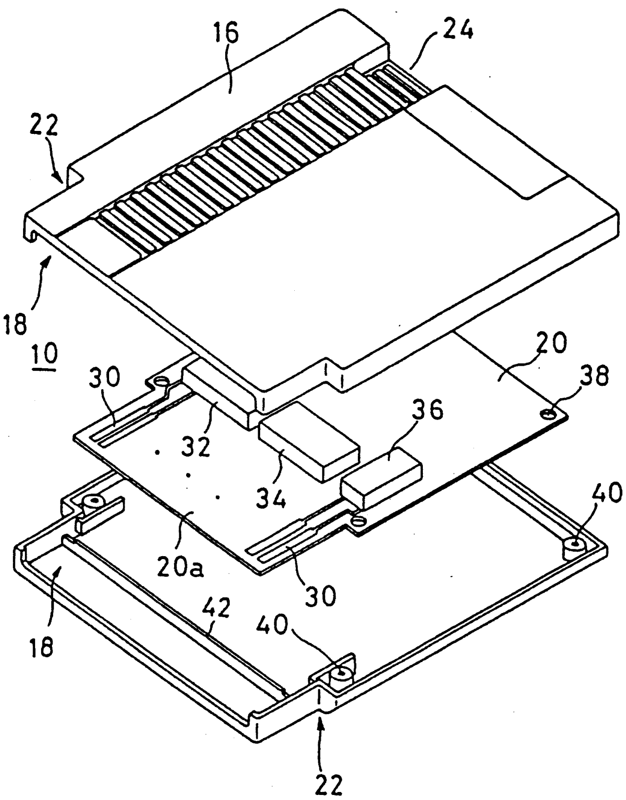


Figure 4.7: A Schematic of a NES Cartridge (Nakagawa and Yukawa 1987)

The same was true for nearly every medium on which console games were distributed. In 1996 64MB memory modules were not cheap, nor was it easy for consumers to create them or place them into plastic cartridges, which would fit into the N64. In part this was one reason Nintendo opted to continue using cartridges in their systems, despite Sony's move to use CD-ROMs in the competing Playstation. Of course this does not mean that either company did not also take precautions despite the cost or difficulty of producing these items. Knowing full well that an all-out copyright assault would be expensive, drawn-out, and difficult, each company again employed the use of secretive and patented mechanisms to prevent production of unauthorized duplicates.

4.3.2 Patenting the Lock and Key, Watch, Change, Repeat

Take the GameCube (GC) and its more recent counterpart, the Wii, for example. Each made use of a mechanism referred to as "burst cutting" to create authorized game disks, despite both using DVD-based technologies for the disks themselves.

> "Burst Cutting Area [BCA] Extraction Method and Device...
>
> A BCA is an area arranged on the inner periphery of an optical disk. A BCA-Code, which is a series of low reflectance stripes, is formed in the BCA. The BCA is formed on a disk using a laser cutting process after the fabrication of the disk. Thus, the manufacturer can record desired information in the form of the BCA-Code, on the disk. For example, the serial number of the disk or anti-counterfeit information can be recorded to the disk. ...
>
> It is therefore a primary objective of the claimed invention to provide a method for automatically recovering the time width T of one channel bit in order to read the BCA-Code quickly and easily." (Yen and Shen 2003)

Nintendo is attempting to ensure that every disk read by their system is one that they have produced. It isn't mentioned that consumer DVD burners have no capability by which to burst cut, since the lasers, which perform this operation, are too powerful and costly. Which is to say nothing of the Content Scramble System (CSS), an encryption algorithm that all DVD based formats use. Despite the compromise of this encryption algorithm, Nintendo's mechanism of using BCA verification remains intact. Sony too made use of similar tactics. It was widely believed that Sony's method was to make the assumption that most CD and DVD burners would write the correct "checksum" or error detection mechanism on disks. Rather than writing a true checksum to their PS1 and PS2 disks, they would have 0s (zeros) written instead. This was actually not the case, Sony was well aware that their system would need the power of the Patent system behind them if they were to enforce their control over these mechanisms. Rather than using the approach of Nintendo, embedding special lock and keys in their systems, Sony developed a system to encode particular 4-Character sequences on their disks, one for each of the regions in which they were distributing their consoles. This mechanism was patented, though with a name much less obvious than Nintendo's (Akiyama et al. 2001). Each company was essentially accomplishing the same goal, marking legitimate disks in a way which could later be identified, but also ensure that others could not similarly mark disks.

This entire situation was complicated by the development of "Region Encoding" or more popularly known as "Regional Lockout," which prevents the use of a given medium in a geographic location other than the manufacturers desired delivery point. This is especially bizarre in a global moment where users move from one geographic location to another, and it would seem logical that they might continue wanting video games or other media from a different "regional" area. Among early console systems region lockout was accomplished simply by changing the size and shape of connectors or cartridges. Adapter devices readily counteracted this. Much like copy-protection systems console manufacturers desired legal recourse against users who would seek to import games from locations around the world. To enforce this, they began to develop and patent systems, which would later be incorporated into other devices, like DVD, UMD, HD-DVD, and Blu-Ray media disk formats. But in many respects the video game industry birthed the practice that has now spread to other media and software distribution practices.

4.3.3 "Copyright Infringement" and Copyright Infringement by Users

To counteract these limitations, users have been willing to open up their consoles and install "Mod-Chips" which have three effects. One which allows a user to legally import a game and play it (circumventing the regional lockout) and its secondary effect also allows users to illegally "burn" or copy game disks.[48] A third and less publicized effect is that it is typically a requirement for those interested in developing games for consoles outside the networks of access. Truly "independent" game development, work disconnected from the networks of secrecy and access requires breaking the law. Companies who offered services to assist users in this process were subsequently taken to court by console manufacturers for violation of copyright (Nintendo and Lik Sang 2003; Sony Computer Entertainment and Lik Sang 2003). Oddly in these cases the copyright of the console manufacturers was not being violated, but rather their mechanisms for control were being circumvented. The issue is even foggier in countries like India where a particular console may not be available in the first place. What region is the user a part of? If they import a game console to play, they must ensure that all games they buy subsequently are from the proper region. In a well-known case amongst game developers the site "Lik-Sang" was shut down because of continued harassment from console manufac-

48. In many respects I see MOD or remix culture (Lessig 2005) embodied by these activities of hackers to be a serious complication to any notion of "closure" or "stabilization" as used in the Social Construction of Technology (SCOT) approaches to studying technology (Pinch and Bijker 1989). Even thinking of technological systems as "durable" (Latour 1991) in any kind of lasting way without connections to systems which enforce durability seems to be problematic. Of course this isn't to say that barriers are not in place, which encourage durability or stability, merely that an increased interest by users of technology in MODing or remixing them increases the complexity of the situation.

turers, despite their service to the game development community. Lik-Sang was the leading provider of adapters for console controllers so that developers could use them on PCs during the process of development. Some of the products were even being used by smaller development studios to supplement the number of available DevKits within their company. This case is also indicative of a broader problem of litigation practice, where companies with more money, despite possibly having an invalid case, can cause those that seek to interrupt or alter networks of access to simply bleed to death from monetary loss.

> Lik-Sang.com Out of Business due to Multiple Sony Lawsuits
> Tue Oct 24 2006 21:58:51 Hong Kong Time - Corporate Info
>
> OUT OF BUSINESS NOTICE
>
> Hong Kong, October 24th of 2006 - Lik-Sang.com, the popular gaming retailer from Hong Kong, has today announced that it is forced to close down due to multiple legal actions brought against it by Sony Computer Entertainment Europe Limited and Sony Computer Entertainment Inc. Sony claimed that Lik-Sang infringed its trade marks, copyright and registered design rights by selling Sony PSP consoles from Asia to European customers, and have recently obtained a judgment in the High Court of London (England) rendering Lik-Sang's sales of PSP consoles unlawful.
>
> As of today, Lik-Sang.com will not be in the position to accept any new orders and will cancel and refund all existing orders that have already been placed. Furthermore, Lik-Sang is working closely with banks and PayPal to refund any store credits held by the company, and the customer support department is taking care of any open transactions such as pending RMAs or repairs and shipping related matters. The staff of Lik-Sang will make sure that nobody will get hurt in the crossfire of this ordeal.

A Sony spokesperson declined to comment directly on the lawsuit against Lik-Sang, but recently went on to tell Gamesindustry.biz that "ultimately, we're trying to protect consumers from being sold hardware that does not conform to strict EU or UK consumer safety standards, due to voltage supply differences et cetera; is not - in PS3's case - backwards compatible with either PS1 or PS2 software; will not play European Blu-Ray movies or DVDs; and will not be covered by warranty".

Lik Sang strongly disagrees with Sony's opinion that their customers need this kind of protection and pointed out that PSP consoles shipped from Lik-Sang contained genuine Sony 100V-240V AC Adapters that carry CE and other safety marks and are compatible world wide. All PSP consoles were in conformity with all EU and UK consumer safety regulations.

Furthermore, Sony have failed to disclose to the London High Court that not only the world wide gaming community in more than 100 countries relied on Lik-Sang for their gaming needs, but also Sony Europe's very own top directors repeatedly got their Sony PSP hard or software imports in nicely packed Lik-Sang parcels with free Lik-Sang Mugs or Lik-Sang Badge Holders, starting just two days after Japan's official release, as early as 14th of December 2004 (more than nine months earlier than the legal action). The list of PSP related Sony Europe orders reads like the who's who of the videogames industry, and includes Ray Maguire (Managing Director, Sony Computer Entertainment Europe Ltd), Alan Duncan (UK Marketing Director, Sony Computer Entertainment Europe Ltd), Chris Sorrell (Creative Director, Sony Computer Entertainment Europe Ltd), Rob Parkin (Development Director, Sony Computer Entertainment Europe Limited), just to name a few.

"Today is Sony Europe victory about PSP, tomorrow is Sony Europe’s ongoing pressure about PlayStation 3. With this precedent set, next week could already be the stage for complaints from Sony America about the same thing, or from other console manufacturers about other consoles to other regions, or even from any publisher about any specific software title to any country they don’t see fit. It’s the beginning of the end... of the

> World as we know it", stated Pascal Clarysse, formerly known as the Marketing Manager of Lik-Sang.com.
>
> "Blame it on Sony. That's the latest dark spot in their shameful track record as gaming industry leader. The Empire finally 'won', few dominating retailers from the UK probably will rejoice the news, but everybody else in the gaming world lost something today." (Lik-Sang.com 2006)

While patents keep companies from using the same mechanisms which a console manufacturer uses to create unauthorized companies, copyright is brought to bear when any other mechanism compromises these methods of control. Of course when the accusation of copyright infringement is false, often the one being sued will attempt to placate the attacking corporation. Taken to its eventual conclusion, this technique seems to only lead to endings like that which befell Lik-Sang.

4.4 World 4-4: Coordinating the Local - The Ruling Relations

All of this regulation has altered the way games get developed. How could one create these games without the same technologies in place on PCs? Thus, the NES also heralded the birth of a now ubiquitous game development technology, the "DevKit."[49] DevKits were introduced so that game developers could create games for consoles where the hardware differed significantly from that of PCs. Nintendo developed technologies to bridge the gap between the PCs where code was typically written and the consoles,

49. The "DevKit" is distinct from "development kits" as defined by some authors (Postigo 2003, p. 603). There is a slippery and important language to keep in mind. SDKs or software development kits are distinctly different, though intertwined with DevKits. DevKits typically have accompanying SDKs. However, it is possible for companies to release SDKs without having DevKits. The hardware of the DevKit is in part what distinguishes it from an SDK. It is also access to documentation and other resources like online discussion forums.

which ran the code. The complexity of these devices has increased dramatically as the complexity of consoles has risen.

DevKits are also distributed with the software packages that simplify the process of game development. These range from SDKs that provide a set of software resources that developers can draw on, to software tools that combine art, code, and data into a format, which can be delivered and run on a DevKit. It also includes very basic technologies like compilers, IDEs, and debuggers, which are indispensable tools for game developers. Without these resources, the process of creating games can be much more complex. On a simple system like the NES or Game Boy Advance, hobbyists can overcome some of these limitations, but the resources typically available to developers, are off limits. NDAs accompany these technologies, which further limit game developers. These agreements prevent developers legally from distributing or sharing knowledge of or resources for these systems. While technically companies covered by the same NDA could share knowledge and resources, it is prevented by the overarching emphasis on secrecy. This means that an unconnected producer cannot create something for one of these devices; it requires the permission of the manufacturer.

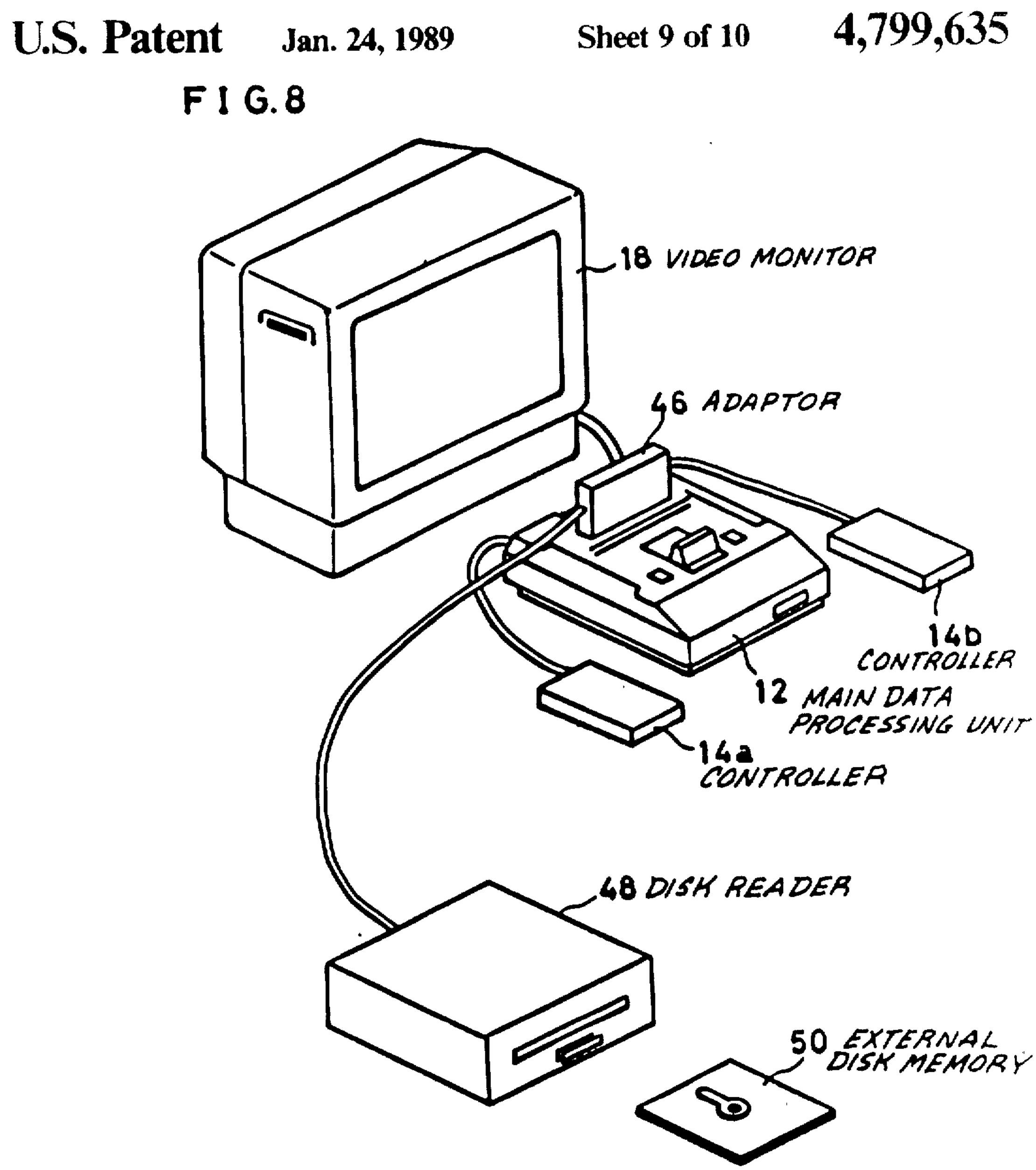


Figure 4.8: The Famicom Combined with a Disk Drive Served as a NES DevKit.

And while the following comment reflects on distribution explicitly, it does not emphasize clearly that development necessitates these kinds of technologies. Without them, games can be developed and prototyped, but massive changes will be required in the end to support the highly specific characteristics of a game console. While distribution may become more accessible, it is still structured by access to hardware platforms and development tools, as indexed by Game Studies scholars and independent game developers.

> "Both physical and digital distribution rely on independent developers' ability to make games for the new platform in the first place. While Nintendo has been quite vocal about its intention to support independent developers, including offering Wii dev kits for under US$2,000, Nintendo of America has also said that it won't start reviewing independent developer applications until January 2007—which means that only those developers with publishing contracts or special invitations actually have them." (Bogost 2006)

Even in summer of 2007 independent developers have continued to be unable to acquire these devices, despite announcements that indicate otherwise. The rhetoric used in this announcement harkens to the one used by Microsoft with the announcement of their XNA Express endeavors mentioned in World Three's Boss Fight. However, when pressed the reality seems to indicate that very little is actually changing.

Which really means that the announcement was made to indicate that Nintendo was intending to distribute independent and original games on their online network, much like Microsoft and Sony had already been doing on their network. The mechanisms for controlling production will continue for the time. So while Microsoft's XNA technologies are actually creating opportunities for developers to share resources and technologies, Nintendo's approach to production control will remain the same, despite the similarities in the rhetorical framing of their press releases.

If the ability to gain access to the technologies to develop games for these systems has been difficult for developers in established industries, it is only more complex in countries like India with emerging industries. These legal, technological, and political maneuverings unflatten the global playing field. At least while I was in India, and when I left at the beginning of 2007, there were precisely two Nintendo DS DevKits in all of India. Both were at the same company, which had recently been acquired by a rapidly globalizing publishing company. The engineers and artists were trying to get their heads around this more limited technology with SDKs they had never seen before. It was one of those odd moments where my past developing video games suddenly became useful during fieldwork. Having worked with Nintendo SDKs during the time of the N64, I quickly recognized some of the techniques and standards that were being used. For the developers in India, these were new concepts, cursorily documented and often in ways that were less than useful for developers making games. When asking questions on the private forums used for developer discussions, they frequently encountered responses like, "How can you not know this?" So I spent some of my time working with engineers offering what aged and blurry knowledge I did have to assist in their development work. In some cases it was sitting with engineers writing code. Other times it was whiteboard talk. In others it was asking artists how they were generating their art and getting it into the game. How did they play the game? This precisely illustrates the point that these structures have not only provided the ability to control production, but prevent any circulation of knowledge regarding what it takes to create games for console systems.

This also has the secondary effect of encouraging Indian game development companies to focus on the production of art assets. Because the tools of game art production, Max and Maya, are relatively standard, developers in India were already familiar with these tools and able to produce artwork for U.S. and Western European companies. However, because the tools, which would allow engineers to create code for other systems and begin generating tools for designers to bring these aspects together, were unavailable, many companies were forced to focus on those aspects most amenable to offshore outsourcing. This meant that rather than being able to bootstrap themselves into the global game industry, they were structurally positioned to act as art production houses. Because of this, many of the Indian game studios shifted their focus to the production of game titles for mobile cell phone devices, currently the smallest market within the broader game industry.

Many of the other technologies that compromise consoles and PCs have actually converged, the "lockout chip," 10NES or otherwise, perseveres. It is the main limitation between the worlds of developers and getting their games onto consoles.[50] It is the limitation, which prevents new publishers from challenging existing ones, and even when they do, the standard approach is consolidation and acquisition. The distribution of DevKits fits nicely into the way the networks have been structured throughout the video game industry. The big players lease these DevKits for large sums of money. They use them or distribute them to developers making the games for consoles. Even in consoles purported to utilize "standard CD-ROM" drives (Malliet and Zimmerman 2005), place similar limitations on developers.

Regardless, if a technology being touted as a controller of distribution or production, the net-effect is the same. These technologies and their connection with political and legal structures disable the ability of producers from using, learning from and shar-

---

50. It also strikes me as uncanny current "trusted computing" endeavors by PC and Software companies as making an appeal to the 10NES worlds of game development where you can be much more sure that a user has paid for what they are playing. "The 'trusted' part of this system is that this device obeys rules established by the copyright owner when they first make the song available." ... "The rhetoric is classic command-and-control, a far cry from the delicate balance of copyright" (Gillespie 2004, p. 241).

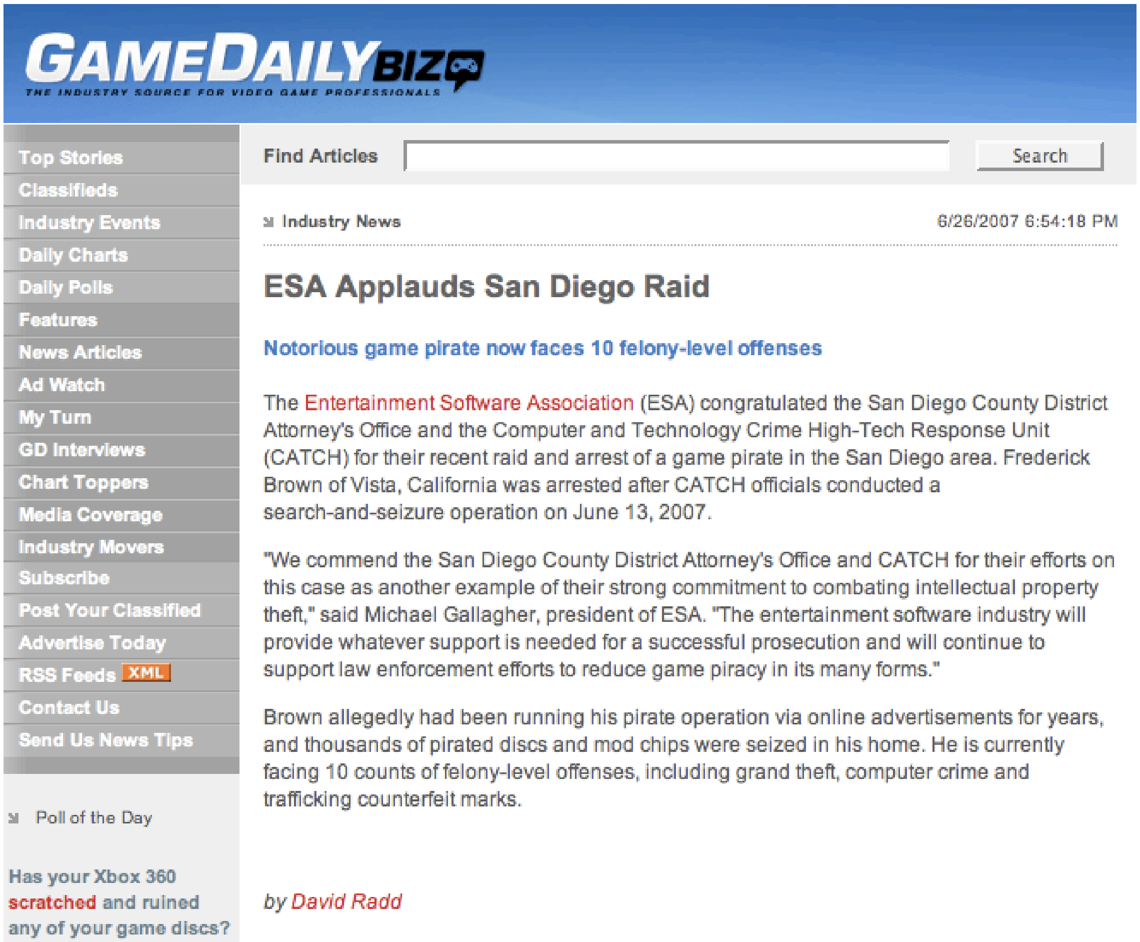
ing experiences with one another. It reinforces the idea that the work of game development is somehow different and separate. It encourages the maintenance of secret societies. While corporations scream that the market must be allowed to function, they simultaneously use the very mechanisms they protest to alter the market specifically in their favor. Only those who have been approved and are part of the networks should be allowed to speak as broadly as possible. It is an inherently anti-market approach. However, corporations cannot enforce these systems without the intervention of the state.

4.5 Boss Fight: The Coercive Moment and the Corporatization of the State

On the other hand, when the accusations of copyright infringement are true, frequently no amount of legal maneuvering by a company can dissuade those individuals perpetrating the theft of these items other than force. Which is why continually companies are looking for new mechanisms to use the State to perform those activities that they cannot. In particular, corporations have come to desire the prerogative capacities of the State. These are the only ways in which they can ensure that their networks of access and secrecy are not compromised. Media corporations in particular have become quite adept at mobilizing these forces to meet their needs, rather than the interests of users and producers, who are left to work within or break these assembled structures.

> In its protection against unauthorized circumvention, the DMCA does much more than protect digital copyright; it will be the guardian at the gates of the trusted system, ready to patrol the boundaries of this massive control mechanism. And by emphasizing access rather than copying, it can sanction violations of the trusted system that have nothing to do with copying, but are rather about accessing materials without following the proper channels, i.e. paying for it, and following the rules prescribed by that commercial relationship. CSS and the trusted system proscribe behavior in intense detail and design other behaviors out of existence, and then depend on the law to ensure that consumers use the system as

recommended, risking the threat of criminal penalty if they attempt otherwise. (Gillespie 2004, p. 244)


GAMEDAILY BIZ
THE INDUSTRY SOURCE FOR VIDEO GAME PROFESSIONALS

Top Stories
Classifieds
Industry Events
Daily Charts
Daily Polls
Features
News Articles
Ad Watch
My Turn
GD Interviews
Chart Toppers
Media Coverage
Industry Movers
Subscribe
Post Your Classified
Advertise Today
RSS Feeds XML
Contact Us
Send Us News Tips

Poll of the Day

Has your Xbox 360 scratched and ruined any of your game discs?

Find Articles Search

Industry News 6/26/2007 6:54:18 PM

ESA Applauds San Diego Raid

Notorious game pirate now faces 10 felony-level offenses

The Entertainment Software Association (ESA) congratulated the San Diego County District Attorney's Office and the Computer and Technology Crime High-Tech Response Unit (CATCH) for their recent raid and arrest of a game pirate in the San Diego area. Frederick Brown of Vista, California was arrested after CATCH officials conducted a search-and-seizure operation on June 13, 2007.

"We commend the San Diego County District Attorney's Office and CATCH for their efforts on this case as another example of their strong commitment to combating intellectual property theft," said Michael Gallagher, president of ESA. "The entertainment software industry will provide whatever support is needed for a successful prosecution and will continue to support law enforcement efforts to reduce game piracy in its many forms."

Brown allegedly had been running his pirate operation via online advertisements for years, and thousands of pirated discs and mod chips were seized in his home. He is currently facing 10 counts of felony-level offenses, including grand theft, computer crime and trafficking counterfeit marks.

by David Radd

Figure 4.9: A Screen Shot of San Diego Piracy Raid Report (Radd 2007)

Some companies take this particular interface between the State and corporation very seriously. Nintendo is one such example. Recently they have come out in strong support of the U.S. stance on anti-piracy measures. But there is an implicit conflation of actual illegal activity and potential illegal activity, and certainly no mention of the legal and legitimate uses of these technologies. What makes these efforts troubling is that like all coercive state-based efforts, they suffer from the inability to determine "the difference between false deference and real deference ... how can we distinguish compliance under force from mystification and fatalism" (Scott 1976, p. 230)? The growing mobilization of the prerogative state power on the part of corporations is impressive, as this game industry news report indicates, with new "processes" in which corporations offer their input on the "adequacy" of state intervention in the space of intellectual property rights enforcement.

> "Commented Nintendo in a statement: 'Despite the millions of counterfeit Nintendo products seized from retailers and manufacturing plants in China through the years, there has only been one criminal prosecution. Numerous factories, where tens of thousands of counterfeit Nintendo products were seized, escaped with only trivial fines or no penalty at all. And often these production sites continue to operate after products are seized. In order to avoid punishment, many counterfeiters are sophisticated and keep stock levels below the criminal thresholds and avoid keeping sales records.'
>
> "Each year Nintendo participates in the annual Special 301 process, by which the U.S. Trade Representative office solicits views from the industry and makes judgments about the adequacy of intellectual property laws and enforcement in foreign countries, including not only China, but Hong Kong, Brazil, Mexico and Paraguay as well." (Dobson 2007)

As time has gone by, the situation has become more complex. With the introduction of the Digital Millennium Copyright Act (DMCA) and the use of encryption schemes in console game systems, attempting to circumvent the limits on production has become a criminal activity (DMCA 1998). While the DMCA has come under particular scrutiny recently because of its relationship with Digital Rights Management (DRM) in the context of digital movie players, there has been very little broader public scrutiny of the practice and legislation. In many respects DRM technologies are actually a collective invention of the video game industry. Users interested in playing music files on different devices seem to have gotten the bulk of the attention[51], game content has long been restricted and many users fail to see their rights extend to game content. While this is good for the video game industry, one might posit that our inconsistency or inattention to this detail is especially problematic because it erodes the foundations of the argument. Some in the video game industry have noted this, though seem unsure what it means precisely (Fahey 2007). The "postmodern" state is not one in which the state is strictly on the decline, particular components are on the decline, as others are in accidence, and particularly those dealing with restrictions: the police, surveillance, and military action.

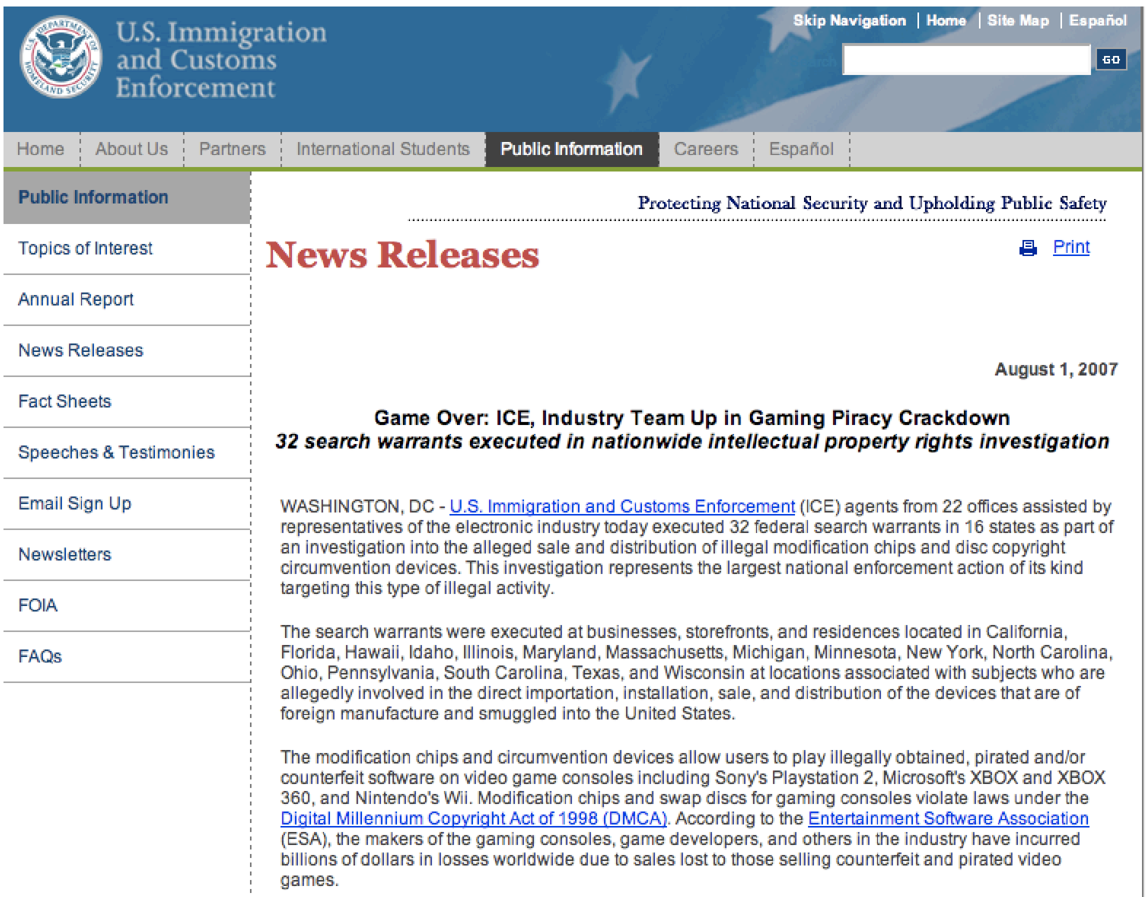
> The state, constituted as a coercive system of authority that has a monopoly over institutionalized violence, forms a second organizing principle through which a ruling class can seek to impose its will not only upon its opponents but upon the anarchical flux, change, and uncertainty to which capitalist modernity is always prone. The tools vary from regulation of money and legal guarantees of fair market contracts, through fiscal interventions, credit creation, and tax redistributions, to provision of social and physical infrastructures, direct control over capital and labour allocations as well as over wages and prices, the nationalization of key sectors, restrictions on working-class power, police surveillance, and military repression and the like. (Harvey 1990, p. 108)

Recently the same mechanisms have been deployed in a previously unprecedented manner. In tandem with the U.S. Immigration and Customs Enforcement agency, the game industry has leveraged the punitive powers of the state to execute search warrants

---

51. Since the controversy surrounding the sharing of music files on Napster, music files have dominated the public awareness of where DRM and the DMCA impact their lives (Gillespie 2007). Unfortunately the impact is far broader, and much of the emphasis remains on the technology creators rather than the copyright holders who are most often responsible for making the digital lock-down demands in the first place.

in 16 states across the U.S. Despite the possible legitimate uses of technology, the risk of possible illegal activity become the motivation for very real action. A preemptive piracy strike one can only presume. The scale of these recent actions are of particular note, verging on the military rather than on localized police actions.



U.S. Immigration and Customs Enforcement

Skip Navigation | Home | Site Map | Español

Home | About Us | Partners | International Students | Public Information | Careers | Español

Public Information

Topics of Interest

Annual Report

News Releases

Fact Sheets

Speeches & Testimonies

Email Sign Up

Newsletters

FOIA

FAQs

Protecting National Security and Upholding Public Safety

News Releases

Print

August 1, 2007

Game Over: ICE, Industry Team Up in Gaming Piracy Crackdown

32 search warrants executed in nationwide intellectual property rights investigation

WASHINGTON, DC - U.S. Immigration and Customs Enforcement (ICE) agents from 22 offices assisted by representatives of the electronic industry today executed 32 federal search warrants in 16 states as part of an investigation into the alleged sale and distribution of illegal modification chips and disc copyright circumvention devices. This investigation represents the largest national enforcement action of its kind targeting this type of illegal activity.

The search warrants were executed at businesses, storefronts, and residences located in California, Florida, Hawaii, Idaho, Illinois, Maryland, Massachusetts, Michigan, Minnesota, New York, North Carolina, Ohio, Pennsylvania, South Carolina, Texas, and Wisconsin at locations associated with subjects who are allegedly involved in the direct importation, installation, sale, and distribution of the devices that are of foreign manufacture and smuggled into the United States.

The modification chips and circumvention devices allow users to play illegally obtained, pirated and/or counterfeit software on video game consoles including Sony's Playstation 2, Microsoft's XBOX and XBOX 360, and Nintendo's Wii. Modification chips and swap discs for gaming consoles violate laws under the Digital Millennium Copyright Act of 1998 (DMCA). According to the Entertainment Software Association (ESA), the makers of the gaming consoles, game developers, and others in the industry have incurred billions of dollars in losses worldwide due to sales lost to those selling counterfeit and pirated video games.

Figure 4.10: Screen Shot of the "ICE" Press Release on the "Piracy Crackdown"

"Illicit devices like the ones targeted today are created with one purpose in mind, subverting copyright protections," said Julie L. Myers, Assistant Secretary of Homeland Security for Immigration and Customs Enforcement. "These crimes cost legitimate businesses billions of dollars annually and facilitate multiple other layers of criminality, such as smuggling, software piracy and money laundering."

...

> Between fiscal years 2002 and 2006, ICE agents arrested more than 700 individuals for IPR violations and dismantled several large scale criminal organizations that distributed counterfeit merchandise to nations around the globe. At the same time, ICE investigations into these networks resulted in 449 criminal indictments and 425 convictions. Together, ICE and CBP seized more than $750 million worth of counterfeit goods from fiscal year 1998 through fiscal year 2006. (ICE 2007)

The lack of response by potential producers more broadly is, "what passes as deference 'is ritualized and habitual' or even calculating. ... There may in fact be a large disparity between this constrained behavior and the behavior that would occur if constraints were lifted. The degree of this disparity would be some index of the disingenuousness of deferential acts. The very act of deferring may embody a certain mockery" (Scott 1976, p. 232). In many respects this may become the case amongst a new generation of media producers, those with an eye towards ironic re-interpretation or "remixing" (Lessig 2005). As more users understand themselves as capable producers, the current legitimacy of this state regulatory action will be called into question, as has already been demonstrated in ares like digital music downloads (Gillespie 2006).

> The tension between the fixity (and hence stability) that state regulation imposes, and the fluid motion of capital flow, remains a crucial problem for the social and political organization of capitalism. This difficulty is modified by the way in which the state stand itself to be disciplined by internal forces (upon which it relies for its power) and external conditions - competition in the world economy, exchange rates, and capital movements, migration, or, on occasion, direct political interventions on the part of superior powers. (Harvey 1990, p. 109)

The DMCA has further extended the ability of corporations to incarcerate people who regardless of their intentions enable others to circumvent those copy protection mechanisms, which companies create. The conflation of "hacking" and "cracking" derives from the tension between the rights of users to legally do with technology as they please ("hacking") and users illegally attempting to copy or redistribute the property of corporations ("cracking"). Which is precisely why the DMCA is so problematic, because its foundations are rooted in the assumption that anyone interested in doing something with digital data, which it was not originally intended to do, is attempting to make illegal copies. These same limitations extend to anyone interested in producing media for console video games. There is no acknowledgment that these technologies structure producers in ways far more limiting. Production of new media is only talked about vaguely as "home brew" and both denigrated and extolled in different press releases. The erroneous assumption of illegality is encourage by companies precisely because it gives them power to enforce their existing technologies and legal structures with new ones not in the interest of users.

4.5.1 "Hacking" Bowser's Castle for the Right to Speak

Returning to the division between crackers and hackers, it is important to recognize that in many ways hackers have had a great deal to offer the video game industry throughout the years. Though most industry leaders continue to conflate the difference, using a single word to reference two very different activities. The following quote from a game industry executive indicates the refusal to acknowledge the nuance between those activities that are actually crucial to their survival.

> "Unfortunately, hackers will try to exploit any hardware system software," SCEA spokesperson Dave Karraker told GamesIndustry.biz.

> "The best we can do as a company is to make our security that much stronger and aggressively pursue legal action against anyone caught trying to use an exploit in an illegal manner."
> ...
> Every hardware launch brings with it a race for hackers to defeat the system's protections, whether for the technological challenge, to run copied software, or to allow for homebrew games.
>
> Despite Sony's attempts to prevent its emergence, the PSP has a strong homebrew community - and hackers are doubtless hoping to establish a similar base for PS3. (Androvich 2007)

What is the difference between exploiting a hardware system to do what you would like it to do, and trying to use it in an illegal manner? I would answer that this is precisely the distinction between the hacker and the cracker, yet companies continue to combat them as if they were the same entity.[52] Apparently producing for a console in an unlicensed manner fits this description. There is a huge difference between homebrew software and illegally copied software. "Homebrew" software, which is developed by amateurs at home, hobbyists hoping to tinker with the devices they have already paid large sums of money for, or aspiring students and developers hoping to learn about game development practice. Though the same exploits may contribute to both activities, it seems premature to pursue both as if they were the same thing. Some companies like Nintendo largely ignore homebrew developers, Sony actively combats them, and Microsoft has embraced them.

---

52. While I treat the distinction between hackers and crackers as relatively clear, it is actually a rather complex division, even amongst hackers (Coleman 2005, pp. 53-54). In part I do so as a reaction to the alternatively muddy conflation, which is frequently tossed about in which all hackers are bad. This erring on the side of upstanding hacking activity is done with the reasonable assumption that not every hacker has criminal intent.

One story in particular stands out as an exemplary example of how homebrew development benefits the video game industry. Nintendo's GameBoy Advance (GBA) handheld system has enjoyed one of the longest lives of any console game system, in part because of its relatively low cost and large library of video games. More importantly, it developed a large community of homebrew developers who invested significant time and energy into making the system accessible and open to new developers. Even the Nintendo Dual Screen (DS), the logical successor to the GBA will play cartridges made for that system. GBA development in its later years benefited greatly from the development of an open source and homebrew project called VisualBoyAdvance[53], a project which began as an emulator for the GBA on PCs. As the project matured, so did the development tools that the software included. Built-in map viewers, sprite viewers, memory viewers, palette viewers, and visual debugging tools were all integrated into one package. Not even Nintendo had provided such a host of tools for developing games for the GBA. Even the software libraries associated with GBA homebrew began to surpass those supplied by Nintendo. Licensed developers began developing tools with "hacker" made software. While it is possible for someone to download the emulator and then download ROM files created from GBA cartridges, this is not the only possible use of the technology.

Emulators, decompilers, and numerous other technologies which might be labeled as dangerous to the video game industry, are not obviously so. Artwork is being generated based on these technologies. Visualizations of how systems work are being created. Each one of these is based on technology that at one time was considered illegal

53. While it is still possible that some day in the future this project will be dismantled by Nintendo, at this point it seems unlikely. The project can currently be found at: http://vba.ngemu.com/ with tools available for most PC platforms like Windows, Mac OS X, and Linux.

by many in the video game industry. In part this is due to the conflation of hackers and crackers. The other complication is that as far as most companies are concerned, even if you do own the ROMs that you're playing on an emulator, they would rather be able to re-sell you that content than you make use of it yourself. It also dilutes their branding. Nintendo's Mario on a Sony PS3 or Microsoft Xbox360 does not help their brand building initiatives. Even "open" consoles, which have attempted to break down these barriers for developers, has economically failed with more development work by hobbyists put into emulation than new games. In practice, emulation actually gives console manufacturers more reason to combat homebrew rather than embrace it.

4.5.2 I'm Sorry Mario, but our Princess is in Another Castle

There seems to be something particularly distinct occurring in this space, something which I believe cuts to the core about why coercive State action is being mobilized on an unprecedented scale against entirely non-violent citizen action. The (re)productive capacity of users rather than consumers is integral in this crisis for those seeking to control productive capacities. Play and playfulness is not something that corporations or the State has learned to work within yet, and as a consequence you find strong rhetoric against those that seek to leverage technologies to meet their desires rather than those they have been expected to consume.

At which point you are going to begin noticing a pattern. We have made some progress, but we are not quite there yet. There are a lot of rigid rules in which we are muddling through, very real structures that shape what gets created and by whom. We understand a bit better now about whom the players are, what kinds of objects are in our world, and the rules that run the system. In short, we are starting to get the hang of how this game is played. This is of course when we head to the next World.

PART II. MOD(IFY)-ING GAME DEVELOPMENT WORLDS

CHAPTER 5
GAME DEVELOPMENT PRACTICE: A POSTMORTEM

World Five takes the form of a "postmortem." In many respects the "what went well" and "what went poorly" format of postmortems 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, p. 22). Worlds Five and Six also mark the transition to the concluding chapters and as such these chapters move from a descriptive frame to a normative one. This World deploys postmortem articles from Gamedeveloper Magazine over the years to connect Worlds One and Two with the experiences of other game developers. These two Worlds described how elements of work/play tend toward excess, frequently resulting in collapse or a propensity for "crunch." These same modalities express themselves in an overwhelming urge towards interactivity, which when pushed too far transitions from means to end.

While there are no silver bullets to improve the numerous aspects of the game industry that informants have hoped I could offer insight, I offer four particular interventions. Two are presented in World Five and two in World Six. Each recommendation is made in the interest of seeing my informants become more effective at their work, able to pursue more what drives them, rather than what prevents them from pursuing it. I believe these suggestions provide new opportunities for creativity and collectivity rather than a "churn and burn" approach to game development. The critique and recommendations are done with the greatest sense of care for an industry that I believe has a potential to be something other than what it has become. In some respects the "reinvention" of the game industry from time to time makes it more permeable, more subject to significant change.

It is also done in the hope of fostering new levels of "flatness." I believe much of the dissertation demonstrates that the game industry is exemplary of how the New Economy remains dramatically uneven. Game developers in India will have much more to offer the game industry if they given access to technologies, tools, and the experiences of US developers. The structure of the industry currently encourages only timid steps towards this end. Game developers in emerging industries are encouraged only to partake in the aspects of game creation that industry has deemed economically necessary or safe enough to reduce the sacred secret barriers. The technological, economic, and creative endeavor that is game development is better than placing restrictions on developers due to simply locale. Developers have not reduced these barriers enough to enable game developers in new locations to learn the practice of game development. Instead developers in these locations are more limited, because the small amounts of information that are shared remain frequently embedded in embodied human social networks. For an industry which places such importance on Market solutions to problems, it seems only natural to encourage greater "real" competition, rather than artificially restrained involvement.

However, I do not see this broadening of access to game development as simply enabling "more competition," but rather new sites for new forms of game development practice. For those seeking a market solution, or with a commitment to the "long tail," that may be enough on its own. A flatter game industry is good for developers, both foreign and domestic. Expanded access means new opportunities for collaboration and cultural expression.

World Seven returns the analytic framework developed in Worlds One through Four, connecting them to the overarching argument about Creative Collaborative Practice. All of the concluding Worlds or Levels (five, six, and seven) are at their core about collaboration and the kinds of structures that do or do not support it. This particular formation of structures while specific to game developers is emblematic of knowledge work more generally in the New Economy. Creative collaborative practice has become central to the functionality of New Economy workplaces and as such, this particular configuration can offer insight into those configurations that enable or disable it.

Despite being part of the same secret society, game developers have continually muddled their ways into the same situations time and again. The continued emphasis on the uniqueness or distinctiveness of video game development enables game developers to disconnect their worlds of work from other forms of labor. Experimentation and interactivity both propel and hinder the maturation of game development work. Most importantly, however, is the continued dominion of secrecy. Secrecy plugs into numerous aspects of game development work. The lack of collaboration and sharing within studios and between studios is at the core of the industries inability to mature. The orientation towards secrecy prevents developers from connecting their islands of operation with one another; for fear that surely others are waiting to poach your ideas, your hardware, your software, etc.

It is this World that game developers must rethink in order for the work of video game development to mature. The secret society ideology must be weakened in order for broader interest and new ideas to make their way to video game studios. The desire for real-time interactive feedback must be tempered by the ability to gain experience and reflect on situations, not simply plunging forward with the next milestone breathing down your neck. Interdisciplinarity takes work and cannot be dictated by the gate-keeping demands of implementation. These are all things that would benefit from more connections to other worlds, not fewer. Connections demand new forms of openness and collaboration that go beyond those already broadly conceptualized in game studios. This kind of world requires significant change and an activist orientation on the part of game developers and game studios. Because an activist orientation is crucial to encouraging change, this progression is necessary.

5.1 World 5-1: The Secret Society of Game Developers

As detailed in World 1-1, secrecy surrounds numerous aspects of the video game industry. NDAs and corporate agreements both structure the ability for game developers to talk about their work. However, it is not simply a matter of structure that prevents developers from talking about what it takes to create games in practice. Game developers deliberately do not talk about many of their practices more broadly at the level of detail needed to learn from one another. Abstract concepts are useful once you have made games. Prior to that moment, they are merely concepts, disconnected from the activity of making games.

5.1.1 What Went Right

The President and Founder of Insomniac Games, the creators of *Ratchet and Clank* for the Playstation 2 noted how the sharing of technology between companies significantly improved their ability to make games. During a time when most Playstation 2 games were taking two to three years to produce, *Ratchet and Clank* was developed in only a year and a half.

> Sharing technology with Naughty Dog. ... Naughty Dog didn't want anything from us other than a gentlemen's agreement to share with them

> any improvements we made to whatever we borrowed plus any of our own technology we felt like sharing. In an industry as competitive as ours, things like this just don't happen. (Price 2003, pp. 55-56)

In every case that I have heard developers talk about sharing technology and ideas at a very concrete level, it has been productive for both parties. This quote comes from the experiences of developers at Insomniac collaborating with those of Naughty Dog. While not every piece of advice, code, or art is useful, the practice itself of working with other teams is a productive opportunity. It creates the chance for new avenues of collaboration or new ways of thinking about problems that have already been encountered or those they have yet to encounter. This sharing goes beyond what is typically seen at GDC or in the documentation provided to most aspiring game developers. Rather than a demonstration of how to do something as basic as rendering a pyramid to the screen, a demonstration of how the same activity might be done with an eye looking forward. In creating a video game you would not want to "hard code" every vertex (a point in 3D space) or color for every model in your game, especially considering that most models for games will have thousands of vertices. The animation and the code required to generate a pyramid using OpenGL is illustrated in Figure 5.1 and 5.2 respectively.

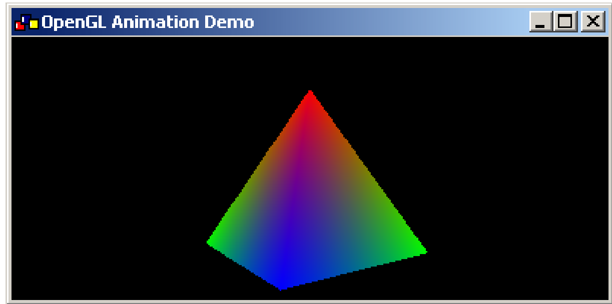


Figure 5.1: Screen Shot of a Simple OpenGL Animation Window

```
// Clear the GL Context
glClearColor(0, 0, 0, 1.0);
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glLoadIdentity();

// Get Elapsed time in Seconds Since Last Update
double rElapsedTime = wxGetElapsedTime(TRUE)/1000.0;

// Draw Some Stuff Based upon Time... (Borrowed from Nehe's Tutorial #5)
// http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=05
glTranslatef(0.0f,0.0f,-4.0f);

// If you just want it to spin, use this line.
//m_rRotationAngle += 1.0;
// If you want to update it based upon time...
// m_rRotationRate is the number of revolutions per second of the triangle.
// the default rate is being set in the constructor.
m_rRotationAngle += (m_rRotationRate*360.0) * rElapsedTime;

glRotatef(m_rRotationAngle,0.0f,1.0f,0.0f);

glBegin(GL_TRIANGLES);                  // Start Drawing The Pyramid
    glColor3f(1.0f,0.0f,0.0f);          // Red
    glVertex3f( 0.0f, 1.0f, 0.0f);      // Top Of Triangle (Front)
    glColor3f(0.0f,1.0f,0.0f);          // Green
    glVertex3f(-1.0f,-1.0f, 1.0f);      // Left Of Triangle (Front)
    glColor3f(0.0f,0.0f,1.0f);          // Blue
    glVertex3f( 1.0f,-1.0f, 1.0f);      // Right Of Triangle (Front)

    glColor3f(1.0f,0.0f,0.0f);          // Red
    glVertex3f( 0.0f, 1.0f, 0.0f);      // Top Of Triangle (Right)
    glColor3f(0.0f,0.0f,1.0f);          // Blue
    glVertex3f( 1.0f,-1.0f, 1.0f);      // Left Of Triangle (Right)
    glColor3f(0.0f,1.0f,0.0f);          // Green
    glVertex3f( 1.0f,-1.0f, -1.0f);     // Right Of Triangle (Right)

    glColor3f(1.0f,0.0f,0.0f);          // Red
    glVertex3f( 0.0f, 1.0f, 0.0f);      // Top Of Triangle (Back)
    glColor3f(0.0f,1.0f,0.0f);          // Green
    glVertex3f( 1.0f,-1.0f, -1.0f);     // Left Of Triangle (Back)
    glColor3f(0.0f,0.0f,1.0f);          // Blue
    glVertex3f(-1.0f,-1.0f, -1.0f);     // Right Of Triangle (Back)

    glColor3f(1.0f,0.0f,0.0f);          // Red
    glVertex3f( 0.0f, 1.0f, 0.0f);      // Top Of Triangle (Left)
    glColor3f(0.0f,0.0f,1.0f);          // Blue
    glVertex3f(-1.0f,-1.0f,-1.0f);      // Left Of Triangle (Left)
    glColor3f(0.0f,1.0f,0.0f);          // Green
    glVertex3f(-1.0f,-1.0f, 1.0f);      // Right Of Triangle (Left)
glEnd();                                // Done Drawing The Pyramid

// Flush the GL Pipeline
glFlush();
```

Figure 5.2: Screen Shot of the C++/GL Code Necessary to Generate Figure 5.1

Instead, a demonstration of reading the pyramid from a file, or reading an arbitrary number of points and colors from a file would be more useful for aspiring developers. Instead, pointing would-be game developers toward conceptualizing the practice of video game development collaboratively would be more productive. Encouraging engineers to think about the place of artistic and design practice. The same would be true for sites focused on artistic practice, encouraging the engagement of engineering and design practice. In the game development workplace, this is why data is read from files rather than hard-coded into the code of a game. The "m_rRotationRate" (the rate at which the 3D pyramid will rotate in space) in the above sample would likely be something a designer would like to edit, and as such would be a likely candidate to also have been parsed from design data. The model (pyramid in this case) would instead have been read from a file produced by an artist. It quickly becomes simpler for an engineer to read data produced by artists and designers than to place the information themselves into the code for a game. It is about enabling collaborative practice and giving prospective developers a sense of this cooperative environment.

Even the online manifestations of collaborative practice must be more indicative of the workplace. The answer to forum or email questions needs to foster a community rather than shut it down. Rather than answering a question with, "RTFM" (Read the F-ing Manual), "Can't you use Google," or "Search the Forums," should be gentler. "This has been covered in this previous forum topic here. If that isn't what you're looking for, try to be more specific, with what you are asking." Just because one person has managed to teach themselves does not mean that everyone has been instructed on how to teach herself or himself. Enabling this process rather than shutting it down fosters growth of the game development community.

Concrete examples are needed. In the years of postmortems available, one of the most honest and detailed comments came from a team of developers moving a PC game title they had written over to a game console. The following are two nearly identical comments made four years apart by two different development teams. The first quote comes from the Lead Programmer on Lucas Arts' *Star Wars Starfighter* for the Playstation 2, a project which took two and a half years, and the second comes from the CEO and Lead Designer of Infinitive Interactive, which created the game *Puzzle Quest*.

> It's not a surprise that most of the code suffered from a bad case of "PC-itis." I use this term to refer to programming practices that, while potentially portable to a console, are definitely not console friendly. ... For starters, we relied on the STL for all of our container classes. On one hand we benefited from a bug-free and robust set of standardized collection classes. As an integral part of the C++ Standard Library, the STL contains a powerful toolset for development. ... Unfortunately, depending on what containers you decide to use, the STL is notorious for making many small memory allocations. ... I urge those PC developers making the switch to consoles to take this lesson to heart. (Corry 2003, pp. 232-233)

> Even simple things we take for granted, such as using STL containers, can cause nightmares on platforms where they are poorly implemented. (Yes std::list, I'm talking you!) (Fawkner 2007, p. 48)

Both are extremely detailed and specific recommendation to game developers. These comments exemplify what kinds of even "generic" information can be useful to your fellow developers. But there is an unwritten aspect to this comment as well. Why would one use a "container?" A few additional lines explaining why containers are particularly useful for game developers and how they are being used would be helpful. Typically a container is a collection of objects in memory. These are dynamic constructs, which can expand or contract to accommodate various components. This might be useful for keeping track of all the models stored in memory in the game, or possible animation sequences. Information, which experienced developers instantly recognize, but is rarely explained or demonstrated for aspiring game developers. Containers can be used to store those series of vertex points and colors being read from files above. Yet the same recommendation comes four years apart.

This process takes not only less secrecy on the parts of developers everywhere, but it also takes an insistence on paying attention to the work and experiences of others. Sharing specifics of these experiences is important, but taking those experiences and learning from them requires attention by other game developers. In the end, this may have more to do with the state of the game industry developing professional practices, but many of the existing structures actually disable professional formation.

5.1.2 What Went Wrong

Even veterans of the video game game industry, like the Vice President of Operations and Development for Big Huge Games, which created *Rise of Nations* for the PC, a project which took more than three years and nearly 1800 files to create, express their frustrations over the lack of institutional memory.

> Not listening to all the other Postmortems ever printed in Game Developer. The Postmortems are the most widely read feature in Game Developer around Big Huge, and yet somehow we still managed to make many of the mistakes developers are cautioned against in these pages. (Train 2003, p. 40)

Secrecy, while also preventing the disclosure of outside influences, encourages developers to discount the experiences of others as distinctly unique from their own experiences. This developers comments indicate while they may not be directly applicable or importable into each and every studio, there is something to be learned from those experiences in the context of any game development company. This also requires that developers build in the time to address these concerns and perhaps even incentivize these practices during their development schedules.

Secrecy also hinders the passing on of information to those interested in joining the video game industry, be they U.S. or Indian game developers. Because common practices are not documented or circulated broadly, students, hobbyists, and independent developers are left to reinvent the wheel and relearn practices that ought to be commonplace in game development practice. The Lead Designer of the game *Asheron's Call*, which took four years to create, comments on the lack of experience and communication difficulties which arise between disciplinary groups in the workplace.

> Many of the employees were students immediately out of college, or even college students completing a work-study program. ... It was nearly impossible for team leads to give realistic schedule estimates for tasks, since few of us had experience in professional software development. It was also initially difficult to get different teams from the programming, art, and design departments to communicate regularly with each other. (Ragaini 2003, p. 307)

As sociological studies of science indicate, estimation in particular requires experience (Pinch et al. 1997), if developers made it clear that this was something that a developer should be thinking about when approaching tasks, it would be more transparent. An educational website could read, "Before you begin this task, estimate how long you think it will take you to complete the exercise." Being honest about practice opens it up to further discussion. If estimates are routinely ignored because they are either dramatically under- or overestimated, how does that affect project deadlines? New conversations about practice can begin.

By making the practice of game development more accessible, or "open," you have the potential to draw new people into the world of video game development. What does game development practice look like at our company? What does a designer do, and how do they do it? How do engineers, artists, and designers interact with one another? How do their efforts come together to make a game? These practices may very well interest new kinds of engagement by new groups of people. There must also be an implicit understanding of the importance of each aspect of game development work.

5.2 World 5-2: Instrumental and Experimental Work

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 as to what these technologies do, and 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 until someone has begun working in the game industry. The Supervisor on the game *Final Fantasy XII* for the studio and publisher Square Enix notes the importance of tools which provide experimental or trial-and-error approaches to design.

> 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, p. 24)

This aspect of game development is largely unknown and unexplored by those looking to enter the video game industry. Even more disturbing is that these resources are most frequently kept from companies doing offshore outsourcing work for video game studios. "Real-time previews" of game's content inside of a "game's rendering engine" is 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.

This is an "old way" of developing games, processing everything by hand in ways that require an artist (or a designer) to "hassle" someone (frequently an engineer) to see their work operating within the game. Most game developers work this way 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 amongst developers notes the difficulties that not having these tools creates. *Diablo II* took more than three years to develop and required a 12 month crunch period. The Producer for *Crackdown* on the Xbox 360, a project which took nearly four years, also notes the significant lag times created by poor experimental tools.

> 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, pp. 88-89)

> 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, p. 30)

Until experience indicates otherwise, this process remains indicative of the industry at large. Aspiring game developers and newly created video game companies continue to function this way. For developers in India, or young developers in the US, there is no "new way" until experience indicates that there ought to be. This does not however provide the kind of experimental environment, which seems to be necessary for the creation of video games. New mechanisms are necessary for this kind of work.

5.2.1 What Went Right

Flexible technologies have become the new key component of game development. They offer the ability to alter characteristics of a game by artists and designers. While engineering must frequently create these tools, it is at the core of what makes game development practice "work." The Lead Engineer on *Battle Engine Aquila*, a game which was developed for the Playstation 2 and Xbox over the course of two and a half years notes the importance of flexible and modifiable systems.

> 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, p. 51)

At their best, flexible technologies provide members of a game development team to work independently enough that they are not constantly dependent upon the work of others to continue progressing with their own work. The ability to experiment with ideas during the constantly shifting set of properties that make up a game is essential. In the end it becomes good design practice; the ability to expand or contract game components without the requirement of intervention on the part of several people. An individual can experiment with one component while others do so simultaneously.

These technologies also interface with the numerous disciplines that birth them, in game development studios. Without engineers who interface well with designers and artists, you will end up with mutant 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 three years and 4,000 files to create, suffered similar 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, p. 26)

These tools are, however, frequently written in the "spare" time of engineers, when other demands are not being placed on them to provide basic game functionality. These tools, if poorly designed, become hazardous to the health of a project. More 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.

5.2.2 What Went Wrong

But even when tools function properly, the ability to experiment, if taken too far, becomes hazardous. It is the limit of experimental systems that becomes problematic for game developers. At some point limits must be placed on projects and experimentation must be given direction. Again, the Lead Engineer on *Battle Egine Aquila* and the Project Manager on the game *Resistance: Fall of Man*, one of the release titles for the Playstation 3, comment on the double-edged character of these tools.

> 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, p. 58)

> 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, p. 35)

Interactivity and experimentation taken as a goal disconnected from the broader aims of a project, or without a plan or trajectory, proves 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 are derived.

Interactive work has limits. People must have time to work and think about their tasks. While 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*, 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.

> 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, p. 46)

This is the core problem of interactive and experimental systems. Both at 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 based on those experiences. Instead meetings are scheduled, instant messages pass back and forth, emails 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, become particularly detrimental.

5.3 World 5-3: Playing Interactively Cross/Intra/Interdisciplinarily

Interdisciplinarity is at the core of video game development, and as such it needs to be presented this way, not merely as a world dominated by engineers. The broader imaginary about what game development work is and who game developers are shapes who becomes interested in making games, and thus the kinds of games that are made. This interdisciplinarity is tempered by the ability to communicate and work across disciplinary divides, a process that has been aided by the creation of new categories of specialization, tools engineers and technical artists. This can be difficult in young, small, or studios just starting to explore the work of game development. The importance of these new categories of specialization is rooted in experience.

As game companies grow they rediscover the importance of this process. Yet more broadly it is not communicated or expressed that it is an integral component of game development work. Despite rosy collaborative pictures like the one painted below, interdisciplinary work takes time and the acceptance that the ideas brought by each area of expertise is worth considering. If any one component of the collaborative effort is unwilling to recognize this, the system breaks down. The Project Manager on *Resistance: Fall of Man* comments on the importance of interdisciplinary collaborative practice and how it was something that their studio, Insomniac, had to continually work at during the more than two year development cycle.

> Insomniac grew from a company of 40 people to around 160 in a few short years. In order to keep the business running smoothly, a new layer of management structure was introduced, which worked surprisingly well. But Insomniac quickly became more departmentalized. People began to focus more on the needs of their department than how their department related to the ultimate goal: the game. ... In a world where we deal in the qualitative rather than the quantitative ... By the end of the

> project, it was common to see animators sitting next to gameplay programmers, going over get-hit timings and whatnot. In a collaborative environment where each person brings ideas for improvement and innovation, getting the right people together is the key to creating quality. (Smith 2007, p. 36)

Many game development studios remain departmentalized, which provides collaborative resources within disciplines. Both interdisciplinary and intradisciplinary collaboration is productive given different circumstances. The key is finding the time to pursue collaboration cautiously. Interactivity can be cautious, rather than plunging blindly ahead.

5.3.1 What Went Right

The ability to interactively work across disciplines to solve problems was almost universally cited by informants as a "useful" or "necessary" aspect of game development. Fast cycle times and feedback loops that allowed small groups of developers to rapidly find solutions to problems, which they faced in the development of a game were found productive. The Project Lead of *Bioshock*, again, notes how interdisciplinary collaborative teams were crucially important throughout the development process.

> Over the course of development, we created multidisciplinary strike teams to work on a wide variety of problems, including AI, animation, visual effects, and cinematic. The results of those teams were universally better than the previous non-iterative process. (Finley 2007, p. 24)

At the same time, this process, when under pressure can result in reckless cycling, or rapid changes that result in a more chaotic structure. The ability to "iterate" on a problem with a team is productive, but when that iterative structure is put under immense time pressure, it becomes less productive.

5.3.2 What Went Wrong

As noted in my fieldwork, "iteration" and fast feedback loops when disconnected from those structures that attempt to keep them under control are bypassed, unpredictable results may occur. An engineer working on Microsoft's game *Age of Empires*, developed by Ensemble Studios commented on a phenomenon very similar to experiences of my informants noted in World Two's Boss Fight.

> The lead is the go-to person when someone outside has new requests for the team. As the development of AOE progressed and the pressures rose, adherence to this system broke down as people went direct to get their needs filled quickly. We paid a price for it. People didn't know about programming changes or new art that was added to the game, and the level of confusion rose, creating a time drain and distraction. We all had to stop at times to figure out what was going on. (Pritchard 2003)

Interactive interdisciplinarity disconnected from those sites where knowledge about the safety valves between feedback loops can result in system failure. Emergent forms of structure within these groups must be considered as important as the more rigid and formalized structures. These forms must also be communicated amongst team members, a process that is frequently neglected when time pressures are imposed, disconnected from knowledge about where a team is, or how it is functioning.

It is also difficult when small number of individuals working at the margins of disciplines, the tools engineers, technical artists, and leads, are not provided with the human and time resources necessary to prevent breakdowns between groups. Again, the Project Manager from Insomniac Games comments about the tendency towards excess these systems can exhibit.

> Adding to the confusion, only a small number of programmers had the knowledge required to debug the problems, and these people were overwhelmed with requests for help. If it weren't for their inhuman effort and long hours hunched over keyboards, we would have never hit launch date. (Smith 2007, p. 35)

Unfortunately this is frequently solved by asking or assuming that employees will stay late to make up the slack created by overwhelmed feedback loops, ineffective interactivity, interdisciplinary breakdowns, or disrespect of emergent forms of structure. Time spent doing these tasks is seen as separate from the actual work of making a game. This can no longer be the case if the work of game development really does include this interdisciplinary work, then it must take into account the structure and process of game studios.

5.4 World 5-4: Questing after Process

While "process" generically conceptualized has been the "solution," which many developers have arrived at, "scrum" has become the most common. However, much of what makes game development process particularly difficult is rooted in secrecy, the fetishization of interactivity, and the interface with external demands made by publishers and console manufacturers. Scrum or any process will remain inherently limited without mechanisms for institutional learning. The Producer of the game *Crackdown* notes how the constant movement from platform to platform created significant difficulties during the development process.

> Over the course of its four-year development Crackdown moved from PC (where it was prototyped), to Xbox (where it was initially intended to stay), back to PC (in preparation fro move to Xbox 360), to Xenon Alpha, then Xenon Beta, and at last to Xenon/360 Final. Even on the final hardware, we continued to take hits from significant system software

> updates every few months. When at last the platform stabilized during the last year of development (post hardware launch), development efficiency increased massively. (Wilson 2007, p. 29)

Process can improve difficult situations, but it cannot account for the myriad of other influences that shape the worlds of game developers. More importantly scrum requires localization at the level of each studio. It is a conceptual framework, which must be worked out by each studio and must frequently be modified for each project. The continued norms of secrecy demand that every game developer be able to experimentally figure things out on their own, make the broader success of projects difficult for game developers. While scrum offers promise, it must interface with numerous other aspects of the video game industry on a daily basis, just as game developers must.

5.5 Boss Fight: Forging More Connections, not Fewer

More than anything, the game developers need to rediscover the importance of sharing and collaborating across corporate divides. They need to reinvigorate their ability and desire to write, talk, and share details of their work, which they take for granted. In the nearly 30 years of video game development, game developers have not managed to share more broadly the reality of their work practice, despite the demands that people entering the game industry already be aquatinted with "making games."

This individualistic barrier will require developers to give up on the secret society. Game developers can no longer afford to perceive themselves or those around them as independent rock-stars. Indeed, many of them are extremely intelligent and hard working, and deserving of recognition for their creative work. The current system tends to recognize too few, favoring the recognition of a few figureheads. With these changes, more developers can be recognized as integral to the process.

This will also necessitate change in how game development companies recruit new talent. Rather than demanding that aspiring youth "break into" the game industry, there must be mechanisms by which they can be encouraged into the industry. "Talent" and innovation must be fostered rather than demanded from the outset. If developers allow themselves to maintain the existing structure where only those who have already "figured it out" are authorized, they will continue to get more of the same. Many have come to enjoy the caché of working in the video game industry, which may not entirely disappear. Instead, developers must become more accessible. They must begin to share more publicly, and more with one another. They must begin to think of themselves collectively rather than as individuals or individual studios. A sense of "the profession" and culture of the game industry must become something that people actively engage with and consider.

5.5.1 Defying the Cult of Secrecy

The game industry needs to become more open. Ultimately this must occur both at the lowest and highest levels. Game developers must be able to converse broadly about the practice of game development. Publishers and manufacturers need to be able to differentiate between talking about how one goes about making games and "giving away" a game. Many software companies have made numerous aspects of their work and work processes available online to foster a community of practice. The important difference is that for game companies this goes beyond releasing the "source code" of a game. It is also about how artists and designers went about creating and working within the source code of a game. How they created content and data, which then results in a game. Samples of real data that artists worked on and their process to get it into the engine should be provided. Designers should be able to document and explain how data was combined with artistic assets and how it mobilized the source code to create a game.

Much of this information already exists, in studio and corporate Wiki sites. The argument that developers simply do not have enough time to do this work is simply incorrect. They already do it internally. Sharing the information more broadly can only make them more effective at the practice of game development. Portions of their internal Wikis can be released more broadly, perhaps delayed until after the release of a game title. These Wikis can serve as the foundation for fostering new developers interested in working with those practices. Young developers interested in becoming part of these game studios can become the intermediaries between the company and its community. Rather than seen simply as a training or proving ground for new unpaid talent, it can be used as a space for developing collaborative skills amongst new paid developers. Learning how to use the tools and work with others both in and outside the company.

This same process will also encourage a broader understanding of what goes into making games. Artists, designers, engineers, and managers, who already participate in the generation of these resources, will make the work of game development more visible. The cross-disciplinarity of the endeavor can be made explicit. The importance of process, tools, communication, and collaboration can be made clear. The imagination of game developer as computer scientist will no longer reign supreme.

Teams will have the opportunity to make the numerous design decisions and their impacts visible. Making the impacts of sudden shifts of scope or design dictates from other interests more transparent can provide insight into the world of game development. Collectively this information may encourage developers to work with particular manufacturers and publishers in favor of others that detrimentally affect the working worlds of developers. Transparency may also help publishers and manufacturers understand why developers are resistant to dictated shifts or changes. Improved visibility could provide publishers and manufacturers insight into when and why studios or development teams are not moving forward successfully. Transparency cuts numerous directions.

Transparency will begin to demystify the game development process. New conversations can begin about these processes, ones which are explicit and clear, rather than general and vague. Companies can discus aspects of game development, which have remained closed. Aspiring game developers can uses these insights in their own quests to create video games. Rather than making the same day-to-day mistakes or misunderstanding how the process functions, they can learn from some of the lessons of the nearly 30 years of video game development history. More than anything, opening up will encourage game developers to think of themselves in a broader collective context, rather than individuals and individual studios scrapping against all odds against their fellow developer.

The unlocking of the video game industry creates new opportunities for creativity and entrepreneurship from new locations around the world. If U.S.-based game development companies are willing to work with globally located companies for their artistic production, then these companies should be provided the access necessary for further engagement.

5.5.2 Standards, Tools, and Practices

The game industry needs standards put in place. Having standards does not mean that individual companies cannot go above and beyond those industry standards. This is not to suggest that only Nintendo, Sony, Microsoft, or some other large, well established company must be the only player in town. This is the reason analysts have shut down these calls in the past, through misinterpretation or simply poor "analysis."[54] There needs to be some base level which is approachable and deployable broadly. This is where Microsoft's XNA efforts fall short. The continued lock-in to proprietary systems like C# and DirectX contradict the call for standardization. There are numerous "industry standards" that end up competing with other proprietary standards. Developers frequently pick and chose based upon those with the newest features rather than on those that have open standards. Continual secrecy is chosen over openness. Ultimately this recommendation is dependent upon implementation of the first recommendation. Developers have to be willing to take a stand and support those more open technologies, which ultimately will improve their ability to do game development work. Hegemony works in multiple directions. Counter-hegemonic projects can force accommodation to their demands.

From a more open foundation, new tools, practices, and processes can be developed. This is precisely where Free/Libre and Open Source Software (F/LOSS) has proven its ability. F/LOSS has dramatically influenced much of the broader software development world. Because the game industry has assumed itself different, it has not yet been able to grasp the importance of openness in the ability for companies to push their

54. Periodically a call for some kind of standardization goes out in the game industry (Waters 2007). The response is typical, "I do not think the industry will ever resort to one console. It would be bad for the industry. I could understand the argument for a single development standard, but not a single hardware standard. ... However, in regards to a single console, it would hinder innovation and consumer choice." Standards are instantly conflated with a single console, which is not the point. Either that or the idea is instantly tossed out as impossible, "The fact is, as long as Sony and Nintendo are alive and kicking, one platform will never happen. If there were one platform, the manufacturer would have all of the leverage, unless it offered open architecture. Not likely. Nintendo and Sony [would] both insist upon a proprietary standard. Microsoft has a proprietary online business. It sounds wonderful, but so does world peace" (Wen 2007).

technologies, practices, and stability further. F/LOSS in particular has proven itself adept at supporting unified technologies across numerous hardware platforms and at the production of resources that developers can draw on to reduce their dependence on self-developed code. The constant recreation of software often results in bugs or deficiencies due to the process of software development. F/LOSS has made scalability a particular concern, one that the video game industry could dramatically benefit from. Scalability, the ability for similar APIs to work across numerous devices ranging from full-scale computers to very limited custom hardware has been a core aspect of the movement.

Ultimately much of what game development companies pay for now as "Middle-ware" is software, which could be more effectively developed in an open and cooperative manner. This would also allow developers to work with similar tools even in locations where licensing agreements are unattainable because of cost or due to restrictions by hardware manufacturers. However, they would be able to retain more of their work when licensing was provided. What is at stake is not the "giving away" of free DevKits, but rather the opening up of the SDKs that allow developers to create games for these systems. Production pipelines for designers and artists will emerge around these SDKs. Console manufacturers will have to provide some mechanism by which games can be tested without DevKits (these devices are already being provided outside the law, and will be used in the implementation of the game described in World Eight). Surely aspiring developers would be more likely to purchase "approved" Nintendo, Sony, or Microsoft version of the device. Developers want to be included in these networks. When a developer has gotten approval to move forward with a manufacturer or publisher, then developers can then benefit from agreements that make the more "interactive" DevKits available.

The practice of video game development needs standards. These are embodied in tools and practices, neither of which have been stimulated in the current setting. The continued opacity and closed character of the industry has lead developers to continually forge the same path over and over again. Having once traversed a course, developers have been unable to share in detail these routes because of restrictive NDAs and licensing agreements. Console manufacturers and publishers continue to hold the keys technologically and legally to the means of distribution, and will continue to do so. The obsession with control over the production of games is a relic of the past that only does a disservice to developers and the industry. This must change for the industry to truly "mature." The maturation of the video game industry goes hand in hand with improvements in Quality of Life (QoL) and sustainability; the major concerns which occupied the minds of my U.S.- and India-based informants.

Perhaps more importantly, game studios need to learn how to "carry string with them" during the construction of their labyrinths. This process has no technological fix. It is a deep-seated social component that necessitates taking time to reflect on the activities of game development, and not simply at the end of a game title's development. Slowing down long enough to talk about how and why something should be done gives development teams a better understanding of why they are progressing down one path and not another. More connections, detailed sharing, and openness will result in more mature video game studios able to work within the numerous structures of access. Ultimately however, for this kind of revised video game development work to be possible, it will require significant change in the numerous structures that shape the worlds of video game developers. Demands that developers must make themselves. They are the ones, which work to make video game titles possible, and for too long have allowed the forces of the video game industry more broadly shape their worlds. These are castles that must be hacked and MODed.

CHAPTER 6
THE GAME INDUSTRY GALAXY: A POSTMORTEM

The connections between video game development work, the video game industry, and the political-economy more broadly are the focus of this World. Similar to World Five, it takes the form of a postmortem, though more has gone wrong here than has gone right. In many respects it seems remarkable, given the secretive context within which game studios reside, that so many games get developed each year and such significant dollar amounts exchange hands. The limitations which the industry places on itself and those who must work to create the games that sustain it build upon and sustain a culture of secrecy. Secrets are extended and enforced through inter/intranetworks of relations. Companies deliberately conflate distribution and production in the effort to ensure the access restrictions of those networks. Copyright and patent law are deployed and extended in an effort to mobilize the force of the State as one more nail in the coffin of collaboration. Corporations simultaneously praise the market and subvert it with these same machinations. Even those that attempt to break out of the mold do so only as far as it entrenches their vested interests.

Troubling from an analytic perspective is that the majority of game developers only perceive and discuss a small number of these institutions as being ones that impact their worlds. In part because this structure extends and draws on ideas about the desirability and distinctness of video game development work from other forms of production in the New Economy. This is precisely what has prevented long-term change and growth of the game industry, as is demonstrated by the repetition of mistakes and lack of institutional development over the last 30 years. While the focus tends to land on relationships with publishers and manufacturers, game developers must refocus some of their critiques on broader political-economic issues that also structure their worlds. The relationships with publishers and manufacturers are enabled and reinforced through their connections with these structures; constructions that game developers are capable of influencing.

6.1 World 6-1: The Good, The Bad, and the Ugly Inter/Intranetworks

The most apparent structures that game developers encounter on a regular basis are their connections with publishers and console manufacturers. These relationships are heavily managed and controlled. While these networks of (in)access frequently delineate those studios that have proven their ability to develop video games, it does not indicate that those who fall outside lack those capacities. The Lead Designer of the game *Puzzle Quest*, notes how despite the breakout hit of the PC based demo, they faced constant roadblocks from publishing companies. The game has since been ported to the Nintendo DS, Sony Playstation Portable, and Xbox 360's Live Arcade. A game that almost didn't make it out of prototyping because of publisher resistance has gone on to be hugely successful.

> No matter how much experience we had developing games, and no matter how many previous titles we could show off that had won Editor's Choice awards and 90 percent review ratings, no publisher wanted to deal with a PC developer in the console space until we had a game that was about 75 percent complete and that we could demonstrate. (Fawkner 2007, p. 42)

The ability to participate in the broader video game industry is structured by (in)access to console hardware and distribution markets. The console space dominates the imagination of game developers in part because of the massive amount of money that circulates through it, but also because it is a social indicator of having breached the intranetworks of the video game industry. It is a status marker. Despite a game studio's success at creating games, its access to these worlds may be severely restricted. These relationships can provide resources game developers can draw upon. They can also be used to constrain and push developers into situations where destructive relationships internally and externally wear down the talent and drive of those working on games.

6.1.1 What Went Right

In some cases the relationships between publishers and developers can be quite productive. This is often the case when vision and direction are communicated frequently and accurately between developer and publisher. It requires an honest working relationship between studio and publisher. Publishers will make requests that developers must be willing to refuse or qualify to clarify the game's progress is at a given moment and the impact changes will have on that progress. Otherwise, the situation can, "only end in tears." Project Manager for Insomniac Games talks about the importance of managing the demands for demos by console manufacturers and publishers can dramatically shape the daily lives of developers on a project.

> Losing time due to hacking demos can only be diminished by defining internal deadlines and coinciding focused demos with these. By focused, I'm talking about demos with clearly communicated (both internally and externally) goals. Everyone working on the project as well as the publisher (or whoever will be seeing the demo) knows what will and will not be included. Word of advice: Don't demo elements of your game that aren't ready. It can only end in tears. (Smith 2007, p. 35)

Demos embody a problematic aspect of the game development process. As previously mentioned, demos are often referred to as "vertical slices" of a game or polished and complete "slices" of a game that provide a visual and playable sample of the progress of a game's development. However, these slices often require more to be done than frequently can be completed. Because publishers control access to the intranetworks that game developers have invested so much time and labor in gaining entry to, rather than respond realistically, they push to have game components complete earlier than is actually possible. When sections are not available, they are added hastily, and frequently in ways that will be unusable as development progresses. The Executive Producer for the critically acclaimed game *Psychonauts*, which was developed over four and a half years and made use of nearly 3,500 files notes how publisher relationships can dramatically shape daily work worlds and the necessity of "crunching."

> One early publisher milestone required that we demonstrate multi-pass effects before the renderer was completed. In another case, it was only after a milestone had been submitted that we learned of content that was required before the delivery would be approved and a payment released. (Esmurdoc 2005, p. 32)

While publishers exploit the position of game developers, independent, and third party studios in particular, demanding more functionality than was agreed to, or to view aspects of the game not yet completed, it becomes the responsibility of producers, managers, and studio heads to protect their employees from these situations. While a publisher can conceivably withhold payment, managers must learn to push back against these tactics. Managers should indicate that the publisher is risking the loss of people, time, and money by making demands that were not agreed to in the contracts. Contracts must be made clear and game developers must learn to share more with one another and collaborate about how they have been (un)successful with any given publisher. For studios owned by publishing companies, the task is simpler, management must communicate the desires of publishers to their workers, and pay attention to what is transpiring in their own studios. They must respect the knowledge at each level of understanding in their organizations, following the flow of information between the emergent structures of individuals. They must talk to producers, leads, and rank and file developers. Yet, managers must do so in ways that leave employees with enough time to work and reflect on their approaches to tasks.

Ultimately, however, this process is hindered by the (in)ability for game developers to communicate and learn from one another. The constant need for developers to solve problems on their own, disconnected from the work of others, makes it impossible for publishers and developers to enter into relationships that are mutually beneficial. Rather, publishers end up partially funding learning curves, which are not inevitable or always necessary. Certainly there are times where "mistakes" or "course corrections" will be necessary, that is part of an experimental process. Though as developers frequently note this is not a part of the experimental process, but rather one based on access to networks being opened or closed. During the development of *Psychonauts*, one event caused the entire project to move from the Xbox, to the PC when licensing was pulled, and then later ported to the Playstation 2.

> In February 2004, at what seemed to be our peak productivity, a time when we felt most confident about shipping on schedule, Microsoft decided to discontinue its development of Psychonauts. Microsoft had funded years of mistakes, course corrections, and learning curves, but it drew the line at underwriting the remaining game development [now] that Double Fine was finally on track. (Esmurdoc 2005, p. 32)

However, developers and publishers must both be honest about the current status of their work. If a course correction is necessary, developers must explain what they have learned, and if possible document and communicate those experiences more broadly. Those reflections will ultimately encourage growth and learning in the game industry.

6.1.2 What Went Wrong

Publishers and manufacturers must recognize that ultimately they are in positions of power that allow them to make unfair demands of developers and development studios. Those demands have direct impacts on the working lives of game developers. Game developers need to help them understand how those actions affect their worlds. If a publisher or manufacturer's demands cause massive crunch in an office and endanger the success of a project, it must be communicated. Especially when publishers control hardware platforms and access to networks, frequently the intersection of these conditions have the most direct consequences for rank and file employees.

> Yet with mounting pressure from the publisher to get on Xenon Alpha hardware (in itself a mistake given its tenuous relationship to the Xbox 360 proper) (Wilson 2007, p. 30)

> The new publishing terms meant foregoing additional planned hires without the benefit of scaling back the design. The convergence of these factors led to the most insane crunch I have ever witnessed. (Esmurdoc 2005, p. 32)

While it is true that publishers take significant risks and losses on many video games, their gains significantly outweigh their losses. In many respects the losses are contributed to by the very networks of (in)access they so actively police. Broader maturity will make risks less expensive. Publishers currently externalize most risk by depending on independent developers to create new IPs, which are then purchased through acquisitions. This is a far more costly practice than fostering the development of IPs internally. The risk is borne by those least able to pay. Independent developers, excited about the pursuit of new game titles will work much harder in an effort to create their games, though frequently they make the same mistakes that have been made time and again in game studios owned and operated by a publisher.

Because publishers and manufacturers control the networks of (in)access, they are often viewed as the "bad guys" of the video game industry. Yet publishers are one of the few corporations in the game industry that have the power to encourage change by console manufacturers to further open network access. While many focus their ire on the perceived risk-averse character of publishing companies, it is only the most apparent aspect of the (in)access story. It is more broadly rooted in institutional practices, technologies, and legal structures.

6.2 World 6-2: Disentangling Distribution and Production

The (in)ability for game developers to learn and share information about game production practices severely limits the capacity for the industry to mature. This is largely due to the restrictive practices surrounding the tools necessary for the production of video games. NDAs prevent developers from sharing source code, tools, or information about how to navigate and apply these devices. These NDAs are disguised under the category of "licensing" at the level of the publisher and console manufacturer. Ultimately it is the manufacturer who demands that information about the console not be shared more broadly, though these agreements can also be passed down from manufacturer, to publisher, to developers. So while the following SEC filing excerpt indicates the risks associated with working in relation with these licensing agreements, it neglects to indicate the effect that these agreements have on the practice of game development more broadly. As even Securities and Exchange Commission filings note, the position of development studios and even publishing companies is dramatically affected by the control of hardware platforms.

> The video game hardware manufacturers set the royalty rates and other fees that we must pay to publish games for their platforms, and therefore have significant influence on our costs. If one or more of these manufacturers adopt a different fee structure for future game consoles, our profitability will be materially impacted.
>
> In order to publish products for a video game system such as the Xbox 360, PLAYSTATION 3 or Wii, we must take a license from the manufacturer, which gives it the opportunity to set the fee structure that we must pay in order to publish games for that platform. Similarly, certain manufacturers have retained the flexibility to change their fee structures, or adopt different fee structures for online gameplay and other new features for their consoles. The control that hardware manufacturers have over the fee structures for their platforms and online access makes it difficult for us to predict our costs, profitability and impact on margins. Because publishing products for video game systems is the largest portion of our business, any increase in fee structures would significantly harm our ability to generate revenues and/or profits. (Electronic Arts 2007, p. 53)

Concern about cost structures over production practices is indicative of the fundamental disconnect the game industry needs to make between production practice and the secrecy which surrounds those devices. Instead of attempting to make significant modifications to these systems of relation, developers who have gained access to these production networks willingly trade their ability to share and learn for an opportunity to make more money and produce another game title. The continued dominance of the console game market, which represents the "largest portion" of the game development business, also symbolizes a critical point of access, which must be examined. Even online distribution networks, which remove a significant aspect of the risk associated with publishing game titles, remain closed and unavailable except to those companies that work within the licensing agreements of console manufacturers.

6.2.1 What Went Right

The intention of production control, at least initially, was to control the quality and supply of games entering the market, as well as a means to supplement the costs associated with selling console hardware systems at a loss. The lack of standards and access to the resources to produce or distribute game titles for a system allows manufacturers to prevent "undesirable" material from appearing on or being played on their consoles. It provides the opportunity for brand management. It forces companies to abide by market driven solutions to the ratings of games. A console manufacturer will not distribute an unrated game even if the producer has managed to bypass the production control mechanisms implemented in the console, a process that if distributed will now carry criminal consequences under the jurisdiction of the DMCA.

There is an aspect of quality control which console manufacturers are able to exert over the games playable on their console. Final quality assurance and testing of games is executed by the manufacturer. It is out of a concern of maintaining the image of the console's brand rather than out of care for the kinds of games being positioned on the console. But the key here is distribution. Distribution and production have become entangled in the current game industry regime. The distribution networks have been disciplined by patent and legal structures. The reason production remains so tightly controlled seems problematic. One of the three console manufacturers, Microsoft, has recognized this disconnect and opened up a partial production path to the public. A strong community has risen up in this newly opened space, and developers have begun to share tools and practices more broadly. Unfortunately, this production path is exclusive to Microsoft's console. In opening up game development on their console, they have closed off the possibility of opening up those paths across console devices.

6.2.2 What Went Wrong

Developers gaining access to these networks must frequently take a game that is already significantly developed and attempt to move it to consoles. This cannot be accounted for earlier, because the production mechanisms are closed, open only to the select few authorized by the manufacturer or publisher. Even when a developer is authorized to move a game to a new hardware platform, individual idiosyncrasies make a massive difference, as a Lost Toys Studio's engineer notes.

> Our code structure was aimed toward making the porting process as painless as possible, but we hadn't counted on the extent of the limitations of the console platforms relative to the PC.
>
> ...
>
> The Xbox port of the game had the advantage of being based on DirectX, and hence the majority of the code was shared with the PC version. The Playstation 2 port, however, required an entire graphics and sound engine to be coded from scratch - a mammoth task for our two Playstation 2 programmers, one of whom had never actually written any code for the machine before this project and was still supporting a significant amount of code on the PC tool-chain and Xbox sides of the project. (Carter 2003, p. 56)

Ultimately this process impacts developers more than publishers or manufacturers. In the end, developers must re-create what they have already made, or could have already accounted for retrospectively and under significant constraints made by other corporate entities. Processes that every game development studio must create are re-made and re-learned constantly throughout the process of development. Developers frequently talk about processes that are not unique to their game, like "baking" data, a process that in many respect could be reasonably standardized, but has not.

> The content baking process for the console was time-consuming and difficult to troubleshoot. Frequently the only way to either identify or resolve a bake problem was to re-bake at the cost of up to an hour of work, and if the tools were actually broken in some way, it would take at least another bake cycle to be able to work effectively again. (Finley 2007, p. 26)

Publishers and manufacturers make no effort to encourage sharing or collaboration across the industry; even across studios they own and manage. Manufacturers have no incentive to make the process of game development flow more smoothly. Too many developers are throwing themselves at publishers and manufacturers hoping to breach the gates of access networks. Established developers cannot mobilize efforts on their own for risk of criminal violation of the DMCA, violation of NDAs, or for fear of being cut out of the networks, which allow them to work in these spaces.

The entanglement of production and distribution significantly limits the game industry and ultimately affects the working lives of game developers worldwide. Sharing and collaboration take a back seat to simply keeping a development studio alive and functioning within the limits placed on production.

6.3 World 6-3: The Breakdown of Copyright and the Patent Systems

A significant problem exists when corporate SEC filings begin to indicate as a key "risk factor" the litigious character of existing U.S. copyright and patent systems.

> If patent claims continue to be asserted against us, we may be unable to sustain our current business models or profits, or we may be precluded from pursuing new business opportunities in the future.

> Many patents have been issued that may apply to widely-used game technologies, or to potential new modes of delivering, playing or monetizing game software products. For example, infringement claims under many issued patents are now being asserted against interactive software or online game sites. Several such claims have been asserted against us. We incur substantial expenses in evaluating and defending against such claims, regardless of the merits of the claims. In the event that there is a determination that we have infringed a third-party patent, we could incur significant monetary liability and be prevented from using the rights in the future, which could negatively impact our operating results. We may also discover that future opportunities to provide new and innovative modes of game play and game delivery to consumers may be precluded by existing patents that we are unable to license on reasonable terms.
>
> ...
>
> Other intellectual property claims may increase our product costs or require us to cease selling affected products.
>
> Many of our products include extremely realistic graphical images, and we expect that as technology continues to advance, images will become even more realistic. Some of the images and other content are based on real-world examples that may inadvertently infringe upon the intellectual property rights of others. Although we believe that we make reasonable efforts to ensure that our products do not violate the intellectual property rights of others, it is possible that third parties still may claim infringement. From time to time, we receive communications from third parties regarding such claims. Existing or future infringement claims against us, whether valid or not, may be time consuming and expensive to defend. Such claims or litigations could require us to stop selling the affected products, redesign those products to avoid infringement, or obtain a license, all of which would be costly and harm our business. (Electronic Arts 2007, p. 54)

Copyright and patent infringement claims against established companies in the video game industry is only an index into a more substantial problem. These companies have the money and experience that allows them to be more able to deal with infringement claims. Smaller and newly created development studios looking to establish themselves actually have more to lose in this environment than companies like Electronic Arts, who index the problem. Ultimately copyright was designed to "promote the progress of science and useful arts," (Sprigman 2002) of which it has done very little for the video game industry broadly.

6.3.1 What Went Wrong

At a practical level, copyright and patent infringement claims have come to impact the daily worlds of game developers. A pervasive environment of conservatism surrounds the legal analysis of video games. The assumption is that if a patent or copyright might apply, then it ought to be preemptively licensed, purchased, or the game altered. The Lead Designer of the game *Tony Hawk's Downhill Jam* for the Nintendo Wii talked about dealing with corporate legal teams during development work.

> We also had a number of changes to make due to a fear of potential lawsuits. This exchange is a prime example:
>
> Activision Legal: "You'll have to change that restaurant's name."
> Development Team: "But it's called Dim Sum. That's, like, totally generic."
> Activision Legal: "But if you type Dim Sum into Google, you'll find many actual, real-life restaurants called 'Dim Sum.' It's safer just to change it."
>
> I failed to realize how frightening the legal climate is at present. The fear of being sued is so pervasive that artistic freedom is being compromised, and conservative, safe decisions are routinely made even when there is no legitimate legal infringement. (Schadt 2007, p. 34)

Ultimately this conservatism enables the continued failure of copyright and patent law to promote "progress" and instead encourages regress. As more and more allowances are made, claims of fair-use and public domain knowledge are diminished, based simply on fear of litigation. Numerous patent and copyright infringement claims are made with the knowledge that companies will be more likely to pay off the claimant than actually attempt to fight or correct the broader problem. This is in part because those same companies have a vested interest in being able to pursue similar litigation against other large or small studios. The volley of infringement claims is so pervasive that several game industry lawyers have indicated in conversations with me that new studios need to assume that within their first year of activity, they will be taken to court by one corporation or another.

Console manufacturers also make significant use of the slippery slope of copyright and patent law to control production and distribution systems. Copyright claims are a mechanism for shutting down retailers of "pirate" technologies. Those same technologies are integral to truly independent game development, or game creation entirely outside of the networks of the video game industry. Again, the ability to control distribution (copyright) becomes conflated with the ability to produce (speak). As such, many of these attacks on "copyright infringement" need to be reframed as attacks on speech by those who are being silenced. The ability for people to learn, investigate, and share information about these devices via mechanisms which game companies do not control is not purely indicative of piracy or violations of copyright. In many cases it is simply a demand by the user/producer that they would like to work on these new devices. In a world now characterized by technologies that blur the lines between user and producer, the ability to investigate, experiment, and tinker is especially important. The world of "Web

2.0" or "Participatory Culture" is endangered by the continued attacks on the very core mechanisms that support it.

6.4 World 6-4: The "Market" and the Prisoner's Dilemma

More broadly, the game industry must decide how it would like to position its relationship between the State and the Market. The contradictory appeals to the independence of the game industry and the importance of the Market is subverted by continued appeals to the State to provide protection and enforcement on activities that can arguably be defined as "piracy." The mobilization of the State as a means to enforce artificial controls on the Market, which ultimately impact the worlds of game developers, is problematic. Those same activities severely constrain the ability of game developers to share and collaborate. This results in game developers spending significant amounts time reinventing the same components and never having the time necessary to share specific details about game development. While the double-speak creates particular opportunities for console manufacturers to manage and control the game industry, the same activities discourage any kind of maturation or the encouraging broader participation in game development worlds.

6.4.1 What Went Wrong

These contradictions weaken the game industry's claims that market-based ratings systems like the ESRB are effective. If the game industry cannot determine whether the Market or the State should be the dominant mode of disciplining their production methods, then claiming that only a market-based solution to the consumer's understanding of games ratings is subverted. While criminalization of the sale of "mature" or "adult only" rated games to minors seems a foolhardy approach to managing the situation, the game industry and the ESRB have not been particularly effective at coming up with alternative approaches to how to address the problem. The criteria and varying moralities at play in the ratings of games ultimately goes unquestioned.

> I take issue with the fact that to get an E10 rating, we had to change our beloved secret bonus character's name from Armondo Gnuetbahg to Armondo Ootbagh. Am I missing something here, or is Gnuetbahg a new curse word that the hip kids are using? Yes, in our tutorial we teach kids to "clobber 10 people before time expires" but we aren't allowed to say "Gnuetbahg." (Schadt 2007, p. 34)

In many respects the failure of the Market and the State is visible in the kinds of games that are produced and distributed in the game industry. The continued consolidation and risk averse activities of resulting conglomerate companies encourages an adherence to the status quo. Only a small number of individuals within those corporations are given the freedom to push the game industry in new directions. Aspiring game developers are not given an opportunity to fully participate in the Market of the game industry because they are constrained by the activities of established companies.

Ultimately the relationship the industry has with the Market and the State becomes a game of prisoner's dilemma. Will the players cooperate or defect? The unfortunate answer is that, as the game has currently been constructed, the players will tend to defect even though greater reward would be had if they had cooperated. The same becomes true for the game industry.

6.5 Boss Fight: Opening Up the Inter/Intranetworks and Risk

"Have fun stormin' the castle," was a statement made by a character named "Miracle Max," played by Billy Crystal in the movie *Princess Bride*. His wife, a moment later asks, "Do you think it will work?" to which he replies, "It will take a miracle." I am certainly no Miracle Max, and this is the video game industry. Yet the ironic play of the statement holds true, there is more than one castle to storm and there are many levels.

Part of the irony stems from the compelling nature of the functionalist argument, "Yes, but if it is so broken, then why does it seem to work so well?" Despite all of the contradictions, the video game industry broadly speaking has a compelling irrational stability and adaptability. Perhaps more importantly for some, it continues to bring in massive amounts of money. Why on earth would I want to storm such a formidable castle? The answer is of course part personal and part analytic. At an analytic level, I believe that the changing relationship between "users" and "producers" as examined in this World signify a significant schism for media producers. The rising use of coercive State power examined in World Four ought to be an indicator of this critical moment. If the game industry does not adapt to this changing relationship, it will not continue to exist as it does now, and much pain and strife will come to workers in the game industry before it comes to those in positions of power. There are also limits to the desire machine which has driven the game industry for so long. As users become more capable of producing and fulfilling their desire produce games of their own, the continual churn-and-burn attitude that the industry has had for employees will no longer be sustainable. At a personal level, the answer is more esoteric. Just because something functions does not mean that it isn't capable of functioning in ways that are more respectful, nimble, and nurturing. For so many that believe video games are an art form, I can only hope that a commitment to that form will encourage game developers to seek changes to their communities of practice.

For too long, game developers have seen their worlds as purely Western or Japanese. The time for broader participation is at hand. Web 2.0 as broadly presented is an indicator of this changing relationship. While I may quibble with the particularities of what it is precisely, at its core it is a changing relationship between "user" and "producer." There is a fundamental difference, but that difference is dependent upon ideas that the game industry has not yet embraced: access, standards, openness, participation, and remixability are its core modes of operation.

Sony's Phil Harrison continues to speak about "Game 3.0" as the video game industry's version of Web 2.0, the reality remains something quite different. It remains vetted to a broadcast model of game design, development, and distribution. "Community" and "customization" remain limited to the rather small sandbox provided for players. This is quite different from Web 2.0, where companies may not always be happy at what their users produce or reproduce.

Despite the slide delivered at GDC 2007 containing the words, "open, extendible, customizable, collaborative, audience-driven, localized," the reality is that the only openness that has been realized in the year since its delivery is perhaps "content creation" and "commerce," and even those continue to remain significantly limited by the networks of connection to Sony. Web 2.0 depends on a backbone of open technologies, formal standards, and a community of developers whose actions reflect the terms, not mimic them. Even post-dot-com bust, the World Wide Web remains the realm of innovation, more so than the game industry center.

Figure 6.1: Sony's Phil Harrison Speaks of "Game 3.0" at GDC 2007

His slide was attempting to appeal to the draw of Web 2.0 and capture the excitement that was being generated by a similar, but very different image.

Figure 6.2: Web 2.0 Graphic Released Under the Creative Commons Copyright

Numerous words appear in this image, which do not appear in Sony's vision of Game 3.0. Most importantly, words like, "standardization, open APIs, data driven, design, and participation." These words do not appear in Sony's understanding of what the future holds. Regardless of the "reality" of Web 2.0, something is happening online. YouTube, Google, and Really Simple Syndication (RSS) refocus media on the user as co-conspirator in the design and development process. The secret society of "production" is always already in doubt in this world, yet it is dependent upon openness, standards, and remixability. This is the world the game developers and the game industry must embrace.

6.5.1 Demanding Change to Copyright and Patent

The game industry, like many new artistic/creative endeavors is under serious threat from existing copyright and patent law in the United States. The fact that both of these institutions fall into the category of, "Risk Factors" in the SEC filings of major corporations should be extremely unsettling.

> Our business is subject to many risks and uncertainties, which may affect our future financial performance. If any of the events or circumstances described below occurs, our business and financial performance could be harmed, our actual results could differ materially from our expectations and the market value of our stock could decline. The risks and uncertainties discussed below are not the only ones we face. There may be additional risks and uncertainties not currently known to us or that we currently do not believe are material that may harm our business and financial performance. (Electronic Arts 2007, p. 51)

"Risk and uncertainty" caused by the current state of the patent and copyright institutions points to a fundamental problem for the future of video game production and new media production more broadly. In many respects the risk and uncertainty is exacerbated for those companies, who unlike Electronic Arts, does not have the available capital to defend itself from the numerous copyright and patent claims that could conceivably be brought against them. On a global scale it is troubling that despite the problematic character of intellectual property law in the U.S., it remains the standard. More than that, new legislative efforts, funded by corporations, push hard against ideas like fair use or a creative commons which will only create more risk and uncertainty for new media producers.

Game developers have a particularly interesting understanding of the complex connections between creative interdisciplinary work that is necessary for the production of new media. They should involve themselves in the process of copyright and patent reform, reminding lawmakers that ultimately these decisions impact workers/voters. Game developers need to help a broadly defined public understand the importance of these issues, especially in connection with their ability to think with and comment on cultural forms.

Aspiring, independent, hobbyist, and student game developers also need to assert their rights to speak on devices that have been shut off. The ability for game developers to "speak" is being shut down through the application of patents and copyright. Console systems have been closed to these developers, and homebrew efforts deliberately thwarted with technological and legal approaches. Game developers must assert that this is a violation of their speech. Copyright and patent law has become a threat rather than a cultivator of new speech and speech forms.

Ultimately these changes must occur at the level of policy, and will require the activism of numerous game developers if they hope to make change occur. Likely there will be resistance from console manufacturers and publishers. It is likely that these corporations will use the threat of network access as a means to prevent changes, which they see as disadvantageous. This is when it becomes imperative that studio heads use their positions to push for change. A drive of this sort cannot succeed without broad industry support and collaboration.

6.5.2 Reflecting on Content

New forms of interaction and more broad concepts of what constitutes play is necessary for the game industry's future. In particular it needs to think critically about the social context of play. Game developers have largely seen their work as individualistic artistic endeavors, despite the push and pull manufacturers and publishers have managed to exert on this process. Because of this disconnect, when the public vents its outrage over violent or sexual game content, the typical response from the game development community is, "Just like movies, books, photographs, music and other forms of art and entertainment, video games are fully protected speech under the U.S. Constitution" (Bala and Bala 2007). While this may be the case, U.S. media consumers have been willing to authorize government agencies to control media distribution in many cases. This is not a winning position to take for the long term. Because games are seen as media, rather than strictly works of art, this places them in a different category for many users.

Game developers need to become more aware of how game content will be received more broadly. It is necessary to view themselves as connected with broader social norms and practices. The continued placing of the game industry as outside traditional forms of media and art lends itself to special treatment by U.S. citizens and legislatures. It is difficult to argue that games are "speech" when corporate interests dramatically control that speech. If games are speech, then the game industry weakens their position by selectively limiting speech on their devices. Rather, the majority of the game industry is structured around a model of interactive mass media, which users have largely seen as worthy of government intervention and control.

While I do not make significant claims about the content that appears in games, it is shaped by the policies of manufacturing and publishing companies. It is dramatically influenced by the kinds of games that developers play. Game content is obviously shaped by the kinds of people involved in its production. If game developers are limited to a small segment of the population and a severely limited idea of what constitutes "play," then inevitably it will stagnate on certain forms. These forms will be encouraged by publishing companies that know a game will sell. For games to be thought of as speech or art, they must be the product of a more diverse community and aimed at more than what is labeled as an obvious "market." Publishing companies must be willing to take risks with new games, new styles of play, and unclear markets.

Game developers must also reflect on the reception of a game. The social context of a game is as important as the game itself. Despite numerous texts labeled canonical in the emerging discipline of game design indicating that cultural context is crucial, developers continue to see it as disconnected from their work.

> No game is an island.
>
> Games are always played somewhere, by someone, for some reason or another. They exist, in other words, in a context, a surrounding cultural milieu. The magic circle is an environment for play, the space in which the rules take on special meaning. But the magic circle itself exists within an environment, the greater sphere of culture at large. (Salen and Zimmerman 2004, p. 503)

No game is an island, nor is any game studio, neither is the video game industry. Being conscious of the reception of a game does not mean kowtowing to the demands of society, but at least being capable of having a reasonably sophisticated argument for why the content of a game is important. The ESRB, as a market based enterprise and as currently configured, will ultimately not serve the broader needs of game developers and the game industry. The ESRB answers to the ESA, which ultimately answers to the video game manufacturers and publishers. Because of this it ultimately will bend to the demands of the market, rather than how a game will be received or the kinds of arguments it is making.

The continued belief that the video games and their production fall outside the worlds of work and culture will only harm the video game industry in the long run. In the end, it is this desire to be different, to be distinct, that both differentiates and hinders the video game industry. The content and creativity of the work of video game development does set it apart as an important index into new media work. However, it remains intimately connected to work and culture more broadly. It is the complex interaction of law and culture, which places the game industry at the crossroads. Money, interesting projects, new technologies, and the promise of industry fame swirls all around and distracts game developers from these important issues. In the end, fame and money are to be found by a small few, so the majority of developers must change the direction of the industry to ensure that interesting projects and technologies continue to be available, for that is what really drives the video game industry.

6.5.3 Re-Engaging with the Structural

Developers face barriers that are more structural as well. They will require significant changes on the part of console manufacturers and publishing companies. These companies will be required to change practices that have been commonplace since the release of the NES. These companies have already managed to technologically and legally barricade access to distribution channels. The fears that led Nintendo to close off the NES no longer apply. Manufacturing companies have largely been playing the same game for the last 30 years and they will likely be hesitant to play by new rules. In a more open production space, there will be greater competition, and the possibility for content that manufacturers do not endorse or approve. However, this is not any different from the current state. The only difference is that game developers will be more able to share knowledge and resources.

Companies will likely claim that opening up routes of production will increase piracy. However, this does not seem to correspond with Microsoft's experiences on the

Xbox 360 with the XNA Toolkit. Many of the existing efforts to stem the tide of piracy have the double-effect of silencing open lines of production. When not arresting or shutting down companies that make these technologies possible, manufacturers stem the tide of broader involvement by blocking homebrew developers with "updates." Upgraded features for existing devices tempt users to install updated "firmware" (software upgradeable hardware) technologies that render devices with homebrew software unusable or no longer accessible. Despite this, time and again, users are able to bypass new mechanisms, again making them open to their activities. It seems illogical to invest so much time, energy, money, and legal resources into preventing these activities. These developers are bringing new functionality to these devices at no cost to the manufacturer. In the end, the risk is that content that has not been approved or paid licensing fees will make its way to these devices. However, the continued control of development hardware or DevKits will likely encourage most developers to continue to pay licensing fees and retain their connections with manufacturers.

Though developers and companies within the video game industry will both have to give up certain elements that the existing structure encourages, the benefits outweigh the drawback of previous approaches. Now, more than ever, the game industry must come to understand that these changes stand to benefit the sustainability and maturity of the game industry in the long term. To continue down a path lacking standards, emphasizing secrecy, and subject to aging patent and copyright law, which largely does not understand new interactive media, is to progress down a path where the same errors and risks will continue to eat away at the game industry. A path that disrespects the creative work of creating the place that video games have come to occupy in broader global culture.

The Boss Fight for this Level is ultimately the one that game developers and the game industry may be the least willing, but may the most beneficial fight, to play. Creating forums for broader cooperation across the video game industry will create opportunities for learning and sharing. There is the potential for "transgressive" possibilities, which may be viewed as undesirable. This is an inevitable consequence of encouraging more participation in the creation of video games rather than less participation. Ultimately however, existing structures enable corporations to maintain control over their "official" networks.

> The idea of a single platform for the video game console industry has been kicked around nearly every video game cycle. Publishers would gain leverage over console manufacturers or forego licensing payments altogether with a collaborative organization to develop standard hardware and software specs and requirements. I don't believe such a consortium could bring about a single console system. Business models and publisher strategies are too divergent to enable agreement on a hardware platform.
> ...
> You can now read the entire Gamasutra feature, with more from our analysts on the realities of a one console future, and how, by contrast, the PC and mobile sectors could actually benefit more from this line of thought. (Staff 2007)

How would "PC and mobile" sectors, those which have had the most penetration by global players outside the typical networks of access, benefit from these efforts, yet the console game industry game would not? It could be a boon to game developers, if they were to demand a common open reference platform, not a single console device. This is an argument for standards, not total standardization. The continued labeling of the console game industry as different or unique again enters the picture and our conversation. This is where the argument folds back onto the worlds of game developers. They must come to see their worlds as intimately connected rather than outside of traditional forms of work and social-technical practice.

PART III. ANALYTIC CONCLUSION

CHAPTER 7
WORK/PLAY, INTERACTIVITY, NETWORKS, AND CREATIVE COLLABORATIVE PRACTICE

Creative collaborative practice resides at the center of this text's analytic focus and videogame development work serves as an exemplar of this activity. Analyzing creative collaborative practice in the context of videogame development requires analytic attention at different scales. It is the entire system that is important, and it is this system, which the text grapples. Each World or Level has taken a particular scale as its analytic center, but connections remain between each. World One struggled with the interconnections between aspects of work/play and their tendency toward excess. These predilections feed into the interactive work practices within videogame development studios of World Two. Those interactive systems are intimately connected the ability to ability to gain access to the broader corporate inter/intranetworks of the videogame industry. World Three demonstrates the importance of the attention to network access and its connection back to the systems being constructed in World Two. Yet, the disciplining of these networks is done through particular mechanisms, and the means by which networks are disciplined with the prerogative power of the State is the focus of World Four.

This should not be taken to mean that the Worlds simply build on one another, but rather they are intimately connected. World Four, or the extra-local, constantly connects with and impacts the local worlds and activities of videogame developers. The structure of this text is merely one means by which the material can be analyzed. It is with this in mind that World Seven returns to these analytic categories to further tease them out from the empirical material of earlier Worlds. Each Level of this World pulls out aspects of the analytic categories of Worlds One through Four to demonstrate their connections and interplay.

World 7-1 returns to the category of work/play, its subcomponents, and their propensity for excess. World 7-2 steps-into "interactivity" as a means for understanding the technologies and organization of work in the context of new economy work. The ways in which these things come together and how access and protocols begin to structure networks of access are examined in World 7-3. Finally, World 7-4 scrutinizes the rise of corporate mobilization of the State's prerogative powers to ensure control over those networks. The Boss Fight for World Seven sets the stage for World Eight and returns to the importance of the ability to pursue underlying social and technical systems and the importance of this desire to the future possibilities of creative collaborative practice.

This text's focus has been the system; creative collaborative work practice and those things that (dis/en)able it in the context of globalized videogame development work. This system runs on the ability and the desire to get at underlying social and technical systems and structures. I have attempted to make this text an example of the very phenomenon it indexes. It is dependent upon and produced via new modes of collaborative practice. It demonstrates the importance of being able to drill down into (or "step into") and debug those subsystems that make up the system. It does not step into every possible subroutine, but those that emerged from the empirical material this work draws on. It is this ability and the desire for it, which propels my pursuit of the underlying systems and structures. These systems and structures have come to be significant for the worlds of videogame developers. World Eight is one way I have attempted to grapple with how social analysts can come to grapple with systems that cross "scales, variables, and forces" (Fortun 2006, p. 296) in different ways. The game as an analytic tool is one which I think through as a means for understanding the complexity of these formations as well as how they are open to change and (re)interpretation.

7.1 World 7-1: Work/Play

Work/play is perhaps one of the most difficult analytic categories to define in this text and simultaneously intimately plugged-into other Worlds. It is part of the machinery which other piece of the game are propelled by. Secrecy, instrumental play and experimental play are three particular subcomponents of work/play examined in the text, upon which I will further elaborate. Play is a cultural phenomenon and serves a significant cultural function, but as cultural analyst of play Johan Huizinga notes, there is something about play that exceeds the immediate needs of life, there is something that gives meaning to it. One of those aspects is the holding of play apart, the labeling of it as the intermezzo. Many industries that can tap into work/play enjoy the benefits of the intermezzo. It encourages these perceptions, thus marking itself different:

> Even in early childhood the charm of play is enhanced by making a 'secret' out of it. This is for us, not for the 'others.' What the 'others' do 'outside' is of no concern of ours at the moment. Inside the circle of the game the laws and customs of ordinary life no longer count. We are different and do things differently. (Huizinga 1971, p. 12)

This is the first step towards the culture of secrecy, which dominates in many of these spaces. Even those who have managed to access the closed communication networks within these industries, inquiries into the missing links result in responses that questioned the basic intelligence of the asker. It was these "critical silences" and "ridiculous questions" (Haraway 1997, p. 269), which I encountered when I began working with Indian developers attempting to navigate these new language systems.

If you do not know these things already, then likely you are not one of us. This logic of secrecy pervades so thoroughly, that it makes it difficult for many members of the community to learn anything beyond what is learned at the single studio and retained in the minds of workers who likely will last less than ten years. While in other industries this has typically led to educational and credentialing programs, and this seems to be an emerging trend for game development as well. However, secrecy about methods often results in the programs which developers largely feel do not meet the needs of students or practitioners.

Some studios have established new disciplines whose goal is to remove some aspects of instrumentality, providing spaces for both kinds of play. Not every developer is motivated by the "urge to be first" or to "compete for superiority." But there are always "various forms of expression" which can tap into the same drive, be it strength, art, performance, rhymes, or riddles (Huizinga 1971, p. 105). The answer is equally troubled by that it is not simply technology that influences instrumental play, but organizational structure. Further there is the issue of professionalization in the context of what really did start as a game, which has now become work/play.

> Now, with the increasing systemization and regimentation of sport, something of the pure play-quality is inevitably lost. We see this very clearly in the official distinction between amateurs and professionals (or 'gentlemen and players' as used pointedly to be said). It means that the play-group marks out those for whom playing is no longer play, ranking them inferior to the true players in standing but superior in capacity. The spirit of the professionals is no longer the true play-spirit; it is lacking in spontaneity and carelessness. This affects the amateur too, who begins to suffer from an inferiority complex. Between them they push sport further and further away from the play-sphere proper until it becomes a thing sui generis: neither play nor earnest. (Huizinga 1971, p. 197)

Simultaneously, this "[t]ension means uncertainty, chanciness; a striving to decide the issue and so end it. ... Baby reaching for a toy, pussy patting a bobbin, a little girl playing ball - all want to achieve something difficult, to succeed, to end a tension. Play is 'tense,' as we say. It is this element of tension and solution that governs all solitary games of skill and application such as puzzles, jig-saws, mosaic-making, patience, target-shooting, and the more play bears the character of competition the more fervent it will be. In gambling and athletics it is at its height" (Huizinga 1971, pp. 10-11). Or more carefully by the literary and language studies scholar, Avital Ronell, examining the test drive:

> Our extreme submission to the test [or game] – this is what the test requires – runs the risk of wearing down to the point of obliteration the one being tested. ... [W]hat won't kill you will make you stronger. Yet – assuming this peculiar perspective to be viable – one needs to come close to the killing point before suddenly desisting. (Ronell 2005, p. 145)

There is an important relationship with desisting. AutoPlay is the asymptote, the killing point, when we suddenly transition from talking about things like fun, entertainment, self-determination, and wit, to disengagement, crunch, and burnout.[55]

> Such at least is the way in which play presents itself to us in the first instance: as an intermezzo, an interlude in our daily lives. As a regularly recurring relaxation, however, it becomes the accompaniment, the complement, in fact an integral part of life in general. It adorns life, amplifies it and is to that extend a necessity both for the individual - as a life function - and for society by reason of the meaning it contains, its

55. I like to think about the "zone" in relation with Deleuze's Body without Organs. It is "the field of immanence of desire, the plane of consistency specific to desire (with desire defined as a process of production without reference to any exterior agency, whether it be a lack that hollows it out or a pleasure that fills it)" (Deleuze and Guattari 1987, p. 154). The zone cuts both ways.

> significance, its expressive value, its spiritual and social associations, in short, as a cultural function. (Huizinga 1971, p. 9)

For some, the work/play game has become the intermezzo. A process of desiring and pursuit, which actually meets ends, or does not. This is unlike the daily life, which has no clear ends, and certainly for most, no win or loss.

Desirability is also critically linked with secrecy. The fact that one must be part of the secret society imparts cachet to the members. The desirability of particular jobs over others as being more lucrative, intellectually stimulating, imaginative, or otherwise is not necessarily new either. While some scholars of communication might determine that:

> By making work more like play, "employers neatly erase the division between the two, which ensures that their young employees will almost never leave the office." Instead of becoming enraged, or unionizing, multimedia workers "smile and thank their lucky stars for being part of the digital revolution." For employers "it's a sweet deal: you can't buy flexibility like that." Paradoxically, these young multimedia workers, "touted as the most renegade - the most entrepreneurial - generation in years," are actually "amazingly subservient: the ideal post-industrial employees." (Kline et al. 2005, p. 201).

However, this simply does not seem to be a particularly unique property of game development work, or certainly any reason to tap into our new work/play mark as distinguishing it from prior work regimes. The fact that employees are interested in or driven by their work in ways that defy the value of their paychecks does not provide an adequate link to work/play. Even more problematic is the assumption or equation of industrial work to that of game development work. It is important to not elevate New Economy work over that of Industrial Work. There certainly is a difference in the desirability and accessibility of these kinds of jobs. However, "skill," or other abstract labels defining what constitutes an adequate worker, can cloud an entire other aspect of the work/play of the game industry.

You can see the danger of work/play's propensity to spiral out of control here. Too much secrecy results in isolationism. Too little and it no longer becomes "fun" or "interesting," it simply becomes work. Secrecy marks one aspect of work/play in the New Economy and the game industry in particular. While secrecy, imagination, and desirability are not necessarily new, they do plug into the system of work/play, which drives the kinds of work practices we see.

The implications of work/play are still emerging, but the New Economy's ability to leverage it, and in some cases exploit it, has significant implications for the future of work. The implications are the most dramatic in cases like those of my Indian informants who have grown up with a more rigid boundary between work and play spaces. Most American students have already seen the transition to work/play as Text Messaging (or SMS) has invaded the schools, Facebook and Instant Messaging (IM) are constant computer-based cohorts.

While the binary that has sustained much of my argument is the distinction between instrumental work/play and desiring work/play, the distinction is not really so clear. They are part of the same system and crucial for one another. While I may seem to put them in opposition, that is a move I make because I perceive a problem in the functioning of this component of the system. Too often the propensity for excess is the desired outcome of allowing spaces of play, rather than the proclivity for creative collaborative practice.

7.2 World 7-2: Interactivity

Much of New Economy work, and digitized work in particular, is increasingly interactive. It taps into experimental and instrumental play in ways that often encourages the excesses of work/play. World Two, though focused largely on disciplinary distinctions and how interactivity plays out in these spaces was also about the tools that facilitate interactivity. Interactive tools appear at disciplinary boundaries, in part, because they provide means for understanding experimentally those systems and structures, which compel and constrain our interdisciplinary compatriots. However, these new tools are plugged into an assemblage that has largely been used to encourage excess, and as such, so too do the interactive systems tend towards excess.

Creative collaborative work is interactive because it functions through it connections with experimental and instrumental play, as well as connections with networks the provide access to new tools and systems which enable new forms of interactive communication. However, because it is connected with work/play, it can tend towards excess. Interactivity can become a goal in and of itself. Interactivity can provide such levels of freedom and play that understanding the position of the individual within the broader system can become blurred. Systems that were designed for one purpose may be experimentally used to provide new functionality in ways that were not previously envisioned, but perhaps other more effective means are possible. Our interactive capacity quickly becomes a liability rather than an asset.

Interdisciplinary work frequently necessitates interactive systems. They allow for new conversations. While a discussion may reveal some of these differences, it is frequently in the play of work that the full depth of the distinctions made along disciplinary lines become apparent. Interactivity is a means by which these differences are navigated. The ability to interactively negotiate and make meaning of the underlying systems and structures which position the work of others is a crucial component of creative collaborative work. Understanding and the desire to understand those systems and structures within which others work within is key to the creative collaborative process.

Videogame development is also indexical of many of the kinds of interdisciplinary ventures in New Economy work. The products created in these contexts have become increasingly complex and interactive systems provide means for individuals to understand the position of their work in relation to the overall whole. Simultaneously, these interactive systems can be pushed so far that they have the reverse effect. Individuals can become so obsessed with the alteration of minute aspects of a system rather than attempting to understand or pursue the underlying relationship of that aspect within the system. Especially when timelines loom and last minute changes begin to loom for workers in these contexts, it can quickly degenerate into interactive collapse.

Management becomes increasingly difficult in under these conditions. Attempting to understand the volume or complexity of work required for seemingly simple changes to a system may be extremely difficult. Ascertaining what constitutes "legitimate" work or play within these structures can prove impossible. Management and workers become increasingly connected with one another and it can seem as if one has the pulse on the other. What frequently is lost in the translation is that this system is mediated by a large number of interactive tools that bridge disciplines. Those systems and the people who have constructed them must not be forgotten. The importance of paying attention to what is frequently lost to invisible work must be retained.

Yet, interactivity is a new and necessary component of the tools that encourage creative collaborative practice. It is a necessary component. It encourages our experimental and instrumental play. Yet at the same time, we cannot forget that these systems shape our possible futures, and their connectivity to systems of excess must be met with moments of non-interactivity to reflect on where they are more or less productive. It is this reflective moment that has largely been lost. The labyrinth continues to be a useful metaphor for thinking about our tools and the systems we create with them. The notion that carrying string along with us, such that we can reflect on the paths we have taken or not taken and how one might move through the maze differently is important.

The postmortem fulfills part of this goal, but frequently it is too little too late. Planning has already begun on the next project, without insights gained from the previous processes of production. Those reflections may begin to filter in after the fact, but often disconnected from broader connections, which hamper our ability to get at the underlying social and technical systems and structures which shape the everyday worlds of those working in these contexts. Our interactive tools provide the foundation of how we are able to grope at and determine those systems and structures which (dis/en)able our creative collaborative practice.

7.3 World 7-3: Inter/Intranetworks

The network has long been a useful core category used by social analysts looking to understanding productive practice. What this text does differently is take the issue of access and visibility within networks and bring it to the foreground. The Inter/Intranetwork by its marking of "inside" and "outside," denoted by the slash, encourages us to constantly pay attention to the restrictive capacities of networks. Too often networks are talked about as inherently open, better, or different than hierarchical systems, yet networks can be just as hierarchical. There is nothing fundamental about networks that make them naturally flat or more open. They must be constructed in ways that enable flatness or openness.

Again, my argument is mobilized by this distinction not that "inside" is good and "outside" is bad, or vice versa. Both the internetwork and intranetwork are necessary and useful in the context of creative collaborative practice. The good/bad distinction is made more carefully. Intranetworks are expanding at an explosive rate because of corporate consolidation. Internetworks are shrinking and being used as sites for offloading labor and risk. Rather than both being used productively to foster and encourage creative collaborative practice, inter/intranetworks are being used to shore up existing sites of power and control.

The willingness to submit to access and protocol restrictions associated with inter/intranetworks has a great deal to do with the intermezzo, the secret societies of work/play. Because access to new networks and protocols marks one as privileged, elite, or worthy, they are particularly seductive. Thus, our inter/intranetworks connect to work/play. They connect to our other substructures as well. Experimental and instrumental play are subject to access restrictions. Our ability to access new intranetworks provides us with new resources for understanding or groping for new notions of why systems function the way they do. They provide us new resources and protocols by which we can think through and reason about the world. Without access to these networks we are left only with prepositions about the possibility of something being a particular way.

Our inter/intranetworks are also intimately tied to interactivity. Networks control our access to particular tools, or the knowledge that certain kinds of tools are useful for improving our experimental systems. The ability to construct tools at the sites between disciplines may be fundamentally tied to our ability to access information. Our desire to determine how or why something is occurring may be significantly impacted by (lack of) access to network resources. Thus, we see how the particular attention to inter/intranetworks rather than "just networks" is an important one.

This same category also offers social analysts a reason, perhaps even demanding, that they begin paying attention to the structure of networks, technical or social ones. Perhaps drawing parallels between Network Neutrality deployed in a social context rather than simply in a technical one might help illuminate for politicians why it is such an important issue for technologists. While social analysts have found the core category of the network useful, we ought to push the metaphor further. Protocols, routers, switches, hubs, access, and the inter/intra distinction all offer new tools for thinking about power in the context of networks.

Thus, inter/intranetworks become intimately connected with the final analytic core category. While it is possible for networks to be technologically mediated and managed, especially in creative collaborative contexts, those technological limitations are feeble compared to the desire many have to instrumentally and experimentally understand how they work. Instead corporations have turned to the State and mobilized the prerogative aspects of the State to ensure the continued maintenance and control of these networks. Code may be legislation by another means, but code without legislation lacks any real teeth. It is in concert that their power is truly exhibited. It is in this context that I identify the rise of the corporatized State, something quite different from traditional notions of the State in the context of Neoliberalism or the New Economy.

7.4 World 7-4: Corporatization of the Prerogative Power of the State

More than any other subsystem, I was struck by the ways in which the prerogative power of the State (that extra-legal and often violent aspect) has been mobilized to discipline collaborative networks and access to videogame development work. World Four took the State and in particular the prerogative power of the State as its focus. The coercive moment has dramatic implications for the future of creative collaborative practice. The deployment of the prerogative power of the state on the part of corporations to discipline creative collaborative networks of production is telling. In part this is why I return in World Six's Boss Fight to Web 2.0. There is something that has shifted in the production of technical and digital systems in the context of the global New Economy. Work/play, interactivity, and inter/intranetworks are each aspects of this shift. Fundamentally it is a different notion of users rather than consumers. Users have the capacity to become producers and many industries have embraced this notion. The videogame industry however has held it tenuously at arms length. The rising corporatization of the State has more to do with the increasing pressure being placed on "industries" in this context. It is symptomatic of the crisis the very notion an "industry" faces from user/producers rather than consumers.

Thus, the corporatization of the State has more to do with the shifting configurations necessary to be part of the creative collaborative process. Previously the inter/intranetworks of the videogame industry have been disciplined in ways that required only the possible threat of force. Now it requires actual deployment of force to discipline networks. Even in the shifting terrain of digital distribution systems, the desire to maintain delineated boundaries around who can and cannot develop games or who is a legitimate producer of videogames has become so important that the State's prerogative power must be mobilized to ensure control. These mechanisms are brought in chorus with technological mechanisms to prevent user/producers from accessing networks, both literal and figurative, within the industry. This has been pursued so thoroughly that those playful aspects of work/play, despite their propensity for excess, which are crucial to the creative collaborative process.

Each of these systems connect to one another, more so, they are intricately connected. At their core, this game, this system is about the capacity, the ability, perhaps even the right to pursue underlying social and technical systems and structures. Ultimately the corporatization of the prerogative power of the State has done more to damage the future possibilities of creative collaborative practice. When user/producers are consistently told not to pursue underlying technical and social systems and structures, and if they do, they suffer the possibility of arrest and incarceration. This configuration does more to discourage entry into spaces once thought crucial the future of productive practices and ultimately disables the very desires and drives foundational to activities like videogame development.

7.5 Boss Fight: Creative Collaborative Practice and Underlying Systems

The ability to get at underlying technical and social systems is the core mechanic of this game. It is central to the functioning of creative collaborative practice. Without this capacity, the system begins to collapse. Because we are plugged into this system in ways, which previously workers may not have been, this excess quickly becomes personal. Game developers have spent years wondering why the have yet to find a sustainable equilibrium. The answer lies in pursuing those underlying systems and structures which (dis/en)able the creative collaborative practice within their daily worlds. This has been the pursuit of this project. It is my hope that given this understanding that game developers and workers in "industries" more broadly look to how those structures work well and where they fail.

It is for this reason that dichotomies like instrumental work/play versus desiring work/play collapsing are brought forward in this text. It is the reason for the emphasis on the corporatization of the prerogative power of the State. Our very capacity to play these games is being fundamentally dismantled, and in the end, those actions have the most detrimental effects on everyday workers within the videogame industry. The same is true in other professions as well. The dramatic shift in academic publishing structures are one example. The rising deployment of the prerogative power of the State by television, movie, and music corporations or trade groups against citizens. The decline of fair use rights due to things like the Digital Millennium Copyright Act and broad deployment of "encryption" technologies, that despite their circumvention to re-grant users of a technologies the abilities they already had, prevent questing after underlying social and technical systems.

The foundation of creative collaborative practice is the ability and desire to pursue underlying technical and social systems. This is a playful practice, which 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 design that inevitability. It is my contention that by looking at work/play and understanding its tendency towards excess, the importance of and danger of interactivity, being constantly aware of access and protocol rules for inter/intranetworks, as well as how aspects of the State are being mobilized on the part of corporations offers significant insight into understanding creative collaborative practice.

PART IV. EPILOGUE

CHAPTER 8
EPILOGUE - THE VIDEO GAME INDUSTRY GAME

What kind of game would the game industry be? Would it be a fun game? World Eight mobilizes the arguments made throughout the dissertation and synthesizes them by conceptualizing the video game industry as a game. It is an opportunity to ask different questions about the work of video game development. It is a multiplayer game, though not necessarily a "networked" game in the traditional sense like a first-person shooting game. It is both collective and individual. The individual affects and is affected by the collective. At the level of work practice, I hope this exercise makes the point that interactivity, though a valuable tool for game developers, can also be pushed too far. People must be given the time and space to get work done. Crunch in many respects is the product of over-interactivity in concert with poor planning, modified timelines, artificial demands by other interests, and the continued demands for secrecy in the game industry.

The game is designed for the Nintendo DS for numerous reasons. The first is personal; the Nintendo DS has a burgeoning "homebrew" and technically illegal community growing around it. I hope to support this community by offering new arguments in support of their activities as well as the technical resources created during the eventual development of the game. It is my opinion that developing a game for the Nintendo DS in connection with a scholarly project demonstrates the illogical character of criminalizing speech on proprietary technologies. The ability to speak with and through devices that are owned by a user should not be compromised by legislation encouraged by corporations that largely have been unable to prove the value or sustainability of what they promote.

The other reasons the DS is the target platform are more practical. More than any other console device, both "hardcore" gamers and casual gamers alike have accepted the

DS. The DS has a much broader player demographic than typical console hardware. Nintendo has gone as far as marketing the device and a subset of its games at numerous markets, men, women, young, and old. Atypical games have been released for the console and it continues to attract new kinds of gamers. Figure 8.1 seems indicative of this changing trend, and even how gamers have adjusted their thinking about games and these new audiences.

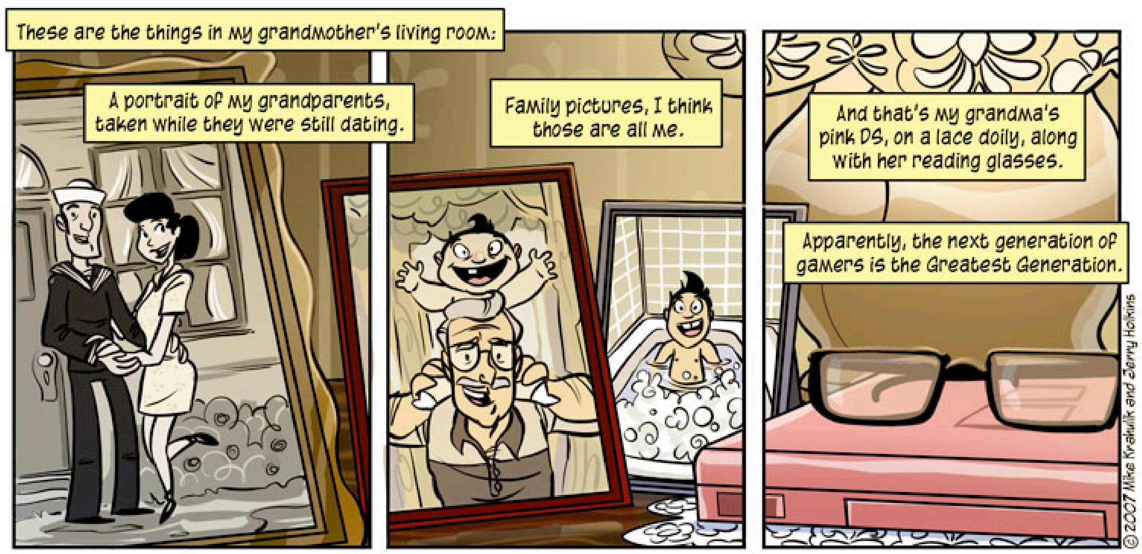


Figure 8.1: Webcomic Penny Arcade on the Diversity of the DS's Market

(Krahulik and Holkins 2007)

Thinking broadly about who could conceivably be or desire to be game developers is an important aspect to changing the structure of their worlds. In some respects, the decision to target the DS is a departure from using this exercise strictly as a demonstration of the structures of the game industry. If that were the case, then I would target only the Playstation 3, the most inaccessible and expensive of the current generation of consoles.

The DS's two screens, stylus input, built-in WiFi, and relatively low price also make it an attractive target for this project. The two screens allow the ability for interaction and presentation of new information simultaneously. The stylus while being an approachable form of input for game-play is also more like the mouse on a personal computer, where most games are actually created. The WiFi capabilities will allow for "teams" of developers to work together on tasks. I do not conceptualize this in "real-time," but rather through a mechanism where individual tasks come to affect others on your team. Each individual developer has their own drives and specialties, which ultimately affects the kinds of games they play, as well as their progression through the game. The graphics are meant to be stylized and again, accessible. Rather than appealing to core demographics, accessibility is emphasized.

Over the course of game-play, it is hoped that the player begins to understand their position in a larger structure, some they can adjust and modify, and others they cannot. It is these structures I hope players will come to question and desire to change. The game is meant to bring a broader understanding of what it means to create video games, as well as appreciate the work which numerous people do to bring games to market, many of whom are unacknowledged or never seen as integral to the development of this class of new media.

8.1 World 8-1: "Vertical Slice" - A Gameplay Narrative

The following is a gameplay narrative imagined by a player of this game. It is thought of as a conceptual or verbal account of what a "vertical slice" of how this game would be experienced by a player.

The first thing I was presented with while playing this game was a screen, which I was asked to enter my name. I presume I could enter whatever I like, but I chose to enter, "Cassie O'Donnell." The second thing it asked me was what my college undergraduate major was. There were several options, I selected, "Computer Science." It told me that currently I was unemployed, that I had managed to escape my undergraduate institution with no debt, but that I was "unfortunately" located in the Midwest, where few game jobs were available.

I was presented with my developer's "status" screen. It indicated that I was currently unemployed and I had "undergraduate" skills in engineering and "low" skills in art, design, and management. My personal status was currently "happy." When I touched that element with my stylus it went into more detail, displaying that my "fatigue" was "low," my mood was "good," number of hours at work were "0," I had $1,000.00 in the bank, I was single, and I had no children.

Three new options presented themselves, "Search for Jobs," "Create Independent Studio," "Join Independent Studio," and "Relocate." I first investigated the "relocate" option, because I had previously been told that my location was "unfortunate." Unfortunately the $1,000.00 I had banked was not enough to finance relocation to anywhere with greater game development job availability. So, with that in mind I began searching for a job. There were several available "software development" jobs which I applied for and was accepted and took on the role, "entry level software engineer."

At this point I was presented with my first engineering puzzle game. It was relatively easy to complete. My goal was to trace the movement of "data" into a system and correct "improper" data movement. Several pieces of data were being improperly placed, which I corrected by adjusting "pipes" on the screen. After completing the first game I was returned to my "status" screen. My money was gradually increasing and my engineering skills were increasing.

I then chose to "Create Independent Studio." I named my studio "Alchemyst Creations." I then entered into a new engineering puzzle game. During this game however, I received a message that my "real" job was demanding my time. I could either respond to the request or continue with my independent work. At first I selected to continue with my independent work, and was then instructed that this would likely result in a poor performance report from my real job. Considering this warning I instead chose to return to my real job. I completed two more engineering puzzles before I was returned to my independent engineering puzzle. After completing that puzzle I closed my console for a break.

Later, when I returned to the console, I was told that I had two waiting real job puzzles to complete. After finishing those I began another independent puzzle. My independent studio status screen indicated that "production" on my first game was 5% done with respect to engineering, but 0% for both "design" and "art." There appeared to be several available options, the first was to "Take on Design Task," "Take on Art Task," or "Find other Developers." I tried the find other developers option, at which time the console attempted to wirelessly find other "developers" in its range. I was by myself however, so it could not find anyone else. It encouraged me to find players with artistic or design skills and "connect" with them using this feature. It also said that I could "manually" add other players using the "Friend Code" option.

In the mean time I took on several art and design tasks. The first art task was to duplicate several line drawings displayed on one screen using the stylus on the lower screen. Other tasks involved attempting to place a texture on a 3D model by selecting points on the lower screen. Many of these puzzles were quite difficult for me. Later that night I emailed a friend from high school, asking if he had seen this game. He had not, but downloaded it and placed it on his console. Since he was an artist he was initially assigned a job at an advertising company, but was able to join my independent studio using the Friend Code option. At this point we both became able to work together on our independent project, when not assigned tasks from our real jobs. He took on the art projects, I did the engineering, and we split the design tasks.

During this time my position at my real job role had improved to "lead software engineer," and I had begun being assigned "management" puzzles. While my management skills were improving, it made it more difficult to improve my engineering skills, which were crucial to my independent project work. I began to seriously consider the relocation option available in the game. It was going to cost me a significant amount of money, but I could relocate to the West Coast where there were several available "jobs" at established game companies. I found that I could search jobs in these locations and even apply for them, though I frequently only received rejections or no response at all.

Eventually I did decide to relocate. At this time I was able to take an "entry level game engineer" position at a company for less money than I had previously been making, but it was an opportunity to be a "real" game developer. Very quickly I was being assigned new puzzle tasks. On several occasions when I returned to the console after having shut it for a day I would find myself with five to ten puzzles to complete before I could return to my independent work with my friend. Then one day the game notified me that game production at the company I was working for was entering "crunch" mode.

I wasn't quite sure what that indicated, but it initially meant that I had puzzles coming at me nonstop. When I attempted to close the console, it flashed a red light, which I assume indicated something, so I opened the console again. It said that if I chose to stop in the middle of this puzzle I would be risking my job, and that I should finish this puzzle and two more before closing the device. I did that, reluctantly, and closed the console. When I returned to the device the next day I had 15 puzzles waiting for me to complete. Quickly I found myself working exclusively on these puzzles. Occasionally other puzzles would interrupt the puzzle I was working on, and not even reduce the number of puzzles I needed to complete before closing the device without risking a poor performance review.

I noticed that my avatar's status was deteriorating. This began to manifest during puzzle activities, where "bugs" or "mistakes" would strike while I was attempting to solve a puzzle. These would frequently make the puzzles more difficult and take more time. Finally I was frustrated enough with this process that I began searching for new jobs. There were plenty of new jobs available, so I tried a new company. I was even hired as a "senior game engineer," but quickly this company too was in crunch mode. However, I had saved enough money up that I quit my job this time and began working exclusively on the independent project.

Unfortunately my friend was not doing much work on the independent project by this time, because he was working for a game company as well. However, I knew a handful of people with the game now, so they joined my company as well and began working on the project. When the game was 50% complete a new option became available. We suddenly had the opportunity to "Shop Your Game Around," to publishing companies. When we did find a publisher willing to fund the remaining development of the game, suddenly new tasks began presenting themselves, primarily management tasks. Because I had started the studio, everyone indicated that I should handle those tasks.

Our independent work had quickly become our own work, the publishing company began also asking for changes to our game, and new engineering, art, and design tasks began presenting themselves as a result. Quickly, in an effort to meet a deadline for the game (part of the deal with the publishing company), I was forced to indicate "crunch" mode for our game. Quickly it became apparent that this was the same game all over again, only I was in charge of the company this time. Some of my fellow players began to bail out, resulting in more tasks for fewer people. I tried to bring on other players, but they too quickly dropped out of the game. Eventually it was just a handful of hardcore players that made sure that the game completed.

While there was a sizable payout at the end of that part of the game, my avatar was left "exhausted," and almost incapable of completing a puzzle due to the frequency of bugs and mistakes. I set the game aside for a while to recuperate myself as well. At some point it seems inevitable that I will have to use the "Leave the Game Industry" button, an option always available, but it seems such a shame after all that I have invested.

8.2 World 8-2: Core Game-play

The work of game development functions as the core game-play mechanic. It is stylized, in the form of puzzle-like tasks that the player navigates. These tasks attempt to approximate the "play" of work as much as possible, though it does not undermine the fact that they can be difficult or complex. These tasks take the form of "mini-games" which the user plays. The overall goal of the game is open ended. Players can determine if their goal is to "create titles," create their own company, climb the corporate ladder, or simply enjoy the tasks which they have the opportunity to work on. While there is no "score" in the traditional sense, several different sub-systems have the potential for being recognized as a "score." The "employment history," "employer," and "titles" sub-systems each have the opportunity to become categories which players place value on as being a score. The game has several underlying subsystems which each impacts the overall game mechanic. These are divided into the following categories:

Sub-System Name	**Description**
Skill Levels	The "skill level" represents skills acquired while working. These skills provide futher options during the play of mini-games as well as the ability to advance or move to other companies.
Personal Status	Personal status represents the state of a worker. Fatigue, mood, number of hours at work, money, relationship status, and number of children are included in this category.
Regional Location	The regional location is randomly assigned to the player at the beginning of the game. The region affects available job opportunities and employers. A player can move if they have enough money to do so.
Employment Status	Employment status represents where a player is employed.
Job Role	Job role is the kinds of primary tasks available to and assigned to a player. Sometimes jobs not associated typically with a role will be available, providing players with the ability to gain new kinds of skills.
Employer	The employer of a player determines the kinds of tasks available to them. Players can create new "independent" companies while employed at other companies, though they must maintain their work level enough to remain employed.
Employment History	The employment history of a player is a record of where they have worked, how long, and kind of work they did.

Titles Published	The titles published are a list of games which the player has been credited for in their work.

Table 8.1: Game Sub-Systems

Some of these sub-systems are more simple than others. Employment history and Titles published are a historical record of what a player has done through the course of the game. Some are more complex. The Skill Level system is one of the primary game mechanics, as it determines the kinds of employers and job roles available to a player. Players are awarded "experience" during the course of playing mini-games. Games conceptualized as "engineer" games will be heavier on engineering skill rewards. The player then has the ability to distribute these points amongst those skills they would like to improve.

Skill Levels	**Category and Description**
Software Coding	[Engineering] Coding skills point at the ability for a player to complete quickly software development tasks. This is not necessarily indicative of the quality of the code being produced.
Software Design	[Engineering] This is the other side of software development skills. Being able to produce "good" or "well designed" code.
Debugging	[Engineering] Debugging is the ability of an engineer to determine where something is going wrong in the process of game development.
Modeling	[Art] Modeling is the creation of objects which can be placed into the game.
Texturing	[Art] Texturing is the creation of skins for models.

Animation	[Art] Animation is the ability to create the animation sequences which textured models depend upon to be put into motion.
Level Design	[Design] Level design is the creation of game environments.
Character Design	[Design] Character design is the generation of compelling game character concepts.
Game Mechanics	[Design] Game mechanics are those rules and systems which underly a games visual presentation.
Scheduling	[Management] Scheduling is the ability to accurately estimate and hit project deadlines.
Resource Allocation	[Management] Resource management is the ability keep a team on track and adequately tasked for a project.
Networking	[Management] Networking keeps developers connected to other developers, publishers, and manufacturers.

Table 8.2: Skill Level Sub System Detail

Each player when starting the game will choose an undergraduate major. These will affect the starting values of a players skills. All players will start out as unemployed, and can either begin searching for work, move if money provides the opportunity, or start their own game companies. These companies of course will only have one role available to begin, "Jack of all trades." Numerous tasks will be available to the player, but they will quickly have to decide on a specialization and then either pursue employment at an established company, or attempt to attract other players to their companies. At any given time a player can only be part of two different companies. This amounts to the idea of a "day job" and a "startup." Of course a player can belong to two startup companies, though this will quickly affect fatigue, mood, and money.

Some game systems provide restrictions on the player. Personal status, regional location, current job role, employment history, and titles published fit well into these categories. The personal status of a player affects the amount of time they can reasonably remain at work each week without their fatigue increasing or their mood falling. These two categories will affect the "accuracy" with which they can complete work related tasks. The more fatigued a player is, or the worse their mood, the more possible it becomes that actions taken in a task will not work quite as desired. This is done to approximate the declining ability to remain focused experienced by fatigued workers.

Figure 8.2: A Conceptual Graphic by Tyson Stecklein for "Burn Out"

The "game is over" when the players mood sinks to "jaded" and their fatigue reaches "hospitalization," or at any point their money reaches zero. This is considered "burn out." A player can later re-enter the industry with a character if sufficient real-time has passed since the game was last played. Players can also opt to "leave the industry," at any time during the game, though for players who are working with that player will suffer in-game repercussions.

The regional location affects the likelihood that particular kinds of employment are available for a would-be game developer. Each employer defined in the game will have regional locations where particular job roles are available to the player. These restrictions will reflect the available job opportunities of a given geographic region. Players will be equally likely to spawn in the U.S., Western Europe, India, or Japan. In locations where game development jobs are particularly rare, developers will have the opportunity to work at companies that border game development work. Engineers will have the opportunity to work for software companies, though they must maintain a company of their own on the side to "remain in the industry."

8.3 World 8-3: Game Environments and Structure

The mini-games represent the game space of the video game studio. Mini-games are tasks which each player must complete. As a player moves into certain kinds of positions or organizations, tasks may become timed. An "estimated" time may be provided, which a player must beat in order to gain all points associated with a mini-game. Timed mini-games may not be paused or stopped by closing the DS without penalty. It is assumed that if a timed task is paused that the human player need a break that would be unavailable to a worker. While the player may pause, their virtual characters fatigue will increase and their mode will decrease. Un-timed tasks may be paused or the DS can be closed and put into its "hibernation" mode.

Crunch mode occurs when a series of tasks are scheduled back to back, and the user has no option to not complete all of them simultaneously. Again, if the game is paused or place into hibernation, the characters mood and fatigue will be affected. If crunch has not occurred recently, a temporary increase in mood will be provided, representing some of the allowances made to employees during crunch times.

Figure 8.3: A Conceptual Graphic by Tyson Stecklein for "Crunch Mode"

Throughout the play of mini-games, "interactive" prompts from other employees as well as other games may interrupt the flow of a current game. Meeting mini-games, comments from fellow workers, email messages, and instant messages will in some cases distract a player. If a player has a significant other or children, then additional mini-games or interruptions may occur, resulting in fewer skill level experience points. Over time this loss of points may result in lower earnings or fewer advancement opportunities.

Players may also be dependent upon the completion of a task by another player. They then have the opportunity to send an IM to that player to check on the progress of the work. These messages will later show up on the screen of the other player in the form of "interactive" prompts. However, these same prompts can distract or affect the work of the other player being done. They are a double-edged sword.

Based upon the "amount of time spent at work," effectiveness at work, and other minimum values, players may be encouraged to leave a company. In many cases however, these players will likely "burn out" prior to this situation.

Other game elements are defined by the employer and role of a player. In some cases, the combination of particular employers, roles, and locations may result in situations where players are pushed more quickly towards burnout. In other companies, roles, and locations more sustainable models may be in place. At first there will be several categories of employers, though this list can be expanded on in the future.

Employer Type	Description
Independent Studio	As previously mentioned, a player can create an independent studio of their own at any time. They will either have to complete all of the tasks associated with game development, or they will have to look for other players to join their company. As these companies mature, they may be approached by publishing companies or manufacturers to do work, at which time the classification of their studio will change. This is simulated by the game. In many cases this will be a necessary step for getting a job with any of the other available employers. These studios may be sold, bought, or merged with other studios, publishers, or manufacturers.
3rd Party Studio	Several third party companies will be created for the game to serve as starting locations for players to work. These will be restricted by geographic location.
1st Party Studio	These studios frequently have more freedoms than third party studios as they have more funding coming from a publishing or manufacturing company. However, these companies can also exert force over the studio during the development of a game.
Publisher	These companies will likely only be accessible to developers with high skill sets. Players will have to ensure that they live in the geographic locations where these companies are, and that their skills are such that they will be hired.
Manufacturer	Similar to publishing companies, but a step higher in difficulty for developers to gain access.

Software Firm	In locations where engineering game development work may be more difficult to find, players interested in being engineers will likely have to work for a software firm while also doing independent game development.
Art Production Studio	These companies only employ artists and typically have fewer "crossover" tasks that allow developers to work on projects that expand their skill set. However, these employers will train artists and pay them enough so that players have mobility if they wish to change geographic region.

Table 8.3: Employer Categories

Once a studio has reached the level of 3rd party or higher, they can begin "researching" tools to assist in the development of their games. These tools must be individually researched. If a player changes companies, these tools will not move with the player. If a player transitions from one company to another, they may gain access to new tools that assist in their work.

It is also possible, however, that during game-play if a tool is used, it may malfunction depending on how frequently it is used by employees. The more frequently a tool is used, the more reliable it becomes. The less used, the less reliable. This means when a tool is first created it may actually make work more difficult. Over time however, with continued use the tool will begin to simplify the work processes. Because tools cannot be shared, the testing of a tool must be done at each individual company, so players may have to experience the "learning curve" or "testing phase" of a new tool several times during the course of their career. Tools may also be purchased from a middle-ware company, while the cost of purchasing a tool may be less expensive than researching a tool, the learning curve and testing phases will still be required.

Tool access can also be restricted by geographic location and employer type. Some middle-ware will simply be too expensive for companies to purchase, while others are unattainable because employers have not yet gained access the those networks which provide them the opportunity to acquire tools or even know that their development is useful.

8.4 World 8-4: Game Elements

Each mini game will be divided into one of the following four categories:

Game Category	Description
Engineering	Engineering tasks involve an assortment of "parsing" and "number crunching" tasks. There are also "debugging" tasks, where the player must identify previously "parsed" or "number-crunched" pieces of work that are incorrect.
Art	Art tasks require the player to create and modify different kinds of artistic elements. This can be the creation of "wireframe" models that approximate goal images. It can be the picking of proper texture coordinates to skin a model, or the creation of animations for a model.
Design	Design games require the user to take game "content" and "mechanics" and solve problems with them. The goal is to make a particular goal event occur through the modification of on-screen objects.

Management	Management work is frequently puzzle work, piecing together disparate pieces of a system. The goal is project completion, or task completion, or the movement of resources from one location to another.

Table 8.4: Mini-game Categories

Different mini-games can be defined for each category. Throughout the game frequently players will have to engage with some games which fall outside their realm of expertise, or they may take a job that requires them to complete tasks outside of their area of expertise. In these cases, games may not behave entirely as expected, in order to simulate the accumulated expertise over time. However, if a player continually plays and does well with particular types of games and expands their skill levels at these games, new jobs, companies, and tasks will become available to them.

While particular mini-games will be encountered more than once, they will also have elements which are generated randomly when a game starts. For this reason, players will not be solving the same game each time. Rather when they encounter the same game, it will begin in a random fashion. Engineering number crunching games will be randomly seeded in such a way that they remain re-playable. All games will have one or more randomized elements which will make the re-playing of games possible.

Because players will also be presented with occasional games outside their area of expertise, it will give each player the opportunity to play different kinds of games. Players can increase their skills in new areas in hopes of receiving new kinds of games. New "moves" will be provided for players as their experience increases. For example, an experienced artist working on a particularly difficult model can push a button that says "basic layout" that moderately decreases the complexity of the model they are working on. In some cases companies will have defined special moves that assist their employees in getting tasks completed.

"Winning" is a category that users can define on their own. Because the game is designed to provide numerous user-defined goals, winning can be either successfully running their own independent studio, working at an established company, to start a company that is acquired and leave the industry with a large bank account, etc. It is intentionally left ambiguous to encourage players to reflect on their motivations and goals.

8.5 Boss Fight: The Credits Roll and Bowser Lives

While I see numerous goals available in the game, which players may take on, either the creation of their own studios, console manufacturing companies, or the amassing of money, I believe that the structure of the game provides a commentary on the state of the video game industry as it currently stands.

It is my hope that it also encourages discussion around certain aspects of the video game industry, like the artificial construction of restrictions that prevent access from developers in the U.S. and other countries. I believe it also makes apparent that things like "crunch" and a culture of overtime are not only undesirable for the game industry, but the product of a particular construction and imagination of the game industry, certainly not an inevitability. I also hope that it encourages developers to appreciate the value of experience and expertise, that both are crucial to the future success and stability of the video game industry. The continued hemorrhaging of talented hardworking individuals, and the lack of collaboration that results from an environment demanding secrecy are problematic for the industry.

This is the real Boss Fight, and it is one that has yet to occur.

APPENDIX

0.1 Fieldwork and Data Sources: Working In/And/Of Corporate field sites

Many of the additional data sources I arrived at during the course of this project were out of necessity. The NDAs which structured my position at my field site made it necessary for me to search out new means by which to write and talk about issues in the video game industry.

0.1.1 Field-sites

During the course of my three years of fieldwork in the video game industry, I was fortunate enough to establish connections with a game development company in Troy, NY whose studio heads were willing to work with me, and assist me in establishing connections with other developer and studios. My initial connections with Vicarious Visions (VV) established the possibility for my future connections. Gaining access to corporate field sites is a long running research rub, one that I managed to navigate without really realizing.[56]

Vicarious Visions (VVEast), Troy, NY, USA was the most recent studio acquisition by Activision, one of the leading video game publishers in the United States. I gained access to this site during a pilot research phase, and have continued research at this site since that time (October, 2004). I have had the opportunity to observe the company's movement from independent developer to in-house developer for Activision. As

56. The "dual constraints on ethnographic researchers who study work: access and time," is of particular concern in the context of the video game industry. All researchers interested in studying the workplace face challenges on both fronts. Access is highly structured and frequently can take a significant amount of time to acquire. Any call for more studies of the production of video games must engage with this complexity. Of particular relevance is the excellent review article: (Smith 2001).

previously stated, studio acquisition is common in the game development setting, though seldom documented or examined by social scientists.

Dhruva Interactive (DI), Bangalore, India is the only video game development company in India that has managed to gain the necessary licenses for doing work on console video games. DI has managed to create industry connections in the U.S., and has produced a large number of cell-phone bound games. While DI has not yet gained permission to develop their own console video game titles, their industry position in India, and its location in a rapidly globalizing industry, makes it an ideal field site to better understand how the game industry responds to the forces of globalization.

FXLabs (FX), Hyderabad, India continues to grow and push the envelope for game studios in India. They actively recruit and hire U.S. game developers who then spend half of their year living and working in Hyderabad and half of their year living and working in the U.S., telecommuting to India. FXLabs has recently begun work on an Archie Comics based game targeted at the Xbox console, as well as licenses for several Bollywood based movie IPs.

RedOctane (RO), Chennai, India is one of the first U.S. held studios in India. The India-based studio was established shortly after the release of Guitar Hero by RedOctane in the U.S. When RedOctane was acquired by Activision in May of 2006, they retained the India based offices of RedOctane. Because of these corporate networks RedOctane has been able to work on the Nintendo DS, as well as several other projects acquired through Activisions corporate networks.

The Game Developer's Conference (GDC) is the primary industry conference held each year. This is often an opportunity for developers to demonstrate their latest work, for console makers to exhibit future console concepts, and for developers to conduct educational seminars. GDC is largely focused on work and work organization.

The Association for Computing Machinery's (ACM) Special Interest Group, Graphics (SIGGRAPH) is one of the most watched "academic" conferences by the video game industry. Often referred to by game developers as, "What we should be doing in two or three years," SIGGRAPH is one of the most widely attended events by academics and video game professionals. SIGGRAPH focuses primarily on the newest technologies and software coding developments.

The International Game Developers Association (IGDA) holds local and global meetings. Local meetings are often held at coffee shops or pubs, with demonstrations or discussions being held. One of the primary goals of local meetings is networking, and the sharing of ideas. National meetings are typically held in conjunction with other large events like GDC. The IGDA focuses primarily on work and work organization.

0.1.2 Supplemental Data Sources

I used numerous other data sources to supplement my fieldwork data, which was sometimes limited by my access to people, or the coverage of NDAs. To complement my ethnographic fieldwork, I drew on numerous other resources which I included with my analysis of data. It was through these experiences looking for other points of access to restricted information that actually helped me understand the ways in which copyright and patents were being mobilized to control production.

Patent Filings can provide insights into the technologies present in the game development workplace, which may otherwise be neglected. In some cases the significance or purpose of a patent is relatively straightforward. In other situations patents may apply to a product, but it may be unclear what products to which it applies. Patents can both constrain and enable certain kinds of developments, and they can help us understand certain technologies, without violation of NDAs.

Court Cases are another source of information regarding the technologies which lie underneath the hardware and software at use in the context of video game construction. When video game companies litigate one another, often times details about the underlying hardware and software are use in the proceedings as evidence. Much like patents, court cases can bring to light details regarding a technology which might otherwise not be disclosed.

Security and Exchange Commission (SEC) Filings offer another level of detail about the interaction of game companies, and the ways in which the industry structures itself. These documents are often the only accurate source of information available regarding the money flows within or between organizations.

Press Releases (PR) are another public document which, when combined with ethnographic data, can help us to understand the internal and external activities of organizations, their interactions, their battles, and their collaborations. Without these documents, many of these activities may be known by only a small number of individuals within an organization. This study will capitalize on the combination of these data sources.

I also used publicly available resources like Game Developer Magazine and enthusiast press websites like Kotaku, Joystiq, and Penny Arcade as well as more developer-centric sites like GameDev.Net, and the IGDA's web-site. Each provided additional resources which I could draw on in my analysis of data. These sites also provided me with data which I could not have received through my networks of connection within the game industry. Frequently game related news sites broke information before press releases were made available, or employees were notified of corporate activities. I continue to monitor many of these sites in an effort to keep my material up to date.

0.2 Data Storage and Analysis

At each field site I maintained field notes in digital form on my computer, which were then then hand-coded using Bookends bibliographic software in concert with a text editing application called Textwrangler. 47 interviews were recorded using an iPod and microphone adapter. These were then partially transcribed using Express Scribe and transcription foot-pedal. These transcripts were also coded and placed in text documents and Bookends. The emerging codes were then used to group data into four primary categories, "Work/Play," "Interactivity," "Inter/Intranetworks," and "Corporatization of the State." These four primary codes were further broken down and coded into the Worlds seen throughout the dissertation. Each of these categories emerged from my fieldwork. All digital research files were stored only on my computer and password protected using Mac OS X's built-in FileVault application which encrypts data using the AES-128bit encryption system. Connecting the ethnographic fieldwork with existing literature was done in a similar fashion. Relevant material was coded for the same categories that emerged from my fieldwork and placed into Bookends. These quotes were then used to situate my work broadly in the literature.

While writing the dissertation I divided it into eight Worlds and four Levels per World, or 32 segments. Each of these segments corresponded to a code or several codes. World 1-1 for example was coded as "secrecy," "play," and "skill." I would search my database of quotes, transcripts, and references for these codes, which had been placed in the keyword field of the database. This made it easy for me to assemble a series of data an literature from which to draw on in writing that section of the dissertation.

0.3 Highlighting Console Manufacturers

The consoles manufactured by four major players, Nintendo, Sega, Sony, and Microsoft have had a profound impact on the video game industry. I include them in the appendix, because their development and progression is not transparent for those unfamiliar with the video game industry. They are an important window into the moving targets that are the major consoles. Throughout the dissertation this can be seen by the changing destinations developers are working to create games for. This too influences the lack of historical memory of video game companies.

Nintendo has been part of the game developer and game lexicon since before the introduction of the Nintendo Entertainment System in 1985 in the U.S. Mario and Donkey Kong have been around for a very long time. But the birth of the NES in 1985 was a telling moment for the future of video games and the video game industry. The NES introduced a host of innovations which took the game developer and gamer worlds alike by storm. From the games to the numerous accessories which were marketed and developed, the NES fundamentally changed both gaming and game development. Not only was Nintendo responsible for revolutionizing the video game industry, they have continued to remain a major player, despite predictions by frequent nay-sayers that they were doomed. Indeed the recent success of the Nintendo Wii is a testament to this.

Nintendo's Console Systems	US Release	Common Name	Distinguishing Characteristics	Notable Games
• Nintendo Entertainment System	1985	NES	"D-Pad" Controller NES Zapper Light Gun Power Pad R.O.B. or Robotic Operating Buddy Power Glove 8-bit Ricoh CPU at 1.79 MHz 8-bit Ricoh PPU at 5.37 MHz 2 KB Memory for Each CPU and PPU 128KB - 4MB Cartridge Size	Super Mario Bros. - Nintendo Duck Hunt - Nintendo The Legend of Zelda - Nintendo Metroid - Nintendo Mike Tyson's Punch-Out! - Nintendo Final Fantasy - Square Enix/Nintendo Contra - Konami Castlevania - Konami Mega Man - Capcom Ninja Gaiden - Tecmo
• Gameboy	1989	GB	Portable Handheld Gaming System Link Cable to Connect Two Gameboys Battery Powered Camera and Printer Accessory 8-bit Sharp x80 CPU at 4.19 MHz 8kB Memory for CPU and PPU 256kbit - 8Mbit Cartridges	Tetris - Nintendo Pokémon - Nintendo Metroid II - Nintendo

• Super Nintendo Entertainment System	1991	SNES	Ergonomic D-Pad Controller "Shoulder" Buttons on Controller Super Scope Light Gun Unreleased CD-ROM Drive Made by Sony Became PlayStation Super Game Boy Adapter 16-bit Ricoh CPU at 1.79 MHz 16-bit Ricoh PPU at 3.58 MHz 128KB of DRAM for CPU 64 KB SRAM for PPU Enhancement Chips for Cartridges 512KB - 4MB Cartridge Size	Super Mario World - Nintendo Super Mario Kart - Nintendo F-Zero - Nintendo Super Metroid - Nintendo Star Fox - Nintendo Legend of Zelda: Link to the Past - Nintendo Pokémon - Nintendo Pilot Wings - Nintendo Final Fantasy II - Squaresoft Mega Man X - Capcom Chrono Trigger - Squaresoft
• Nintendo 64	1996	N64	Analog Stick Controller Controller "Paks" for Game Saves Expansion Pak for Increased Memory Rumble Pak for Force Feedback 64-bit NEC MIPS CPU at 93.75MHz 64-bit SGI GPU at 62.5MHz 4MB - 8 MB of RDRAM 4MB - 64MB Cartridge Size	Super Mario 64 - Nintendo Mario Kart 64 - Nintendo Star Fox 64 - Nintendo GoldenEye 007 - Rare/Nintendo Zelda: Ocarina of Time - Nintendo Super Smash Bros. - Nintendo Mario Party - Nintendo Perfect Dark - Rare Conker's Bad Fur Day - Rare Ogre Battle 64 - Quest/Atlus Co.
• Gameboy Advance	2001	GBA	Portable Handheld System Color Screen Battery Powered Link Cable to Connect Two GBAs Link Cable to Connect to GameCube 32-bit ARM CPU at 16.8 MHz 8-bit Z80 Coprocessor at 8.4 MHz 32KB RAM Internal to CPU 96KB VRAM Internal to CPU 256KB WRAM External to CPU 32MB - 128MB Cartridge Size	Mario Kart Super Circuit - Nintendo Super Mario Advance - Nintendo Castelvania: Circle of the Moon - Konami Advance Wars - Intelligent Systems/Nintendo Tony Hawk's Pro Skater 2 - Vicarious Visions/Activision Metroid Fusion - Nintendo Zelda: The Minish Cap - Flagship/Nintendo Golden Sun - Camelot Software/Nintendo
• GameCube	2001	GC	Two Analog Sticks Two Analog Shoulder Buttons Eight Standard Buttons Donkey Kong Bongos Accessory Network/Modem Adapter IBM "Gekko" PPC CPU at 485 MHz ATI "Flipper" GPU at 162 MHz 24MB SRAM for CPU 3MB SRAM for GPU 8cm DVD Derived Game Disk ~1.5GB Disk Storage Space	Animal Crossing - Nintendo Metroid Prime - Retro/Nintendo Super Mario Sunshine - Nintendo Super Smash Bros. Melee - Nintendo Pikmin - Nintendo Mario Kart: Double Dash!! - Nintendo The Legend of Zelda: The Wind Waker - Nintendo The Legend of Zelda: Twilight Princess - Nintendo Eternal Darkness - Silicon Knights/Nintendo Resident Evil 4 - Capcom Viewtiful Joe - Capcom
• NindendoDS	2004	DS	Portable Handheld System Two Color Screens One Touch Sensitive Screen Included Stylus for Touch Screen Built-in Microphone Built-in WiFi Capability Secondary GBA Cartridge Slot 32-bit ARM CPU at 67 MHz 32-bit ARM CPU at 33 MHz 4MB RAM for CPU 656KB RAM for Video 8MB - 128MB Cartridge Size	Opera Internet Browser - Opera/Nintendo Animal Crossing: Wild World - Nintendo Big Brain Academy - Nintendo Brain Age - Nintendo Cooking Mama - Taito/Majesco Games Heroes of Mana - Brownie Brown/Square Enix Meteos - Q Entertainment/Nintendo New Super Mario Bros. - Nintendo Nintendogs - Nintendo Puzzle Quest - 1ST Playable/D3Publisher SimCity DS - EA Japan/EA Games Tetris DS - Nintendo Viewtiful Joe: Double Trouble - Clover Studios/Capcom
• Wii (Code Name: "Revolution")	2006	Wii	Motion Sensitive Wireless Wii Remote Controller ("Wiimote") "Nun-chuck" Attachment for Wiimote Wireless Connectivity to DS Built-in WiFi Capability Backward Compatible with GC Parental Controls for Game Content IBM "Broadway" CPU at 729 MHz ATI "Hollywood" GPU at 243 MHz 24MB RAM for GPU 3MB Texture RAM for GPU 64MB RAM for CPU Up to 8.5GB on DVD-Based Disks	Wii Sports - Nintendo Legend of Zelda: Twilight Princess - Nintendo WarioWare: Smooth Moves - Intelligent Systems/Nintendo Metroid Prime 3: Corruption - Retro Studios/Nintendo Tony Hawk's Proving Ground - Neversoft/Activision TMNT - Ubisoft Montreal/Ubisoft Red Steel - Ubisoft Rayman Raving Rabbids - Ubisoft NiGHTS: Journey of Dreams - Sonic Team/Sega Geometry Wars: Galaxies - Kuju/Sierra Call of Duty 3 - Treyarch/Activision

Table 0.1: Highlighting Nintendo[57]

57. Each of the tables created for Nintendo, Sega, Sony, and Microsoft were done based entirely on information gathered from Wikipedia, home-brew and hobbyist developer web-sites, and any website that listed "specifications" information regarding a particular console. In part this is a piece of the story. Very little information has been made public about what is actually inside a game console. In some cases detailed information was made available as part of a press release, as was done for the Sony PS3 (Sony Computer Entertainment 2005). For the most part however this data can only be assumed partially accurate. I am attempting to work with each one of these companies to clarify, corroborate, and make more public some of this information.

As can be seen in Table 1.2, Nintendo has remained a force within the video game industry since 1985 through the present. From the introduction of hand held gaming consoles to the push to real time 3D graphics in the N64 to the control innovations of the Wii, Nintendo has consistently pushed the video game industry to think differently about what it is they do and for whom they make those products. Sometimes this has earned them ridicule of other companies, gamers, and game developers. Even failed business dealings with Nintendo have resulted in critically important events; the failed partnership with Sony to produce a CD-ROM drive for the SNES went on to become the Playstation.[58] Nintendo, in addition to arguably being the leader of the game console manufacturers club, has also been one of the most recognized and influential makers of video games. Designers part of Nintendo, like Shigeru Miyamoto, are especially recognized by game developers as having significant influence on the video game industry. Miyamoto was the designer credited with the creation of the characters and games featuring Mario, Donkey Kong, and Zelda, all long standing and immensely successful franchises for Nintendo.

Sega emerged shortly after Nintendo took stage with the NES and quickly became their closest competitor and rival. Each company took turns with marketing jabs aimed at the other. The competition between these two companies created the industry as it now stands in many respects. Each faced the complex concern of staying technologically ahead of their competitor, while also providing game developers with the time and resources necessary for creating games for their systems.

58. Though this claim has never been corroborated by Sony or Nintendo, it is a widely held belief amongst game developers, and is even mentioned in some histories of the video game industry (Malliet and Zimmerman 2005).

In many respects you can see at a miniature level the "MHz arms race" which was soon to grip the computer and software industry more broadly. However, rather than talking about the speed of the processors, the short hand typically was in the number of "bits" a system was. From the "8-bit" era of the NES and SMS through the "16-bit" times of the SNES and Genesis. Of course in many respects this arms race can even be seen in the names of systems, the 32X by Sega and the N64 by Nintendo. For the most part this was the expression of a more broad phenomenon of growing computer power available at significantly lower costs. Each company struggled to out-pace the other while still providing enough margin on the retail sales to not cause bankruptcy for themselves.

Sega too created characters and games that even now continue to thrive on game consoles. Sonic the Hedgehog is one of the most well known Sega characters, or their series of "virtua" games.

Sega's Console Systems	US Release	Common Name	Distinguishing Characteristics	Notable Games
• Sega Master System	1986	SMS	"Phaser" Light Gun 3D Glasses 8-bit Zilog Z80A CPU at 3.54 MHz 8-bit TI Video Display Processor 8 KB Memory for CPU 16 KB Memory for VDP Up to 4MB Cartridge Size	Double Dragon - Technos/Taito Ecco the Dolphin - Novotrade/Sega Ghouls 'n Ghosts - Capcom Ninja Gaiden - Tecmo Out Run - Sega Rampage - Baily Midway Shinobi - Sega Sonic the Hedgehog - Ancient/Sega Ultima IV: Quest of the Avatar - Origin Systems
• Sega Mega Drive	1989	Genesis	Backward Compatible with SMS Contoured Controller with 3 Buttons Controured Controller with 6 Buttons Wireless 6 Button Controller "Menacer" Light Gun Mega-CD ROM Add-On 16-bit Motorola CPU at 7.67 MHz 8-bit Zilog Z80A CPU at 3.58 MHz 8-bit TI Video Display Processor 64 KB Memory for CPU 64 KB Memory for VDP 8 KB Memory for Zilog CPU Up to 4MB Cartridge Size	Sonic the Hedgehog 2 - Sonic Team/Sega Mortal Kombat II - Midway Virtua Racing - Sega Altered Beast - Sega Ecco the Dolphin - Novotrade/Sega The Lion King - Westwood Studios/Virgin Interactive/Sega/ Walt Disney Computer Software Desert Strike: Return to the Gulf - Electronic Arts The Revenge of Shinobi - Sega
• Sega Game Gear	1991	Game Gear	Portable Handheld Gaming System Color Screen Battery Powered TV Tuner Adapter Backward Compatible with SMS 8-bit Zilog Z80 CPU at 3.58 MHz 24 KB Memory for CPU and VDP Up to 4MB Cartridge Size	Sonic the Hedgehog - Sega
• Sega 32X	1994	32X	"Add-On" for Sega Mega-CD Two 32-bit RISC CPU's at 23.01 MHz 256 KB RAM for CPU's 256 KB RAM for VDP Up to 4MB Cartridge Size Up to 650MB CD-ROM Games	Space Harrier - Sega DOOM - id/Sega Virtua Racing Deluxe - Sega Virtua Fighter - Sega
• Sega Saturn	1995	Saturn	Analog Stick Controller Two 32-bit RISC CPU's at 28.63MHz Two 32-bit VDP's at 7.159 MHz Custom "Saturn Control Unit" for Geometry Processing 1MB SDRAM for CPU's 1MB DRAM for CPU's 1.5MB VRAM for VDP's 650MB CD-ROM Drive	Virtua Fighter - Sega Virtua Racing - Sega Virtua Cop - Sega The Need for Speed - Electronic Arts Street Fighter Alpha - Capcom Sonic 3D Blast - Sonic Team/Sega Quake - id/Sega Panzer Dragoon - Sega Nights into Dreams - Sonic Team/Sega Mega Man 8 - Capcom Duke Nukem 3D - 3D Realms/Apogee Software

• Sega Dreamcast	1999	Dreamcast	Visual Memory Unit with LCD Rumble Pack VGA Adapter Mouse and Keyboard Fishing Rod Available Modem or Network Adapter 32-bit RISK CPU at 200 MHz 32-bit PowerVR2 GPU at 100 MHz 16MB RAM for CPU 8MB RAM for GPU 1.2 GB GD-ROM Drive	Bomberman Online - Hudson Soft/Sega Puzzle Bobble - Capcom Dead or Alive 2 - Team Ninja/Tecmo Ecco the Dolphin: Defender of the Future - Sega Jet Grind Radio - Smilebit/Sega Phantasy Star Online - Sonic Team/Sega Quake III Arena - id/Raster/Sega Resident Evil 2 - Capcom Soul Calibur - Namco Spider-Man - Treyarch/Activision Tony Hawk's Pro Skater - Treyarch/Crave Entertainment Unreal Tournament - Epic Games/GT Interactive Shenmue - AM2/Sega

Table 0.2: Highlighting Sega

Things went well for the two console companies as they competed against one another, Nintendo seeming to be on top most of the time, while Sega was able to maintain a technological edge on Nintendo. Sega frequently released consoles with more sophisticated components earlier than Nintendo. The Genesis in particular was a coup for them, providing new and higher detailed levels of graphics than most users had previously seen. This is where Sonic the Hedgehog really took off, with games making use of the technological advances throughout this period.

But this kind of competitive environment always has losers, and it was the entry of a third player into the world of the console video game industry that heralded the death of Sega as a producer of consoles. Sega has of course continued to make games, but the hardware race is an expensive one, and while Sega had been able to justify higher prices for their consoles based on higher quality graphic experiences, this argument began to lose its muster with the introduction of the Saturn which was quickly overshadowed by Sony's PS1. The technologically sophisticated Dreamcast was significantly more expensive than every other console on the market and was, only a year after its release, trumped by Sony's Playstation 2. It was at this time that Sega threw in the towel as a console manufacturer.

While it is likely that Sony had been long hoping to make an entry into the console game industry since the development and marketing of the CD-ROM drive, which had made them significant amounts of money from the music industry, it wasn't until

1995 that Sony took the video game industry by storm. The introduction of the PS1 had a significant impact immediately. The available storage space of a CD rather than a cartridge, and the relatively inexpensive manufacturing costs of CDs in comparison to cartridges positioned Sony well to compete both technologically and economically with Nintendo's aging SNES and Sega's Saturn. It wasn't until a year later that Nintendo answered with their new console, the N64. In the meantime Sony was able to snatch up a significant amount of market share with their new console.

It is also likely that the available tools for game developers had a significant amount to do with the success of the PS1. Up until the release of the Dreamcast, PS1, and N64, most games were written in assembly language, in a way that is directly translatable into machine code. With the introduction of the Dreamcast, developers could write code in C, a higher level programming language which required the use of a compiler to generate assembly code, which was then "assembled" into machine code. However, the Dreamcast suffered from a poorly written compiler which required most developers to continue working in assembly to get the speed necessary for functional games. The PS1 on the other hand shipped ready for developers to make wide use of a C compiler. Sony spent a significant amount of time ensuring that developers had the proper tools for developing games on the PS1. The N64 by contrast also had an impressive set of development tools based on the Silicon Graphics workstations which had birthed it, but the nearly year long delay of the console and expense of producing cartridges for it resulted in fewer developers making games for it.

The relatively inexpensive PS1 disks which while looking like CD-ROMs, were not quite the same[59], but they did make it easier for Sony to cut manufacturing costs

59. Sony was well aware that CD burners and CD-R media were well on their way to market prior to

throughout the lifetime of the PS1. Sony introduced the PS1 at a cost significantly lower than the Saturn. The Nintendo 64 while less expensive to purchase had significantly more expensive cartridges when compared with PS1 disks. Combined with a more liberal licensing scheme than Nintendo, and reduced manufacturing costs, Sony was able to court new development studios with a large degree of success.

Sony's Console Systems	US Release	Common Name	Distinguishing Characteristics	Notable Games
• Sony Playstation	1995	PS1	Distinctive Controller Dual Analog Stick Controller "Dual Shock" Controller with Vibration Net Yaroze Version for Hobbyists 32-bit RISC CPU at 33.87 MHz Geometry Transformation Engine Data Decompression Engine 2MB RAM for CPU 1MB RAM for GPU 650MB CD-ROM Drive	Ridge Racer - Namco Tekken - Namco Mortal Kombat 3 - Midway/Sony Computer Entertainment Resident Evil - Capcom Crash Bandicoot - Naughty Dog/SCEA Tomb Raider - Core Design/Eidos Interactive Gran Turismo - Polyphony Digital/SCEA Spyro the Dragon - Insomniac Games/SCEA Oddworld - Oddworld Inhabitants/GT Interactive Final Fantasy VII - Square/SCEA Driver - Reflections/GT Interactive Tony Hawk's Pro Skater 4 - Vicarious Visions/Activision
• Sony Playstation 2	2000	PS2	Backward Compatible with PS1 Contained PS1 CPU for I/O Control Network Adapter Optional Internal Hard Drive Eye Toy Camera Logitech Headset Guitar Hero Controller 2 USB Ports 2 Memory Card Slots 64-bit CPU at 194 MHz 32-bit FPU Coprocessor 128-bit Vector Processor 128-bit GPU at 147 MHz Multi-pass Rendering Support 32 MB RDRAM 4 MB DRAM for GPU Up to 4.7GB on DVD-ROM Disks	Dead or Alive 2 - Team Ninja/Tecmo Oni - Bungie Studios/Rockstar Games Madden NFL - EA Tiburon/EA Sports NHL - EA Canada/EA Sports Devil May Cry - Capcom Grand Theft Auto 3 - DMA Design/Rockstar Games Tony Hawk's Pro Skater 3 - Neversoft/Activision Final Fantasy X - Square/Square EA Jak and Daxter - Naughty Dog/SCEA Max Payne - Rockstar Toronto/Rocstar Games Ecco the Dolphin: Defender of the Future - Sega Spider-Man - Treyarch/Activsion Sly Cooper - Sucker Punch Productions/SCEA Tekken 4 - Namco Bloodrayne - Terminal Reality/Majesco Rachet and Clank - Insomniac Games/SCEA Shrek 2 - Luxoflux/Activision Viewtiful Joe - Capcom SingStar - Sony Computer Entertainment God of War - SCE Studios Santa Monica/SCEA Guitar Hero - Harmonix/RedOctane Shadow of the Colossus - Team Ico/SCEA Okami - Clover Studio/Capcom
• Sony Playstation Portable	2005	PSP	Portable Handheld Gaming System Widescreen Color TFT LCD Battery Powered 32-bit CPU at 1 to 333 MHz Vector Processing Unit 32MB RAM and 4 MB DRAM 512-bit GPU at 166 MHz 2MB RAM for GPU Up to 1.8 GB on UMD Disks	Death, Jr. - Backbone Entertainment/Konami Gran Turismo 4 - Polyphony Digital/SCEA Hot Shots Golf - Moby Games/SCEA Loco Roco - SCE Japan/SCEA Lumines - Q Entertainment/Ubisoft Metal Gear Solid = Konami Prince of Persia - Ubisoft Ratchet and Clank - High Impact Games/SCEA Ridge Racer - Namco Spiderman 2 - Vicarious Visions/Activision Ultimate Ghosts 'N Goblins - Capcom
• Sony Playstation 3	2006	PS3	Backwards Compatible with PS2 "SIXAXIS" Controller Internal Hard Drives of Various Sizes Wireless Controller WiFi Network Connectivity High Definition TV Connectivity Connectivity to Home Service Downloadable Game Content 128-bit IBM Cell CPU at 3.2 GHz Six "Synergistic Processing Elements" One Power Processing Element" One Vector Procesing Unit Custom NVIDIA G70 GPU at 550 MHz 256 MB of XDRDRAM for CPU 256 MB of GDDR3VRAM for GPU Up to 25GB on Blu-ray Disks	Resistance Fall of Man - Insomniac Games/SCEA Ridge Racer 7 - Namco Madden NFL - EA Tiburon/EA Sports NBA - SCE Studios San Diego/SCEA Tony Hawk's Project 8 - Neversoft/Activision Fight Night Round 3 - EA Chicago/EA Sports flOw - thatgamecompany/SCEA (Downloadable Game)

Table 0.3: Highlighting Sony

the release of the PS1 and as such they took significant measures to ensure that the games bootable on the PS1 were different than those generated by CD burners available to users. I discuss these measures further in World Four, but it worth noting here because I believe it is a significant reason why Nintendo did not employ CDs in the N64.

Sony was able to capitalize on the immense success of the PS1 with the introduction of the PS2 a short five years later. Many developers have said that, in retrospect, Sony could have likely delayed releasing the PS2 for several years, as the original PS1 still had games being actively developed and released until Sony refused to grant any more licenses for the aging console. The PS2 cemented Sony's position in the console video game industry. During the years while the PS1 was on the market and through the introduction of the PS2 Sony was at the top of its game, dominating both hardware and software sales in the game industry. This was a heady time for Sony, virtually unrivaled, they cemented their position and gained a great deal of user support and loyalty. The number of franchises generated for these two consoles was considerable.

However, much like Nintendo, things were about to change, and the entry of a new third console manufacturer was going to have a significant impact on the video game industry. Nearly one year after the introduction of the PS2, Microsoft threw their hat in the ring, releasing the Xbox, sure that Nintendo had been finished off by the relatively poor performance of their GameCube console.

Microsoft, long known in the computer industry, and maker of PC based games did not enter the video game industry until 2001. Many chided the new console maker as not understanding the market, or as simply trying to package a PC in a small case which could be connected to a television. While each of these may have had some amount of truth, what they neglected to take into account was the massive amount of money MS was willing to lose in order to enter this new market. They also knew that a "killer app" was a necessary component for creating demand for their console. For this effort MS bought up long time Apple Macintosh game development company Bungie, whose still in development title "Halo" had garnered a great deal of excitement amongst gamers based on preliminary screen shots and video clips released. While it is true that MS's Xbox was much closer to a common PC than the PS2 or GC, that also made it simpler for developers familiar with MS's development tools and technologies such as DirectX to make a direct movement to the console, something which was not true for the other consoles.

The Xbox enjoyed moderate success in the United States, but managed to capture very little of the overseas market. In particular the Xbox seemed completely unable to penetrate the Japanese market, which was perceived to be the proving grounds for the viability of a new console video game system. The Xbox was louder and larger than its Japanese developed counterparts. This was only complicated by the relatively clunky looking case which seemed to reflect the typical mentality of PC manufacturers of the time, "its what's inside that counts."

Microsoft's Console Systems	US Release	Common Name	Distinguishing Characteristics	Notable Games
• Microsoft Xbox	2001	Xbox	Built in Hard Disk Drive Downloadable Xbox "Live" Content Integrated Network Connectivity 32-bit Intel Celeron CPU at 733 MHz SSE and MMX SIMD Units Vector Processing Unit 64MB SDRAM for CPU and GPU 128-bit NVIDIA GPU at 233 MHz Up to 4.7GB on DVD-ROM Disks	Advent Rising - GlyphX Games America's Army - U.S. Army/Ubisoft Batman Begins - Eurocom/EA Games/Warner Bros. Bloodrayne - Terminal Reality/Majesco Burnout - Criterion/Acclaim Counter Strike - Valve Software/Microsoft Game Studios Crash Nitro Kart - Vicarious Visions/Universal Interactive Darkwatch - High Moon Studios/Capcom Dead or Alive 3 - Team Ninja/Tecmo DOOM 3 - id/Activision Fable - Lionhead Studios/MGS Halo - Bungie Studios/MGS Halo 2 - Bungie Studios/MGS Karaoke Revolution - Harmonix/Konami Leisure Suit Larry - High Voltage Software/Sierra Max Payne - Rockstar Vienna/Rockstar Games Need for Speed Carbon - EA Black Box/EA Panzer Dragoon Orta - Smilebit/Sega Phantasy Star Online - Sonic Team/Sega Spiderman 2 - Treyarch/Activision X-Men Legends - Raven Software/Activision
• Microsoft Xbox 360	2005	360	Backwards Compatible with Xbox Downloadable Content Downloadable "Live Arcade" Games XNA Express and Game Studio Optional Hard Drives Wireless Controller External HD-DVD Drive Integrated Network Connectivity High Definition TV Connectivity 32-bit Three Core IBM CPU at 3.2 GHz ROM in CPU Stores Encryption Keys 1024-bit ATI 2 Core GPU at 500 MHz 512 MB of GDDR3 RAM for GPU/CPU Up to 7GB on DVD-ROM Disks	America's Army - Red Storm Entertainment/Ubisoft Banjo-Kazooie 3 - Rare/MGS Blue Dragon - Mistwalker/MGS Bullet Witch - Cavia/Atari Crackdown - Realtime Worlds/MGS Dead or Alive 4 - Team Ninja/Tecmo Forza Motorsport 2 - Turn 10 Studios/MGS Gears of War - Epic Games/MGS Kameo - Rare/MGS Lost Planet - Capcom Project Gotham Racing 3 - Bizarre Creations/MGS Quake 4 - id/Raven/Activision Rockstar Games presents Table Tennis - Rockstar Viva Pinata - Rare/MGS

Table 0.4: Highlighting Microsoft

While this may have been true, Microsoft actually learned from these lessons and rather than waiting the typical five to six year interval between console systems, developed and released the Xbox 360 only four years later, a full year before Sony and Nintendo's "next-gen" systems. The new system was significantly smaller than its predecessor and had significant aesthetic considerations in its design. Microsoft was even willing to depart from the previous PC in a box approach, instead using CPUs manufactured by IBM, who at the time was supplying rival Apple Computer with CPUs for their computer systems. Ironically, at the same time Apple was beginning to make their transition from PowerPC based computers manufactured by IBM and Motorola to those manufactured by Intel.

Microsoft's entry into the market is not yet proven. They continue to be perceived as the newcomer, and despite their efforts have not yet been able to crack markets outside of the United States and Western Europe. Of all the currently shipping console manufacturers, Microsoft has been the only company to adjust their pricing or marketing approaches to encourage users in developing economies to purchase their systems, recognizing the differences in global monetary systems.[60]

60. Microsoft instituted an Xbox 360 purchase program while I was in India. Financed through India's ICICI bank, customers could walk into an electronics retailer, and for 1000RS or around $20.00 walk out of the store with an Xbox 360. The remainder of the $500+ cost of the console was financed over a six month term.

BIBLIOGRAPHY

Adorno, Theodor W. and Max Horkheimer. 1976. *Dialectic of Enlightenment* Translated by Cumming, John. New York, NY: Continuum International Publishing Group.

Akiyama, Yoshiyuki, Shinichiro Iimura, Hiroshi Ogawa, Kazuo Kuroda, Toshio Suzuki, Akiyoshi Inoue, Shouji Taniguchi, and Minemasa Ota. 2001. U.S. Patent No. 6430133. Washington, DC: U.S. Patent and Trademark Office. Assignee: Sony Corporation, Tokyo; Pioneer Electronic Corporation, Saitama-Ken; Pioneer Video Corporation, Yamanashi-Ken, all of (JP).

Androvich, Mark. 2007. "Sony Threatens to Pursue Legal Action Against PS3 Hackers." Website <http://www.gamesindustry.biz/content_page.php?aid=25750>. Last Accessed: July 20 2007.

Appadurai, Arjun. 1996. *Modernity at Large*. Minneapolis, MN: University of Minnesota Press.

Atari Games Corp. and Tengen, Inc. v. Nintendo of America Inc. and Nintendo Co., Ltd. 1992. 975 F.2d 832 (1992).

Baba, Marietta L. 2003. "Working Knowledge Goes Global: Knowledge Sharing and Performance in a Globally Distributed Team." *Anthropology of Work Review* 24.1-2:19-29.

Bala, Karthik and Guha Bala. 2007. "Video Crackdown Violates Free Speech." *Times Union* Opinion

Bam Entertainment, Inc. 2001. "IPO S-1, EX-10.31, Material Contract."

Barley, Stephen R. 1996. "Technicians in the Workplace: Ethnographic Evidence for Bringing Work into Organizational Studies." *Administrative Science Quarterly* 41.3:404-441.

Barley, Stephen R. and Julian E. Orr. 1997a. *Between Craft and Science: Technical Work in U.S. Settings*. Ithaca, NY: Cornell University Press.

———. 1997b. "Introduction: The Neglected Workforce." in *Between Craft and Science: Technical Work in U.S. Settings,* edited by Stephen R. Barley and Julian E. Orr. Ithaca, NY: Cornell University Press.

Bates, Bob, Jason Della Rocca, Alex Dunne, John Feil, Mitzi McGilvray, Brian Reynolds, Jesse Schell, and Kathy Schoback. 2004. "Quality of Life Issues (http://www.igda.org/qol/open_letter.php)." IGDA. Last Accessed:

Beck, Ulrich. 2000. *The Brave New World of Work*. Malden, MA: Polity Press.

Becker, Howard. 1984. *Art Worlds*. Berkeley, CA: University of California Press.

Bogost, Ian. 2006. "Persuasive Games: Wii's Revolution is in the Past." CMP Serious Games Source. Website <http://seriousgamessource.com/features/feature_112806_wii_1.php>. Last Accessed: July 20 2007.

Bonds, Scott, Jamie Briant, Dustin Clingman, Hank Howie, François Dominic Laramée, Greg LoPiccolo, Andy Luckey, and Mike McShaffry. 2004. "Quality of Life in the Game Industry: Challenges and Best Practices." *IGDA Website*

Bowker, Geoffrey C. and Susan Leigh Star. 1999. *Sorting Things Out: Classification and Its Consequences,* Edited by Wiebe Bijker, Bernard W. Carlson, and Trevor Pinch. Cambridge, MA: The MIT Press.

Brown, Wendy. 1995. *States of Injury: Power and Freedom in Late Modernity*. Princeton, NJ: Princeton University Press.

Bucciarelli, Larry. 1994. *Designing Engineers,* Edited by Wiebe Bijker, Bernard W. Carlson, and Trevor Pinch. Cambridge, MA: MIT Press.

Bucciarelli, Louis L. and Sarah Kuhn. 1997. "Engineering Education and Engineering Practice: Improving the Fit." pp. 210-229 in *Between Craft and Science: Technical Work in U.S. Settings,* edited by Stephen R. Barley and Julian E. Orr. Ithaca, NY: Cornell University Press.

Burghardt, Gordon M. 2005. *The Genesis of Animal Play: Testing the Limits*. Cambridge, Massachusetts: MIT Press.

Carless, Simon. 2007. "Why Consoles are Here to Stay, Yay." GameSetWatch. Website <http://www.gamesetwatch.com/2007/08/why_consoles_are_here_to_stay.php>. Last Accessed: 2007 August 4.

Carter, Ben. 2003. "Postmortem: Lost Toys' Battle Engine Aquila." *Game Developer Magazine* 10.4:50-58.

Cassell, Justine and Henry Jenkins. 2000. *From Barbie to Mortal Kombat: Gender and Computer Games*. Cambridge, MA: MIT Press.

Castells, Manuel. 1998. *The Rise of the Network Society*. Oxford, UK: Blackwell Publishers.

Castronova, Edward. 2005. *Synthetic Worlds: The Business and Culture of Online Games*. Chicago, Illinois: University of Chicago Press.

Chaplin, Heather and Aaron Ruby. 2005. *Smartbomb: The Quest for Art, Entertainment, and Big Bucks in the Videogame Revolution*. Chapel Hill, NC: Algonquin Books.

Clapes, Anthony L. 1993. *Softwars: The Legal Battles for Control of the Global Software Industry*. Westport, CT: Quorum Books.

Coleman, Gabriella E. 2005. "The Social Construction of Freedom in Free and Open Source Software: Hackers, Ethics, and the Liberal Tradition." Dissertation Thesis, Department of Anthropology, The University of Chicago, Chicago, IL.

Corry, Chris. 2003. "LucastArts' Star Wars Starfighter." pp. 223-236 in *Postmortems from Game Developer,* edited by Austin Grossman. New York, NY: CMP Books.

Coupland, Douglas. 2006. *JPod*. New York, New York: Bloomsbury Publishing.

Croal, N'Gai. 2007. "EXCLUSIVE: Nintendo of America President Reggie Fils-Aime Tells Level Up About His Big Plans For a Little WiiWare." Website <http://blog.newsweek.com/blogs/levelup/archive/2007/06/27/reggie-on-his-big-plans-for-a-little-nintendo-wiiware.aspx>. Last Accessed: July 20 2007.

Davis, Galen. 2006. "GDC Rant Heard 'Round the World." Gamespot News. Website <http://www.gamespot.com/news/6120449.html>. Last Accessed: September 14 2006.

Deleuze, Gilles and Felix Guattari. 1987. *A Thousand Plateaus: Capitalism and Schizophrenia* Translated by Massumi, Brian. Minneapolis, MN: University of Minnesota Press.

Deuze, Mark, Chase Bowen Martin, and Christian Allen. 2007. "The Professional Identity of Gameworkers." *Convergence* 13.4:Forthcoming.

DMCA. *To amend title 17, United States Code, to implement the World Intellectual Property Organization Copyright Treaty and Performances and Phonograms Treaty, and for other purposes, 17 U.S.C. §§ 512, 1201–1205, 1301–1332; 28 U.S.C. § 4001; 17 U.S.C. §§ 101, 104, 104A, 108, 112, 114, 117, 701*. 1998.

Dobson, Jason. 2007. "Nintendo Issues 'Strong Support' for U.S. Anti-Piracy Measures." Gamasutra.com. Website <http://www.gamasutra.com/php-bin/news_index.php?story=13466>. Last Accessed: July 20 2007.

Donovan, Tristan. 2006. "Top 20 Publishers." *GameDeveloper Magazine* 11-18.

Downey, Greg. 2001. "Virtual Webs, Physical Technologies, and Hidden Workers." *Technology and Culture* 42.2:209-235.

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* 31.3

Dyer-Witheford, Nick and Zena Sharman. 2005. "The Political Economy of Canada's Video and Computer Game Industry." *Canadian Journal of Communication* 30.2

ea_spouse. 2004. "EA: The Human Story." IGDA. Website <http://www.igda.org/articles/easpouse_qol.php>. Last Accessed: January 27 2006.

Edery, David J. 2007. "Console Demise? Don't Hold Your Breath." Game Tycoon. Website <http://www.edery.org/2007/07/console-demise-dont-hold-your-breath/>. Last Accessed: July 23 2007.

Electronic Arts, Inc. 2007. "Form 10-Q: Quarterly Report Pursuant to Section 13 or 15 (d) of the Securities Exchange Act of 1934."

English-Lueck, J. A. and Andrea Saveri. 2001. "Silicon Missionaries and Identity Evangelists." *Anthropology of Work Review* 22.1:7-12.

ESA, Entertainment Software Association. 2005. "2005 Sales, Demographics, and Usage Data: Essential Facts about the Computer and Video GAme Industry." Entertainment Software Association. Website <http://www.theesa.com/files/

2005EssentialFacts.pdf>. Last Accessed: February 5 2008.

Esmurdoc, Caroline. 2005. "Postmortem: Head Games: Double Fine's Psychonautic Break." *Game Developer Magazine* 12.7:30-38.

Fahey, Rob. 2007. "Locked Away: Do the Death Throes of Music DRM Mean Anything for Games?" GamesIndustry.Biz. Website <http://www.gamesindustry.biz/ content_page.php?aid=24585>. Last Accessed: May 1 2007.

Farmer, Paul. 2005. *Pathologies of Power: Health, Human Rights, and the New War on the Poor,* Edited by Rober Borofsky. Berkeley, CA: University of California Press.

Fawkner, Steve. 2007. "Postmortem: Infinite Interactive's Puzzle Quest: Challenge of the Warlords." *Game Developer Magazine* 14.8:42-49.

Feil, John and David Weinstein. 2006. "Game Industry Crediting: A Snapshot of the Present." IGDA. Website <http://www.igda.org/credit/ IGDA_CreditsSnapshot_Apr06.pdf>. Last Accessed: March 2007.

Finley, Alyssa. 2007. "Postmortem: 2K Games' Bioshock." *Game Developer Magazine* 14.10:20-26.

Forsythe, Diana E. 2001. *Studying Those Who Study Us: An Anthropologist in the World of Artificial Intelligence,* Edited by Timothy Lenoir and Hans Ulrich Gumbrecht. Stanford, CA: Stanford University Press.

Fortun, Kim. 2001. *Advocacy after Bhopal: Environmentalism, Disaster, New Global Orders*. Chicago, IL: The University of Chicago Press.

———. 2006. "Poststructuralism, Technoscience, and the Promise of Public Anthropology." *India Review* 5.3-4:294-317.

Fortun, Mike and Herbert J. Bernstein. 1998. *Muddling Through: Pursuing Science and Truths in the 21st Century*. Washington, DC: Counterpoint Press.

Frost, Paul. 2003. "Postmortem: The Tools Development of Turbine's Asheron's Call 2." *Game Developer Magazine* 10.9:42-48.

Fry, Ben. 2003. "deconstructulator." MIT Media Lab. Website <http:/ /acg.media.mit.edu/people/fry/deconstructulator/>. Last Accessed: April 8 2008.

Galison, Peter. 1997. *Image and Logic: A Material Culture of Microphysics*. Chicago, IL: The University of Chicago Press.

GDC. 2007a. "Session Listings for GDC 2007." Website <https://www.cmpevents.com/ GD07/a.asp?option=C&V=1&SB=4>. Last Accessed: November 14 2007.

———. 2007b. "Session Listings for GDC 2008." Website <https:/ /www.cmpevents.com/GD08/a.asp?option=C&V=1&SB=4>. Last Accessed: November 14 2007.

Gill, Rosalind. 2007. *Technobohemians or the New Cybertariat? New Media Work in*

Amsterdam a Decade after the Web. Amsterdam: Institute of Network Cultures.

Gillespie, Tarleton. 2007. *Wired Shut: Copyright and the Shape of Digital Culture*. Cambridge, MA: MIT Press.

———. 2006. "Designed to 'Effectively Frustrate': Copyright, Technology, and the Agency of Users." *New Media & Society* 8.4:651-669.

———. 2004. "Copyright and Commerce: The DMCA, Trusted Systems, and the Stabilization of Distribution." *The Information Society* 20.4:239-254.

Gourdin, Adam. 2005. "Game Developer Demographics: An Exploration of Workforce Diversity." International Game Developers Association. Website <http://www.igda.org/diversity/report.php>. Last Accessed:
Hakken, David. 2000a. "Resocializing Work? Anticipatory Anthropology of the Labor Process." *Futures* 32.767-775.

———. 2000b. "Resocializing Work? The Future of the Labor Process." *Anthropology of Work Review* 21.1:8-10.

Hall, Stuart. 1996. "Gramci's Relevance for the Study of Race and Etnicity." pp. 522 in *Critical Dialogues in Cultural Studies,* edited by David Morely and Kuan-Hsing Chen. New York, NY, USA: Routledge.

Haraway, Donna Jeanne. 1997. *Modest_Witness-AT-Second_Millennium.FemaleMan©_Meets_OncoMouse™ : Feminism and Technoscience*. New York, NY: Routledge Press.

Harvey, David. 1990. *Condition of Postmodernity: An Enquiry into the Origins of Cultural Change*. Oxford, UK
Cambridge, MA: Blackwell Publishing.

Huizinga, Johan. 1971. *Homo Ludens: A Study of the Play-Element in Culture*. Boston, MA: Beacon Press.

Hyman, Paul. 2007. "For Better or Worse: A Quality of Life Update." *GameDeveloper Magazine* June/July 7-11.

ICE, U.S. Immigration and Customs Enforcement. 2007. "Game Over: ICE, Industry Team Up in Gaming Piracy Crackdown: 32 Search Warrants Executed in nationwide intellectual property rights Investigation." Website <http://www.ice.gov/pi/news/newsreleases/articles/070801washington.htm>. Last Accessed: August 2 2007.

Immersion Corporation v. Sony Computer Entertainment America, Inc and Sony Computer Entertainment, Inc. 2007. U.S. App. LEXIS 7984.

Informant and Casey O'Donnell. 2006. "Email - Re: Press Start."

———. 2005a. "Field Notes Amongst Engineers."

———. 2005b. "GDC 2006 Abstract Submission: Where is my Make Art Button?

Improving the Artist/Programmer Relationship."

———. 2005c. "Interview with an Artist."

———. 2004. "Interview with an Engineering Group Manager."

———. 2007. "Interview with an Engineer."

———. 2004. "Interview with a Tech-Artist."

Johns, Jennifer. 2006. "Video Game Production Networks: Value Capture, Power Relations, and Embeddedness." *Journal of Economic Geography* 6.151-180.

Kazemi, Darius. 2007. "Breaking In: Then and Now." Blogger.com. Website <http://tinysubversions.blogspot.com/2007/08/breaking-in-then-and-now.html>. Last Accessed: August 7 2007.

Kelly, John D. 2006. *The American Game: Capitalism, Decolonization, World Domination, and Baseball*. Chicago, IL: Prickly Paradigm Press.

Kent, Steven L. 2001. *The Ultimate History of Video Games: The Story Behind the Craze that Tourched our Lives and Change the World*. New York, NY: Three Rivers Press.

Kline, Stephen, Nick Dyer-Witherford, and Greig de Peuter. 2005. *Digital Play: The Interaction of Technology, Culture, and Marketing*. Québec, Canada: McGill-Queen's University Press.

Knorr-Cetina, Karin D. 1983. "The Ethnographic Study of Scientific Work: Towards a Constructivist Interpretation of Science." pp. 115-140 in *Science Observed: Perspectives on the Social Study of Science,* edited by Karin D. Knorr-Cetina and Michael Mulkay. Beverly Hills, CA: Sage Publications.

———. 1999. *Epistemic Cultures: How the Sciences Make Knowledge*. Cambridge, MA: Harvard University Press.

Krahulik, Mike and Jerry Holkins. 2007. "Old School." Penny Arcade. Website <http://www.penny-arcade.com/comic/2007/12/03/>. Last Accessed: April 8 2008.

Lashinsky, Adam. 2007. "Search and Enjoy (Cover Story)." *Fortune Magazine* 155.1:70-82.

Latour, Bruno. 1987. *Science in Action: How to Follow Scientists and Engineers Through Society*. Cambridge, MA: Harvard University Press.

———. 1991. "Technology is Society Made Durable." in *A Sociology of Monsters: Essays on Power, Technology and Domination,* edited by John Law. New York, NY: Routledge Press.

———. 1999. *Pandora's Hope: Essays on the Reality of Science Studies*. Cambridge, MA: Harvard University Press.

Latour, Bruno and Steve Woolgar. 1986. *Laboratory Life: The Construction of Scientific Facts*. Princeton University Press.

Law, John. 1989. "Technology and Heterogeneous Engineering: The Case of Portuguese Expansion." pp. 111-134 in *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology,* edited by Wiebe Bijker, Thomas P. Hughes, and Trevor Pinch. Cambridge, MA: MIT Press.

Lessig, Lawrence. 2005. *Free Culture: The Nature and Future of Creativity*. New York, NY: Penguin Press.

———. 1999. *Code and Other Laws of Cyberspace*. Jackson, TN: Basic Books.

Lik-Sang.com. 2006. "Lik-Sang.com Out of Business due to Multiple Sony Lawsuits." http://www.lik-sang.com. Website <http://www.furl.net/search?search=cache&id=19501972&url=http%3A%2F%2F209.85.165.104%2Fsearch%3Fq+%3Dcache%3AKTQmcrxScBwJ%3Awww.lik-sa+ng.com%2F+lik+sang%26hl%3Den%26ct%3Dclnk%26cd%3D1%26gl%3Dus%26client%3Dfirefox-a>. Last Accessed: March 22 2007.

Lyotard, Jean-François. 1984. *The Postmodern Condition: A Report on Knowledge,* Edited by Wlad Godzich and Jochen Schulte-Sasse. Translated by Bennington, Geoff
Massumi, Brian. Minneapolis, MN: University of Minnesota Press.

Malliet, Steven and Eric Zimmerman. 2005. "The History of the Video Game." pp. 23-46 in *Handbook of Computer Game Studies,* edited by Joost Raessens and Jeffrey Goldstein. Cambridge, MA: The MIT Press.

Martin, Emily. 1997. "Managing Americans: Policy, Work, and the Self." pp. 239-260 in *Anthropology of Policy: Perspectives on Governance and Power,* edited by Cris Shore and Susan Wright. New York, NY: Routledge.

Microsoft. 2006. "Microsoft Invites the World to Create its Own Xbox 360 Console Games for the First Time." Microsoft Press Release. Website <http://www.microsoft.com/presspass/press/2006/aug06/08-13XNAGameStudioPR.mspx>. Last Accessed: August 13 2006.

Mizrachi, Eliot and Robert Khoo. 2006. "ESRB and Penny Arcade Team Up to Create New Gamer PSA Campaign." Website <http://www.esrb.org/about/news/downloads/psa_gamer_campaign.pdf>. Last Accessed: April 8 2008.

Murata, Taku. 2007. "Postmortem: Final Fantasy XII." *Game Developer Magazine* 14.7:22-27.

Nakagawa, Katsuya. 1985. U.S. Patent No. 4799635. Washington, DC: U.S. Patent and Trademark Office. Assignee: Nintendo Co., Ltd.

Nakagawa, Katsuya and Masayuki Yukawa. 1987. U.S. Patent No. 4865321. Washington, DC: U.S. Patent and Trademark Office. Assignee: Nintendo Co., Ltd.

Neff, Gina. 2005. "The Changing Place of Cultural Production: The Location of Social

Networks in a Digital Media Industry." *The Annals of the American Academy of Political and Social Science* 597.134-152.

Neff, Gina and David Stark. 2004. "Permanently Beta: Responsive Organization in the Internet Era." pp. 173-188 in *Society Online: The Internet in Context,* edited by Philip N. Howard and Steve Jones. Thousan Oaks, CA, USA: Sage Publications.

Neff, Gina, Elizabeth Wissinger, and Sharon Zukin. 2005. "Entrepreneurial Labor among Cultural Producers: "Cool" Jobs in "Hot" Industries." *Social Semiotics* 15.3:307-334.

Nintendo. 2007. "Nintendo's WiiWare Paves the Way for Fresh Games, Cool Consumer Experiences." Website <http://www.nintendo.com/newsarticle?articleid=lJUgYjCGf4pVbYMAU2qmwF3wp7DiOK5k&page=newsmain>. Last Accessed: July 20 2007.

———. 2003. "NES Licensed Game List." Nintendo. Website <http://www.nintendo.com/doc/nes_games.pdf>. Last Accessed: June 6 2007.

Nintendo Co. LTD. and Nintendo of America Inc. v. Lik Sang International, LTD. 2003. HCA 3584/2002.

Nishiumi, Satoshi, Kazuo Koshima, Shigeru Miyamoto, and Yasunari Nishida. 2005. U.S. Patent No. 7102618. Washington, DC: U.S. Patent and Trademark Office. Assignee: Nintendo Co., Ltd.

Nishiumi, Satoshi, Kazuo Koshima, and Nobuyuki Nonaka. 2001. U.S. Patent No. 6676520. Washington, DC: U.S. Patent and Trademark Office. Assignee: Nintendo Co., Ltd.

O'Donnell, Casey. 2004. "A Case for Indian Insourcing: Open Source Interest in IT Job Expansion." *First Monday* 9.11:http://www.firstmonday.org/issues/issue9_11/odonnell/index.html.

Ogata, Hiroki, Toru Akazawa, and Akihisa Ono. 1998. U.S. Patent No. 6171191. Washington, DC: U.S. Patent and Trademark Office. Assignee: Sony Computer Entertainment, Inc.

Omi, Michael and Howard Winant. 1994. *Racial Formation in the United States: From the 1960s to the 1990s*. New York: Routledge.

Ong, Aihwa. 1991. "The Gender and Labor Politics of Postmodernity." *Annual Review of Anthropology* 20.279-309.

Orr, Julian E. 1991. "Contested Knowledge: Work, Practice and Technology." *Anthropology of Work Review* 12.3:12-17.

———. 1996. *Talking about Machines: An Ethnography of a Modern Job,* Edited by Stephen R. Barley. Ithaca, NY: Cornell University Press.

Pentland, Brian T. 1997. "Bleeding Edge Epistemology: Practical Problem Solving in Software Support Hot Lines." in *Between Craft and Science: Technical Work in*

U.S. Settings, edited by Stephen R. Barley and Julian E. Orr. Ithaca, NY: Cornell University Press.

Perrow, Charles. 1999. *Normal Accidents: Living with High-Risk Technologies.* Princeton, NJ: Princeton University Press.

Pickering, Andrew. 1995. *The Mangle of Practice: Time, Agency, and Science*. Chicago, IL: University of Chicago Press.

Pinch, Trevor and Wiebe Bijker. 1989. "The Social Construction of Facts and Artifacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other." pp. 17-50 in *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology,* edited by Wiebe Bijker, Thomas P. Hughes, and Trevor Pinch. Cambridge, MA: MIT Press.

Pinch, Trevor, Harry M. Collins, and Larry Carbone. 1997. "Cutting Up Skills: Estimating Difficulty as an Element of Surgical and Other Abilities." in *Between Craft and Science: Technical Work in U.S. Settings,* edited by Stephen R. Barley and Julian E. Orr. Ithaca, NY: Cornell University Press.

Pinckard, Jane. 2007. "Game Girl Advance: Jade Raymond is for Real." http://www.gamegirladvance.com/. Website <http://www.gamegirladvance.com/archives/2007/11/16/jade_raymond_is_for_real.html>. Last Accessed: November 16 2007.

Postigo, Hector. 2003. "From Pong to Planet Quake: Post-Industrial Transitions from Leisure to Work." *Information, Communication & Society* 6.4:593-607.

Price, Ted. 2003. "Postmortem: Insomniac Games' Ratchet & Clank." *Game Developer Magazine* 10.6:52-60.

Pritchard, Matt. 2003. "Ensemble's Age of Empires." pp. 63-74 in *Postmortems from Game Developer,* edited by Austin Grossman. New York, NY: CMP Books.

Radd, David. 2007. "ESA Applauds San Diego Raid." Ed. Libe Goad GameDaily.Biz. Website <http://biz.gamedaily.com/industry/news/?id=16634>. Last Accessed: July 20 2007.

Ragaini, Toby. 2003. "Turbine's Asheron's Call." pp. 299-309 in *Postmortems from Game Developer,* edited by Austin Grossman. New York, NY: CMP Books.

Rheinberger, Hans-Jörg. 1997. *Toward a History of Epistemic Things: Synthesizing Proteins in the Test Tube*. Stanford, CA: Stanford University Press.

Rogers, Everett M. and Judith K. Larsen. 1984. *Silicon Valley Fever: Growth of High-technology Culture*. New York, NY, USA: Basic Books.

Ronell, Avital. 2005. *The Test Drive*. Chicago, Illinois: University of Illinois Press.

Ross, Andrew. 2003. *No-Collar: The Humane Workplace and Its Hidden Costs*. Jacson, TN: Basic Books.

Salen, Katie and Eric Zimmerman. 2004. *Rules of Play: Game Design Fundamentals*. Cambridge, MA: MIT Press.

Scarselletta, Mario. 1997. "The Infamous "Lab Error": Education, Skill, and Quality in medical Technicians' Work." in *Between Craft and Science: Technical Work in U.S. Settings,* edited by Stephen R. Barley and Julian E. Orr. Ithaca, NY: Cornell University Press.

Schadt, Toby. 2007. "Postmortem: Not Your Typical Grind - Tony Hawk's Downhill Jam for Wii." *Game Developer Magazine* 14.1:30-37.

Schaefer, Erich. 2003. "Blizzard Entertainment's Diablo II." pp. 79-90 in *Postmortems from Game Developer,* edited by Austin Grossman. New York, NY: CMP Books.

Schüll, Natasha Dow. 2005. "Digital Gambling: The Coincidence of Desire and Design." *The Annals of the American Academy of Political and Social Science* 597.1:65-81.

Scott, James C. 1976. *Moral Economy of the Peasant: Rebellion and Subsistence in Southeast Asia*. New Haven, CT: Yale University Press.

Sheff, David. 1993. *Game Over: How Nintendo Zapped an American Industry, Captured Your Dollars, and Enslaved Your Children*. New York, NY: Random House Inc.

Shirai, Ichiro. 1985. U.S. Patent No. 4687200. Washington, DC: U.S. Patent and Trademark Office. Assignee: Nintendo Co., Ltd.

Simon, Roger. 2001. *Gramsci's Political Thought: An Introduction*. London, UK: ElecBook.

Smith, Dorothy E. 1999. *Writing the Social: Critique, Theory, and Investigations*. Buffalo, NY: University of Toronto Press.

Smith, Marcus. 2007. "Postmortem: Resistance: Fall of Man." *Game Developer Magazine* 14.2:28-36.

Smith, Vicki. 2001. "Ethnographies of Work and the Work of Ethnographers." pp. 220-233 in *Handbook of Ethnography,* edited by Paul Atkinson, Amanda Coffey, Sara Delamont, John Lofland, and Lyn H Lofland. Sage Publications.

Snow, Blake. 2007. "The Future of Video Games According to 1982." Joystiq. Website <http://www.joystiq.com/2007/01/30/the-future-of-video-games-according-to-1982/>. Last Accessed: January 30 2007.

Sony Computer Entertainment, Inc. 2005. "Sony Computer Entertainment Inc. to Launch its Next Generation Computer Entertainment System, Playstation 3 in Spring 2006."

Sony Computer Entertainment, Inc v. Lik Sang International, LTD. 2003. HCA 3583/2002.

Sprigman, Chris. 2002. "The Mouse that Ate the Public Domain: Disney, The Copyright Term Extension Act, and Eldred v. Ashcroft." FindLaw. Website <http://writ.news.findlaw.com/commentary/20020305_sprigman.html>. Last Accessed:

April 25 2007.

Staff. 2007. "Analysts: Single Console Future 'Bad for the Industry'." Gamasutra. Website <http://www.gamasutra.com/php-bin/news_index.php?story=16277>. Last Accessed: November 16 2007.

Stengers, Isabelle. 1997. *Power and Invention: Situating Science,* Edited by Sandra Buckley, Michael Hardt, and Brian Massumi. Minneapolis, MN: University of Minnesota Press.

Suchman, Lucy. 1995. "Making Work Visible." *Communications of the ACM* 38.9:56-64.

Suchman, Lucy, Jeanette Blomberg, Julian E. Orr, and Randall Trigg. 1999. "Reconstructing Technologies as Social Practice." *American Behavioral Scientist* 43.3:392-408.

Sutton-Smith, Brian. 1998. *The Ambiguity of Play* . Cambridge, MA: Harvard University Press.

Taussig, Michael. 1992. *The Nervous System*. New York, NY: Routledge.

Taylor, T. L. 2006a. *Play Between Worlds: Exploring Online Gaming Culture*. Cambridge, MA: MIT Press.

———. 2006b. "Does WoW Change Everything?: How a PvP Server, Multinational Player Base, and Surveillance Mod Scene Caused Me Pause." *Games and Culture* 1.4:318-337.

Terranova, Tiziana. 2000. "Free Labor: Producing Culture for the Digital Economy." *Social Text* 18.2:33-58.

Train, Tim. 2003. "Postmortem: Big Huge Games' Rise of Nations." *Game Developer Magazine* 10.7:36-41.

Traweek, Sharon. 1988. *Beamtimes and Lifetimes: The World of High Energy Physicists*. Cambridge, MA: Harvard University Press.

———. 2000. "Faultlines." pp. 21-48 in *Doing Science + Culture: How Cultural and Interdisciplinary Studies are Changing the Way We Look at Science and Medicine,* edited by Roddey Reid and Sharon Traweek. New York, NY: Routledge Press.

Tsing, Anna Lowenhaupt. 2005. *Friction: An Ethnography of Global Connection*. Princeton, New Jersey: Princeton University Press.

Turkle, Sherry. 1997. *Life on the Screen: Identity in the Age of the Internet*. New York, NY: Simon & Schuster.

Ueda, Hiroo and Hiromitsu Yagi. 1987. U.S. Patent No. 4824106. Washington, DC: U.S. Patent and Trademark Office. Assignee: Richo Co. Ltd. and Nintendo Co. Ltd.

Varma, Roli. 2006. *Harbingers of Global Change: India's Techno-Immigrants in the*

United States. Lanham, MD: Lexington Books.

Wark, McKenzie. 2007. *Gamer Theory*. Cambridge, MA: Harvard University Press.

Waters, Darren. 2007. "EA Wants 'Open Gaming Platform'." bbc.co.uk. Website <http:/ /news.bbc.co.uk/1/hi/technology/7052420.stm>. Last Accessed: October 23 2007.

Weber, Tim. 2007. "Games Industry Enteres a New Level." BBC News Website. Website <http://news.bbc.co.uk/2/hi/business/6523565.stm>. Last Accessed: June 4 2007.

Wen, Howard. 2007. "Analyze This: Will There Ever Be One Console to Rule Them All?" Gamasutra. Website <http://www.gamasutra.com/view/feature/2012/ analyze_this_will_there_ever_be_.php>. Last Accessed: November 16 2007.

Whalley, Peter and Stephen R. Barley. 1997. "Technical Work in the Division of Labor: Stalking the Wily Anomaly." in *Between Craft and Science: Technical Work in U.S. Settings,* edited by Stephen R. Barley and Julian E. Orr. Ithaca, NY: Cornell University Press.

Williams, Dmitri. 2002. "Structure and Competition in the U.S. Home Video Game Industry." *The International Journal on Media Management* 4.1:41-54.

Wilson, Phil. 2007. "Postmortem: Realtime Worlds' Crackdown." *Game Developer Magazine* 14.9:26-31.

Wong, Nicole. 2006. "Exclusive: Nicole Wong Reveals Identity of EA Spouse." The Mercury News. Website <http://blogs.mercurynews.com/aei/2006/04/ exclusive_nicol.html>. Last Accessed: April 25 2006.

Yee, Nick. 2006. "The Labor of Fun: How Video Games Blur the Boundaries of Work and Play." *Games and Culture* 1.1:68-71.

Yen, Chung-Tien and Po-Jen Shen. 2003. U.S. Patent No. 6950379. Washington, DC: U.S. Patent and Trademark Office. Assignee: MediaTek, Inc.

Zabusky, Stacia E. 1997. "Computers, Clients, and Expertise: Negotiating Technical Identities in a Nontechnical World." pp. 129-153 in *Between Craft and Science: Technical Work in U.S. Settings,* edited by Stephen R. Barley and Julian E. Orr. Ithaca, NY: Cornell University Press.