

Citation: Lowood, Henry. "Found Technology: Players as Innovators in the Making of Machinima." *Digital Youth, Innovation, and the Unexpected*. Edited by Tara McPherson. The John D. and Catherine T. MacArthur Foundation Series on Digital Media and Learning. Cambridge, MA: The MIT Press, 2008. 165–196. doi: 10.1162/dmal.9780262633598.165

Copyright: © 2008 Massachusetts Institute of Technology. Published under Creative Commons Attribution-Noncommercial-No Derivative Works Unported 3.0 license.

Found Technology: Players as Innovators in the Making of Machinima

Henry Lowood

Stanford University, University Libraries

... while to adults the Internet primarily means the world wide web, for children it means email, chat, games—and here they are already content producers. Too often neglected, except as a source of risk, these communication and entertainment-focused activities, by contrast with the information-focused uses at the centre of public and policy agendas, are driving emerging media literacy. . . . Bearing in mind that the elite realm of high culture has already been breached, who is to say that this form of content creation counts for little?—Sonia Livingstone¹

The focus of this essay is a new narrative medium, called machinima, which has sprung out of computer game technology and play since the mid-1990s. Machinima is “filmmaking within a real time, 3D virtual environment.”² This means producing animated movies with the software that is used to develop and play computer games. The growth of machinima as a creative medium within digital game culture certainly reveals much about the creativity of players. Homing in on machinima as a form of player-driven innovation also teaches us lessons about how and what players learn from game technology and gameplay, how they share knowledge about what they learn, and how these engagements lead them to come to grips with issues that threaten to limit the potential of machinima as an expressive medium. Player creativity and innovation then is the focus of this investigation of machinima as an outgrowth of computer game culture. Computer games are providing new opportunities for performance and expression mediated by computers and networks; players enjoy game-based replay movies and machinima videos, and they are equally fascinated by the activities of the player behind them as they try to understand the mix of skills and tricks that go into making them. Put another way, the development of machinima over roughly a decade is about how players have learned to master computer technology, gameplay, and performance practices. How players have learned to be creative may well be the most significant story in game culture today.

Introduction: Koulamata’s “The French Democracy”

For many observers outside game culture, perhaps the first visible example of the expressive power unlocked by machinima was provided by a game-based movie created in November of 2005, Koulamata’s “The French Democracy.”³ The story behind its creation shows us that community players—players who create and circulate game-based performances within communities of game players—can contribute to public discourse about current events. This is an important enough claim, but just as significantly, “The French Democracy” provides a convincing example for another assertion: machinima has become accessible, putting a



Figure 1
Koulamata (Alex Chan), "The French Democracy" (2005).

powerful technology for real-time animation production in the hands of a widening circle of players. In other words, digital games and game technology have a place among new media forms that have translated entertainment technologies into engines for the creation of content. Sonia Livingstone has described the impact of networked media in empowering children and young adults as "content producers." The example of Koulamata's "The French Democracy" suggests that game-based moviemaking is capable of empowering a variety of age groups as moviemakers.

Let us begin by taking a closer look at Koulamata's project. In late October 2005, riots broke out in the largely African and Arab Parisian suburb of Clichy-sous-Bois and other parts of France after incendiary remarks by Interior Minister Nicolas Sarkozy and the electrocution of two teenagers fleeing from police in an electrical substation. These riots spread and smoldered for about two weeks, reaching a peak between November 6 and 8, followed by the declaration of emergency powers on November 9 to quell the violence. Concurrently, on November 8, Peter Molyneux's Lionhead Studios in England released a game called *The Movies*, part Hollywood studio simulation, part toolkit for making animated movies by setting up simple scenes and manipulating actor avatars. The obvious reason for players to make these movies is that as virtual studio heads, that became their job within the simulation space. However, many players found a different reason to play the game: they figured out how to move straight to the movie studio and the production of movies. This was no surprise to Molyneux, who revealed that "one of the dreams for the game was that as you play, you realize you could direct a movie of your own."⁴ His vision of the player as creator was confirmed within days of the game's release, as dozens and then hundreds of movies were posted to Lionhead's community Web site. About two months after the game's release, more than 3,000 movies were already available for viewing on this site⁵ alone. Over a year later, in November 2006, movies were still being uploaded to this Web site at a rate of nearly seventy per day, not to mention those posted to viral media and other distribution points on the World Wide Web.⁶

Back in the Parisian suburb of Seine-Saint-Denis, twenty-seven-year-old Alex Chan, a freelance industrial designer with absolutely no experience making movies, decided to buy *The Movies* and make his own filmic statement about the French riots. It was still

mid-November 2005; the turmoil was just subsiding around Paris and the game was brand new. These two unrelated facts were connected by Chan's drive to craft an immediate response to the events around him. As Chan put it himself, "through these tools you can get some more spontaneous reaction or reflection, not from mass media, but from a simple citizen like me."⁷ Remarkably, under the name Koulamata, he was able to post his thirteen-minute movie to the Lionhead site on November 22, hardly two weeks after both the end of the riots and the release of the game. His movie told several stories about the victimization of French minority groups through harassment, job discrimination, and daily events in an attempt to explain the tensions and emotions that fueled the riots. Despite awkward English-language subtitles and cinematography—or perhaps because of these qualities—the movie comes across as immediate and authentic, directly striking a heartfelt note of sorrow for the loss of French historical ideals such as liberty and fraternity. MTV News admitted that the first film about these events came "courtesy of a video game."

"The French Democracy" was rapidly distributed through numerous game Web sites, blogs, and movie download sites on the Web and ignited a passionate discussion on "The Movies" Web site and elsewhere. On the Lionhead site, for example, more than 400 viewers rated the movie within the first six weeks of its release; a year later, in November 2006, viewers were still posting comments. The main thread for discussion of the movie offers a similar number of comments, both in French and English, from the very first posting on November 22 ("Magnifique, tout simplement! Very great job!") to reflections that document the continuing resonance of this project around the world nearly two months later ("I think those events can be linked to our own here in Quebec in 1970 when our own country called the army on us.") These examples document the roles that gameplay and the online forums in which communities of game players gather can powerfully extend participation in public discourse about current events in "real life," not just game culture. They also suggest the new modes of informal learning that were taking place in these forums, as participants traded ideas about the events and the project in ways set up by engagement with the game. News of Koulamata's project also flowed into mainstream media channels. It was covered by a range of outlets from *The Washington Post*, *USA Today*, MTV.com, and *Business Week* to the important Socialist daily, *Libération*.⁸ It was shown widely via viral media and machinima sites on the Web and continued to be shown during 2006 at venues such as the Centre Pompidou's Flash Festival in May and the World Wide Short Film Festival the following June in Toronto.⁹ By then, Chan's movie had become something more than a commentary on recent events in his own country. It had become a poster for the potential of machinima as a medium. Paul Marino, a veteran machinima artist and executive director of the Academy of Machinima Arts & Sciences, concluded that Chan's movie had proven "that Machinima can be a powerful medium—showing that it can extend the thoughts of individuals into areas the game developers might not be so quick to embrace."¹⁰ Xavier Lardy, founder of the leading French Web site devoted to machinima, told *USA Today* that "there has never been a machinima with such a clear and prominent political message."¹¹

During the unrest and in its aftermath, Jacques Chirac publicly bemoaned a lack of diversity and minority representation in French media. He insisted that mainstream media must "better reflect the reality of France today" precisely as Koulamata was learning to use *The Movies* in order to answer this challenge in his own way.¹² Subsequent discussion in France during December 2005 focused on the specific solution of the hiring of more minority television reporters and newscasters. This proposal was important, but it bypassed equally critical issues surrounding point of view and the power to create new narratives. In this sense, the public

discussion missed the contribution of “The French Democracy.” Indeed, Chan told reporters that he made a game-based movie in order to bypass mainstream media as well as to correct the errors carried by them. So, the story with regard to Chan’s project is that it filled a gap not only in minority representation, but also by utilizing a new format that while under the mainstream media radar, nevertheless responded immediately, was widely viewed, and offered a point of view that was different from what could be seen on established news outlets. The Web-magazine *Alterités* of the Cité nationale de l’histoire de l’immigration, which has highlighted media-related issues of concern to French immigrant groups, pointed out in December that from “blogs to videogames,” new media were dedicating themselves to “all the youth of color in the world.” Mogniss H. Abdallah cited “The French Democracy” in an article for *Alterités* specifically as evidence that “technological innovations are being used to satisfy the thirst for public expression (expression ambiante).”¹³

It is of course important to acknowledge that the production and circulation of game-based content might open up new means of public and creative expression, and not just for gamers. And yet, reducing a positive assessment of gameplay to its potential for leading to “more serious” forms of activity is neither the only nor the most compelling rationale for turning our attention to machinima. From a strictly historical perspective, this move wrenches the motives of game players out of the original context of their innovative uses of game technology. Overvaluing “serious” goals as an end point of the creative use of entertainment technologies leads to an anachronistic, even Whiggish justification for these uses of technology: the value of those activities *then* is seen through a lens that only shows us their recontextualized value *now*. If that were the only problem, however, perhaps only historians would have reason to be upset. If the payoff of projects like “The French Democracy” adds value to public discourse or creates new outlets for artistic expression, so what if we misread the interests and motivations of the players who created the techniques that can be exploited in this fashion? In fact, there really are important reasons to pay attention to the lessons we might learn from machinima. The first is that it illustrates the full reach of the do-it-yourself culture that has emerged so powerfully from game culture, not just for modifying game technology and content, but also as a move across media technologies from interactive digital games to animated filmmaking. Second, if we fail to contextualize the emergence of the practices associated with machinima production inside player communities, we lose the essential association of these practices to a mix of technology, gameplay, and public performance inside game culture. Perhaps most importantly, the history of machinima provokes discussion of frequently debated points in the valuation of new media associated with “fans, bloggers, and gamers,”¹⁴ as well as others of their ilk: in unlocking game-based moviemaking, what have players learned about connecting innovative uses of technology to the emergence of new cultural modalities made available to them?

This essay addresses these questions by presenting three takes on the development of machinima. *Take One*: What do the origins of game-based moviemaking tell us about the relationship among learning, demonstration, and performance in the uses of technology associated with computer games? The story about how game players learned to make machinima becomes easier to understand when it becomes more than an answer to the question, “who invented machinima?” It also must show how some players learned to use this technology and develop practices of extroverted performance and production around it, as well as how others acquired skills as spectators and began to watch these movies. *Take Two*: How has the status of machinima creators as players and users of game technology redefined them as players—or as “moviemakers”? How has the dependence of machinima on game engines,

that is, on a particular technology, interacted with the context of machinima as emerging from game culture and player communities? Following the lead taken by recent technology studies, this take explores not only whether players as consumers matter, but also how *use* matters.¹⁵ I will explore the notion of the “game engine,” stabilized in games like *DOOM* and *Quake* that also propelled the creation of replays and machinima, as a “found technology,” ready-made for appropriation by players. *Take Three*: As machinima emerges from the geeky edge of game culture, does the context of its creation offer special advantages, or alternatively, noteworthy constraints for convergence with other forms of cultural expression? We know that today a young Frenchman can acquire an animation studio in the form of a commercial game costing about \$50. Might “The French Democracy” be a signal that game technology and culture has established new means for cultural and political expression? Or will limitations embedded in this culture ranging from steadfast self-referentiality to scant regard for intellectual property throttle the expansion of machinima spectatorship to a wider audience?

Public Demo: Origins of Game-Based Moviemaking

In December 1993, id Software released the first-person action game *DOOM*. The authors were a group of programmers led by John Romero and John Carmack, who had founded id in February 1991 to develop shareware games for personal computers. From the beginning, Carmack in particular focused on the development of cutting-edge game technology as a platform for frantically paced, competitive action games. *DOOM*'s gameplay, for example, consisted largely of slaughtering (shooting) demons or opponents in the game world. What was innovative about id's games was that they depicted this world as the player's character would see it, from a 3D first-person perspective that was rapidly redrawn as that character moved through the environment. This innovation established a game genre that would dominate the development of game technology during the 1990s: the “first-person shooter” (FPS).

DOOM immediately left its imprint on almost every aspect of computer gaming, from graphics and networking technology to styles of play, notions of authorship, and public scrutiny of content. Id followed up, in June 1996, with the release of *Quake*, built on the modes of competitive play introduced with *DOOM*. *Quake* was a technological tour de force. Its built-in client/server networking was an improvement over the peer-to-peer networking in *DOOM*, greatly stimulating the popularity of internet-based multiplayer games. *Quake* also offered what is generally acknowledged as the first graphics engine capable of providing true 3D graphics.¹⁶ The reception of *DOOM* and *Quake* offers an object lesson in the importance of distinguishing public reactions to game content from practices associated with the use of technology provided by the same games. The subject matter of *DOOM* (bloody, rapid-fire killing of demons in outer space), its moody graphics and audio, and the vocabulary associated with its gameplay (“shooters,” “death match”) increased public attention to the levels of violence depicted in computer games. Coincidentally, the game was released just as congressional hearings convened by senators Herb Kohl and Joseph Lieberman were getting underway to examine media violence and its influence on children, and of course a few years later *DOOM* came under particularly close scrutiny in connection with the tragic shootings at Columbine High School. The portrayal of computer games such as *DOOM* or *Quake* as “murder simulators”¹⁷ and “bloodlust” games obviously reflected on players, who were often portrayed as disconnected and desensitized, or as engaged in mindless, repetitive, or addictive play.

I do not mean here to enter debates about the presentation of violent content in computer games or its effects on players. I am looking instead at how *DOOM* and especially *Quake* opened up the medium of the computer game to unexpected kinds of player creativity. Practices associated with FPS play and the technical configuration of these games together made new modes of creative play possible, among which were making animated movies using id's real-time graphics engines. What was it about these games' technology that inspired player creativity? In order to answer this question, we have to begin with the idea of the "game engine" as worked out by Carmack. Inspired by programming hacks that altered games, such as the "Barney patch" for Silas Warner's original *Castle Wolfenstein* (Muse Software, 1981), he had often altered computer games as a teenager. He built modifiability into *DOOM*, but in a manner that simplified the process and didn't require such hacks. He did this by separating the core "game engine" from the code for specific "levels" of the game defined by maps, objects, monsters, graphics, sound, and so on, which came to be called the "game assets." The explicit intention of this separation was to make it possible for players to create their own content by designing their own maps or "levels." Level-specific information was captured in so-called "wad" files, which were loaded separately into the game to play these levels; editing or creating wad files changed a game's content without hacking at the game engine. It thus protected critical software code that produced fundamental functionalities of the game, such as graphics or physics. Carmack's design decision to separate game engine from assets spawned independent and third-party level design, as id expected, and encouraged the development of software tools to make new content, whether by id, other companies, or players.¹⁸ The game engine also represented a new business model for id, now positioned as a game technology company as well as a game developer. Standardizing game production also opened up the gates for a flood of player-created content, which in turn would extend the commercial life of the game. According to id's own corporate history:

The team of innovators also made *DOOM*'s source code available to their fan base, encouraging would-be game designers to modify the game and create their own levels, or 'mods.' Fans were free to distribute their mods of the game, as long as the updates were offered free of charge to other enthusiasts. The mod community took off, giving the game seemingly eternal life on the Internet.¹⁹

Beginning with *DOOM*, much of the fun began after an FPS game was published, as players (and other independent designers) would create their own levels and maps, or modify other game assets. Computer games such as *Quake* essentially could function as design tools, and learning how to use id's software to make mods, even the smallest personalization of the game, became part of the player's engagement. This also meant a "letting go of authorial control" by game developers that media curator Randall Packer has called a distinguishing characteristic of computer games, in contrast to other artistic or entertainment media.²⁰ And why not give up some of this control? Carmack was particularly skeptical about the game designer as auteur, arguing in one widely circulated statement that "there is not a hell of a lot of difference between what the best designer in the world produces, and what quite a few reasonably clued in players would produce at this point."²¹ Indeed, players came to see customization of *Quake* as another way to show other players the superiority of their skills.

Carmack's configuration of the game engine and assets was not the only aspect of his games that unexpectedly led to game-based moviemaking. The other was more closely tied to gameplay than game development. *DOOM* could be played as a single player or even as a cooperative game, but it also introduced a new mode of competitive play devised by id cofounder John Romero called "deathmatch." Players connected via Local Area Networks to



Figure 2
The Rangers, “Diary of a Camper” (1996).

compete by recording kills (or “frags”) against opponents. *Quake* was fundamentally a multiplayer and competitive game, and was playable over the Internet. Immediately after *Quake*'s release, players formed groups held together by its improved multiplayer connectivity and chat options. Like hacker gangs dissecting the intricacies of computer networks, these *Quake* Clans shared techniques of high-performance gaming, both play *and* programming. Multiplayer was not just linking up to frag opponents online; it also described chat, discussion, and sharing of exploits *among* players. Competition led to community; networked death match led to networks of players. The Ranger Clan provides a telling example. Arguably the most famous clan of all, its top-notch players also helped shape the *Quake*'s technically inclined community. They had participated in the first prerelease test of the *Quake* engine. One member designed the original Capture the Flag mod; another founded one of the major sources of information about *Quake* development, Blue's News; in all, about half the twenty-five or so members remained active in game development or went on to work in the game industry.²² Being in such a clan meant being a community player; it meant visibly performing skills and demonstrating abilities, showing how to do things, spreading information, and building software tools and content to share with other players. With their reputation for stellar performances as players and programmers firmly established, the Rangers surprised the *Quake* community in October 1996 with a new kind of exploit. Barely a month after its commercial release, they circulated a movie made inside *Quake*, “Diary of a Camper.” It established id's first-person shooter as the first platform for game-based movies, known for several years simply as *Quake* movies.

These *Quake* movies were closely related to a form of in-game performance known as the demonstration or “demo” movie. The demo as a means of performance, skills certification, and training runs through the history of computer game culture. It goes back to *Spacewar!*, arguably the first modern computer game. Developed at M.I.T. in 1962 by a group led by



Figure 3

NoSkill, *DOOM2* demo, 1995.

Steve Russell, *Spacewar!* established the computer game as a method for demonstrating the capabilities of computers. Russell's group disdained previous methods for showing off what a computer could do and believed that a good demonstration "should involve the onlooker in a pleasurable and active way—in short, it should be a game."²³ *Spacewar!* did just that, showcasing the new PDP-1 computer, its graphics, I/O, and display technology; the M.I.T. group confidently told the new PDP users' community that *Spacewar!* "amply demonstrates the real-time capabilities of the PDP-1" and verified its "excellent performance."²⁴ But at the same time as it showed what computers could do, this game demonstrated the technical mastery of programmers and hardware hackers. An immensely popular competitive game available in any U.S. computer science laboratory of the 1960s and 1970s, *Spacewar!*'s community of programmers and players grew rapidly. Stewart Brand, writing in the early 1970s, considered this game to be a "flawless crystal ball of things to come in computer science and computer use." He cited it as evidence of a new culture, part coproduction, part player performance. In "days of batch processing and consumerism," it was a "heresy, uninvited and unwelcome. The hackers made *Spacewar!*, not the planners."²⁵

So public demonstration as a mode of computer-based performance was associated with computer games right from their origins in academic computer science laboratories. *Spacewar!* players demonstrated computer-mediated performance through play. Recasting the player as a performer settled into not one, but two predominant modes: the superior player, the God of the joystick and mouse, and the player-programmer able to hack into game code and show off mastery of the technology. Both sorts of performance occurred as public demonstrations in ways that would later influence the use of games to make movies, while also illustrating the degree to which important kinds of experimentation and informal learning have long accompanied computer gaming.

Technically, *DOOM* demos were essentially replay files, saved sequences of instructions from a previously played game that, when executed by the game software, would show the same game from the same (first-person) perspective of the original player. As one guide put

it, “in the *DOOM/DOOM II* universe, the term “demo” refers to a file that contains a recorded session of gameplay.”²⁶ *DOOM* demos demonstrated player skills. The rapid action of *DOOM* as a multiplayer, competitive game; the growth of a networked player community, and the technology of the replay facility all came together in the demo to establish a performer–spectator relationship. A critical aspect of this relationship was a learning experience. *DOOM* required skills. Some players excelled in marksmanship, others in movement tricks, others in stealth and the psychology of stalking their opponents. Star players emerged, and mere mortals wanted to see these “*DOOM* Gods” play, to gather insights into their play tactics and learn how to improve the quality of their own play. As BahdKo, a veteran of the *DOOM* demo scene points out, “Use of demos for their educational value has been going on since almost the beginning.” Demonstrations of skill by admired players such as NoSkill, XoLeRaS, and Smight circulated widely. In a typical use of these movies, “a new player who wants to get better requests that a game with a higher-skilled player be recorded, and then the new player watches the demo (where presumably he lost) from the higher-skilled player’s point of view, hoping to learn ways to improve his own skill. Such a player is then able to plainly compare his own movement, aim, and possibly strategic ideas with those of the higher-skilled player, enabling him to practice on his own in order to improve or otherwise attempt to adjust his own performance.”²⁷ Single-player, death match, and clan demo movies certified the status of star players while helping everyday players compare and improve their abilities. Beginning in 1994, the Doom Honorific Title (DHT) Program, a game rating system, became “the means by which good players can objectively prove to the world that they are as good as they claim.”²⁸ The certification process explicitly promoted the performance of gameplay through demo movies. Another project, COMPET-N, started in 1994 to collect demos and a variation of them known as speedruns (completing a game or game level as quickly as possible), joined before long by the Public Demo Archive of the Non Gods, DOOMed Speed Demos Archive, and other collections of demo movies established so that players could find demonstrations of any skill or game level. Establishing a basis for spectatorship through recorded gameplay fed back on the growth of the player community as well. *DOOM* clans sought to establish reputations through demos just as individual players did. These various practices of showing, viewing, and learning via *DOOM* replays showed that multiplayer collaboration could be derived from competition.²⁹

Evidently, the notion of the demonstration is fraught with implications for the computer game as a site for skill performance, certification, and acquisition. This was again the case in the distinctive scene associated with a particular kind of demonstration program known simply as the “demo.” These demos were noninteractive animated movies, in the sense that they combined sights and sounds, and like *DOOM* and *Quake* demos, they were typically generated in real time. The main difference between demo movies and demos is that the so-called demoscene emerged from practices of game piracy and hacking of the 1980s, particularly on home computers such as the Apple II and Commodore 64. By then, the open, cooperative culture of game design associated with computer science laboratories, games such as *Spacewar!* and *Adventure* or the people’s computing movement of the 1970s, had collided with a business culture founded on proprietary development and the closed technology of the game cartridge. The failure of the Atari generation of console manufacturers coincided with the success of games such as *Pac-Man*, intellectual properties controlled by closed industrial studios that produced games to be played not toyed with. The next generation of companies, led by Nintendo, carefully guarded their console technology and intellectual property. Games published for home computers followed this business model for the most



Figure 4
A crack screen.

part, but with the important difference that it was possible, and often acceptable (at least among players) to copy software acquired on formats such as audio cassettes and floppy disks. This provided an opening for players to create an ironically clandestine sort of open culture around computer and video games. The availability of BASIC interpreters built into most of these machines provided the often irresistible temptation, particularly among younger users of home computers, of not just copying, but often altering software as a form of resistance, the 1980s version of taking computer power to the people.

As players acquired the chops for disassembling and rebuilding programs, defeating copy protection schemes or cracking code became means for modifying games or simply for unlicensed, free distribution. Some crackers, as they came to be called, with names like the German Cracking Service, AEK Crackware Essen 2099, 1103, or JEDI, acquired reputations as masters of code that circulated with the copied games; we have already seen that reputations are important in the social networks of players. Crackers documented their prowess when they added credits or load screens to the beginning of games they marked as having been opened up by them. Like tattoos on the body or sprayed tags on a city wall, these introductions became personal (or group) signatures. They also became a basis for competition among crackers. They played a new game with the goal of being the first to post a newly cracked version publicly. This competition intensified during the late 1980s as copy protection systems began to provide more intelligent opponents on this playing field. After cracking a new game, groups celebrated each exploit with ever more elaborate and visually impressive title or load screens, including graphics, sounds, and even animations. These cracktros (cracker intros) became a self-standing form of hacker performance, the cinematic "demo," on emerging multimedia platforms such as the Commodore 64 and Amiga, the Atari ST and the PC.

Demoscene groups began to compete publicly, especially in Europe and California. Their programs were turned into real-time multimedia shows of coding prowess, not just intros



Figure 5
Axios Copy Party Announcement, 1988.

but standalone graphics-and-sound productions lasting roughly as long as a music video or even longer. At a live demoparty (originally known as a copyparty), greatness was no longer demonstrated simply by being the fastest to crack software, it also meant putting on a good party, sharing and socializing; being the best was not just about cracking the most games, it also meant providing an exquisite, yet tightly coded presentation, doing it within prescribed constraints such as available memory or a particular console or home computer platform, and sharing it.³⁰ The founder of the cracker group Sledgehammer (SLH, interviewed by JS) described its social network:

JS: How old are you?

SLH: I am 22 years old (22-10-65)

JS: When did you found Hotline?

SLH: I founded HOTLINE in beginning of August 1986.

JS: How many members does HTL have at the moment?

SLH: HOTLINE has now 15 members!

JS: What was your first crack?

SLH: It was DONKEY KONG from OCEAN!

JS: How many swappartners do you have?

SLH: 10 real ones! 4 in Holland 3 out of them are Hotline members, 1 of the SRG, POPEYE in Denmark and 5 guys in England. I swap by express with them!

JS: How many packages do you send per week?

SLH: 4 express ones and 26 or 27 to the rest!³¹

The demoscene fed off the same impulses as other forms of game-based skill demonstration, while showing another move from playful competition to exhibition, and from mastery of computing technology to creation of new social and performative spaces defined by computer games and their players. As a few years later with game-modding or machinima, players learned about demos from demos, and they kept right on learning when they gathered together (whether in parties or online). Curiously, this very public demo culture within various player communities remained one of its best-kept secrets, because the cost of entry was exactly the high level of involvement, motivation, and time commitment that nurtured the demo as the center of an informal learning network. It was probably not until the mid-1990s perhaps that modding connected demo culture both to mainstream game technology (via the standardized game engine) and to the game industry (via targeted recruiting). But demo coders nevertheless learned skills of direct relevance to eventual careers in the game industry, and many of them moved in that direction. As one coder put it, “at the time I did it for ‘the love of the code,’ not because I planned to get into the games industry. Being part of the scene you also get to discuss your ideas with other people, and show off, which all helps improve your skills. Also, you get some experience with building something through working with a few other people, both other coders and ‘arty types,’ which is just like the real world.”³² At the same time, these same practices set the stage for *Quake* movies and machinima by completing the idea of the demonstration program in an important way. In the demoscene, the hacker was no longer demonstrating technology; he was demonstrating the hacker, and it was his performance that mattered. The public demo as skills demonstration set the stage for game-based moviemaking.

From *Quake* Movies to Machinima: Players as Moviemakers

So, was machinima invented by game developers or players? Did the player clans who made the first *Quake* movies simply connect the dots from real-time play to real-time moviemaking, or did they only see this possibility through immersion in the practices and community of multiplayer gaming? Was machinima an invention or an innovative use of existing technology? We have seen so far that game-based moviemaking emerged from an existing culture of performance that included showing, sharing, and competing in forms enabled by real-time competition, replays, and skill demonstration. But is it necessary to separate practices emerging from what we typically call the “culture” of computer games from those directly enabled—or constrained—by innovation in computer game technology? Not at all. Game-based performance and spectatorship associated with machinima depended on specific design decisions (e.g., Carmack’s architecture of the game engine), but at the same time they were unlocked by players. These players were users of game technology who discovered rather than invented new uses for it.³³ Their excitement about exploiting, hacking, and demonstrating uses of game technology emerged out of engagement with computers—a form of engagement that can only be described as playful, but can also be recognized as a powerful mode of informal learning.

Following the model introduced by id Software, game developers produce or utilize software called “game engines” to manage sophisticated real-time graphics, physics, lighting, camera views and other facets of their games. Games are interactive; prerendered animation has limited applicability in software that must immediately respond and redraw the screen in response to player actions.³⁴ Fast-paced action games drawn from the dynamic perspective of the player/avatar—first-person shooters being the exemplar—significantly increased

the technical challenge for game developers. In order to immerse the player in the rapid action of the game, FPS developers had to render a compelling 3D space from the player's point of view. Software (later, with the assistance of specialized graphics boards) constantly recalculates and rerenders at high frame rates as the player's character moves through that space. The insight that led to Quake movies and machinima was to see this technology as providing a robust system for real-time animation production, as opposed to the painstakingly drawn (whether by hand or server farm) animation delivered, say, by artists at Pixar or Disney studios.

This is not to say that game developers and programmers missed or ignored the potential of their games for making animated movies. They were busy enough making computer games, to be sure, but more fundamentally, moviemaking with games was a user-driven innovation. It was another form of play that evolved out of the particular context of multiplayer, competitive games and the forms of demonstration, learning, and performance that grew out of technologies and communities associated with those games. We can see the impact of competitive play more clearly in an idea that predated Quake movies, that of the "game film." This notion opens game-based moviemaking up to replay, and from there to other forms of mediated or archival performance, such as televised sports spectatorship (introduced as Ampex's "instant replay," first utilized by CBS in a football game telecast in 1965). It is also connected to practices of "proto-performance,"³⁵ such as rehearsal and training, both in dramatic performance and, especially, in sports (such as the football team that spends endless hours "studying game film"). Dani Bunten Berry, the designer of the head-to-head multiplayer strategy games *Modem Wars* (1988), *Command HQ* (1990), and *Global Conquest* (1992), brought this idea to game development. Berry designed these games to reward hand eye coordination and interface mastery as well as strategic thinking, so that "each person had their own specialized style of play." The technical design made it possible to store data from which replays, or "game film" as Berry called it, could be created. Players could watch these movies to rerun and study their performance. Supporting this kind of informal learning was precisely the role that replays played later in the *DOOM* community. Berry also noticed the ease with which players slipped back and forth between the replay as learning material and as a story about games remembered. She was amazed at "how people used this opportunity the game films offered to rationalize their loss and to create stories out of the intense and ephemeral experience of the battle." Berry believed that player communities would thrive on replay's capacity to make "legends out of their best performances."³⁶ Game film was included in both *Command HQ* and *Global Conquest*, but the vision would not be realized until multiplayer social and computer networks required for making the reputations of community players were more closely synchronized with replay technology, that is, until *DOOM* and *Quake*.

Over and over, already, we have seen that demonstrating skills through competitive player performance was behind the replay captured as a demo movie; learning about gameplay through replay implied practices of spectatorship, witnessing, and certification. They both provided means and motives for community players—again, players who put on their own display for other players—to learn more about how to record their gameplay. Not only were replays usually circulated as skill demonstrations, but the very code that made *Quake* movies possible was embedded in something called a "demo" format and saved in a DEM file (with a filename ending .dem). This convention was carried forward from *DOOM* to many other games: *Quake/Hexen*, *Half-Life*, *Unreal Tournament*, and so on. The Rangers' "Diary of a Camper" resembled the demo movies of *DOOM* gameplay, with short bursts of frantic

action punctuated by flying blood and bits of body parts. Yet, "Diary of a Camper" broke with the *DOOM* demo movie in one essential respect: the independence of the spectator's view from that of any player/actor. Their movie was not "shot" from the first-person perspective of the shooter. An independent camera view framed the action. This innovation illustrates *Quake's* significance as a platform for high-performance play. Strictly speaking, *DOOM* demos were not really movies. Rather, they were sequences of commands or scripts that told the game engine what to do essentially by repeating the effects of keyboard and mouse input in the same sequence as the player had. One consequence of the separation of game engine from asset files in Carmack's architecture was that the demo or "intro" movie was stored in a discreet file with its own format, the LMP ("lump") file. This was a game asset in the form of the gameplay movie shown when a player started up the game. Players could also record their own game sessions as demos and play them back inside the game by loading and running these LMP files. Due to the design architecture, making these movies required no hacking of the game engine, in effect creating a new performance space. This technical change differentiated the makers of *DOOM* and *Quake* demo movies from previous modes of game-based performance.

The real-time animation engine inside the computer game was discovered by players, who found the technology and learned to use it to produce animated movies in a new way. Certainly, it was the game developers who built this technology, but they did not foresee this application of it, which was instead discovered, tested, and improved by those players. Uwe Girlich became the leading technical authority on *Quake* moviemaking; he found in his analysis of its new demo format that "player coordinates and the camera positions may be different," in other words, that different camera views could be presented in replay demos. This discovery led him to observe that, "for people with too much spare-time *Quake* can replace a full 3D modelling system for cartoons or the like."³⁷ When the Rangers figured out how to move cameras in *Quake*, they programmed their own tools to do this, thus demonstrating coding skills alongside gameplay. Girlich showed how this approach could be taken a step further, focusing on careful analysis of *Quake's* architecture and operation, especially the demo format and network protocol; he revealed what he discovered to the player community and provided tools such as the "Little Movie Processing Center" for replay recording in *DOOM* and other games. Others followed his lead, notably David "CRT" Wright, who wrote the Keygrip and Keygrip2 utilities for "no compromise demo editing," that is, postproduction work on *Quake* movies.³⁸ Clearly, these contributions to *Quake* moviemaking were another mode of game-based performance, as these code mechanics opened up the hood of the game engine and revealed their talents by unlocking capabilities hidden inside the *Quake* programming code. *Diary of a Camper* used the Rangers' homebrew tools, but later projects drew increasingly on a reservoir of knowledge and skills shared with a growing community of players and moviemakers. Players were once again demonstrating multiplayer learning derived from demonstration, competition, and collaboration.

During the first year after *Quake's* release, projects such as the "Quake done Quick" team's filmed "speedruns" and Clan Undead's "Operation Bayshield" showed how playing with the camera in computer games could be enhanced by techniques such as "recamming" (editing demo movies to alter the camera view) and postproduction editing with player-created software tools.³⁹ Every new project seemed to introduce a technique or push game-based moviemaking forward in some unanticipated way. Once players learned how to redeploy sophisticated game engines to make movies, they began to see themselves as moviemakers. They discovered that their knowledge of play *and* technology translated into acting and

directing, literally turning players into “cameras” to make these animated movies inexpensively on the same personal computers with which they fragged monsters and friends in *DOOM* or *Quake*. Id continued to open up access to *Quake* in order to make it easier for players to customize or modify the game. Carmack supported open discussion of id’s technology as beneficial, because “programming is not a zero-sum game. Teaching something to a fellow programmer doesn’t take it away from you.” Abrash, the 3D programming expert who helped Carmack with *Quake*, followed this “enlightened attitude.” He published articles about their programming tricks “even before *Quake* has shipped” and noted “when it’s legally possible, sharing information benefits us all in the long run.” He called this philosophy, “learn now, pay forward.”⁴⁰ These values trickled down to *Quake*’s player community, including those who made *Quake* movies. Just as FPS players shared knowledge about how to modify games, progressing together from map and level design to add-ons and eventually to full-scale total conversion mods, machinima makers likewise helped each other. The Rangers learned how to change the camera view, “Operation Bayshield” was the first *Quake* movie to utilize custom artistic assets, and so on. At each step, the Rangers, Clan Undead, and other players shared their techniques as they learned them. As other players did for game mods, they created skins for avatars, designed visual effects, added sounds, and produced game graphics (for sets, characters, etc.) in order to make more ambitious movies. Projects following on their work through the late 1990s taught methods for revising game assets as surely as they demonstrated gameplay and provided entertainment.

The impact of the modification of game software and assets on the cultural economy of game design is well documented, particularly with respect to modding. Id’s “enlightened” attitude about discussion of its innovative programming techniques encouraged the player community to dive into investigation of the game software and eventually to produce tools and assets, especially with each eagerly anticipated (and sometimes unanticipated) release of source code. But even if sharing is caring, are these practices sufficient to take players into a sphere of activity we might call codesign or cocreation of the technology itself (as opposed to the media content produced by that technology)? Put another way, when players use game technology to make mods and movies, in what sense are they innovators? Perhaps these activities are better depicted as an extension of their use of technology rather than part of the process that invents it? As players figured out how to use computer games to make animated movies, were they inventing a new technology for real-time animation production or coming up with a new way to play these games? And, if they were not coinventors, does that mean that they were just playing around?

Historians of technology for a long time focused on the process of invention in a way that paid far more attention to invention, design, and even production than to how technologies are continuously redefined, stabilized, and redefined again through use. In a variety of disciplines encompassing the history of technology, business, innovation, and media, more critical attention has been paid in recent years to consumers and users, especially in terms that emphasize how they take part in shaping, modifying, and repurposing the technologies that they use. Studies of technological innovation are paying attention not just to *users* as consumers, but also to *use* and the practices associated with it as part of the process of innovation.⁴¹ This perspective emerged during the late 1980s in technology studies under the rubric of “social construction of technology” (SCOT). The SCOT manifesto was a group of essays published as *The Social Construction of Technological Systems* in 1987.⁴² As an approach to innovation, SCOT initially focused on interactions among producers (including inventors, designers, etc.) and users as a process of negotiation leading to “stabilization”

of a technological artifact. Space here does not permit a detailed history of demand-side innovation studies. However, it is important to note that during the 1990s, criticism of SCOT's focus on the stabilization of design through interactions between suppliers and the demand side (consumers and users, treated interchangeably) shifted to an approach that encompassed users of mature technologies and producers. In brief, SCOT began to address "how technologies are actually used in practice."⁴³

This "turn to the users"⁴⁴ is allied with similar impulses from other disciplines. Cultural studies, fan studies, innovation studies, business history, and a broad range of writings on contemporary media have shifted the focus of creativity and innovation to the user or consumer of technology.⁴⁵ This diverse set of approaches includes a variety of models to describe users as active and creative, rather than just passive consumers, ranging from resistance to distributed innovation. An important common characteristic of these approaches has been the idea of *negotiation* as part of an overarching process that shapes technologies as well as their users from innovation and design through consumption. In studying this process, researchers have worked out a claim made by Kline and Pinch in their study of early rural use of the automobile in the United States, "that the use of an artifact or system has not only resulted in unforeseen consequences, but that users have helped to shape the artifact or system itself."⁴⁶ In working out the dynamics of this process, it is crucial to identify different participants in such a process of negotiation and the meanings they attach to a particular technology under discussion. SCOT proposed the "interpretive flexibility" of an artifact as a way of articulating that it can have different meanings for various interested social groups. This notion was coined for a model that showed how this flexibility eventually led to "closure" as technologies stabilized and disparate notions faded away. Later studies explored "how interpretive flexibility can reappear at the use stage of technology."⁴⁷

Of course, as business historian JoAnne Yates has pointed out, the "very notion of closure in interpretive flexibility is problematic" in the realm of software.⁴⁸ Machinima provides a good example. Certainly, Carmack achieved a dominant design for game engines, but players opened up this design by using it as a real-time animation engine for making movies; they reshaped the technology by establishing needs and desires for a new group of users (moviemakers, let's call them) with a different set of expectations for how the technology would be used. This may be conceived as users pushing back on the configuration of a technology established by designers, but this tells us little about how the technology likewise configured the users, that is in the case of machinima, how the use of game engines for making movies created a certain kind of player and moviemaker.⁴⁹ As we have seen, at least through 1998, there was no machinima community, only a *Quake*⁵⁰ movie community. This is important to acknowledge, because even though the movies produced were growing in scope and ambition, the community that produced and consumed them was limited to players of a particular, albeit very popular, computer game. Whether creator or spectator, if you knew about *Quake* movies, you were a *Quake* player; indeed, the demo format required access to the game even to play the movies.

The popularity of the game was drawing in new players, however, and some of these players imagined that game-based moviemaking could lead to a meaningful convergence of interactive gameplay and linear, story-driven entertainment forms. They began to think of themselves not so much (or not always) as players, but as moviemakers. Players with training or professional experience in video, improvisational comedy, film or television production began to explore the possibilities of moving "beyond the world of *Quake*,"⁵¹ notably the Ill Clan in the United States and Strange Company in Great Britain. For example, the Ill Clan's



Figure 6
The Ill Clan, "Hardly Workin'" (2001).

"Apartment Huntin'" (1998) and "Hardly Workin'" (2000) added voice talents, improvisational comedy, and custom artwork to skills derived from *Quake* gameplay, resulting in a comedic short good enough to win the Best of SHO and Best Experimental Short awards at the Showtime Network's Alternative Media Festival. Paul Marino, the director of "Hardly Workin'," cited these awards as an indicator of "the recognition of Machinima by a mainstream media venue."⁵² Only viewers familiar with *Quake* (or *Quake II* in the case of the second movie) would have recognized that the Ill Clan's movies had anything to do with a computer game. But that realization pointed perhaps to a problem in terms of increasing the visibility of these movies; if only *Quake* players could see them, what difference did it make if nonplayers would find them entertaining to watch?

By the late 1990s, players had learned that they could make animated movies from computer games. While on the one hand, it could be said that game engines were ready-made for making movies, on the other hand, finding that the technology could be used in this way did not feed back significantly on the subsequent development of the game engine as a technology. This is an important issue for a better understanding of how users of game technology (players) interact with roles traditionally understood as inventors and producers (programmers, developers), whether by changing game content or reshaping game technology. One might think that if making movies with game engines is such a compelling idea, producers such as game developers, makers of 3D modeling and rendering software or even graphics processor manufacturers would have responded with a dedicated technology, a flexible real-time animation studio perhaps. There was in fact little incentive for game developers, or anyone else for that matter, to spin off such animation engines as a new technology just for players who chose to make movies using *Quake*.⁵³ The question of why game-based

moviemaking technology has not emancipated itself from game technology compels a closer look at how platform-independent machinima “broke out” from the relatively closed world of *Quake* movies. Furthermore, these matters focus attention on whether players making movies aspired to break out from the culture of computer games to a broader media audience, that is, to cross over into other media and engage with an increasingly diverse set of issues and genres in their movies. In fact, the dependence of *Quake*-based moviemaking on id’s game engine development played an important and unexpected role in stimulating the development of machinima beyond *Quake* movies.⁵⁴

Quake II, the successor to *Quake*, was released in 1997 and continued as the dominant platform for making demo-based movies. Productive groups, such as Ill Clan, Zarathustra Studios, and Strange Company used the game for influential projects, some of which (such as Strange Company’s multipart *Eschaton*, which took place in the fantasy world of H. P. Lovecraft’s “Chtulhu mythos”) spanned development in *Quake* and *Quake II*. At the end of 1999, id released *Quake III Arena* (henceforth, *Quake III*), published by Activision. With an updated and improved version of the *Quake* engine, such that *PC Magazine* called it “the current state of the art in gaming graphics,”⁵⁵ and id’s apparent encouragement of player-created content, the new game seemed to offer nothing but benefits for game-based moviemaking. The initial enthusiasm among demo moviemakers soon dissipated, however. *Quake*’s developers had become concerned about cheating in competitive versions of the game. The potential effect on the success of the game was obvious; as one player of FPS games put it, the knowledge that other players were cheating online made him “ready to walk away from the game in disgust and tell every one else to do the same.”⁵⁶ It turned out that as players gained more knowledge about the network code underlying multiplayer games, they learned not only how to make mods and movies, but also to cheat more effectively. Girlich had done more than anyone to reveal and disseminate the intricacies of the demo format, but even he had to admit that “existing demo formats are only variants of the network data flowing between the server and the clients. So any existing demo format description leads directly to the network protocol. Many cheaters misuse this information to write proxy-bots.” Indeed, the network messages sent back-and-forth by the *Quake* server and clients were identical to those used in demo files. Girlich concluded that he could no longer provide information about the *Quake III* demo format to the moviemaking community, noting that there were also “legal reasons” for this decision.⁵⁷ Wright, the creator of *Keygrip* and *Keygrip 2*, likewise announced that there would be no new version for *Quake III*.⁵⁸ An expert on *Quake* movies concluded that “it looks highly unlikely that any tool will ever be released with the capacity to edit *Quake III* demos.”⁵⁹ Suddenly, it looked like the neat linkage of peer-to-peer learning and multiplayer innovation accidentally nourished cheating technology. The social organization of players into competitive clans and communication patterns that promoted the sharing of these techniques for high-performance play also abetted those who would propagate cheats drawing upon this knowledge. Clan members shared these hacks as they would other programming exploits that publicized the superior skills of their clan. Lone players were more likely to keep such tricks to themselves.⁶⁰ The *Quake* movie scene threatened to grind to a halt.

From id’s perspective, what was a game developer to do? The company decided to throttle access to *Quake III*’s network code to protect the game. Demo-editing tools such as Wright’s *Keygrip* and Girlich’s *LMPC* that made use of this information were out of luck. Now that circulating detailed information about this aspect of the game was forbidden, to the extent that id threatened legal action against anyone who revealed the code or how it worked, these

tools would not be available for making movies with *Quake III*. The crisis initiated by id's decisions with regard to *Quake III* technology forced a response from those moviemakers who until then had found the game engine so hospitable. Now they began to see the gift of a found technology as a limitation on their creativity. As Paul Marino, a member of the Ill Clan put it, "This started a turning-point for the movie community. The newness of imbuing fantastical environments with mundane situations and commentary seemed to wear off. Simply said, the joke was getting old. This coincided with id's decision to protect *Quake III Arena's* network code, and hence the amount of productions slowed to a crawl. The community was forced into survival mode—either reinvent itself or succumb to the harsh realities that the allure of *Quake* movie production would slowly fade away."⁶¹

The *Quake* movie community reinvented itself around a more general conception of game-based moviemaking as a new media form. Hugh Hancock, founder of Strange Company, and Anthony Bailey, renowned as a founding member of the *Quake done Quick* team that had popularized the class of *Quake* movies known as speedruns, coined a term that would redefine the community's work: machinima. Bailey, in a private e-mail to Hancock during May 1998, had devised the term as a portmanteau of machine and cinema, so he originally spelled it "machinema." In the original e-mail he noted that "it's a bit of a contrived term . . . but what in general *are* we going to call these pieces of cinema that are made using 3D engines?" He also predicted accurately that, "Not only is 'Quake movie' an ugly and confusing term, it's also fast going to become outdated as other technologies become relevant."⁶² For a while, the term was rarely used in public forums.⁶³ At the end of 1999, about when *Quake III* would be published, Hancock, while working on the Web site for Strange Company, proposed "machinema" as an improvement. It suggested animation generated by machine (game engine). He registered the domain name and shortly afterwards launched the machinima.com Web site at the beginning of 2000; almost immediately, it became the de facto community site and resource for machinima makers.

Bailey and Hancock's conscious use of "machinema" to describe an expanded scope of game-based moviemaking not limited to the *Quake* engine coincided roughly with id's decision to restrict access to the demo format in order to protect its network code. Players and developers pulled their oars in synch with each other, but without a common notion of their destination. However, machinima continued to include projects created with the various versions of *Quake* technology, and id supported such player-created content in principle, even if its design decision had the effect of driving projects away from the game. The first movie to be made with *Quake III* was Triton Films' "Quad God," and the effects of these crosscurrents are illustrated by the circumstances of its creation and release. Work on the project began in June 1999, long before the game's release, and the movie was released in early January 2000. The Triton team had responded to a contest set up by Katherine Anna Kang of id Software to create the most original work of art making use of the *Quake III* logo. They made more than fifteen custom maps and thirty character skins on the way to releasing a thirty-three-minute movie made with the test engine for the game (as it had not yet been released). Even though "Quad God" was clearly a *Quake III* movie, the circumstances of its release previewed characteristics of a more widely accessible medium. It was one of the first movies to be released via Hancock's then brand-new machinima.com site; just as important, it was released not as a demo movie that required viewers to own and execute the game, but encoded in a conventional video format that could be played using easily available media players. One could now with a bit more sleep deprivation encode game movies as video files that could be viewed almost by anyone.

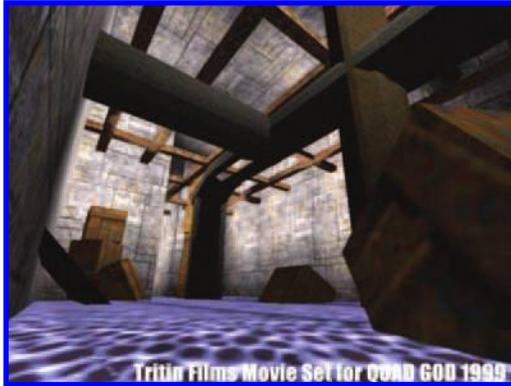


Figure 7

"Tritin Films Movie Set for QUAD GOD 1999." Source: Tritin Films.⁶⁴

The shift away from *Quake* movies and the gradual demise of the demo format accelerated during 2000. The crisis gave way to the emergence of machinima. More movies appeared that were based on other engines, such as *Unreal* and *Half-Life*.⁶⁵ Ill Clan's "Apartment Huntin'" at the end of 1998 was the first machinima piece encoded in the RealMedia format, but two years later video files could be distributed over the internet (at almost no cost) either as the original game replay file or in an encoded movie format. A few years later, nearly all machinima movies were released in one of the common video formats, and today nearly every one of the major viral media sites includes a channel devoted to game movies. The decline of the *Quake* movie can be seen as a consequence of the tight connection between game-based moviemaking and a particular technical configuration of the game engine and demo format. The unexpected impulse id's decision gave to *Quake* moviemakers was that it pushed them harder to think of their work as defining a new and more generally accessible cinematic medium with the potential to be viewed as meaningful by a wider audience than *Quake* players. Therefore, what we have here is an example of how the stabilization of use by players does not necessarily follow from what technology studies might describe as closure of design. Machinima resulted from the breaking of the connection between the game engine designed by id and the practices of moviemaking created by players. The innovation of game-based moviemaking was not achieved through the stabilization of design and use to create a new artifact (technology). Instead, players found that they could extend the practices and culture learned from these movies—in fact, the essence of what they had created—by other means.

Screen Capture: The Politics of Found Technology

Players have learned through making machinima that the dissemination of accessible tools—even if they are not necessarily easy-to-use—creates opportunities for the emergence of unexpected content. Player culture values playful experiments and throwaway pieces alongside startling and original instances of creative expression, much as other forms of viral media today.⁶⁶ In this sense, we can think of machinima as exploiting a found technology. I borrow this idea of a found technology from that of the found object (*objet trouvé*) of the art world, the readymade, everyday, recycled object that is appropriated or reinterpreted as a

form of artistic expression, ready to use, but also readily discarded for something else. Marcel Duchamp coined the term “readymade” for the found objects he used for works such as “Bicycle Wheel” (1913)⁶⁷ and “Fountain” (1917). Whether or not they remained in their original form, however, the essential point was that the artifact has not been created for or by the artist, nor generally for the purpose of display or artistic statement. The artist selects unexceptional objects and gives them a new context or purpose. The bicycle wheel chosen by Duchamp is like any other. Even if the artist lays no claim to having created the artifact itself, he creates something new with the visual or conceptual statement that emerges from its recontextualization.

Depicting machinima as a found technology emphasizes certain of its characteristics as a player-created use of computer games, such as the availability of game technology as “readymade” for a purpose other than making movies, or the ease with which machinima creators move to new game engines or from demo to screen capture techniques. Hugh Hancock of Strange Company has been fond of citing William Gibson in this regard, that “the street finds its own uses for things.”⁶⁸ At the same time, there is an important difference. In the artistic notion of the found object, the process of reworking elevates the object as artwork and infuses it with new meaning. The artist offers a reenvisioning of the artifact. In machinima, the transformation stimulated by the artistic process revises the technology that produces the artifact. Not by altering the game engine, but by turning the found technology of the game engine into an animation engine, that is, by finding a new purpose and context for it. When technologies are revised in this unanticipated way, the user operates in a hybrid space, mixing practices learned from the familiar, found technology and reworking them as an application of these skills to new purposes. Paul Marino, looking back at his experience making Quake movies as part of the Ill Clan, observed that the work on these projects was “a combination of playing multiplayer *Quake* and shooting a live film.” The language used to describe machinima techniques became a pastiche of gaming and performance terminology, such as making sure that “key configs were set and that we knew our marks in the game maps.”⁶⁹ The recontextualization of the computer game led to a related layering of practices. Early machinima “definitely took some modding skills”⁷⁰ as well as the use of tools to capture, edit, and reassemble files that could be viewed with game software; later work in games such as *World of Warcraft* typically called upon facility with model viewers, compositing techniques, and postproduction video editing to radically reassemble game assets. As Hancock put it, machinima makers do not create proprietary technology, but rather they “take a bunch of disparate and varied technologies—mostly things that have come of the Digital Video or gaming world—and weld them together, using their various capabilities to make films.”⁷¹ Casting machinima as found technology helps us to move away from an idea of user creativity focused on invention of new artifacts to one that traces new player commitments and uses, often by learning how to mash up practices imported from other activities and contexts.

The hybrid nature of machinima as a found or readymade technology carried important implications as the new medium broke out into a variety of media spaces less closely tied to the hard-core computer game culture of FPS games. As we have seen, video files are easily distributed over the internet at virtually no cost. Significantly, the machinima movies that became most popular with the largest audience relied on a mode of production quite different from *Quake*’s demo movies. Instead of capture in the game, editing with special tools, recompilation into the demo format, and playback inside the game, these movies followed the path pioneered by Tritin’s “Quad God,” capturing the screen display to video, editing

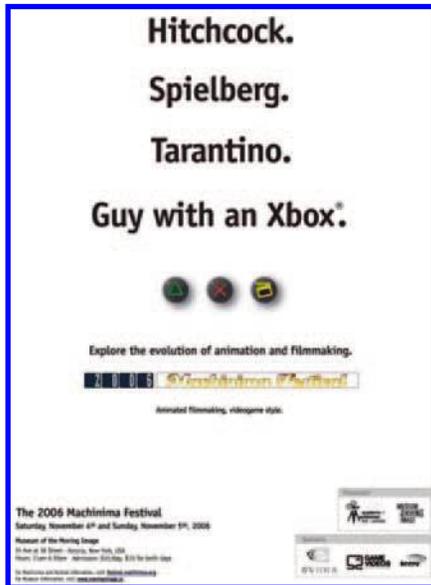


Figure 8

2006 Machinima Festival announcement, *Computer Gaming World* no. 268 (November, 2006): 73.

the video (with nonlinear video editors such as Adobe Premiere), and then distributing and viewing the movies in common video formats. Instead of FPS games, console games and massively multiplayer online games became the dominant modes of machinima production based on screen capture. The breakout title was Rooster Teeth's "Red vs. Blue," a comedy series now in its fifth year based on the Xbox games *Halo* and *Halo 2*. In 2006, huge numbers of machinima titles are produced in massively multiplayer games (*World of Warcraft*), virtual worlds (*Second Life*) or games that export directly to video (*The Movies*); in each case, the predominant mode of production is direct capture to a video format. The issues raised by screen capture as the dominant method of machinima technology are manifold. How should the screen content, which clearly includes content and intellectual property owned by developers, be licensed and marketed? Who owns an artistic product based on contributions from a mix of sources and creators? What constraints will a potentially contested ownership of the assets (not just the technology) used to create these movies place on freedom of expression?

Players have made changes and improvements to game software and assets since *Spacewar!* and *Adventure*. Such modifications have been a permanent part of the business culture of game design since the release of *DOOM*. This is undeniably a quality of game culture that has inspired some observers to value its creativity as a kind of resistance to passive consumption, or as a participation in content creation. In fact, I have been less concerned with an aesthetic evaluation of machinima than with the ways in which this new medium reveals modes of creativity and innovation associated with playing computer games, learning how to play them better, and figuring out what the technology that produces them can do. It is important to see the ready availability of game-based moviemaking evident in the making of "The French Democracy" not only as a validation of computer games as being culturally expressive. Rather, we should dig deeper and discover why playing these games leads

some players to want to make movies for other players and to explore found technologies independent of gameplay. While the creativity expressed through authorship, invention, or intellectual property is important, other aspects of computer games as technology, as the basis for player communities, and as a site for performance techniques and practices have given game movies their particular significance. Those players whom I call “community players” are not only creative and theatrical, but they also take care to exhibit mastery of technology and cyberathletic skill. It is important to recognize the extroverted and expressive play performance of the community player for what it is. Part stage actor, part activist, the community player plays for other players. The complex cocreation of machinima and other game-based movies benefits from consideration of the player not only as a “consumer who also produces,” but also as a “user who also innovates” and the computer game itself as a found technology capable of being appropriated for the new application of real-time moviemaking. An important issue illuminated by the development of machinima is the capacity for users of a technology not only to shape its development, but also to transform its purpose and, in effect, to create a new technology out of an existing one through redeployment for a new purpose. The payoff was more than the delivery of entertaining video clips, even more than creating a new moviemaking technology; it included showing how to *do* it. In this sense, the core innovation was the *demonstration* itself as a paradigm for multiplayer learning fueled by engagement with digital play.

We have seen that the replay culture of game film and screen capture supplemented the demo format as the basis for game-based moviemaking. This has made possible a broadening of appeal, primarily through the replacement of game files by downloadable or streamed movie formats that anyone can view. Of course, conflicts can arise when the free donation of time, effort, and invention is entangled with, indeed dependent upon, technology and content developed by commercial enterprises and professional artists. In machinima, this occurs when a game provides virtual locations, studios, and cameras, or a pop song serves as the soundtrack for a dance video. To get at these conflicts, I will revisit a machinima project that I have covered elsewhere, Tristan Pope’s “Not Just Another Love Story,” made with (and in) the *World of Warcraft* (WoW).⁷²

Until moviemakers learned to extract models and maps with new tools such as *WoW* Model Viewer, server-based games such as *WoW* denied direct access to code and even to game assets outside their use in-game; moviemaking was limited to edited screen captures. This means that early *WoW* movies were essentially replays grabbed via screen capture, then cleverly edited, perhaps with voice-overs or painstakingly lip-synced dialogue. A player dubbed JuniorX made the first *WoW* movies to be widely distributed.⁷³ His movies were essentially training films, introducing potential players to the game, giving an indication of its pace, challenges, and tactics through lengthy recordings of adventures encountered in the game, little more than unadorned gameplay. His movie on the hunter class, for example, showed every moment in the career of a dwarf character up to level ten in the game; more than an hour long and claiming nearly 400 MB of storage space, it was nonetheless downloaded more than 11,000 times from *warcraftmovies.com*, a new Web site devoted entirely to *WoW* machinima and replays. These movies get us back to the skill training that motivated *DOOM* demo movies or the replay sharing among players of other Blizzard games, *Starcraft* and *Warcraft III*. JuniorX recast the training replay as entertainment when he made “Dancemovie” and “Dancemovie 2.” He combined the discovery of dance movements built into the game, the presence of other players as cop performers or spectators, and the showcasing of neat tricks and exploits (such as being able to activate dance movements during combat, a

fleeting “feature” quickly eliminated during the beta test) to put together his recorded performances. Dance videos became a staple of the *WoW* movie scene, remediating MTV music videos through gameplay set to music, with particular attention to matching lyrics and images, synchronization of character movements to soundtracks, and elaborate choreography of players. For the new *WoW* player community, the novelty of these videos dovetailed with replay culture to show *WoW* players how to perform for other players in a virtual world.

WoW movies, from game film and dance videos to memes of viral culture, such as Pals for Life’s “Leeroy Jenkins,” became an integral part of the *WoW* player culture. Moviemakers in *WoW* have discovered, however, that increasing popularity magnifies the creative constraints on their work due to issues of intellectual property, social dynamics, and politics. The production and reception of Pope’s “Not Just Another Love Story” illustrates the new performance politics of the community player, suggesting both the payoffs and problems he or she faces. Its reception pointed to the potential for creative conflict beneath the surface of the relationship between machinima and game developers. Pope, like JuniorX, was an early *WoW* player. He released his first *WoW* movie, “I Surrender,” near the end of the beta period. Completed after playing *WoW* for only three days, he was inspired by other beta period dance and party movies available around that time and soon created the Crafting Worlds Web site to facilitate the distribution of his projects to the *WoW* community. With each of his movies from “I Surrender,” released in November 2004, through “Onyxia Eliminated,” completed in April 2005, he worked through remediations of various movie and even game replay formats, such as the music video, sketch comedy, and guild demo. In April, he coyly introduced a more ambitious project, “Not Just Another Love Story”:

I want to give you a full description of this movie, but that would ruin the surprise.
I’ll give you a hint: I only executed what the pixels in *WoW* suggest . . .
And it has something to do with something that was removed in patch 1.3.
Ok, that’s all you get!⁷⁴

Beginning with the disclaimer that “this movie contains material that may not be suitable for all ages,” Pope’s movie tells a Romeo-and-Juliet story, but with a game-specific twist. It sets up the story by showing his Troll Rogue character, Tristanmon, heading off to work in the desert for another day of monotonous creature kills. In the middle of combat, he notices and falls head over heels in love with a human female who can match him kill for kill. Alliance and Horde characters do not mix in this world, but despite such taboos, they become engaged and marry. Pope uses editing, character positioning, and carefully chosen camera angles to depict the pair consummating their love in various ways. The highlight of Pope’s movie is a spectacular rave during which the Troll emerges from his shell and is fully transformed by love into a wildly dancing party animal. The masterfully choreographed series of scenes would not have been possible without dozens of player-actors, choreographed actions and spell effects, cleverly chosen locations, and immense preproduction planning. In a stunning reversal, the troll’s new life is later shattered by the death of his spouse in combat. His luck holds, however. She is resurrected by an equally attractive human female, and the threesome live happily ever after.

But there was more to this love story. Pope had purposefully sharpened the narrative edge to give voice to the player community. The story specifically drew attention to issues of creative ownership of the story world. Since the first *Warcraft* game, subtitled “Orcs vs. Humans,” the narrative momentum pushing forward the single-player campaign was



Figure 9

Tristan Pope, "Not Just Another Love Story" (2005).

faction and racial hatred. While the opposed races and their relative moral elevation could be remixed from version to version of the game (such as the focus on the reawakened nobility in the Orcs of *Warcraft III*), the role of relentless and unremitting conflict in shaping the history of the fictive world remained constant and fundamental. As players descended from the strategic perspective of the RTS games to play on the ground in *World of Warcraft*, they discovered that these conflicts had been built into their characters. This fundamental fact of *Warcraft* life translated into the inability of Horde and Alliance characters to communicate directly in-game through language. Chat was impossible, and shouted speech was rendered as unintelligible gibberish; the game software even recognized and filtered out subversive attempts to communicate by embedding text in descriptive gestures, known as "emotes." Beginning in the beta version of the game, players discovered that the language of game culture provided the key for unlocking a system of universal speech. They found that it was possible to embed the number- and special character-based misspellings of "1337 speak" ("leetspeak," or elite speak) in emotes, thus bypassing Blizzard's text filters and making it possible for, say, trolls to speak with their human enemies. Just as an earlier generation of hackers and gamers had used it to circumvent mail and bulletin board language controls, they used it to speak across an in-game cultural divide and they shared the workaround. This was a clear transgression of Blizzard's control of the relationship between gameplay and story world, so in the aforementioned 1.3 patch of the game the development team announced that henceforth "numbers and punctuation will not be passed through chat communication to members of the opposing faction."⁷⁵

In the context of this assertion of Blizzard's control, Pope's depiction of the marriage of Troll and Human characters, as well as the massive collaboration of Horde and Alliance players evident in the movie itself, put an alternative vision of the game world into play. In the movie, Pope directed a final comment to Blizzard after the credits had wound down and the waning notes of The Darkness' "I Believe in a Thing Called Love" had faded away:



Figure 10
Tristan Pope, "Not Just Another Love Story" (2005).

"Even without leet speak you cannot take away our love!" The mature content creatively constructed through character positions and camera angles in the video intensified this point, but it also sharpened the ensuing controversy. An often-overlooked implication of the *WoW* moviemaker's inability to modify the server-based MMO was, as Pope argued with a wink, that he had merely showed "what *WoW*'s pixels imply ☺." Even sexual imagery, therefore, was nothing more than a rearrangement of what Blizzard's artists had drawn, or more accurately, what its game engine generated during gameplay. Rather than asserting his right to subvert the game's content, Pope reasoned that he had in fact not created anything on the screen, merely captured it.

Since the release of this phenomenally successful game, Blizzard had eagerly sponsored *WoW* events such as screenshot and stunt competitions, encouraging players to use the game's visual assets as Pope had as a means of generating new content as part of the shared culture of its millions of players. (Indeed, Pope went on to become a Blizzard employee and played a key role in the machinima portions of *South Park*'s notorious "Make Love, Not Warcraft" episode.) Community managers encouraged announcements about game movies in official *WoW* forums, allowing creators to provide links to facilitate downloading of video files. Pope was allowed to post such a link, but within two days the volume of complaints, flames, and counterflames about "adult" scenes in "Not Just Another Love Story" caused Blizzard to cite the user agreement concerning language or images that are "pornographic in nature" and lock the discussion thread about his project. It barred links to any of the movie's download sites in subsequent discussion threads.⁷⁶ The marketing of in-game creativity collided head-on with the game's demographics and success, which by then had brought many young players to the *WoW* player community. Players argued over the implications; some supported Pope, others supported Blizzard: "How can making an IN GAME movie with only IN GAME animations, on a forum about THAT GAME be inappropriate?" or "Let me go take Ken and Barbie at Toys R Us and pose them in sexual ways, and say 'But whoamgod~ their joints BEND that way so its not sexually suggestive or inappropriate for us to advertise that way!!!' Plus you're overlooking the simple fact that there ARE forum rules prohibiting these things."⁷⁷ Pope conceded Blizzard's right to some measure of control and understood that



Figure 11
Ruby (Tilda Swinton) in Second Life (April 2006).

rights to freedom of expression on the forums were not governed “by the Constitution,” but at the same time he argued that “I also don’t want censorship over something that took what is already in game and just made it more provocative.”⁷⁸ By acting as the lightning rod for commentary on the contested boundary between developer and player control of a complex, multiplayer game world, Pope’s “Not Just Another Love Story” showed that game movies could function as a medium for negotiation of issues important to the player community.

Pope’s movie brings us back to an optimistic view of the potential for game-based moviemaking implicit in “The French Democracy.” It suggests that players are learning to use digital media that matter to them in order to express ideas and opinions that are equally important in their lives. Note well that we have seen that these issues of importance extend from the politics and conflicts of real life to those of virtual worlds and game communities. We have also seen that the expressive potential of machinima brings into focus the ways in which player-generated practices and uses of digital technology may be limited by the use of digital games as a found technology for moviemaking. Whatever the accuracy of the substantial claims I have made on behalf of player-driven, innovative use of game technology, the fact remains that most of the technology and many of the images, music, and, in some cases, even texts remain contested intellectual property, thanks to a wild mix of benign neglect by game developers, player-created content, and other sources of content, such as commercially published music. At this early stage in the historical development of real-time animation and game-based moviemaking, the bottom line is that players are learning how to deploy technologies from computer games to develop new practices for expressing themselves through game-based performance. The capacity of a medium based on computer games to venture forth beyond the self-referential is at the heart of player practices that, as we have seen, encourage a sense of co-ownership (however misled it may be in legal terms), and the freedom to replay, reinvent, and redeploy. These practices include

performing for other players, skills certification, showing by doing, and multiplayer learning across networks of players. In particular, performance practices around the demonstration as a key moment show us that “high performance play”⁷⁹ is capable of taking players from their deep engagement with games and game technology to the discovery of new uses of digital technology for learning, as well as cultural production.

Notes

1. Sonia Livingstone, The Changing Nature and Uses of Media Literacy, *Media@LSE Electronic Working Papers* 4 (2003): 13–14. Retrieved September 2006. http://www.lse.ac.uk/collections/media@lse/pdf/Media@lseEWP4_july03.pdf.
2. Paul Marino, The Machinima Faq, (updated 8.03.2005), Academy of Machinima Arts and Sciences. Web site. <http://machinima.org/machinima-faq.html>; for the history of machinima, see Henry Lowood, High-Performance Play: The Making of Machinima, *Journal of Media Practice* 7, no. 1 (2006): 25–42. Also to appear in: Andy Clarke and Grethe Mitchell, eds., *Videogames and Art: Intersections and Interactions* (Bristol, UK: Intellect Books, 2006).
3. The French Democracy. A film by Koulamata. *The Movies*. <http://movies.lionhead.com/movie/11520>.
4. Mike Musgrove. Game Turns Players into Indie Movie-makers. <http://washingtonpost.com> (1 Dec. 2005): D1.
5. <http://www.lionhead.com/themovies/>, with the movie database at <http://movies.lionhead.com/movies.html>.
6. 460 movies uploaded in the last 7 days. <http://movies.lionhead.com/recent.html>.
7. Stephen Totilo, First Film about French Riots Comes Courtesy of a Video Game. *MTV News* (5 Dec. 2005).
8. For example, Musgrove, Game Turns Players . . . ; Totilo, First Film . . . ; Joelle Diderich, French Film about Riots Draws Applause, *USA Today*, December 15, 2005; Carol Matlack, Video Games Go to the Movies, *BusinessWeek online* December 9, 2005. Retrieved January 2007. http://www.businessweek.com/technology/content/dec2005/tc20051208_639203.htm; Marie Lechner, La cité animée d’Alex Chan, *Libération*, December 12, 2005.
9. http://www.worldwideshortfilmfest.com/schedule/programme_details.php?program=Machinima; <http://iarvers.free.fr/pages/evenement.html>.
10. Paul Marino, The French Democratizer. Post to Thinking Machinima blog (16 Dec. 2005). Retrieved January 2007. <http://www.machinima.org/paul.blog/2005/12/french-democratizer.html>.
11. Quoted in Diderich, French Film
12. “Je rencontrerai également l’ensemble des responsables de l’audiovisuel. Les médias doivent mieux refléter la réalité française d’aujourd’hui.” Déclaration aux Français de Monsieur Jacques Chirac, Président de la République, November 14, 2005, Présidence de la République Web site, http://www.elysee.fr/elysee/francais/interventions/interviews_articles_de_presse_et_interventions_televisees./2005/novembre/declaration_aux_francais.32000.html.
13. Mogniss H. Abdallah, Quand les banlieues s’embrasent. Entre révolte “ethno-religieuse” ou colère sociale, une opinion publique divisée, *Alterités (Médias)* Web site, December 22, 2005, http://www.alterites.com/cache/center_media/index.php.
14. Henry Jenkins, *Fans Bloggers and Gamers: Media Consumers in a Digital Age* (New York: New York University Press, 2006).

15. Nelly Oudshoorn and Trevor Pinch, *How Users Matter: The Co-Construction of Users and Technology* (Cambridge, MA: MIT Press, 2003).
16. Designed by John Carmack and optimized by Michael Abrash. A good summary of the technical innovations in *Quake* is provided by Abrash, *Ramblings in Realtime*, esp. chap. 6: *Quake's 3-D Engine: The Big Picture* (2000), gamedev.net Web site, <http://www.gamedev.net/reference/articles/abrash/abrash.pdf>. These essays were revisions of chapters in Michael Abrash's *Graphics Programming Black Book: Special Edition* (Scottsdale, AZ: Coriolis, 1997).
17. Jack Thompson, *Bloodlust Video Games Put Kids in the Crosshairs*, *Denver Post*, May 30, 1999, G1.
18. However, the original version of *DOOM* was not released as open-source software; rather, id issued a "Data Utility License" that allowed modification of the game software under strictly defined conditions. With the release of *DOOM II* in 1994, Romero released more information about the game program. Carmack released the *DOOM* source code as a Christmas present to the player community in December 1997.
19. Id Software, n.d. id Software Backgrounder, n.d. From: id Software Web site, at: <http://www.idsoftware.com/business/home/history/>.
20. Randall Packer, *Net Art as Theater of the Senses: A HyperTour of Jodi and Grammatron*, *Beyond Interface* (2004). http://www.archimuse.com/mw98/beyond_interface/bi.fr.html (accessed June 2007).
21. John Carmack, *Re: Definitions of Terms*. Discussion post to Slashdot. 2002. <http://slashdot.org/comments.pl?sid=25551&cid=2775698> (accessed January 2007).
22. Hugh Hancock, *Ranger Gone AWOL*. *machinima.com*. http://www.machinima.com/articles/Ranger_Gone_AWOL/index.shtml.
23. Stewart Brand, *SPACEWAR: Fanatic Life and Symbolic Death Among the Computer Bums*, *Rolling Stone*, December 7, 1972. Also available from <http://www.wheels.org/spacewar/stone/rolling.stone.html>.
24. D. J. Edwards and J. M. Graetz, *PDP-1 Plays at Spacewar*, *Decuscope* 1, no. 1 (1962): 2–4. See also J.M. Graetz, *The Origin of Spacewar!*, *Creative Computing*, 1, no. 1 (1983): 78–85.
25. Brand, *SPACEWAR*.
26. Ledmeister, *What is a DOOM/DOOM II Demo (or LMP File)?* *Classic Doom*. <http://www.classicdoom.com/>.
27. E-mail from Laura "BahdKo" Herrmann to Henry Lowood (28 Jan. 2004). Interestingly, it has long been suspected that some star players abstained from the practice of delivering demo movies of their gameplay in order to maintain their competitive edge. However, such reluctance seems to have been unusual and the absence of demo movies from players such as Thresh (whose fame is based on his dominance in *Quake* death match) is now seen as evidence that they were not as active in the *DOOM* scene as word-of-mouth would have it. This shows the necessity of spectatorship and community, as well as skill, in achieving the status of a "God-like" player.
28. Welcome to the *DOOM* Honorific Titles! *DOOM Honorific Titles*. <http://www-lce.eng.cam.ac.uk/~fms27/dht/dht5/#dht5>. An excellent compilation of demo movies of all sorts can be found at the *Doom2.net* Web site, <http://www.doom2.net/>, along with recording and training tools.
29. Within limits, of course. Some players were known not to reveal all their best tricks.
30. Extensive documentation on the history of copyparties and demoparties has been lovingly preserved by the demoscene itself. For one example, see "CSDb: The C-64 Scene Database," <http://noname.c64.org/csdb/>.

31. Interview of the Month! This time: SLH/Hotline. *Illegal: The Cracker Magazine 2* (1987). This, appropriately, is from an unauthorized excerpt of the English-language sections that apparently circulated as a text file, though the original was printed in Germany. CSDb, entry for Illegal #22, <http://www.df.lth.se/~triad/triad/ftp/Illegal/illeg22.txt>
32. Saxon Druce, quoted in Vincent Scheib, Introduction to Demos & the Demo Scene: How They Relate to Games, and Their Appearance at SIGGRAPH, *Gamasutra* (2001). <http://www.gamasutra.com/features/20010216/scheib.01.htm#a1>.
33. E.g., Uwe Girlich's unlocking of the *DOOM* and *Quake* demo formats.
34. Examples of pre-rendered games would be Don Bluth's *Dragon's Lair* or the enormously successful *Myst*. But these are exceptions that prove the rule with regard to games based on 3-D engines.
35. Richard Schechner, *Performance Studies: An Introduction* (London: Routledge, 2002), 191.
36. Berry, n.d.
37. Uwe Girlich, The Unofficial DEM Format Description Version 1.02, July 30, 1996: 3.2 and 3.4. <http://www.gamers.org/dEngine/Quake/Qdem/dem-1.0.2-3.html#ss3.2.a>.
38. Cf. the Keygrip 2 Web site, <http://keygrip.planetquake.gamespy.com/>.
39. For more detail on this phase in the early history of machinima, see Henry Lowood, 2006; Andy Clarke and Grethe Mitchell, 2006.
40. Michael Abrash, *Michael Abrash's Graphics Programming Black Book. Special Edition* (Albany, NY: Coriolis, 1997), xxxii, xxxiii, 1190.
41. JoAnne Yates, How Business Enterprises Use Technology: Extending the Demand-Side Turn. *Enterprise and Society* 7 (2006): 423
42. Wiebe E. Bijker, Thomas P. Hughes, and Trevor Pinch, eds. *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology* (Cambridge, MA: MIT Press, 1987).
43. Nelly Oudshoorn and Trevor Pinch, *How Users and Non-Users Matter* (2003). See Ronald Kline and Trevor Pinch, Users as Agents of Technological Change: The Social Construction of the Automobile in the Rural United States, *Technology and Culture* 37 (1996): 763–95 for a pioneering study.
44. Oudshoorn and Pinch, 4.
45. A small sample of important texts, in addition to those already cited: Stuart Hall, Encoding/Decoding, in *Culture, Media and Language*, ed. Stuart Hall (London: Hutchinson, 1980), 128–38; Henry Jenkins, *Textual Poachers: Television Fans and Participatory Culture* (New York: Routledge, 1992); John Fiske, The Cultural Economy of Fandom, in *The Adoring Audience: Fan Culture and Popular Media*, ed. Lisa Lewis (London: Routledge, 1992), 30–49; Susan Strasser, Consumption Conspicuousness: Transgressive Topics go Mainstream, *Technology and Culture* 43 (2002): 755–570; Cornel Sandvoss, *Fans: The Mirror of Consumption* (Cambridge, England: Polity, 2005); Eric von Hippel, *Democratizing Innovation* (Cambridge, MA: MIT Press, 2005); Jenkins, *Convergence Culture: Where Old and New Media Collide* (New York: New York University Press, 2006).
46. Kline and Pinch, *Users as Agents*, 765.
47. Kline and Pinch, *Users as Agents*, 767.
48. Yates, *How Business Enterprises Use Technology*, 442.
49. Cf. Keith Grint and Steve Woolgar, eds. Configuring the User: Inventing New Technologies, in *The Machine at Work: Technology, Work and the Organization* (Cambridge, UK: Polity, 1997), 65–94; Hugh

Mackey, Chris Crane, Paul Beynon-Davies, and Doug Tudhope, Reconfigure the User: Using Rapid Application Development, *Social Studies of Science* 30 (2000): 737–757.

50. Including *Quake II*, released in 1997.
51. Strange Company, Company History (2005), <http://www.strangecompany.org/page.php?id=19>.
52. Ill Clan press release, February 12, 2001, <http://www.illclan.com/ILL%20Clan-Showtime-PR.htm>.
53. Exceptions include the Lithtech Film Producer/Player, co-developed with the machinima group Strange Company, and Epic's sponsorship with NVidia of the "Make Something Unreal" machinima contest.
54. Other reasons for the demise of the *Quake* movie included aspirations among machinima makers to produce for Web video or other media and improvements in game technology introduced by other game systems.
55. Michael E. Ryan, Quake III Arena, *PC Magazine*, June 26, 2001: 176.
56. Matt Pritchard, How to Hurt the Hackers: The Scoop on Internet Cheating and How You Can Combat It, *Game Developer*, July 24, 2000. http://www.gamasutra.com/features/20000724/pritchard_01.htm.
57. Uwe Girlich, The OpenDemo Project: Mission Statement. <http://www.machinima.com/opendemo/od-about.html>. His only recourse was to invent his own protocols through this OpenDemo project.
58. Hugh Hancock, Upcoming Technology: 2/00, *machinima.com*, <http://www.machinima.com/article.php?article=403>.
59. Hancock, *Upcoming Technology*.
60. Pritchard, *How to Hurt the Hackers*.
61. Paul Marino, *3-D Game-Based Filmmaking: The Art of Machinima* (Scottsdale, AZ: Paraglyph, 2004), 11.
62. Machinima, *The Word Spy*, Posted August 9, 2002. <http://www.wordspy.com/words/machinima.asp>.
63. The only exception I have found is Bailey's use of "machinema" in "Suggested Quake1 toolkit," (28 Jan 1999), ReMaic Web site, <http://www.machinima.com/remaic/index.shtml>.
64. <http://www.machinima.com/tritin-films/>.
65. Though *Half-Life* was based on *Quake* technology.
66. I am indebted for this line of thinking to Galen Davis of the Stanford How They Got Game Project. He is preparing an article on machinima and performance theory for publication. On camera phones and the emergence of new content, see: Daisuke Okabe and Mizuko Ito, Camera Phones Changing the Definition of Picture-Worthy, *Japan Media Review*, August 29, 2003. <http://www.ojr.org/japan/wireless/1062208524.php>, also Justin Hall's reports on weblogging and camera phones in *The Feature*, such as Rehearsing the Future: First International Moblogging Conference Report, *The Feature*, July 8, 2003. <http://www.thefeature.com/article?articleid=24815&ref=30721>.
67. From The Sidney and Harriet Janis Collection on view at Museum of Modern Art, New York, http://www.moma.org/collection/browse_secondary_images.php?criteria=O%3AOD%3AE%3A81631&page_number=1&template.id=1&sort_order=1&sec_img=1.
68. Strange Company, *Company History*.
69. Xavier Lardy, An Interview with Paul Marino, [www.Machinima.fr](http://www.machinima.fr), February 11, 2006. http://www.machinima.fr/index2.php?option=com_content&task=view&id=16&pop=1&page=0&Itemid=41.

70. Xavier Lardy, An Interview with Paul Marino.
71. Hugh Hancock, Making Machinima: Part 1, *machinima.com*, July 31, 2002. <http://www.machinima.com/article.php?article=302>.
72. Much of what follows on *WoW* movies is taken from my Storyline, Dance/Music, or PvP? Game Movies and Community Players in *World of Warcraft, Games & Culture* 1 (Oct. 2006): 362–382.
73. United Canadian Alliance Web site. <http://clanuca.ca/index.php?content=about&PHPSESSID=2383368b5ff886e3c04ba1f0f1baab7c>.
74. Tristan Pope, Crafting Worlds Web site (2005). <http://www.craftingworlds.com/>. Movies available at <http://www.craftingworlds.com/videos.html>.
75. *World of Warcraft* 1.3 patch notes (2005). <http://www.worldofwarcraft.com/patchnotes/patch-05-07-04.html>.
76. Caydiem, *Subject: Re: Rated M for Mature: LOCKED Was a good run!* Posted April 22, 2005 to General Discussion Forum. 2005. <http://forums.worldofwarcraft.com/> (accessed May 2005).
77. Necrotus, *Subject: Re: Rated M for Mature . . .* Three posts dated April 22, 2005, and response by Fairon dated April 23, 2005. General Discussion Forum. 2005. <http://forums.worldofwarcraft.com/>.
78. Before the controversy ran out, two discussion threads devoted to it would gather nearly 800 replies with more than 200,000 views. Pope. 2005. *Subject: Rated M for Mature . . .* Posted April 22, 2005 as “Tristanmon,” in General Discussion Forum. <http://forums.worldofwarcraft.com>.
79. Henry Lowood, 2006; Andy Clarke and Grethe Mitchell, eds., *Videogames and Art: Intersections and Interactions* (London: Intellect, 2007), 59–79.