High-Performance Play: The Making of Machinima

Henry Lowood
Stanford University

<DRAFT. Do not cite or distribute. To appear in: *Videogames and Art: Intersections and Interactions*, Andy Clarke and Grethe Mitchell (eds.), Intellect Books (UK), 2005. Please contact author, lowood@stanford.edu, for permission.>

Abstract: Machinima is the making of animated movies in real time through the use of computer game technology. The projects that launched machinima embedded gameplay in practices of performance, spectatorship, subversion, modification, and community. This article is concerned primarily with the earliest machinima projects. In this phase, *DOOM* and especially *Quake* movie makers created practices of game performance and high-performance technology that yielded a new medium for linear storytelling and artistic expression. My aim is not to answer the question, "are games art?", but to suggest that game-based performance practices will influence work in artistic and narrative media.

Biography: Henry Lowood is Curator for History of Science & Technology Collections at Stanford University and co-Principal Investigator for the How They Got Game Project in the Stanford Humanities Laboratory. A historian of science and technology, he teaches Stanford's annual course on the history of computer game design. With the collaboration of the Internet Archive and the Academy of Machinima Arts and Sciences, he is currently working on a project to develop The Machinima Archive, a permanent repository to document the history of Machinima moviemaking.

A body of research on the social and cultural impacts of interactive entertainment is gradually replacing the dismissal of computer games and videogames as mindless amusement for young boys. There are many good reasons for taking computer games[1] seriously. Is the claim that computer games have emerged as an art form one of them? Henry Jenkins, for one, has suggested that it is indeed time to think about video games as the "art form for the digital age." Some will find this thought difficult to reconcile with *Pong* and *Pokemon,* or with images of children staring vacantly into a Game Boy. Jenkins observes that such reactions "tell us more about our contemporary notion of art—as arid and stuffy, as the property of an educated and economic elite, as cut off from everyday experience—than they tell us about games."[2] It is tempting to respond to the rejection of games as an artistic medium by citing the narrative and social aspirations, technical mastery, visual storytelling, strategic depth, or simulated reality found in the palette of game genres today. For computer games today do cover an astonishing range of entertainment, simulation, artistic, competitive, and narrative experiences. Tempting as it may be to tout the artistic status of games in this way, it says little about the direct impact on contemporary artistic practice —demonstrated and potential— of performance rooted in computer games. In this article, I will narrow the focus of this relationship even further to the history of Machinima, a new narrative medium that has sprung almost whole out of computer game technology and play.

Machinima movies transform gameplay through performance, spectatorship, subversion, modification, and player communities. The ways in which early machinima projects defined the "convergence of filmmaking, animation and game development" that became machinima are instructive.[3] They certainly tell us something about the impact of improvements in computer graphics and game technology, but the history of machinima is more than a story about the rise of real-time animation techniques since the mid-1990s. Like the cell phone camera craze, we also learn from machinima how the dissemination of accessible tools—even if they are not necessarily easy-to-use—creates opportunities for the emergence of unexpected content in a postmodern environment that places playful experiments and throwaway pieces alongside startling and original instances of creative expression. [4] Machinima reminds us that the nature of computer games as software allows for an almost limitless flexibility of content, the potential of which has yet to be fully explored.

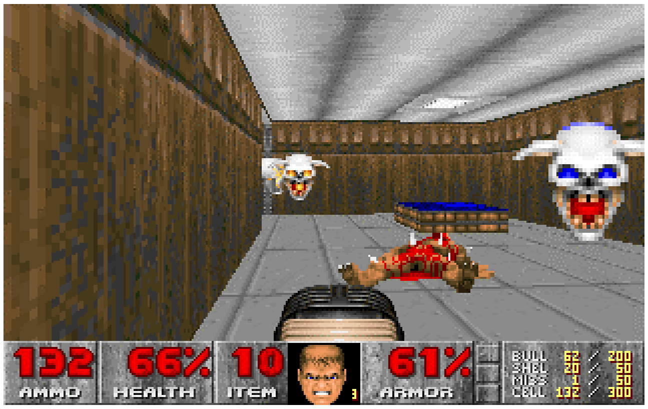
This article will be concerned with machinima as an episode in the history of game performance, design, and technology. Depicting machinima as *high-performance play* stems from its emergence from inter-relationships of play, spectatorship, technical virtuosity, and storytelling in computer games. Each of these factors played a role in defining the practices of machinima as practices of game performance. Already in its early history (1996-1997), we can identify elements of machinima that are particularly indicative of narrative and artistic performance generally: play as performance, modification of content, and community-based tools and content development. At the end of this article, I will visit the question of what machinima tells us about the appropriation of game performance (both play and technology) as an art form.

The word "machinima" was derived from "machine cinema."[5] A more apt derivation might be "machine animation."[6] Whether we think of machinima as cinema or animation, it means making animated movies in real-time with the software that is used to develop and play computer games. 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; pre-rendered animation has limited applicability in software that must immediately respond and redraw the screen in response to player actions.[7] Games such as "first-person shooters" raise the stakes of this technical challenge. In order to immerse the player in the rapid action of the game, they must render the virtual environment as a 3-dimensional space seen from the player's point of view, constantly re-rendering at high frame-rates as the player "moves" through that space. Since the early 1990s, specialized hardware and software solutions have been found to the computationally intensive problem of re-drawing these views on the fly (in "real time"). Machinima makers have learned how to re-deploy this sophisticated software for making movies, relying on their mastery of the games and game software. Beginning as players, they found that they could transform themselves into actors, directors and even "cameras" to make these animated movies inexpensively on the same personal computers used to frag monsters and friends in *DOOM* or *Quake*. They recorded their actions, generally in real time, as game files (logs of keystrokes or other information capable of being saved and replayed). These files could then be distributed over the Internet at almost no cost, either in the original game replay files or in an encoded movie format. Recently, as machinima has matured technically and artistically, attention to it has focused on its significance as a low-cost, efficient way to produce animated films that compete with hand-drawn or digital frame-based techniques. For example, the New York Times reports that, "Machinima movies, which range from short comedies to science-fiction epics, are produced entirely on computers, eliminating the need to buy costly equipment, rent spectacular locations or hire glamorous actors. The films are then distributed free over the Internet."[8] In another article that appeared in *New Scientist,* Hollywood animators, machinima makers, and game developers discussed the limitations and advantages of real-time animation as a possible business threat to intensive, frame-based methods such as those used by Pixar and Disney. Tom Sito, a distinguished Disney and DreamWorks animator, encapsulated the attitude of Hollywood animators to machinima by insisting that "real-time performance-based animation will never replace traditional methods." [9]

Portraying machinima as a potential threat to the tried-and-true methods of the movie industry has directed attention to its cost, technical quality, and scalability. By focusing on the trade-offs in abandoning time-intensive, frame-based animation for computationally intensive real-time rendering, the debate has obscured the implications of machinima emerging from game performance, a notion that encompasses cyber-athleticism, theatricality and the elevation of the player to producer.[10] It is therefore important to recall that the origins of machinima lie not in content production, but in gameplay.

Without a doubt, "play was the thing" in the early history of game movies. Id Software's first-person shooter games were the first stage. The authors of these games were a group of programmers, level designers and artists led by John Carmack and John Romero. They had founded id[11] in February 1991. By that time, Carmack had established himself as a uniquely talented graphics programmer. His first significant exploit along these lines was figuring out how to recreate the first level of *Super Mario Brothers* , a side-scrolling platform game, on a personal computer. The demonstration program was titled "Dangerous Dave in 'Copyright Infringement'", because other than the protagonist, it was a virtually perfect reproduction of one of Nintendo's flagship games, emulating its technology and game design. Despite the playful name, the project implied far more than homage or imitation; it signaled an impending shift in technical leadership and practice for game developers. During the 1990s, games developed primarily for personal computers, not for proprietary game consoles, would dominate innovation in the development of graphical game engines, the software that controlled the real-time generation of imagery and game physics.

By 1992, Carmack had locked with laser-like intensity onto the problem of solving the vast programming challenges associated with building realistic and immersive virtual worlds as settings for three-dimensional action games.[12] Having turned his full attention to 3-D graphics, Carmack achieved a major milestone with the graphics engine for id's *Wolfenstein 3-D,* released in May 1992. It depicted the game's action (mostly shooting) as the player's character would presumably see it by simulating the player's perspective into a three-dimensional virtual world. "But it could be better," as the *Book of id* proclaimed about *Wolfenstein 3-D* a few years later. "The games weren't real enough yet."[13] *Wolfenstein 3-D* set the stage for *DOOM* as the breakthrough achievement of this game genre, which would be called, appropriately, the "first-person shooter." Released in December 1993, *DOOM* featured completely revised software with numerous technical and design improvements over *Wolfenstein 3-D*: a superior graphics engine, fast peer-to-peer networking for multiplayer gaming, a modular design that let authors outside id create new levels, and a new mode of competitive play devised by Romero called "deathmatch." This batch of innovations immediately transformed personal computer-based gaming. *DOOM* established competitive multiplayer gaming as the leading-edge genre of PC games during the mid-1990s.



Doom, screenshot from pre-beta "Press Release Version," dated Oct. 1993. Source: http://www.johnromero.com/lee_killough/versions/prnotes.shtml#shots. With permission of id Software.

Just as important as the improvements in graphics and networking technology, *DOOM* revised notions of authorship by allowing for game modifications, third-party level design, and the creation of independently-developed software tools. According to its own corporate history,

> “id Software didn't stop there, 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. In fact, id discovered many of their current employees and development partners based on mods that were created and distributed over the Internet.”[15]

Id’s self-conscious advocacy of the modification of its own software by the player community reminds us that the loss of authorial control is not generally a dilemma for game designers.[16] Indeed, they embrace subsequent modification of their work by others as the seed of an ongoing relationship with the player community. As the media artist and museum curator Randall Packer reminds us, computer games occupy a salient position in accepting the role of the player as co-producer of content:

> "While theater begins with the notion of the suspension of disbelief, interactive art picks up where theater (and film) leave off with branching, user-driven non-linear narrative. The letting go of authorial control has been the big dilemma of interactive works as an art and/or entertainment medium, games being the exception."[17]

Not only was the loss of authorial control not a dilemma; id openly embraced it. As developers of first-person shooters, defined even more than other game genres by actions rather than by fixed narrative content, Carmack and Co. focused on technology as the foundation for game development and openly encouraged the player community to modify their games.[18]

The nature of computer games as software leads to another sort of variability of content, one that has generally been limited to PC-based games. I am referring to what Lev Manovich has called the new "cultural economy" of game design introduced by *DOOM*.[19] When a computer game is released today, it is as much a set of design tools as a finished product. Game developers often provide software utilities for modifying their own games, sometimes including these tools in the packaged release of the game, or soon after publication for downloading from websites.[20]

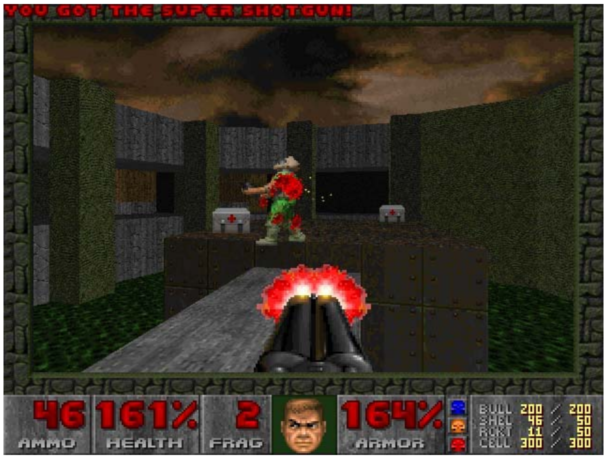
Independent level, scenario and mod designers take over from there, creating “mods” (modifications), ranging from cosmetic changes (new “skins” for avatars, models, textures, etc.) and add-ons to “total conversions” (essentially wholly new games). Websites and discussion lists build the community of mod authors and provide means for distributing their work, as well as exchanging information about content or tools. Manovich contrasts modifiable games to the characteristics of a more customarily authored game like *Myst,* which he describes as "more similar to a traditional artwork than to a piece of software: something to behold and admire, rather than take apart and modify."[21] In the game world, it was the modifiable first-person shooter, not the lavishly pre-rendered puzzle game, that created the means for machinima to become a new narrative medium.

The introduction of fundamentally new styles of play *and* modes of content development set the stage for early game movies using *DOOM.* These were largely demonstrations of gameplay made by recording actual matches. *DOOM*'s unprecedented success as a platform for competitive play heightened interest in the feats of stellar players, especially as word got out about their prowess in the growing player community. Players had competed publicly since the early days of computer games, whether in the Spacewar! Olympics at Stanford in the early 1970s or in Nintendo-sponsored tournaments of a later day.[22] But the modes of networked play by modem or in local area networks made possible by *DOOM'*s peer-to-peer networking architecture intensified multiplayer competition. Established teams were formed, and players operated networked games in offices, local area networks, and by connecting up directly with opponents in cyberspace. As id proudly noted in its press release announcing *DOOM,* it “is the first game to really exploit the power of LANs and modems to their full potential. In 1993, we fully expect [it] to be the number one cause of decreased productivity in businesses around the world.” Id also hinted in this announcement that their high-performance game technology—encompassing unprecedented immersive realism (3-D graphics) *and* multiplayer interaction—would profoundly affect virtual spectatorship as well as gameplay.

> “You can see the other player in the environment, and in certain situations you can switch to their view. This feature, added to the 3-D realism, makes *DOOM* a very powerful

> cooperative game and its release a landmark event in the software industry."[23]

As released, *DOOM* offered options for viewing games, and players soon took full advantage of the ability to record "demo movies" of gameplay. These demos were distributed as discreet files and replayed by other players with a copy of the game. Thus, *DOOM* linked unprecedented multiplayer competition, reproduction of gameplay as demo movies, and a context for spectatorship through the creation of a player community that would distribute and replay these movies. The result was nothing less than the metamorphosis of the player into a performer.



NoSkill, Screenshot from *DOOM* demo movie.

The intensity and rapid action of *DOOM*'s multiplayer deathmatch established a basis for the performer-spectator relationship. *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 everyone wanted to see them play, to gather insights into their play tactics and possibly learn a trick or two. 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."[24] If the goal was to see the game through their eyes, demo movies did just that. But they also certified the status of these star players. 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." The certification process explicitly promoted the performance of gameplay through demo movies:

> "An exciting feature of the game is the ability to record the player's input in a form that can be replayed later, like a movie. The file containing the recording, known as a LMP, can then be sent to other people for viewing.
>
> The DOOM Honorific Titles, based on LMP recording with an authentication mechanism, are the means by which good players can objectively prove to the world that they are as good as they claim. The DHT system also has the beneficial side effect of promoting the production of amazing LMPs—if you want to see some superior *DOOM* action, turn to the ever-growing repository of DHT exam files."[25]

These recordings duplicated the first-person perspective of the player, in essence providing a series of screenshots.[26] In the early *DOOM* demo movies, the player was not only a performer, he was quite literally the camera. Just as the protagonist's weapon was seen as emerging into the virtual space of the arena during competition, the spectator saw the same image—an appropriate projection of the player/actor in gameplay, but somewhat dubious as an intrusion blocking the passive spectator's full view into the screen. Establishing a basis for spectatorship by recording gameplay further encouraged growth of the player community, as well. Individuals and regular teams of players joined together as semi-official "clans" and established reputations based on superior play. Naturally, demo and game movies were used to put exploits on exhibit and document the skills of players and clans. Years later, the surviving demo movies put viewers in the shell of the ghosts of players. One of the best surviving series features perfect reproductions of matches recorded as early as May 1995; these recordings allow us to look through the eyes of one of the first "game gods," NoSkill, having been preserved on the memorial site of this now deceased player.[27]

The machinima community always begins its historical reflections with the publication of *Quake* in mid-1996, the successor to *DOOM*. Work on *Quake* had begun in earnest during 1994. In August of that year, Carmack wrote:

> "BTW,[28] *Quake* development has started. It will run on (and use) arbitrary resolutions and color depths. It will support dozens of concurrent players and persistant game servers. It will be the coolest thing anyone has ever seen ..."[29]

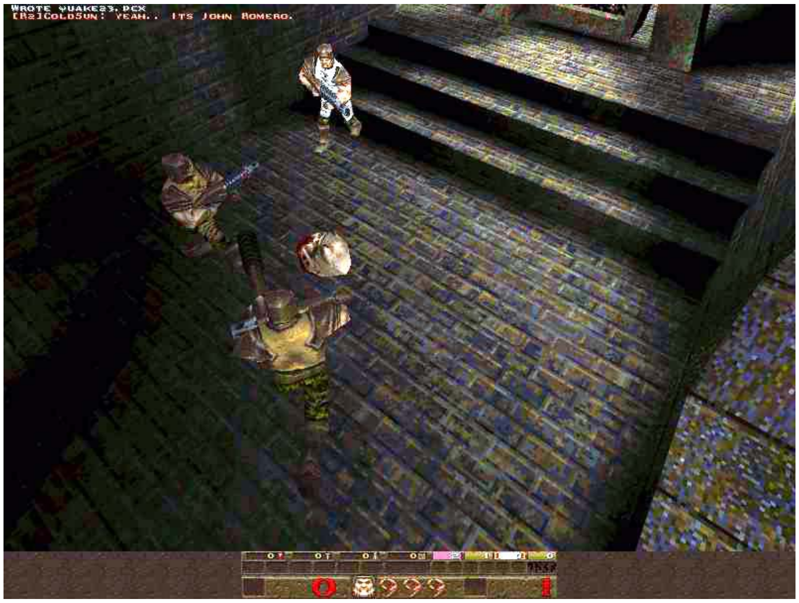
When it was released in June, 1996, *Quake* did indeed prove to be an enormous technical leap over *DOOM*. But not in its game design. It preserved *DOOM*'s fundamental modes as a competitive game, which had the effect of establishing genre boundaries of the first-person shooter. The technical improvements were more than incremental. For example, *Quake* introduced built-in client/server networking, thus stimulating the popularity of Internet-based multiplayer games, and it featured the first implementation of a genuinely 3-D graphics engine written by Carmack and optimized by Michael Abrash. This put demands on personal computer hardware, and *Quake* thus became a major factor in the rise of a consumer market for add-on video cards. The complexity of creating fully 3-dimensional game levels may have daunted many enthusiasts, but during the development of *Quake* it became clear that id would provide more information and tools for modifying *Quake* than they had for *DOOM*. A Usenet discussion group, rec.games.computer.*Quake*.editing, was launched in January 1996, and the participants filled it with discussion of how *Quake*'s levels could be edited and the game modified. In April, Carmack issued information about a new editing tool, *Quake*Ed, followed in May by dissemination of some *Quake* source code, and in June by the shareware version of the game.[30] The retail version would include the *Quake*C script programming language for modifying *Quake* itself, along with source code for *Quake*C. Access to *Quake*C, *Quake*Ed, or the tools created by programming talents in the *Quake* community made it possible to design new "skins" for the avatars or textures for the environment, to devise programming exploits such as the creation of ever more competent "bots" (robot opponents controlled by computer AI), to be the architect of new levels, and even to modify the game itself (as in the introduction of "team fortress" and "capture the flag" modes of play). Customization of *Quake* would become a new arena for demonstrating one's skills.

Quake, as software, was far more complex than *DOOM*, but it had also been made more accessible. This mixture of qualities set the stage for a new wave of high-performance game practice. As Douglas Thomas has noted in his study of hacker culture, programming feats alone do not a hacker make. The hacker emerges into public view by affiliating with other elite programmers, sharing information and refining skills in groups such as the infamous Legion of *DOOM* and Masters of Deception.

> "A second and related means by which hackers made their reputations was by sharing information in public forums, either by disseminating text files through the underground or by publishing in underground journals such as *2600* or *Phrack*. …These files served as means to solidify hackers' reputations, illustrating the degree to which they understood the systems they infiltrated."[31]

Moreover, as a "performance of technology,"[32] hacking hovers between a discourse of technological mastery and one of subversion, between practice as expertise or exploit. While the *Quake* player community operated in a different technical realm, its participants operated under similar practices and tensions of technical performance. Immediately after *Quake*'s release, players formed groups in response to the vast improvement of multiplayer connectivity and chat options over *DOOM*. Like hacker gangs dissecting the intricacies of computer networks, these *Quake* Clans provided a setting for sharing techniques of high-performance gaming, both play *and* programming. The Ranger Clan provides a telling example. Arguably the most famous clan of all, the Rangers' top-notch players contributed visibly to the technical community that grew around the game. They had participated in the first pre-release test of the *Quake* engine distributed to the *Quake* community. 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 25 members or so members remained active in game development or went on to work in the game industry.[33] With their reputation for stellar performances as players and programmers already firmly established, they surprised the *Quake* community in October 1996—barely a month after the commercial release of the game--with an exploit of another sort: the first machinima movie, “Diary of a Camper.”

The Rangers’ animated short resembles the demo movies of *DOOM* gameplay, with short bursts of frantic action punctuated by flying blood and bits of body parts. With spare visual reference to the *Quake* storyline, the plot offers little more than a brief sequence of inside jokes—the “camper” (a player who gains game advantage by camping in a prime location) is reduced literally to a headshot that reveals him to be none other than John Romero. Yet, “Diary of a Camper” breaks with the demo movie as documented gameplay in several important respects. First and foremost is the independence of the spectator’s view from that of any player/actor; the movie is not “shot” from the first-person perspective of the shooter. An independent camera view now frames the action. This innovation illustrates *Quake*’s significance as a platform for high-performance play: It could be exploited as a “found technology” for purposes other than those envisioned by the designers. Uwe Girlich, the leading technical authority on *Quake* movie-making, noted in his analysis of its new demo format before *Quake*’s release that “the player coordinates and the camera positions may be different.” He added wryly that “for people with too much spare-time *Quake* can replace a full 3D modelling system for cartoons or the like.” Even more was now possible, he claimed: “The demo file can contain console commands, which the client runs during replay. With this feature it should be possible to write a screenshot after every time stamp in the demo file. This makes it very easy to create a MPEG movie out of a DEM file.” [34] In other words, he had found an opportunity hidden inside the *Quake* programming code. One could now with a bit more sleep deprivation encode their game movies as video files that could be viewed even by those lacking the game software.



```
WROTE QUAKE23.PCX
[R2]COLDSUN: YEAH.. ITS JOHN ROMERO.
```

The Rangers, "Diary of a Camper" (1996).

The Rangers began work on "Diary of a Camper" before software tools for movie production had been built to realize Girlich's vision.   This *Quake* clan's motto, "Rangers Lead the Way," could be applied to their coding skills, as well. They devised their own programming hacks for editing the *Quake* Demo format. But Girlich made his inside knowledge known to the *Quake* community. As in hacker culture, sharing knowledge about the otherwise hidden aspects of the software would lead to impressive technical achievements, such as building tools that non-programmers could use to analyze and edit *Quake* demo files. Girlich had previously written the "Little Movie Processing Center" for *DOOM* and several other games that used the earlier LMP format for gameplay recording; he wrote a capable editor for *Quake* movies*,* as well. These coding exploits provided a foundation for David "CRT" Wright's influential Keygrip and Keygrip2 programs, which became the most widely used utilities for editing and other post-production work on machinima movies. Wright was a mod coder known among *Quake* players for the Rocket Arena series of one-on-one dueling games and later developed Gamespy Industries' Gamespy Arcade system for networked multiplayer matchmaking. Girlich and Wright thus epitomized the high performance player as programmer, performing exploits neither of gaming nor film-making prowess, but of technical mastery.

Like the hackers' exploits, making movies with game software required a mixture of expertise and subversion. The second sense in which play is at work in machinima comes from a notion I am borrowing from Katie Salen, that of "transformative play." She writes:

> "Because the creators of emergent systems, like generative music or games, can never fully anticipate how the rules will play out, they are limited to the design of the formal structures that go on to produce patterns of events. Sometimes the forms of play that emerge from these structures overwhelm and transform, generating rich and resistant outcomes. Sometimes, in fact, the force of play is so powerful that it can change the rule structure itself."[35]

Salen argues that the "free movement of play" allowed within robust game systems is capable of generating unexpected rules that fundamentally alter the nature of the game, in effect generating a new creative space altogether. The result is unexpected forms of play.[36] The Rangers give us one example in their transformation of competitive play into theatrical play. Another example of transformative, high-performance play that set the stage for machinima is the variant of *DOOM* and *Quake* known as speedrunning.

The *Quake* done Quick team, speedrunners *par excellence*, have defined their work/play as "a project to record runs of multi-level sequences from *Quake* and its friends in the fastest times possible. Any short-cut except cheating is approved of, and Nightmare difficulty is assumed."[37] This is neither deathmatch competition nor gladiatorial combat like Rocket Arena; rather, it is a single-player show that combines virtual gymnastics, game engine analysis, trickery and expert gameplay. In the words of one of the leaders of the *Quake* done Quick team, "speed-running offers another way to compete at *Quake*."[38] Speedrunners dash through game levels by all means available, including stepping on grenades to jump up or past enemies or igniting rocket launchers to shoot themselves rapidly through the virtual space. Runs are timed, then documented as demo movies and often edited together with other runs to make complete movies that race all the way through the game, level by level. As new tricks are invented that cut down on the time in this or that level, runs are re-recorded and the movie re-edited. Speedrunning was not unknown in *DOOM,* but the constraints on camera control, recording and playback limited the "filming" of speedruns, just as they did demo movies. Opening up the hood of the *Quake* game engine to programmers and players transformed speedrunning.

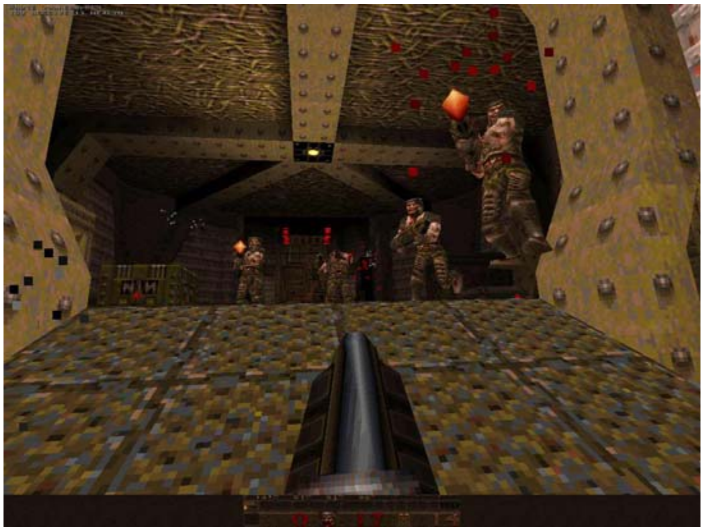
Like the Rangers, the *Quake* done Quick team got to work shortly after the release of *Quake*. "In late 1996," a graduate student in computer science at the University of Manchester named Anthony Bailey viewed a speed-run by Yonatan Donner. Donner had already made a name for himself by speedrunning through *DOOM* levels. Watching this run, Bailey "caught the bug" immediately. "Figuring I could maybe barter some programming skills and ideas for a piece of the action, I wrote to Yonatan a couple of months back with a few ideas about how the time might be improved."[39] They banded together with two other *Quake* players to make speedrunning movies. In order to provide an independent, floating-camera view of the action, Bailey programmed a software utility to "recam" (re-camera) *Quake* demo movies, building on Girlich's work on LMPC. He called the program Remaic ("remake"), because it was now possible to revise the camera view on portions of a speedrun recorded in realtime without re-recording the run. Recamming did not literally mean refilming; *Quake* recorded the movements of player-actors, rendering them into data for the game engine's built-in replay facility. The software only needed to manipulate the data to produce a new shot. The *Quake* done Quick team then compiled and recompiled these data using Girlich's LMPC program. The "director" enjoyed virtually complete freedom of moving the camera view. Bailey used Remaic and other available software utilities to shift camera views, connect runs into a continuous movie, or add sounds and text. In early 1997, the team finished "The Elder Whirled." This short ran a little over five minutes and would later become the fourth episode of "*Quake* done Quick." Put together as an entry in the *Quake*lab Multimedia Contest, the run was never released separately, but "The Elder Whirled" was the first use of Remaic to recam a *Quake* speedrun, thereby becoming the Team's first "refilmed movie."[40]

When it was released in June 1997, "*Quake* done Quick" demonstrated more than impressive playing skills or the technical wizardry of its makers. It signaled a shift from cyberathleticism to making movies and the emergence of a new form of play. As if to underscore the transition, the

Team released two versions of the complete set of speedruns, which lasted nearly 20 minutes after stitching together the individual runs for each level of the game. The first was visually a conventional demo movie viewed from the first-person perspective of the player; the second was the recammed movie. The technical performance involved in recording separate demos and patching them together to make either version, all while preserving a smoothly integrated whole, was of course non-trivial. So was the performance itself. As Bailey put it when describing his work on the project, preparation for a perfect speedrun meant "trying to understand more about how the engine underlying the game works so that we can turn it's little nooks and crannies to our advantage." Even though he considered his skills at conventional *Quake* play "woefully lacking," this was "experimental physics in a whole new universe," finding anomalies or quirks in the world that might help to shave a few seconds off the time to get through a level. But unlike real-world physicists, Bailey could also examine the code underlying the world and build tools to monitor its physics during every run.[41] The *Quake* done Quick collaboration thus represented a meeting of programming and play, a dual performance. But that was not all. Just as speedrunning itself was a reformulation of the rules of play, the collaborators also found that the making of movies became a meta-game of another sort:

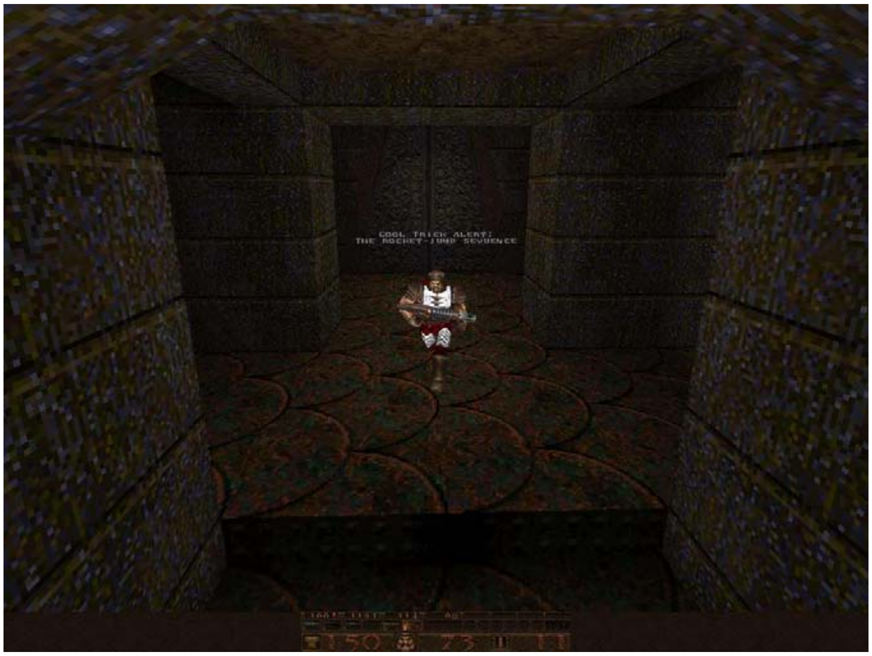
> "Continuity is very important to us; a demo made simply by pasting together unrelated recordings of each level in turn would not be a proper run of *Quake*. Continuity adds a lot of strategy to the game, since items that one finds in earlier levels can be used in later ones, but those that one avoids to save time cannot. Is it worth the deviation to get that grenade-launcher? Can one afford to sacrifice health at some point to save time? Many subtle interconnected decisions have to be taken. It's great fun!"[42]

Indeed, the game systems interacted. In his write-up of the making of "*Quake* done Quick," Bailey commented on the suspicion that "some degree of demo hackery" would be required to stitch together runs of such perfect gameplay. Neither cheating nor hacking were acceptable to the group, only "spectacular human play," not "bot-driven hackery." He argued that, "this is as close to a 'proper' run through *Quake* as we believe it is possible to get."[43]



“Quake done Quick” (first-person version). Screenshot

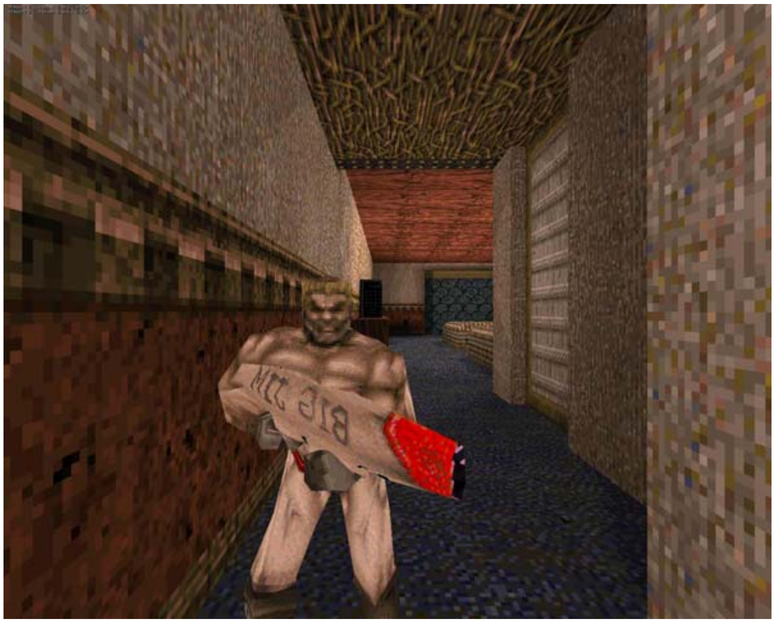
The second version of “*Quake* done Quick” revealed the third leg of the project’s stool, adding cinematic to technical and game performance. Having “wondered if maybe we could alter the camera position in our demos to make them more movie-like to watch,” Bailey used Remaic to create a new recammed version of the speedrun. It was the same run, but now the camera floated behind, alongside, above, and in front of the protagonist, sometimes zooming in or zooming out. “*Quake* done Quick” thus joined “Diary of a Camper” in breaking the visual identity of player, camera, and spectator that had characterized *DOOM* demo movies. Machinima (as such *Quake* movies would come to be called) in turn revised the presentation of game performance, as replay movies would be recammed as well. When legendary *Quake* 2 players Thresh and Billox went up against each other at the British *Quake*adelica tournament in 1998, the recording of their one-on-one deathmatch became one of the most popular *Quake* movies and contributed to Thresh’s reputation as the world’s best competitive gamer. Originally captured from a spectator perspective, the unedited demo movie was recammed into its canonical version by Phil “Overman” Rice, the founder of a machinima production company, Zarathrustra Studios.[44]



“Quake done Quick” (version with floating camera view). Screenshot.

“Diary of a Camper” and “*Quake* done Quick” established the *Quake* movie as high-performance play. In the year after the full release of id’s game, these movies were joined by a third project that incorporated elements of traditional animation and performance. “Operation Bayshield,” released by Clan Undead in January 1997, completed the trilogy of seminal machinima projects. “After seeing what the Ranger Clan had done,” this group of friends decided on New Year’s Eve to “make something we thought would be fun,” namely, a “larger comedy film in the *Quake* engine.” [45] They departed from the reliance of “Diary of a Camper” and “*Quake* done Quick” on gameplay as in demo movies. Instead, Clan Undead borrowed extensively from traditional linear media, both in conventions of storytelling and playful allusions to popular media such as television and cinema. The title and plot referred to the popular television series “Baywatch,” while the title credits (a remediation[46] in machinima of movies) parodied traditional movie openings and soundtracks.

“Operation Bayshield” offered two noteworthy innovations that would become standard feature of machinima. The first was in customization of *Quake*’s content. Clan Undead dressed the characters (avatars) in new skins, manipulated the character images to simulate lip synching, and added visual effects. While the movie appears to have been recorded in a small number of continuous runs, these modifications of game content required pre- and post-production work, largely programming of scripts in *Quake*C. In the absence of available utilities at that time, the makers of "Operation Bayshield" performed not only as actor-players, but also as coders in the tradition of level designers and *DOOM* hackers. Id's invitation to look under the *Quake* engine's hood encouraged these efforts and may have inspired Clan Undead to release the source code for their movie scripting in April 1997.[47] Like the larger *Quake* community, machinima clans were encouraged to publicize their exploits not just as players, but also as content developers. Just as after *Quake* the modification of games progressed from scenario to level design to full-scale total conversion mods, machinima makers likewise grew bolder as they learned more about the graphics and programming resources *Quake* provided. The techniques that produced custom artistic assets for machinima movies were essentially identical to those used to create game mods: creating skins for avatars, designing new visual effects, adding new sounds, and, in time, producing new models and graphics.



"Operation Bayshield." Screenshot.

Clan Undead's second major innovation could be found in their singular attention to the movement and mannerisms of virtual actors. They devised tricks to suggest crude lip movements during character dialogue, and when a character laughed at a joke, the avatar seemed to shake his body appropriately. In "Operation Bayshield," the onscreen "actors" spoke their lines, an advance Clan Undead probably achieved by dubbing audio encoded as .wav files and activating them through *Quake*C's sound command at the appropriate points in the movie. This was a first step towards what Paul Marino, one of the founding members of the Ill Clan, would call the virtual puppetry of machinima, that is, the careful synchronization of avatar actions (moving, speaking) to voice actors/game players via keyboard bindings.[48] Unlike "Diary of a Camper" and "*Quake* done Quick," Clan Undead suggested that theatrical play in *Quake* movies need not be wholly bound to gameplay, that a suite of performance gestures could be more closely mapped to narrative and character. This freedom of theatrical performance could not be achieved, however, without programming prowess and playing skills.

The early history of Machinima illustrates a number of themes in the appropriation of game technology to create a new narrative, even artistic medium. I would identify these as technologies of modification, subversion, and community-developed content. Id Software's decision to embrace and extend the player community's role in creating new *DOOM* levels set the stage for the unprecedented degree to which it opened up access to the game engine inside *Quake*. Not only did providing an editor and scripting language stimulate modification and extension of the game, it encouraged the development of tools for unforeseen purposes, such as the editing of demo movies and, eventually, the making of animated movies using real-time techniques of gameplay as performance. While these modifications were sanctioned by id, they were also subversive. In this regard, I am insisting that Salen's notion of transformative play applies to the underlying technology of computer games as well as to game design. In other words, the technology became a field of play, but not just in order to play the game of optimizing game performance; less than a year after *Quake*'s release, game software was used—playfully—as a technology for making movies. As speedrunning became a new game form within the structure of play provided by *Quake,* machinima makers subverted the game system altogether, turning it into a performance technology. Machinima meant narrative or experimental movie making, not competition. Just as important, machinima benefited from the strong social network spawned by multiplayer gaming. Knowledge of the capabilities built into *Quake* and access to independently-developed tools disseminated rapidly in the virtual community of *Quake* players. The clans and project members deploying this knowledge added to it in every one of the early machinima projects, in turn publicizing a body of work that consisted of movies, software tools and techniques. Exploits of high-performance gameplay, programming, and storytelling were not isolated achievements or acts of creativity; performers crave spectators, and the existence of a gaming community engaged at every level of their work—clans of players, teams of movie-makers, or virtual networks of programmers and tool builders—cannot be underestimated as a factor in high-performance play.

The founding trilogy of machinima--"Diary of a Camper," "*Quake* done Quick," and "Operation Baywatch"—were completed less than a year after *Quake* launched. By comparison to early film, these experiments in the use of computer games to produce animation rank somewhere between the Lumière Brothers footage of a train pulling into a station (1895) and Georges Méliès' "Le Voyage dans la lune" (1902). They are milestones in the evolution of a new narrative medium, pointing forward to new possibilities of expression. Superior techniques and software resources have since been added to the machinimist's box of tools, paced by the rapid evolution of 3-D graphics software and hardware available on personal computers. Machinima is now capable of producing animated shorts (such as the Ill Clan's "Hardly Workin'"), music videos (Ken Thain's "Rebel vs. Thug" or Fountainhead Entertainment's "In the Waiting Line"), feature-film length animated movies (Jake Hughes' "Anachronox"), *avant-garde* shorts (Dead on Que's "Fake Science" or Fountainhead's "Anna"), cutscenes and trailers for video games (*Metal Gear Solid)*, episodic comedy programs à la television ("Red vs. Blue") and even live improvisational comedy (Ill Clan's "Common

Sense Cooking with Carl the Cook"). It is safe to predict that game developers will soon put robust but easy-to-use machinima tools directly into the hands of an increasing number of players.[49] The "family movies" produced by players of *The Sims* foreshadow this trend, and we can expect to see machinima embedded in Peter Molyneux's upcoming Hollywood studio simulation, *The Movies*. The rapid technical and artistic progress of machinima in the nearly eight years since the initial trilogy of projects has brought it to the point of becoming a viable threat to traditional frame-based animation.

© 2001 The ILL Clan, Inc.

The ILL Clan, "Hardly Workin'" (2001).

But is machinima art? What are we to make of claims like those made by Katherine Kang, the founder of Fountainhead Software:

> "Machinima's the new kid on the block so it has yet to prove itself. If game technology moves forward at the pace I believe it will, machinima will revolutionize animation. Like CGI [computer graphic imagery], machinima will have a place in animation history. We will have CGI, stop motion, claymation, anime, and machinima as the primary styles of creating animated features if you're not into traditional cell animation."[50]

For Kang, machinima is the new art form. Whether computer games can lay the same claim is unimportant, but at the same time it is clear that the progress of game technology will pace the growth of machinima as an art form. Ironically, contemporary machinima projects such as Kang's own "Anna" have made such tremendous strides in graphics and narrative expression that traces of game performance and practice have become difficult to discern. But that is exactly the legacy of computer games in machinima; we can imagine a real-time animation engine without them, but not the same historical culture of modification and experimentation. When a computer game is released today, it is as much a set of design tools as a finished game design. PC game developers are routinely releasing their development tools for experimentation and play, that is, play with technology and play with animation, stories, graphics and movies as much as play with games. With the increasing popularity of mods, game developers routinely put impressive editing and cinematic tools in the hands of the player community, encouraging everything from the creation of new graphics to initiation into game development as a career. The contemporary game scene pulses with the energy of player communities that use game engines to create something new.

A little over two years ago, in the Wattis Theater of the San Francisco Museum of Modern Art, Benjamin Weil of SF-MOMA hosted an event called ArtCade. A panel of speakers, including Will Wright, spoke about the possibility that video and computer games might be considered an art form. Before the talks, the speakers and attendees gathered in a room equipped with two small installations, the ArtCade itself. The first consisted of a small collection of video games and the second of digital art inspired by game technology or culture. I learned from this prelude to the ArtCade lectures that the technologies and practices of game design were beginning to have a huge impact on artists, particularly new media artists. But I could not help noticing that while visitors confidently walked up to game screens and immediately immersed themselves in the screen action--even while balancing a drink in one hand--at the other end of the room they stopped and ruminated about what they were seeing, joked about meaning, and puzzled over the intentions of the artists.

It seems to me that too much thinking about the status of games as artistic works, while stimulating and useful, will distract us from what might really be the lead story on the interplay between artistic expression and the new medium of interactive computer and video games. Perhaps we should stop worrying so much about this question. Rather, it may be worthwhile to shift attention to the technologies and practices emerging out of computer games as a medium, and how they might provide new avenues for cultural, artistic and social expression, including performance art. Perhaps we should not bother too long with the question, "are games art? Instead, we should ask high performance play is capable of transforming our notions of how art is created.

NOTES:

I would like to thank several people for their helpful comments. Galen Davis gave the paper a particularly careful reading and, as always, was generous in sharing his insights about performance theory and machinima. From the How They Got Games Project, Tim Lenoir, Casey Alt, Douglas Wilson and, on *Quake* especially, Rene Patnode have helped me think about games as technology and as performance medium. I am also grateful to Anthony Bailey, Paul "ILL Robinson" Marino, Laura "BahdKo" Herrmann for their comments, time and permission to use screenshots. Finally, thanks to Adrian Carmack of id Software for permission to use screenshots from *DOOM* and *Quake*.

[1] I use the term "computer game" as a catch-all phrase for many forms of software-based electronic entertainment, principally videogames, PC games, handheld games and arcade consoles.
[2] Henry Jenkins, "Art Form for the Digital Age," Technology Review (Sept.-Oct. 2000). Available here: http://www.techreview.com/articles/oct00/viewpoint.htm
[3] Frank Dellario, "What is Machinima?" Website of the Academy of Machinima Arts and Sciences. http://www.machinima.org. Dellario is Ill Bixby of the machinima team known as the Ill Clan.
[4] 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* (29 Aug. 2003). URL: 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* (8 July 2003). URL: http://www.thefeature.com/article?articleid=24815&ref=30721.
[5] "Machinima," *The Word Spy* (posted 9 Aug. 2002). URL: http://www.wordspy.com/words/machinima.asp.
[6] Just as, following the name of Fountainhead Software's recently released machinima software tool, we might be tempted to recast machinima as "machinimation."
[7] Examples of pre-rendered games would be Don Bluth's *Dragon's Lair* or the enormously successful *Myst.* But these are truly exceptions that prove the rule.
[8] Matthew Mirapaul, **"**Computer Games as the Tools for Digital Filmmakers," *New York Times 151 (22 July 2002): E2.*
[9] Celeste Biever, "The Animation Game," 180 *New Scientist* (25 Oct. 2003): 28.
[10] On the "player-as-producer," see the chapter on "Games as Open Culture," in Katie Salen and Eric Zimmerman, *Rules of Play: Game Design Fundamentals* (Cambridge, Mass.: MIT Press, 2003): 537-53.
[11] In this article, I will refer to the company as id Software and the game as *DOOM*; other versions of capitalization are sometimes used, even by the company.
[12] On id Software, Carmack, and *DOOM,* see David Kushner, *Masters of DOOM: How Two Guys Created an Empire and Transformed Pop Culture* (New York: Random House, 2003).
[13] *The Book of id.* (Mesquite, Texas: id Software, 1996). This booklet was issued as part of the "id Anthology."
[14] *The Book of id.* (Mesquite, Texas: id Software, 1996). This booklet was issued as part of the "id Anthology."
[15] "id Software Background." URL: http://www.idsoftware.com/business/home/history/

[16] It should be noted here that the original version of *DOOM* was not released as open-source software, and in fact id's initial stance toward editing of the game code was not as encouraging as it has often been depicted. However, id did issue a "Data Utility License" that allowed modification of the game software under strictly defined conditions. With the release of *DOOM II* in 1994, id (particularly, John Romero) released more information about the structure of the game software. Carmack eventually did release the *DOOM* source code as a Christmas present to the player community in December 1997.
As Romero pointed out in an e-mail dated 11 Dec. 1997, "... Regarding the reason why *DOOM* is an "open" game: the reason is because of Wolf3D -- people figured out how to make maps for it without our help, plus change all the graphics, etc. and we were so impressed that we knew that *DOOM* just *had* to be modifyable. That is the real reason. We never wanted to sue anyone -- just protect ourselves a little." See "*DOOM* Editing History," a collection of e-mail documentation on this topic. URL: http://www.johnromero.com/lee_killough/history/edhist.shtml.

[17] Randall Packer, "Net Art as Theater of the senses: A HyperTour of Jodi and Grammatron," in: *Beyond Interface*. URL: http://www.archimuse.com/mw98/beyond_interface/bi_frpacker.html. This is the introduction to Mark Amerika's Grammatron, a work of hypertext web art.

[18] A sense of Carmack's position with regard to an *auteur* notion of game design can be gleaned from one of his posts to slashdot, which elaborated upon comments he made about John Romero's post-id company, Ion Storm (and its motto, "Design is Law"). According to Carmack,

"There is not a hell of a lot of difference between what the best designer in the world produces, and what a quite a few reasonably clued in players would produce at this point. This is the 'abstract creativity' aspect. This part just isn't all that valuable. Not worthless, but it isn't the thing to wrap a company around.

The real value in design is the give and take during implementation and testing. It isn't the couple dozen decisions made at the start, it is the thousands of little decisions made as the product is being brought to life, and constantly modified as things evolve around it. If you took two game designs, one good and one bad, and gave them to two development teams, one good and one bad, the good dev team could make a good, fun product out of a bad design, but the bad dev team could ruin the most clever design. The focus should be on the development process, not the (initial) design."

John Carmack, "Re: Definitions of terms," (2 Jan. 2002). Discussion post to Slashdot. URL: http://slashdot.org/comments.pl?sid=25551&cid=2775698

[19] The term is found in Lev Manovich, "Navigable Space" (1998). URL: http://jupiter.ucsd.edu/~manovich/docs/navigable_space.doc

[20] Prominent recent examples of massively modified games include *Dungeon Siege, Neverwinter Nights,* and *Operation Flashpoint.*

[21] Manovich, "Navigable Space."

[22] On the Spacewar! Olympics, see: Steward Brand, "SPACEWAR: Fanatic Life and Symbolic Death Among the Computer Bums," *Rolling Stone* (7 December 1972).

[23] "*DOOM* Press Release" (1993). Preserved in "Lee Killough's Legendary *DOOM* Archive" by John Romero as part of the "Planet Rome.ro" website. URL: http://www.rome.ro/lee_killough/history/*DOOM*pr3.shtml. According to *The Book of id,* the release was issued of Jan. 1993, about a month before the alpha release of *DOOM.*

[24] 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* deathmatch) is now seen as evidence that they were not as active in the *DOOM* scene as word-of-mouth would have it. Thus the importance of spectatorship and community, as well as skill, in achieving the status of a "God-like" player

[25] From "Welcome to the *DOOM* Honorific Titles!," *DOOM* Honorofic Titles website. URL: http://www-lce.eng.cam.ac.uk/~fms27/dht/dht5/#dht5.

[26] Technically, demo movie files were not screenshots, but scripts that could reproduce the movements and actions of players in the game, thus re-generating the images. Later, the ability to edit demo movies by moving the camera, "recamming," exploited the main advantage of this format, that the data could be used by the game

engine to generate other views. The disadvantages of the method usually had to do with moving the camera to locations the player's character could not "see" in gameplay.

[27] "NoSkill Memorial Site, April 22, 1978-Dec. 26, 2001." URL: http://www.*DOOM*2.net/noskill/index.htm

[28] "By the way."

[29] "Re: *DOOM*: NeXTstep's Most Successful App," Posting by John Carmack to comp.sys.next.advocacy, 14 August 1994. Source: Google Groups.

[30] John Carmack, "*Quake* Editing Tools Information," 5 April 1996. E-mail to Bernd Kreimeier. URL: http://www.gamers.org/dEngine/*Quake*/*Quake*Ed/qedit_info.html.

[31] Douglas Thomas, *Hacker Culture.* (Minneapolis: Univ. of Minnesota Press, 2002): 90.

[32] Thomas, *Hacker Culture,* 47-52.

[33] Hugh Hancock, "Ranger Gone AWOL," machinima.com. URL: http://www.machinima.com/articles/Ranger_Gone_AWOL/index.shtml.

[34] Uwe Girlich, "The Unofficial DEM Format Description" Version 1.02 (30 July 1996): 3.2 and 3.4. URL: http://www.gamers.org/dEngine/*Quake*/Qdem/dem-1.0.2-3.html#ss3.2.

[35] Katie Salen, "*Quake*! *DOOM*! Sims! Transforming Play: Family Albums and Monster Movies," Walker Art Center, 19 October 2002. URL: http://www.walkerart.org/gallery9/qds/

[36] On transformative or emergent play, see: Katie Salen and Eric Zimmerman, "Games as Open Culture," *op. cit.*

[37] From the *Quake* done Quick website. URL: http://www.planet*Quake*.com/qdq/news.html.

[38] Anthony Bailey, "Zigzagging through a Strange Universe," Planet Quake (12 Oct. 1997). URL: http://www.planetquake.com/qdq/zigzag/.

[39] For details on the making of "*Quake* done Quick," see "QDQ_1949. '*Quake* done Quick' the movie, by Yonatan Donner, Matthias Belz, Nolan Pflug and Anthony Bailey, 10 June 1997.' URL: http://www.gamers.org/pub/idgames2/demos/j-q/qdq_1949.txt.

[40] "*Quake* done Quicker." Version History. URL: http://www.planet*Quake*.com/qdq/qdqr.html.

[41] Anthony Bailey, "Zigzagging through a Strange Universe." *Op. cit.*

[42] "QDQ_1949. '*Quake* done Quick' the movie, by Yonatan Donner, Matthias Belz, Nolan Pflug and Anthony Bailey, 10 June 1997.' URL: http://www.gamers.org/pub/idgames2/demos/j-q/qdq_1949.txt.

[43] Ibid.

[44] Rice made the award-winning *Quake* 2 machinima comedy, "Father Frags Best."

[45] "Showcase: Operation Bayshield," machinima.com. URL: http://www.machinima.com/articles/Featured_Film_Op_Bays. "Interview: Tom 'Paradox' Mustaine," machinima.com. URL: http://www.machinima.com/articles/interview_fakk_2

[46] Jay David Bolter and Richard Grusin, *Remediation: Understanding New Media* (Cambridge: MIT Press, 1999).

[47] As announced in Blue's News (13 April 1997) and the *Quake* Zone (April 1997). URLs: http://www.shugashack.com/archives/a18-a12.htm and http://dooley.ca/*Quake*Zone/apr_97.htm.

[48] It was technically possible for Clan Undead to use keyboard bindings via *Quake*C scripting, but I have not yet been able to determine if they used this technique.

[49] While this article was in progress, Fountainhead Entertainment did just that by releasing its Machinimation software for use with *Quake* III.

[50] "Sidrial: Not Just a Game Anymore." Interview with Katherine Kang. URL: http://www.gamers.com/s/feature/001205-sidrial/index.