



Game Engines and Game History

Henry Lowood
Stanford Libraries, Palo Alto

Introduction

I have a problem: it's called preaching to the choir. Nobody here lacks faith in the history of games, and I can only hope that my talk will not reduce your enthusiasm. That realization eliminates the need for a homily on the value of historical studies. Instead, I will talk about the possibilities and promises facing the history of games as we plunge forward. What does history offer game studies and what might a history of games give back? These questions guide my thoughts at the end of this conference, which has given so much food for these thoughts.

So, What is historiography?

For a historian like myself, the word historiography has a two-fold meaning. As the word suggests, it means writing about history and what it means to do history. Second, it encompasses the methods and materials of historical work. Historiography sometimes is used as another way of saying "historical literature," but let's not worry about bibliography today. While preparing for this talk, I checked the Oxford English Dictionary to find an example of usage, which led me to a *Wall Street Journal* book review from a few years ago. I quote: "The book is an example of *historiography*, the study of the principles and techniques of history—a discipline that is usually dryness itself." (Gordon, D10) Oh well.

It should come as no surprise to any of you that historians tell stories about the past. I mean, look at the word! HI-STORY. Historiography then might be described as writing about writing about history. Recent historiography has been mightily influenced by Hayden White, author of the much-discussed *Metahistory: The Historical Imagination in Nineteenth-Century Europe*, first published in 1973. White argued that history is less about a particular subject matter or source material than about how historians write about the past. The historian does not simply arrange events in correct chronological order. Such arrangements are merely chronicles. The work of the historian only begins there. Historians instead create narrative discourses out of sequential chronicles by making choices. White puts these choices into the categories of argument, ideology and emplotment. Without getting into the details of every option open to the historian, the desired result is sense-making through the structure of story elements, use of literary tropes and emphasis placed on particular ideas.

White says that history is writing a certain kind of way, not writing about a certain kind of thing or using evidence according to a certain kind of method. In his book *Figural Realism: Studies in the Mimesis Effect*, he writes about the "events, persons, structures and processes of the past" that "it is only insofar as they are past or are effectively so treated that such entities can be studied historically; but it is not their pastness that makes them historical. They become historical only in the extent to which they are represented as subjects of a specifically historical kind of writing." (White 2) The takeaway here is that it opens up the possibility that history can be interpreted as a form of literature, that writing history is, say, like writing a novel - this has been the most controversial implication of White's historiographical writing.

My purpose in bringing Hayden White to your attention is to introduce game studies to this "historical kind of writing." Again, this writing is neither characterized by the object of inquiry nor by source material such as archival records or oral histories. The historical kind of writing is a narrative *interpretation* of something that happened in the past. Let me sharpen the implications by paraphrasing a line written by the late, great John Hughes and

spoken by Steve Martin in *Planes, Trains and Automobiles*, "... you know, when you're telling these little stories? Here's a good idea - have a POINT. It makes it SO much more interesting for the reader!" Game history also needs to have a point, and this conference has given us a chance to consider what that point might be.

So I am going to suggest a few points that game studies can make. In order to spice things up, I will lean on my current historical project: The game engine. Like most historical studies, this one is not necessarily universal in its implications, but I believe it is significant. More important for us today, the history of the game engine provides a story through which we can explore the kinds of narratives that historical game studies might deliver. Let me begin by setting the stage.

The Game Engine

Before the 1993 release of its upcoming game *DOOM*, id Software issued a news release. It promised that *DOOM* would "push back the boundaries of what was thought possible" on computers. This press release is a remarkable document. It summarized stunning innovations in technology, gameplay, distribution, and content creation. It also introduced a term, the "*DOOM* engine." This term described the technology under the hood of id's latest game software. The news release promised a new kind of "open game" and sure enough, id's game engine technology became the motor of a new computer game industry.

The "Invention of the Game Engine" was only half the story. John Carmack, the lead programmer at id, did not just create a new kind of software, as if that were not enough. He also conceived and executed a new way of organizing the components of computer games by separating execution of core functionality by the game engine from the creative assets that filled the play space and content of a specific game title. Jason Gregory in his book on game engines writes, "*DOOM* was architected with a relatively well-defined separation between its core software components (such as the three-dimensional

graphics rendering system) and the arts assets, game worlds, and rules of play that comprised the player's gaming experience." (Gregory 11)

Before I circle back to historiography, let's look more closely at the chain of events that produced the game engine and the decision to package assets separately.



Google's Ngram viewer allows us to analyze historical usage of the term “game engine” in texts available in the Google Books database. This analysis confirms id's bravado. The term first appeared in print during the year after *DOOM*'s launch. It occurs frequently during 1994 in fact. We find it in André LaMothe's *Teach Yourself Game Programming in 21 Days* or *Tricks of the Game-Programming Gurus* by LaMothe and John Ratcliff, articles in *PC Magazine* and the inaugural issue of *Game Developer*, and -- no surprise -- *The Official DOOM Survivor's Strategies and Secrets*, by Jonathan Mendoza. Google Books identifies citations before 1994, but these have turned out to be unrelated to game development or simply false hits. Fun fact: I bet you didn't know that the term “game engine” first appeared in Richard Burn's *The Justice of the Peace and the Parish Officer* in 1836! Well, it didn't really. The erroneous result was caused by the appearance of the word “engine” in an annotation in the right margin one line before the word “game” was printed on the left margin of the text itself. A more serious candidate for earliest published

use of the term is Douglas A. Young's *Object-Oriented Programming with C++*, published in 1992, but the Engine it describes is an object in a demonstration program of a Tic-tac-toe game. It is the prime mover in the game and controls the computer's intelligence, but the term is specific to this one program, and does not refer to a class of software. Interesting perhaps, but not relevant. Google's tool also suggests that the new term "game engine" kept pace with the increasingly prevalent "game software" while "game program" declined as a description of the code underlying computer games. These analytics suggest that "game engine" was a neologism of the early 1990s and support the conclusion that the invention of this game technology was a discrete historical event of the early 1990s.

Where in fact did the term "game engine" come from? The answer to this question begins with the state of PC gaming circa 1990 and its invigoration by id Software. It is fair to say that as the 1990s were beginning the PC was not where the action was. Videogame consoles dominated, while the popular 8-bit and 16-bit home computers of the 1980s were on the last legs of a phenomenal run. The strengths of the PC as a platform for game design had not yet proven their worth. By the time id released *DOOM* on one of the University of Wisconsin's FTP servers in late 1993, the decade of the PC game was about to begin. *DOOM* was the technological tour-de-force that heralded a "technical revolution" in the words of id's news release, a preview issued nearly a year before the game itself. *DOOM* showcased novel game technology and design: a superior graphics engine that took advantage of 256-color VGA graphics, peer-to-peer networking for multiplayer gaming, and the mode of competitive play that id's John Romero named "death match." It established the first-person shooter and the PC became its cutting-edge platform, even though *DOOM* had been developed on NeXT machines and cross-compiled for DOS execution. Last but not least, *DOOM* introduced Carmack's separation of the game engine from "assets" accessible to players and thereby revealed a new paradigm for game design on the PC platform.

The release of *DOOM* was a significant moment for the chronicle of events, whether we are focused on game technology or the history of digital

games more generally. Before we can build a historical narrative that takes this event into account, we obviously have to know more about the historical contexts for it. Let's begin with the dramatis personae. Many of you know id's story, so I will focus on elements related to the history of the game engine and its impact. In 1990, Carmack and Romero were the key figures in a team at Softdisk tasked with producing content for a bimonthly game disk magazine called *Gamer's Edge*. They had come to the realization that advanced machines running DOS were the future and they would need to “come up with new game ideas that ... suit the hardware.” (Romero “Oral History”¹) Following the release of MS-DOS 3.3 in 1987 and the maturing of the hardware architecture based on Intel's x86 microprocessor family, the PC was poised to become an interesting platform for next-generation computer games.

Romero and Carmack cut their teeth as young programmers on the Apple II. They had perfected their coding skills by learning and working alone. As they thought about future projects at SoftDisk, they also realized that they would need to make software as a team. The plan was to divide and conquer. Carmack would focus on graphics and architecture, Romero on tools and design. Their first project was Slordax, a Xevious-like vertical scroller. Over the course of his career Carmack has displayed a knack for figuring out fundamental programming innovations while working on specific game projects. As Slordax was taking shape, he easily showed how to produce smooth vertical scrolling on the PC, for example. Romero was modestly impressed, but he “wasn't blown away yet.” He challenged Carmack to solve a more difficult problem: NES-like *horizontal* scrolling.

Some of you know the story, I'm sure. Carmack answered the challenge, coding furiously while Romero egged him on. Then one morning Romero found a floppy diskette on his desk. With help from Tom Hall, Carmack had passed the previous night creating a frame-for-frame homage to *Super Mario Brothers* that ran on the PC with smooth horizontal scrolling. They had used

¹ Quotations below from unpublished oral history interviews with John Romero were conducted by the author at the Computer History Museum. Quotations without specific citations are from these interviews. Publication is forthcoming.

some graphics created for Romero's next *Dangerous Dave* game to create the demo, so they called it "Dangerous Dave in Copyright Infringement," acknowledging that it was a blatant copy of Nintendo's game. *Now* Romero was blown away. "I was like, Totally did it. And I guess the extra great thing is that he used Mario as the example, which is what we were trying to do, right, to make a Mario game on the PC."

The story I just recounted both connects and separates Nintendo and PC games as creative spaces. This theme is worth pursuing. For now, let's stick with the development of id's game technology. Horizontal scrolling was the team's ticket out of Softdisk. They realized as Romero recalled in his oral history that, "We can totally make some unbelievable games with this stuff. We need to get out of here." The separation from SoftDisk turned out to be gradual. Their new company, id Software, continued to produce games for their old company. While this was happening Romero came into contact with Scott Miller of Apogee Software, a successful shareware publisher. After showing Miller the *Dangerous Dave* demo, id began to produce games for him, too. These overlapping commitments produced a brutal production schedule, even without taking into account independent projects and technology development. Id met this schedule, in part, by producing games in series.

The first series of *Commander Keen* games, "Invasion of the Vorticons" was published between late 1990 and the middle of 1991. It featured Carmack's smooth horizontal scrolling. He used other projects to make progress with 3D rendering, which seemed like a promising technology for a second *Keen* trilogy. Carmack and Romero began to call the shared codebase for these three games the "Keen engine." The engine then was a single piece of software that produced common functionality for multiple games. The idea of licensing such an engine as a standalone product to other companies emerged quickly. Id briefly tested the idea by offering a "summer seminar" in 1991 to potential customers for the *Keen* engine. They demonstrated the design of a Pac Man-like game during the workshop and waited for orders to pour in. They received one ... from Apogee. Romero recalls that, "so they get the engine, which means they get all the source code to use it." Apogee made one game with id's

engine, *Biomenace*, then made their own engine after gaining access to id's code. "And then," as Romero put it, "they didn't license any more tech from us."

While the trial balloon of the licensing concept was a failure, the game engine stuck as a way of designating a reusable platform for efficiently developing several games. Romero later recalled that, "I don't remember, at that point, hearing of an engine, like you know, Ultima's engine, because I guess a lot of games were written from scratch." In other words, game programs had been put together for one game at a time. But why call this piece of game software a game *engine*? Carmack and Romero were both automobile enthusiasts and, as Romero explained, the engine "is the heart of the car, this is the heart of the game; it's the thing that powers it ... and it kinds of feels like it's the engine and all the art and stuff is the body of the car." We can now make a more precise entry in our chronicle: Id Software invented the game engine around 1991 and revealed the concept no later than the *DOOM* press release in early 1993.

So that was a short excerpt from the Chronicles of the Game Engine. We know roughly where our time-line begins and can put a few events in order, such as the creation of id, the invention context, and the enunciation of the Game Engine as a game technology.

DOOM's game engine is a significant event in the history of game software. Jason Gregory recalls in his introduction to *Game Engine Architecture* that when he bought his first system, the Mattel Intellivision, in 1979, "the term 'game engine' did not exist." Of course, we knew this already. He also observes in the same sentence that games of that era were "considered by adults to be nothing more than toys" and these games were created individually for specific platforms. Not so today. Gregory marvels that "games are a multi-billion dollar mainstream industry rivaling Hollywood in size and popularity. And the software that drives these now-ubiquitous three-dimensional worlds – *game engines* like id Software's *Quake* and *DOOM* engines, Epic Games *Unreal Engine 3* and Valve's *Source* engine – have

become fully featured reusable software development kits that can be licensed and used to build almost any game imaginable." (Gregory 3) In other words, the development of engine technology traces the growth and maturation of the game industry.

I have encountered the game engine in various historical projects. In one of them, I came across a John Carmack interview from several years ago. He remarked that *DOOM* was a "really significant inflection point for things, because all of a sudden the world was vivid enough that normal people could look at a game and understand what computer gamers were excited about." (Carmack, "DOOM 3: The Legacy") Contrast this remark to Gregory's observation about adults having no clue what was going on with digital games during the early home console years. *DOOM* and the game engine technology that powered it marked the beginning of the modern computer game, not only as a technical achievement, but as the springboard for a whole host of changes in perception and play: along with games like this we got networked player communities, modifiable content, fascination with the sights and sounds of games, and concerns about hyper-realistic depictions of violence and gore.

Most of us would agree with Carmack about the inflection point in computer gaming ca. 1993 and its after-effects. A specific example close to my heart is the history of Machinima, and this topic represents another of my encounters with the game engine. Before there was Machinima, there were *Quake* movies, which Carmack's separation of the game engine from assets made possible. Demo files were a particular kind of asset file in both *DOOM* and *Quake*. A few players figured out how to change these files and produce player-created movies that the game engine could then play back. Id had not anticipated movie-making, but enabled it as an affordance of their game technology.

Recently, thinking about how game engines interact with assets as input has even informed my work on game software preservation in the second Preserving Virtual Worlds project. The separation of game engine from assets in *DOOM* suggested a possible solution to the problem of auditing software in

digital repositories. It turns out that the version-specific ability of a game engine to play back demo files constitutes what media preservationists call a significant property of computer games. Put another way, we can check the integrity of game software stored in a digital repository by seeing if it can run a historical data file such as a DOOM demo. The game asset provides a key for verifying the game software, and the game engine pays back the service by playing back historical documentation such as replays. Our case study for this work was DOOM.

These touch-points lead me back to DOOM as a historical moment. Gregory's memory of videogames before the modern game industry and technology is probably a fairly typical one. So let's turn now from the chronicle to the historical narrative. We have seen that the game engine concept was in place in 1991, two years *before* DOOM was released. The Keen Engine served efficiency of serial game development and raised the possibility of licensing to external parties in order to create new games. During the two years between 1991 and 1993, Carmack worked feverishly on new technology and games and turned his attention especially to the problem of 3-D rendering. DOOM showed what he had accomplished during that period.

Lev Manovich has described the impact of *DOOM* as nothing less than creating “a new cultural economy” for software production. He had in mind the full implications of its software model of separated engine and assets. Manovich described this new economy as, “Producers define the basic structure of an object and release a few examples, as well as tools to allow consumers to build their own versions, to be shared with other consumers.” (Manovich 245) Carmack and Romero opened up access to their games in a fashion that might be construed in other media as giving up creative control. And yet, id's move was not a concession; they embraced its implications as the company focused increasingly on technology as a foundation for game development. They encouraged the player community and worked with third-party developers who modified their games or made new ones on top of id's engine. Carmack's support for an open software model can be explained in part by his background as a teenage hacker. Now he had created a robust model of content creation

that would allow players to do what he had wanted to do: Change games and share the changes with other players. Carmack's attention shifted as a result to improving the technology, rather than working on game design.

Users – players that is – played a significant role in shifting id's focus to the game engine as a content creation platform. When id released *Wolfenstein 3-D* in 1992, the efforts of dedicated players to hack the game and insert characters like Barney the Dinosaur and Beavis & Butthead made an impression on Carmack and Romero. Michael Adcock's "Barneystein 3D" patch and others like it documented the eagerness of players to change content, even though the game did not offer an easy way to do this. Romero has recalled in his oral history that *Wolfenstein 3D* demonstrated that players wanted "to modify our game really bad." He and Carmack concluded about their next game that, "We should make this game totally open, you know, for people to make it really easy to modify because that would be cool." Carmack's solution then was a response to a perceived demand. Assets such as maps, textures, and demo movies could now be altered by players without having to hack the engine. Stability of the engine was important for distribution and sharing of new content. Moreover, access to assets encouraged the development of software tools to make new content, which then generated more new modifications, maps and design ideas, and so on. Id's corporate history boasts to this day that after *DOOM* was released "... The mod community took off, giving the game seemingly eternal life on the Internet." (Id Software, "Id Software Backgrounder")

Manovich points to the implications of the changes introduced in *DOOM* in terms of support for content modifications and re-use of the engine, but he does not say very much about the motivations. David Kushner, who wrote a history of id software, says that Carmack's separation of engine and assets resulted in a "radical idea not only for games, but really for any type of media ... It was an ideological gesture that empowered players and, in turn, loosened the grip of game makers." (Kushner 71-72) At the same time, as Carmack and Romero had predicted, it was also good business. Eric Raymond took up this theme in *The Magic Cauldron* by bolstering his case for the business value of

open source software by analyzing id's decision to release the *DOOM* source code. The media artist and museum curator Randall Packer also noticed that a cultural shift had occurred only a few years after the release of *DOOM* and *Quake*. He observed that games had become the exception among interactive arts and entertainment media because game developers did not view the "letting go of authorial control" as a problem. He meant id, of course.

My narrative suggests that the logic of game engines and open design meant that id would become a different kind of game company. The PC, as a relatively open and capable platform during the 1990s, was also conducive to this logic. In line with Carmack's focus, id's game was now technology. A few years after *DOOM* he reflected that technology created the company's value and that there was not much added by game design over what "a few reasonably clued-in players would produce at this point." (Carmack, "re: Definitions of terms") Id's technology was expressed primarily through the game engine while the provisions for modifying assets opened up possibilities for player-generated content. That was the formula. This combination of strategic innovations was strengthened and emphasized in id's next blockbuster: *Quake*, released in 1996. It proved to be an especially fertile environment for player creativity. Expected areas of engagement such as mods were matched by unexpected activities such as *Quake* movies. During the five years from *Keen* to *Quake*, id Software had worked out a game engine concept capable of prodding the rapid pace of innovation in PC gaming during the 1990s.

Narrative

It seems that Jason Gregory's casual observation about the connection between game engines and the history of game development can be expanded into a meaningful historical narrative. As we learned from Hayden White, historical writing incorporates arguments, ideologies and emplotments to build such a narrative. Assembling the story elements I have discussed today provides material for building a history of the game engine on an argument that is familiar in technology studies: It is called technological determinism. A historical narrative does more than answer questions, it compels us to ask more

of them. So here is one: Does the history of the Game Engine support the case for technological determinism? I will conclude this brief history of the game engine with a few thoughts in response to this question.

First, let's refine the determinist argument a bit. In *The Nature of Technology*, Brian Arthur argues that modern technologies are inherently modular. This modularity is an important theme for his theory of technological change. Arthur argues that, "a novel technology emerges always from a cumulation of previous components and functionalities already in place." (Arthur 124) This cumulation is not achieved by replacing one technology with another, but by reorganizing and improving previous technologies. Usually, this process is driven by a novel need or phenomenon. Arthur suggests that this kind of change depends on the modular nature of contemporary technologies, such as the systems and components of a computer, automobile or jetliner. He observes that, "We are shifting from technologies that produced fixed physical outputs to technologies whose main character is that they can be combined and configured endlessly for fresh purposes." Arthur adds elsewhere that "the modules of technology over time become standardized units." (Arthur 25, 38) This historical process is evolutionary rather than revolutionary in character because coherent modules endure as components of other technologies in which they are embedded.

Arthur's way of thinking about technology resonates with the game engine concept, which relies on modularity for a "swift reconfiguration" of game software to "suit different purposes." (Arthur 36) The game engine is modular not just in the general way proposed by Arthur. Modularity was an aspect of the architecture of computer games such as *DOOM* and *Quake*. Like a jet engine, the game engine is a component of a finished product, the computer game, which is completed by adding other assets (modules, if you will) created by other digital technologies. We should perhaps not be at all surprised that a quintessentially "modern" technology was the foundation for the modern game industry, Q.E.D. Recall Carmack's opinion that it derived its value as a company from *technology*, not game design. A historical narrative consistent with these points would argue that the success of *DOOM* and *Quake*

encouraged developers and players to exploit id's new game technology and modular content model. Technological change fueled the dynamic growth of the PC game industry through the 1990s. Ripples became waves as innovations in "middle" technologies such as graphics and sound cards in hardware or AI and physics engines in software kept pace with the expanding game industry, and the intense development of these component technologies sparked secondary innovations as well. For example, it has been argued that the game industry created demand for 3-d graphics hardware, which in turn provided a favorable environment for improvements in computer-aided design.

So, the determinist shoe fits, right?

Well, it was fun to try it on, but before we buy that shoe, let's keep shopping. A reading of communications historian Susan Douglas provides different plot ideas. In her Da Vinci Medal address in 2009, Douglas relied on a "poetic structure" or trope specifically identified in Hayden White's historiographic writing, although she did not refer to him. This structure is irony. White presented historical tropes as a way of categorizing how historians relate language and thought, that is, how they use narrative to relate ideas about history. Influenced by literary scholars such as Northrop Frye, White argued that historical narratives are characterized by a "tropological structure." This means that a particular trope prevails in every piece of historical writing. Tropes also correspond to emplotments. White identified four tropological structures: metaphor, metonymy, synecdoche and irony, which are tied respectively to romance, tragedy, comedy and satire. According to White, this breakdown provides us "with a much more refined classification of the kinds of historical discourse than that based on the conventional distinction between linear and cyclical representations of historical processes." (White 11) The payoff for us is recognizing that there are deep formal similarities between literature and historical writing.

So what? Shouldn't we only be concerned with history's truth value? Does it matter whether a historian is giving us a tragedy or a comedy? While there has been a huge amount of debate about what some have perceived as

White's reduction of history to literary fiction, I would characterize White's objective as consciousness-raising. The historian is as dependent as the writer of fiction on language and the structure of narrative form. White says about irony, for example that, "A mode of representation such as irony is a content of the discourse in which it is used, not merely a form - as anyone who has had ironic remarks directed at them will know all too well. When I speak to or about someone or something in an ironic mode, I am doing more than clothing my observations in a witty style. I am saying about them something more and other than I seem to be asserting on the literal level of my speech. So it is with historical discourse cast in a predominantly ironic mode..." (White 12) Thus irony would be used by a historian who stresses a contrast or disjuncture in historical elements (people, events, movements, etc.) that were thought to be similar or closely affiliated.

Douglas tells us about the "irony of technology." Like many historians weaned on social construction of technology, Douglas rejected long ago the technological determinism a graduate student trained after the 1970s would recognize, say, in the writings of Marshall McLuhan. When she later rethought the relationship between technology and social context, she found "a new attention to what are now called technological affordances" that tell us what "certain technologies privilege and permit that others don't." We might say that as often happens as polarizing debates wind down, the middle position began to look attractive. Douglas concluded that a reasonable take on communications media would be that technologies define a suite of affordances, yet these affordances do not determine actual historical use. The process of "producing often ironic and unintended consequences" is indeed as the social constructionists would have it, a process of negotiation. While the affordances are not to be denied, their impact is critically shaped by business imperatives, use, legal constraints and other messy historical complications. (Douglas 300)

Douglas' contemplation of technology as ironic can be applied to the game engine. The id developers originally considered the game engine as a way to solve the problem of producing computer games more efficiently. Build one

engine to produce a series of three *Commander Keen* games. This is efficiency on the order of Peter Jackson's filming of three *Lord of the Rings* films more or less at the same time. Id's 1991 summer seminar revealed a more general idea, but still an idea associated with efficiency. The licensed game engine could become a platform upon which diverse games would be constructed. In either case the engine developer provides core functionality, that is, a set of affordances, and game developers decide how to deliver new game mechanics and assets using the engine. Games built this way are the product of a modular design concept built around the game engine, but the use of this technology is shaped by constraints ranging from business practices and commitments to user needs and objectives. The history of television is an often discussed problem that exhibits technological irony. McLuhan, for example, had high hopes for TV and the "global embrace" that electronic media technologies would deliver. Instead, we have reality series and the fragmentation of news and ideas, as a variety of non-technical factors have emphasized different affordances than those seen by McLuhan. In the case of the game engine, we might ask if expectations about open software and player-generated content made possible by game technology have been realized as they were imagined. Ideas about game production, design, business models and player creativity have played upon the expectations made possible by the affordances of the game engine; these ideas complicate the deterministic model while raising new historical questions.

More historical work is needed to produce a better understanding not only of the rise of game engine technology, but also of the business and creative decisions that conditioned the use of technology to make new games. I wish to suggest that a binary question -- technologically determined or not? -- often blurs as ironic and other narrative structures add nuance and messiness to our histories.

Conclusion: A future for game history

Musing about the history of games generally and the history of the game engine in particular has led me to call upon my home field, the history of

technology, several times already. In concluding, I am going to this well one last time.

In an essay called “The History of Computing in the History of Technology,” published in 1988, Michael Mahoney reflected on the former subject’s puzzling lack of impact on the latter until then. The importance of computing and informatics was widely recognized and there was a natural relationship between these subjects. So why had the history of computing lagged as a sub-discipline of the history of technology? His analysis of writing in the history of computing showed that the field had been dominated by three kinds of contributions. See if you recognize them in game history. The first was “insider” history. “While it is first-hand and expert, it is also guided by the current state of knowledge and bound by the professional culture,” Mahoney argued. (114) The second group of contributors consisted of journalists, whose work he described as long on immediacy but short on perspective. The third segment of the literature consisted of “social impact statements” and related writings, composed in the service of futurism or policy or studies of social impact; he considered these studies to be more polemical than historical. Finally, Mahoney identified “a small body of professionally historical work.” (115)

Mahoney next considered what he thought of as the big questions in the history of technology. He suggested that as the history of computing matured as a discipline, it would be able to contribute answers to these questions, just as these questions would guide and inform work in history of computing. Here are a few examples of the kinds of questions he had in mind: “How has the relationship between science and technology changed and developed over time and place?” “Is technology the creator of demand or a response to it?” or “How do new technologies establish themselves in society, and how does society adapt to them?” (Mahoney 115-116) Big questions, in other words.

So where am I going with this? I would like to suggest that the history of games is in a similar situation to the history of computing, only twenty-five years later and with a less clear notion of its natural parent discipline. I draw

two points from this comparison, and I would like to add one more from the example of the Game Engine and its history. First, the history of games is in its infancy. As with the history of computing in 1988, most of game history has been written to answer questions that arise from first-hand familiarity, journalism or implications for policy or business affairs. None of this is unwelcome, of course. However, let me remind you that Mahoney said about the history of computing written by computer scientists. He wrote that, “while it is first-hand and expert, it is also guided by the current state of knowledge and bound by the professional culture.” (114) He meant that decisions or results that such an author might take as a given, an outside viewer such as a historian might consider as a choice.

Second, the history of games will develop in more interesting ways if it finds connections to big questions. From my own selfish perspective, we could certainly do worse than reflect on some of the questions that have shaped work in the history of technology. Paraphrasing Mahoney, for our purposes we could ask, “How has game design evolved, both as an intellectual activity and as a social role?” or “Are games following a society’s momentum or do they redirect it by external impulse?” “What are the patterns by which games are transferred from one culture to another?” And so on. Only by asking such questions will the history of games find connections to other areas of historical research and cultural studies, which in turn will invigorate our own work with fresh perspectives and interpretive frameworks.

Finally, my last point derives from the history of the game engine. Consider the contrasting view of this topic that we acquire by backing away from technological determinism and considering Douglas’ notion of technological irony. This contrast encourages us to look more closely at the messy interplay of intentions, users and the marketplace. My point is not that either of these points-of-view is the one and true answer, but rather that we are still so far from creating a critical mass of divergent ideas and perspectives, our last three days at this conference notwithstanding. One last quotation from Mahoney about the history of computing: “What is truly revolutionary about the computer will become clear only when computing acquires a proper history,

one that ties it to other technologies and thus uncovers the precedents that make its innovations significant.” (123) Asking big questions of the history of games and answering those questions in creative and diverse ways strikes me as the right strategy for reaching a similar goal: Not creating a separate enterprise called the history of games so much as finding connections with related fields surrounding this subject, from history of technology to intellectual history. In other words, it will not be enough to create more data points about the history of games. We will also need good questions and big ideas to help us make sense of that history and, ultimately, to have a point.

References

- Arthur, Brian. *The Nature of Technology: What It Is and How It Evolves*. New York: Free Press, 2009.
- Carmack, John. “DOOM 3: The Legacy.” Transcript of video retrieved June 2004 from the New *DOOM* website, <http://www.newdoom.com/interviews.php?i=d3video>
- Carmack, John. “Re: Definitions of terms.” Discussion post to Slashdot, 2 Jan. 2002. Retrieved Jan. 2004, at: <http://slashdot.org/comments.pl?sid=25551&cid=2775698>
- Douglas, Susan J. “Some Thoughts on the Question ‘How Do New Things Happen?’” *Technology and Culture* 51 (April 2010): 293-304.
- Gordon, John Steele. Review of Seth Shulman, *The Telephone Gambit*. Wall Street Journal (16 January 2008): D10.
- Gregory, Jason. *Game Engine Architecture*. Wellesley, Mass.: A. K. Peters, 2009.
- Id Software. “Id Software Backgrounder.” Retrieved Feb. 2004 from id Software website, at: <http://www.idsoftware.com/business/home/history/>
- Kushner, David. *Masters of DOOM: How Two Guys Created an Empire and Transformed Pop Culture*. New York: Random House, 2003.

Mahoney, Michael. "The History of Computing in the History of Technology,"
IEEE Annals of the History of Computing 10:2 (1998): 113-25

Manovich, Lev. *The Language of New Media*. Cambridge: MIT Press, 2001.

Romero, John. *Oral History*. Interviews conducted and edited by Henry
Lowood. Publication forthcoming.

White, Hayden. *Figural Realism: Studies in the Mimesis Effect*. Baltimore:
Johns Hopkins Univ. Press, 2000.