Exploring the Technical History of Games Through Software and Visualization

Eric Kaltman
University of California, Santa Cruz
1156 High St.
Santa Cruz, CA, 95064
+01 510-501-6747
ekaltman@soe.ucsc.edu

ABSTRACT

The history of computer game implementation, driven by source code and development practice, remains largely hidden from view, with only the most dedicated and technically savvy game historians and scholars able to take part in its study. My research hopes to reveal and encourage access to technical history through the analysis of the scholarly practices used in exploring legacy game systems. I will turn to two primary case studies, MUD1 and DOOM, both of which have creator supported archival collections. In proceeding with a historical study of spatial representation in both games, I hope to embed a meta-analysis into my study that involves both defining specifications and toy systems to support scholarly research, and developing visualizations for scholarly presentation. The result is to, a) make a compelling argument about the technical systems behind historical games, b) create systems for better scholarly research of and communication about technical implementation, and c) promote the preservation and storage of historical game development documentation.

Categories and Subject Descriptors

K.8.0 [**Personal Computing**]: General – *games*

Keywords

Game studies, history, visualization, research methods.

1. INTRODUCTION

Understanding the technical implementation of computer games is difficult, requiring expertise in specific languages, architectures, and algorithms outside the normal training for humanist and historical game scholars. As a result, much of the history of computer games is the history of game objects as commercialized and finalized products. My research looks into new methods and approaches to both doing the technical work necessary to understand historical systems, and explaining those systems through more than just historical prose. By leveraging insights from educational visualization, and the massive effort already underway by fan communities to dissect and reverse engineer game software, I present a new approach that embeds software development and visualization into technical historical praxis and presentation. The goal is to show that technical history is an important part of the history of computer games, that it's study can be enhanced through better technical visualization and tools, and that access to development process, through archival collections and online resources, is enabling new avenues for historical games research.

Essentially, can a considered evaluation of the needs of techno-historical game studies scholarship, paired with a methodology and framework for exploration and visualization of technical systems, enable and support new types of historical game studies works?

2. AUDIENCE AND RELEVANCE

Technical implementation and system architecture is usually a highly guarded aspect of any commercial game development process. Lots of time and effort is spent developing complex technical systems that are, by design, supposed to disappear from view upon compilation and release. Much of the history of implementation is still undocumented and under researched, in part due to the hurdles involved in analyzing, accessing, and presenting technical designs. Only through doing the work of technical historical analysis can we encourage game developers to better organize and retain their development documentation. In revealing the technical lineage of game development we can educate future designers and developers in the rich history of the craft. Future scholars could then benefit from the availability of resources provided by historically conscious game creators, and create, through scholarly outputs, a further incentive for historical preservation of game documentation. Additionally, by providing an explanation of how technically minded scholars engage with resources, the work in this proposal can also guide future scholars towards more structured methods for dealing with technical systems and their description.

3. RELATED WORK

This proposal is situated firmly at the intersection of two groups of disciplines. On one side are game studies, software studies, and the history of science and technology, and on the other are educational visualization and the study of scientific and technical representation.

From game studies this proposal draws on the work of other technically focused scholars, like those in the Platform Studies series [1, 2], my own previous investigations of Civilization II's development [3], and seminal game design history work, like [4]. Each of these examples dives into either the technical considerations and affordances of game systems, or the complex, historical connections between different game design methods. The notion of different historical design threads coalescing into modern game design, that are drawn from diverse, and sometimes contradictory, design practices mirrors ideas from the history of science and technology regarding the social construction of artifacts and theories.

Software studies, in revealing the underlying assumptions and decisions embedded in software praxis, is also a ripe field on which to construct a technical historical methodology for games. Some works, most significantly [5], analyze technical systems to reveal the divergence between the perceptions of computational objects, and the underlying technological reality of their design and construction. However, few software studies works make any mention of the scholarly processes used in their investigations. Did they emulate the older systems? What were the challenges in dealing with and organizing legacy documentation and code? What tools or methods would have helped make that process easier? While contributing to the techno-humanist discourse about game systems, I also want this proposal to inform on *how* to do the scholarly work in addition to the what and why revealed by it.

My focus on describing development and scholarly processes is based, in a large part, on the work of archivist's dealing with the records of science and technology. In [6], the Joint Committee on the Archives of Science and Technology lays out guidelines for the preservation of the documentation of scientific research process in addition to published findings. Their methodology, essentially an archival strategy for scientific laboratory records, formed the basis for the SLAC National Accelerator Laboratory Archives, and have proved very amenable to game development documentation as well [7].

The other aspect of this proposal, the explanation of technical system design through novel visualizations, comes from the work of Bret Victor, Alan Kay, and other educational visualization pioneers. Victor's work on Explorable Explanations, and abstraction visualization present methods for explaining processes on different, simultaneous levels, from the underlying assumptions of the code, to user-simulation interaction [8, 9]. That work is based, in part, on the educational goals of Alan Kay, and the desire to introduce non-technical people to computational ideas and constructs. I want the presentation of the proposed work to enable this type of understanding for students of game and development history. Advanced scholars will also benefit from efforts to expand arguments beyond prose, and some efforts in interactive scholarship are already taking the lead [10, 11].

4. METHODOLOGY

The abstracted practice, as shown in Figure 1, involves the feedback cycle between research, analysis, scholarly presentation and publication. Research into technical documentation leads to the creation of software-based tools to help parse and understand the systems on which the documents are based. These tools then contribute to the formation of historical arguments about how the developers and designers implemented their systems. The tools, in a sense, are now part of the argument and make it possible. I'm interested in how software for emulation, code analysis and parsing, and information retrieval (citation, file search, etc.) can enable new perspectives on the historical lineage and influences present in game system implementation.

The next step is to embed some of this newfound technical understanding into my publications. Drawing inspiration from the visualization techniques referenced above [8, 9], I intend to create interactive representations of the sub-components, algorithms, and possibly other, technical implementation details necessary for understanding the systems I describe. This leads to a dual publication strategy through web-based interactive documents, and paper-based (and less visualized) publications.

At each stage of this process I plan on not only doing research into system architecture, design, and their connections with other technical historical practices, I also want to step back and take account of what types of tools and visualizations might help future students and scholars take stock of the complexity present in these systems, and the resources available to understand them.

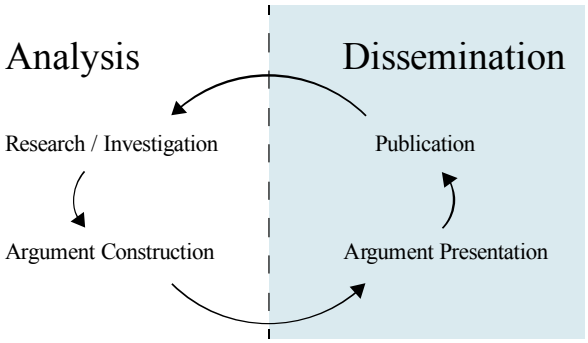

Figure 1. The research loop, ripe for meta-analysis and complimentary tools and visualization.

This meta-analysis is a main contribution of my work, and will inform the other outputs, namely, the historical argument about MUD1 and DOOM (outlined below), the tools developed to analyze them, and the visualizations designed to explain them.

As a base case for this work, I plan on examining Richard Bartle's MUD1 Collection at Stanford University Library, which includes physical development documentation as well as a digital archive of the game's source code, and the DOOM source code, with additionally documentation from John Romero's development archive. The plan is to examine the different strategies employed in creating topological (MUD1) and metric (DOOM) spatial representations and algorithms. The process will involve periods of material study – of code, documents, etc. – followed by reflection on the challenges of that study. This reflection involves development and specification of software for analysis of the source historical materials. Either through code analysis tools, or by leveraging mod community created resources that already enable documentary access. For instance, there are numerous applications for studying DOOM maps, creating mods, and analyzing DOOM data files. I want to look, in part, at what the community has created to see how it could be adopted or co-opted for scholarly research.

In the presentation of arguments and findings, I plan to explicate my historical argument with the aid of computational visualizations in an augmented, web-based presentation. For MUD1, this might involve allowing readers to interact with some subset of the game, while it visualizes their movement through a graph network of the game's locations. In DOOM, given the available knowledge about its maps and their underlying WAD data files, visualization would focus on how the game system interprets and renders its pseudo-three dimensional spaces. The visualizations would attempt to clear up understanding of technical details, applying interactive visualization practice to a historical argument about system structure and design.

5. CURRENT WORK AND SCHEDULE

This proposal is based in my experiences organizing game development documentation [7], and doing historical analysis of the spatial representation systems in Metroid (NES), and Civilization II (Windows 3.1 / 95) [3].

In trying to maintain and organize development documentation for archives in [7], we became aware of the need to provide consistent categorization and storage strategies. Finding relevant research material is incredibly difficult when wading through disorganized

collections of source files and assets. We wanted to raise awareness among developers contributing documentation and broaden expertise among archivists managing it. I plan to leverage my position as a research member of the Game Metadata and Citation Project (GAMECIP) on development preservation, game citation and description as a way to jumpstart the initial tools for exploring technical game systems and their documents. GAMECIP is an attempt to create a set of standard practices and initial tools for the citation, description, and discovery of game-based items and collections in institutions and archives.[1] One outcome is a more robust set of tools for citation of game states, components, and resources not covered in any current scholarly frameworks. I hope to use some of those tools to help reinforce and source my arguments about MUD1 and DOOM.

In studying the spatial systems of Metroid, and Civilization II, I waded into the waters of fan-created tools for game analysis and exploration. Players managed to reverse engineer large amounts of the source code for these games, and provided custom tools for map exploration and visualization. Players created tools to render Metroid's room generation algorithm (Figure 2), or to create and export maps for Civilization II. Generally, the creators left extensive notes on the systems they analyzed and deconstructed the technical processes underpinning the games. This seemed to be akin to the practices of technical software and game studies mentioned above, though there is still little use of these tools for direct scholarly analysis. This proposal is therefore, aimed at providing an overview of how to incorporate this type of fan created work into techno-historical study of games. Is it possible to better incorporate the modding community's work into scholarly practice, given that it is already a form of software study?

One last bit of work was conducted with undergraduates at UCSC [12], where we took [9]'s abstraction visualization technique and applied it to the Pong AI described in [1] (Figure 3), and the Pac-Man AI described in [13]. Our initial demos allowed players to adjust the constraints of the games' AI dynamically, showing players the effects of different variables on the overall system. An expansion of this work is the basis for this proposal's argument for interactive, historically motivated visualization of game technical systems.

The current research schedule is to spend the next six months working with the resources available through the MUD1 and DOOM archives, this includes developing small, custom software tools to analyze and understand the systems and code bases of both games. MUD1 is written in BCPL, so it will require more effort to parse, including collecting legacy resources on BCPL and working with community tools developed to support the language on modern systems. DOOM is an easier case study, and was chosen intentionally because I wanted to tease out the types of resources available for its study. Written in C, there are many versions of the game, and many nuances to discover over the course of its development history. I expect the research of DOOM to be intentionally overwhelming to, again, help develop strategies for dealing with large collections of historical technical information.

After an initial dive, I will surface with further insights into what could make the previous research more accessible to future researchers. I will also begin writing up my findings and framing an historical argument about spatial representation in both systems. This argument will be the test case for the next period (around 4 months) of work on developing custom visualizations of the systems' components, and potential investigation into how to reference the structures I describe in the interactive document. This process will then repeat on a shorter cycle, probably another 6 months, after which I'll have completed one full loop (and meta-analysis) of my technical historical practice. I expect to try and publish results at each step of the project, and to see if issues encountered later in the loop (i.e. presentation and visualization) could be further supported by work on the research side of the process.

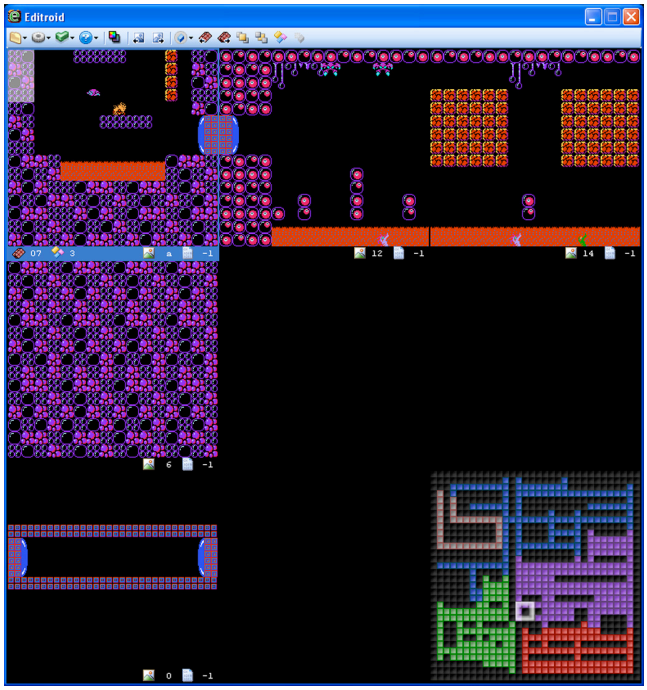

Figure 2. Editroid, a fan-created map editor and viewer for the Metroid (NES) ROM data.

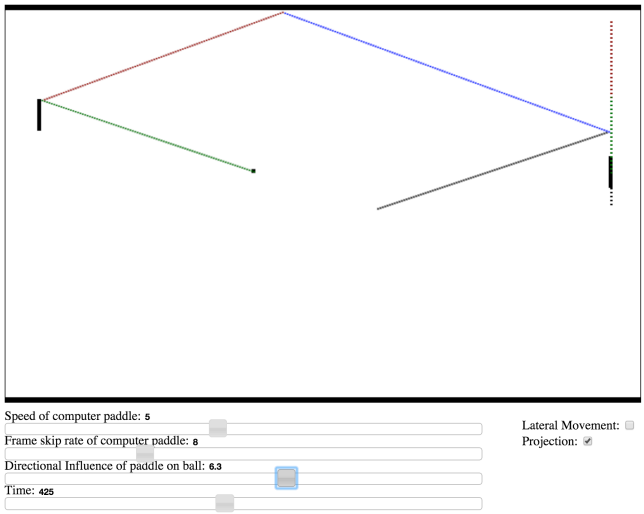

Figure 3. Pong AI Visualization, with dynamic control of system variables.

6. REFERENCES

[1] N. Montfort and I. Bogost, *Racing the beam the Atari video computer system*. Cambridge, Mass.: MIT Press, 2009.

[1] See GAMECIP project site, http://gamecip.soe.ucsc.edu

[2] J. Maher, *The future was here: the Commodore Amiga*. Cambridge, Mass: MIT Press, 2012.

[3] Eric Kaltman. "The Construction of Civilization," *History of Games International Conference Proceedings* in *Kinephanos*, 2014. [Online] Available: http://www.kinephanos.ca/2014/civilization [Accessed: 25-Feb-2015]

[4] J. Peterson, *Playing at the world: a history of simulating wars, people and fantastic adventures, from chess to role-playing games*. San Diego, CA: Unreason Press, 2012.

[5] N. Wardrip-Fruin, *Expressive processing*. Cambridge (Mass.): MIT Press, 2009.

[6] C. A. Elliott, Understanding progress as process: documentation of the history of post-war science and technology in the United States. Society of American Archivists, 1983.

[7] E. Kaltman, N. Wardrip-Fruin, H. Lowood, and C. Caldwell, "A Unified Approach to Preserving Cultural Software Objects and their Development Histories," 2014.

[8] B. Victor, "Ten Brighter Ideas? An Explorable Explanation," 2010. [Online]. Available: http://worrydream.com/TenBrighterIdeas/. [Accessed: 16-Mar-2014].

[9] B. Victor, "The Ladder of Abstraction." [Online]. Available: http://worrydream.com/#!2/LadderOfAbstraction. [Accessed: 03-Dec-2014].

[10] "About Scalar," 2014. [Online] Available: http://scalar.usc.edu/scalar. [Accessed: 25-Feb-2015]

[11] "Vectors: Journal of Culture and Technology in a Dynamic Vernacular," 2013 [Online] Available: http://vectors.usc.edu/journal/index.php?page=Introduction. [Accessed: 25-Feb-2015]

[12] R. Cori and E. Kaltman. "Extensible Pong," 2014. [Online] Available: http://extensiblegames.appspot.com/extensibleGames/default/pong. [Accessed: 25-Feb-2015]

[13] J. Pittman, "The Pac-Man Dossier," 2011. [Online]. Available: http://home.comcast.net/~jpittman2/pacman/pacmandossier.html. [Accessed: 04-Feb-2014].