Critical Essay

A Plea for the Preservation of Early BASIC Game Programs

Jim Gerrie, Cape Breton University

Abstract

Based on a consideration of the outlooks of the Social Construction of Technology (SCOT), Actor-Network Theory (ANT), as well as theories in the philosophy of technology, this critical essay is an argument for a greater recognition of the need to preserve significant early computer games written in BASIC and to provide improved public access to these programs. Although many of these programs have been preserved and presented to the public by hobbyists, there are significant gaps that have the potential to prevent recognition by future researchers of the pivotal role played by non-corporate actors in the early development of the video-game industry. Such informal efforts at curating these items so that they can be easily accessed are also limited by copyright laws. Effectively addressing these issues requires increased support from museums and scholarly institutions.

Keywords: Retrocomputing, Retrogaming, retroprogramming, Software Studies, Software Preservation.

Retrocomputing, Retrogaming and Retroprogramming

The case for a greater recognition of the need to preserve significant early computer games written in BASIC and to provide improved public access to these programs is based on my experience garnered from participation in the "retrocomputing" hobby. In particular, my creation of a website to allow non-specialists to run significant early BASIC game programs has been helpful in understanding the challenges of preservation (Gerrie, 2019). My activities have been focused on collecting early BASIC programs, including many popular "type-in"[1] games from the time period of the early 1970s to the mid-1980s. I have collected hundreds of these programs, many of which are unavailable in runnable form anywhere else online. Recently, I began to consider the possibilities for the formal curation of these materials.

My interest in creating a website was sparked in part by the 50th anniversary of the creation of BASIC in 2014, which prompted a number of authors to discuss their memories of typing in games from computer magazines in the early 1980s (McCracken, 2014). A common refrain of these retrospectives is that these experiences were foundational for later programming careers (Anderson, 2018; Bangeman, 2012; Target, 2018). The willingness of early computer users to type in programs was born of the hardware limitations of early machines. In a time before networked communication and a large-scale software industry, programmers had to rely on the medium of print. Therefore, in the earliest stage of the home computing revolution, roughly 1977-1984, users relied heavily on magazines and books to obtain game software (Anderson, 2018). The recent development of digital archives, such as the *Internet Archive* has also helped spur renewed interest in these kinds of programs. For example, the widespread availability of high-definition scans of computer magazines and books now allows fans to return to aborted attempts to enter programs from their youth or to revisit well recollected programs. My own experience has been typical.

Some Preliminary Challenges for BASIC Retroprogram Preservation

As I have converted found code from scans and archives, I have had to overcome many hurdles relevant to the preservation of early software. For example, one often finds references to programs for obscure computer systems for which no functioning emulator copies are available online, such as the case of the game *Coup D'Etat* for the

Sharp MZ-80 computer, published in the December 1982 issue Computing Today (Venor, 2014). However, sometimes a version might have been ported to a more popular system. But, as was the case with *Coup D'Etat*, these might contain typos and bugs, making them unplayable. Such errors were a product of the necessity to retype these programs to pass them between the largely incompatible file formats used by the different computer systems at the time. This process of retyping combined with the complexities of translating programs from different varieties of BASIC and machine specifications results in a high frequency of errors.

Even with code designed for specific machines, typing errors can be present. BASIC programs often contained a lot of data that could be hard to retype accurately. This problem was further exacerbated by the very limited editing capabilities built into most machines of the era. A common way to view a BASIC program would be to use the LIST command and then halt the listing as it quickly scrolled up the screen. One could use the LIST command with specific line numbers or ranges of numbers specified, but this required a good memory of the layout of the program. Such awkward editing capabilities made it difficult to get a good overview of source code unless one was able to afford a printer, which was a relative luxury for many hobbyists in the period. As a result, the quality of the source one finds preserved today varies greatly according to the typing and cross-platform programming skills of the person who entered it.

Also, the barrier to entry for BASIC programming was low. The language was built into most of the home computers of the time and they typically came with introductory manuals that included type-in programs. This allowed many people beyond professionally trained programmers to join in the computer revolution. It is not surprising, therefore, that there would be quality issues as a result of this immense broadening of access. As a result, unlike commercially produced games found in online archives, there are no guarantees that BASIC source code found today represents a functioning copy. In my experience many programs have errors, with a significant portion having errors that seriously degrade program operation. Such frustrations are likely part of the reason that many retrogamers today tend to avoid type-in games (Figueroa, 2014).[2] If it were not for the necessity to obtain software for my own obscure "orphaned hardware" (Wikipedia, 2021g) computer system, I doubt I would have developed a strong enough interest in these programs to have the patience to test them to a point that they are verified as functional.

Over the last decade of pursuing my hobby I have developed a library of representative BASIC programs from the classic age of type-in software. Most are games, but some are educational and scientific, such as a number of interesting astronomy applications and early attempts at AI, such as G. E. Fisher's 1974 limerick generating program *LIMRK*, Dieter Steinwender's didactic BASIC chess program published in *Computerschach und Spiele* in 1984 (Wall, 2017), and influential Australian programmer Tim Hartnell's Amanuensis (Australasian Digital Heritage Assoc., 2020; Wikipedia, 2021o). When my home internet provider stopped providing webspace I moved the program files to my institutional network storage space. This forced me to consider the complex ethical and legal issues regarding the sharing of copyrighted works (Aufderheide et al., 2018; 2019; Carey, 1972; Carey, 1972; Carey, 1972; U.S. Copyright Office, 2013; University of British Columbia, 2018; Wikipedia, 2021i).

The sharing of copyrighted source code, even if obsolete, is extremely ethically and legally problematic. Many sites, such as the *Internet Archive*, have taken a stance that it is better to beg forgiveness than ask permission when it comes to preserving digital materials. The archive shares old software under the guise of making it available for research purposes, including by using online emulation. They have not yet faced major legal challenge regarding their preservation of classic software, but the practice of posting copyrighted material until someone complains still raises significant ethical concerns (Orland, 2018). The result is that although many classic programs in the form of scans of published materials exist, the repositories holding them exist in a legal and ethical gray zone (Handelman, 2014).

Early Discussions of the Need for Retrogame Preservation

Although there are a growing number of computer museums seeking to preserve the hardware of the early age of personal computing, the preservations and curation of BASIC software poses unique challenges. A piece of hardware can be owned outright, and put on display for public viewing. But copyrighted software can only be owned in the physical form of the medium on which it was distributed. Although easily reproducible, if copyrighted, it should not be shared in a way equivalent to republishing without the copyright holder's permission, either in the form of releasing their work into public domain or some form of license. But there is a genuine interest amongst hobbyists and researchers to see items of the early personal computing revolution, including software, preserved for posterity. As John Aycock observes in his book *Retrogame Archeology*, "It may be odd to consider that code written in the span of a lifetime is now archival document that scholars want to study, but it is" (Aycock, 2016, p. 212). But as he also notes, "One worrisome thing I have found while doing research for this book is how little source code is available for study" (Aycock, 2016, p. 209). Researchers exploring old computer games have come to rely on the plethora of private collections of copyrighted material that are available on the Net.

Aycock, like many hobbyists interested in retrogames, mostly concerns himself with the examination of commercial machine language games. The most popular games of the early home electronics age were for dedicated gaming consoles. Regarding such games Aycock notes "with few exceptions, retrogames had many copies (legitimate and otherwise) widely available; source code had few copies and wasn't distributed at all" (Aycock, 2016, p. 212). This is not the case with the BASIC games developed on the early home computer systems, which typically only contained interpreted versions of the language that require the source code itself to run the program. The problems regarding preservation for early BASIC games are different from those of commercial games. There is a general bias of many retrogamers towards games written in languages other than BASIC.[3] Not only are BASIC games frequently buggy, as noted above, machine language commercial games provide a faster and more exciting experience. James Hague, compiler of the *Giant List of Classic Game Programmers*, puts it plainly: "There's usually a quality gap between 8-bit BASIC games and those written in other languages" (Hague, 2019). Therefore, in my experience BASIC games, beyond a few nostalgic favorites such as the educational game *Oregon Trail*, are generally looked down on as primitive by hobbyists. As a result, finding BASIC sources, even in cases where such games have had a critical influence on the development of major game genres can be difficult.

It is perhaps not surprising then that the term "BASIC" does not even appear in the index of Aycock's book although a few examples of BASIC games are discussed. Nor does it appear in Raiford Guins' *Game After: A Cultural Study of Video Game*

Afterlife (2014). Both of these works are concerned with considering the challenge of preserving early electronic game software and with chronicling emerging efforts in this field. According to Guins:

> A recent shift from what has been designated as the "chronicle era" of game history to what might be characterized as the "collection era" has occurred: an era with clear investment in making historical research possible via the collection, documentation, conservation, and preservation of games and related materials across various cultural institutions, including the labor of private collectors and committed game enthusiasts. (Guins, 2014, p. 4)

Guins, like Aycock, is focussed on machine coded commercially released games produced by major video game companies like Atari. He begins his reflection on the state of retrogaming history by examining the example of Atari's 1979 arcade release of *Lunar Lander*. This game serves as the focus of his introductory chapter, but he only mentions once "Its early design as a text-based simulation," which refers to its origins as the BASIC game "Rocket" on minicomputer systems (Guins, 2014, p. 4). As he observes: "The working program and fully functioning console, computer, or arcade cabinet are most often regarded [by collectors] as the definitive property of the game" (Guins, 2014, p. 5). The collection era is characterized by the growth of private collections of game hardware and software. These collections are the foundation of the current preservation system and are the major source of materials for researchers and museums dedicated to the preservation of early computing materials, such as the Centre for Computing History in Cambridge.

According to Guins, early 2000s scholarly activity was focussed on what media archaeologist Erkki Huhtamo describes as the activity of "amassing and organizing data" (Guins, 2014, p. 25). But in the second decade of the new century the focus has shifted to "not just of gathering data but also of protecting and providing access to it." Guins continues:

> Without dedicated preservation and documentation game historians will lose access to materials vital to their historical research. They will have little recourse but to base future histories on the back of chronicle era works rather than apply their interpretative methods to materials that allow for critical histories that run deeper than fact checking. (Guins, 2014, p. 26)

Scholars interested in software studies, retrogamer archeology and digital humanities have been able to draw on the work of private collectors. But it is now time to begin more systematic efforts at preservation of the materials upon which these disciplines rely. Guins notes that the “virtual world of the game, like all digital technology, is not immune to deterioration and requires dedicated techniques, strategies, and policies to help manage longevity” (Guins, 2014, p. 7). This new focus raises the question “what research materials and subjects are constitutive of games history, and equally vital, constitutive for constructing game history” (Guins, 2014, p. 24)? Early BASIC games are a neglected part of these emerging efforts at preservation.

Some Possible Theoretical Uses of early BASIC Games and Software

According to many of the main philosophical perspectives in the field of the philosophy of technology, understanding the earliest stages of technological development, when new kinds of technological invention move from the hands of designers to users, is critical for our understanding of processes of technological development. Consider the case of the SCOT approach, which suggests that contingent social interests and the motives of specific human individuals are major, if not determining, influences on processes of technological development and diffusion. Sociologist Trevor Pinch and Wiebe Bijker illustrate the power of the values of users in shaping the development process. Their ground breaking examination of the history of the development of the bicycle in the late nineteenth century and the protracted period of the giant Penny-farthings with their “bone shaking” wheels, reveals a history of intense value contestation about cultural factors of a wide array of groups that went far beyond factors of safety and efficiency that preoccupied designers (Pinch & Bijker, 1994). Looking at the development and subsequent use of early computer programs might shed light on similar processes of value contestation that help define pivotal paths of subsequent software development.

Consider the case of "Star Trader” by Dave Kaufman, which is widely recognized as the origin of the “space trading” subgenre of “construction and management simulations” (Wikipedia, 2021a; Wikipedia, 2021l). Despite being the forerunner to the popular *Trade Wars* series and subsequent games of this type, such as *Eve Online*, *Wing Commander* series and the *Elite* series, this program exists on the Net only as a listing for the obscure HP2000F minicomputer system of the 1970s. It comes in two parts that are supposed to be run in a "CHAINed" fashion. The source code mentions "loading the tape into the tape reader" (a storage medium of early minicomputers) as a means of saving and loading games in progress. On a now defunct *Trade Wars* fan site, preserved only on the Archive.today website*,* one finds some tantalizing screenshots of an Atari 8-bit home computer version of the game, but the links to those files are now dead (Archive.today, 2007). I had to work from the listing on the fan site to create a working copy of Kaufman's program.

When I finally got the two programs merged and running, I noticed that there seemed to be something wrong with the economies of the various star systems. They did not develop economically or to increase their demand for products. However, after combing the source for errors, I could find nothing that would explain the weird economic behavior. I began to suspect that there was simply something wrong with the data used to calculate the productivity for each star system. It seemed skewed towards high production of goods, which resulted in extremely low demand for goods from outside the planets. I went in search of a scan of the published listing of Kaufman's program.

The game was included as a type-in program in the *People's Computer Company* (PCC) newsletter. Later it appeared in a book compilation *What to Do After You Hit Return*. I was able to get a PDF of the book from the Net and when I looked at the DATA statements containing the "econometric" information for the simulation, they were different. Swapping in the data from the scan seemed to solve the problem. I do not know why the values were different but my struggle to get this game functioning illustrates the kinds of challenges discussed above that must be overcome to obtain working copies of significant early BASIC games.

Rogers Cadenhead, one of the commentators on *Star Trader* and its influence asked David Ahl, author of the first multi-million selling compendium of early BASIC games, for his perspective on the program. The following is a part of Ahl's response:

> As far as I know, the game Star Traders [sic] originally appeared in People's Computer Company (a newspaper/magazine) published by P.C.C. (an alternative education computer center) in Menlo Park, CA. (It's a bit

> confusing that the physical storefront computer center and the newspaper both had the same name.) The game also appeared in What to Do After You Hit Return subtitled "PCC's first book of computer games." PCC was a rather egalitarian organization, hence credit was only rarely given to the authors of the various games they published and there is no author credit affixed to Star Traders. So it could have been written by any one of the 40-50 people cited in the acknowledgements of the book. The game, written in "standard" HP BASIC, is a monster and, unusual for the time (1973) has both a set-up module and a main playing module. Also unusual for the time, it had the capability of saving the playing data (1) from one session to the next and (2) for different users who logged onto other nodes of the timesharing system. It was this that allowed it to be a multi-player game with games often lasting a matter of weeks or months. The book had some interesting suggestions and ideas for extending and modifying the game to make it more interesting and longer lasting; I believe that over the years many (or most) of these have been implemented. (Cadenhead, 2003)

One can speculate about how early players could have rigged up ad hoc ways to make the program one of the first multiplayer net-based games (tape reels must have been replaced by electronic files at some point). The way the game is programmed requires the users to type “save” at the end of their rounds of trading at the prompt for picking the “next planet to visit.” Doing so saves all the game’s main variables and then returns to the “next planet” prompt. This method of implementing a game save feature seems to assume that both the players will be physically present to enter their next moves.

However, early players could have organized games with other players on early networks that allowed file sharing. What you might do is send the entire game data file to the next player by saving it to a shared file space or by email attachment. Then the player receiving that data file could load it and continue with their turn by typing in the previous player’s selection for the “next planet.” Doing so would allow the current player to begin their round. The process could be repeated for each subsequent player (the game can handle up to 12). In fact, simply sending a message with a planet name might have been the way players let other players know the file was available and that it was their turn. Perhaps games played by a dozen players in this fashion taking place over “months,” as mentioned by Ahl, might also explain the exceedingly low pace of economic development of the data of the DEC minicomputer version. Such questions about the contribution of BASIC to the development of early networked gaming remains to be explored more fully. Often the machine code game *Maze War,* developed on the University of Illinois' experimental "Plato" graphic networked computer system in 1974, is identified as the starting point of multiplayer networked gaming (Downey, 2014, p. 58), overlooking the pivotal role played by BASIC starting in 1969 at Dartmouth College (Rankin, 2018, pp. 48-49).

However, we can see in these speculations about the contribution of the program *Star Trader* a potential example of the central claim of SCOT that non-specialist users often play a pivotal role in determining the direction of processes of technological development. However, such considerations about the possible role of BASIC programmers, working largely for hobbyist reasons, in the development of early online gaming could only emerge through an ability to interpret source code for now defunct varieties of BASIC, or actually running the program (List, 2018). It is also clear that a non-playable digital scan of an archaic hobbyist computer book containing the listing of *Star Trader* on a few sites is a precarious existence for a program with such potential significance to the development of computer gaming. And SCOT is not the only theoretical outlook potentially enlightened by an examination of *Star Trader*.

Theoretical perspectives like that of Steven Woolgar, which look on technologies as texts, might also benefit from consideration of *Star Trader*, as might perspectives that assert as Langdon Winner does that all "artifacts have politics" (Gerrie, 2003; Winner, 1980). One of the reasons that there might be no actual playable versions of Kaufmann's program preserved on the Net, but only scans of the original newsletter and books in which it appeared in the mid-1970s is that there are so many other games inspired by it that use similar names. You can find all kinds of BASIC "Star Trader" or "Star Traders" games preserved in various archives, but they are distinct programs made by other programmers. Many were commercially released programs. In fact, it was my recollections of these other variations that put me on the path to Kaufmann's original version. But these other more well-known variations involve many thematic differences from the original. For example, many involve the player in the buying and selling of stocks in the different space-trading companies. However, Kaufman's version was much less corporate in character, and more in keeping with Isaac Asimov's "Foundation" series of novels with freelance traders navigating pirate infested space between developed galactic core planets and wild frontier planets.

The development of many early BASIC games from classic novels is also potentially illustrative of observation by Marshall McLuhan. In his essay *The Medium is the Message* McLuhan states that "the content of any media is another media" (McLuhan, 1964, pp. 23-24). The early period of BASIC gaming is rich in potential examples of this proposed aspect of technological development, such as *Star Trader*'s connection to Asimov's novel, or games like *Crush, Crumble Chomp* and *Awful Green Things* based on board games (Wikipedia, 2021b). However, awareness of the influence of prior media on subsequent paths of development is likely to be achieved only from the experience of playing games that actually helped establish video game genres.

Unfortunately, there are instances of genre inspiring early BASIC games simply going missing. Aycock for example mentions that the original source of the *Wander* game engine appears to be missing (Aycock, 2016). This game engine allowed for the creation of interactive fiction games and was distributed with a sample adventure “Wander,” from which the program took its name. It is now considered to be the first text adventure game. There was also a near-miss regarding the Fortran source code of what was thought until recently to be the first text adventure, *Colossal Cave Adventure* (Jerz, 2007). These instances illustrate why it is so important not to leave games like *Star Trader* hanging by such precarious threads.

BASIC programs have played major roles in the creation of many game genres. The important role played by Steve Wozniak’s game *Little Brick Out* is illustrative. This program was, according to Wozniak, the first instance of the implementation of a major arcade game written completely in software (Hanson, 2015). Wozniak helped design the circuit board for Atari’s cabinet version of Breakout, which inspired him to design the first Apple computer and its preliminary version of BASIC called Integer BASIC (but originally titled “game BASIC”) to allow for the creation of arcade style games completely in software (Szczepaniak, 2013). But finding a copy of Wozniak’s original Integer BASIC source for *Little Brick Out* and getting Integer BASIC working on an Apple emulator, will pose many obstacles for the uninitiated.

Some Obstacles to Using Early BASIC Programs for Digital Humanities Research

Repositories like the *Internet Archive* are dependent on the interests of the voluntary contributors, such as those who have helped build up its substantial Apple collection. If those contributors are focused on very popular North American computer systems like the Apple II, then those interests get reflected in what is preserved. If those doing the contributing are focused on the commercial games they enjoyed playing when they were young, then those biases will affect which programs are made easily available. For example, one can find a substantial array of commercial breakout-style games for the Apple II on the *Internet Archive*. The result of such multiple biases can skew not only the type of computers and software that end up being made available, but also which countries are represented. Such biases can obscure the highly international character of early BASIC programing and the type-in phenomenon and support a view of an overly exaggerated role for corporations in the creation and production of groundbreaking (and genre inspiring) software. Overlooking the international scope of early BASIC programming seriously undermines the possibility of pursuing a SCOT approach to software studies because that approach requires an ability to engage in cross-cultural comparisons of processes of technological development, especially the earliest stages of these processes. Without this ability it is impossible to discern paths undertaken by people with different evaluative commitments and cultural backgrounds.

Hobbyist programmers using BASIC helped pioneer many genres besides text adventures like *Wander* and arcade games like *Little Brick Out*. However, finding published primary or even secondary sources discussing such contributions can be difficult because of the kinds of biases and practical difficulties discussed above. However, Wikipedia has a substantial listing of video game genres and one can often find tantalizing, if brief, references to BASIC programs at the historical beginnings of many of those genres. Here are nine such references, with supporting references from online commentators who have begun to explore these BASIC games in more detail:

- Joseph Power's *Wizard's Castle* among the first microcomputer Role Playing Games (RPGs). (Barton, 2007; Bolingbroke, 2013)
- According to some commentators the first stealth game was *Manbiki Shounen* (Shoplifting Boy), which was published in November 1979 in Japan as a type-in program for the Commodore PET. (Szczepaniak, 2017; Wikipedia, 2021m)
- Scott Adam's *Adventureland,* the first commercial text adventures for microcomputers and a foundational program for the interactive fiction genre. (Adams & arcadeattack, 2018; Wikipedia, 2021e)
- Based loosely on the simple text game *Hamurabi*, *Santa Paravia and Fiumaccio* was an early God game. (Bolognesi, 2018; Wikipedia, 2020)
- "*Nostromo* was a survival horror game developed by Akira Takiguchi, a Tokyo University student, for the PET 2001." (Grouvee Media LLC, 2019; Wikipedia, 2021n)
- *Cavern 2160*, published as a program listing in the July 1980 issue of *I/O Magazine* in Japan is considered by some to be "the original sandbox game," forerunner to open world games like Minecraft. (Thorpe, 2016)
- *Heiankyo Alien*, is a game created by the University of Tokyo's Theoretical Science Group (TSG) in 1979 and published in various Japanese computer magazines (Jankiewicz, 2014; shmuplations, 2015; Wikipedia, 2021d). It is credited as the inspiration for the commercial arcade game *Space Panic*, which is often credited as the inspiration of the entire platform game genre. (Jankiewicz, 2014; Wikipedia, 2021h)
- *Beneath Apple Manor*, the first "roguelike" RPG that preceded the commercial machine language game "Rogue." (Pepe, 2016; Wikipedia, 2021k)
- Hiroyuki Imabayashi submitted *Sokoban* to a programming contest in 1982, which does not appear to have been preserved. However, a type-in version was published in a Japanese computer magazine. The game is considered to be foundational in the action puzzle video game genre. (Ahab, 2016; Wikipedia, 2021j)

While gaining access to some of these titles is relatively easy, as with the classic game *Hamurabi*, others such as *Star Traders*, *Shoplifting Boy, Cavern 2160, Nostromo,* and *Heiankyo Alien* do not seem to have any playable copies (Grouvee Media LLC, 2019; joafro1, 2016).[4] In his discussion of the seminal role played by *Shoplifting Boy* in the development of the stealth genre, for example, the game commentator "Stealth Docs" had to rely on a Youtube video of someone playing a later version of the game ported to another system, and not a functioning copy of the original Commodore PET game (Szczepaniak, 2017). The case of *Adventureland* illustrates some of the pitfalls that can affect access to original BASIC versions. Although Scott Adams has generously allowed his influential early text adventures to be released as freeware, it is only his later commercial versions, programmed in machine code, that are widely available (Wikipedia, 2021f). Despite the significant role early BASIC games have played in the establishment of many video game genres, popular histories of video games often ignore this role. Ahoy's popular *Youtube* video "History of Video Games" for example leapfrogs from discussing the first arcade cabinets to discussing the influence of the commercial game *Rogue*, despite the fact that gaming historians consider the BASIC games *Beneath Apple Manor, Dungeon of Death* or *Sword of Fargoal* to be the first "roguelike" games even before the commercially developed game *Rogue* (Ahoy, 2013). The pivotal role of hobbyist BASIC programmers in the development of the game industry is occluded in many popular histories of the industry.

The cases of *Star Traders* and *Little Brick Out* illustrate the precariousness and limits of current systems of preservation. Instances like *Wander* and the wargame *Empire* created in 1972 by Peter Langston, illustrate that source code can be lost (Wikipedia, 2021c). The discipline of retrogame archeology, like actual archeology, require more than the preservation of a few popular examples. It requires the ability to trace lines of development from their origins, such as one can find in the extensive genealogy of the BASIC game *Super Star Trek* (Lorenzo, 2017, 86). Unlike machine language games, BASIC programs are easily modified and within the grasp of non-specialists to create their own variations. Such diversity itself embodies critical information about the goals and priorities of programmers and fans and the rapidly changing nature of software culture at a very detailed and early level (Swalwell et al., 2017).

Although applied in a wide variety of social scientific disciplines, Actor-Network Theory is a method of investigation developed by sociologists Michael Callon and John Law, and the philosopher Bruno Latour in the early 1980s. Their investigation takes a less socially deterministic approach to the examination of processes of technological development than the SCOT approach. Its central idea is that technological objects and systems can themselves be actors in processes of technological development, which highlights the position popularly espoused by McLuhan and his catchphrase the "medium is the message." As Law puts it, we "seek to associate entities that range from people, through skills, to artifacts and natural phenomena" (Law, 1994, p. 129). Technical limitations, existing technological systems all potentially skew technological developments in certain directions as much as the conscious evaluative inclinations of specific human actors. The whole complex of such pushes and pulls working together are what create specific processes of development. For example, Aycock discusses how certain hardware limitations in early games like *Atari's Adventure* demanded certain compromises in the design of the game, which once adopted became integral aspects of the genre (Aycock, 2016, pp. 30-33, p. 184).

Many unique obstacles had to be faced by early BASIC text adventure authors working in languages other than English. For example, Tihor Miklós, in his creation of one of the first Hungarian BASIC text adventures, *A Hős Lovag,* recounts how Hungarian programmers were forced to make significant linguistic compromises to duplicate the simple verb-noun format widely used in English text adventures (fubenalvo, 2017). Miroslav Fídler creator of an early Czech BASIC text adventure *P.R.E.S.T.A.V.B.A.* recounts similar difficulties. Another interesting fact about this game was that it was written as an anti-Soviet protest just before the Velvet Revolution protests and actually made a veiled reference to the day and location of the protests (Reed, 2021). I have also translated and made available from French the first installment of the well-known *Citadelle* series from Loriciels, *The Shield of Oran,* in which one finds linguistic difficulties similar to those being addressed by Eastern European programmers. Complex algorithms for parsing and outputting responses based on user input in appropriately grammatically gendered form are spread throughout the code. Multiplied across text-intensive programs like text adventures, this requirement could be prohibitive in the limited memory of the machines of the era. Perhaps this is why one finds relatively few early French BASIC text adventures and why prominent examples, such as *Cavern of the Morlocks* by François Coulon, use a single key command entry system, which is very different from verb-noun parser systems of English text adventurers. The memory limits of early BASIC machines might explain why playable versions of French text adventures seem relatively less numerous in online interactive fiction archives. *The Shield of Oran*, for example, appears to be available as a playable program on only one hobbyist site (Quénard, 2011).

However, there are also many examples of the influence of more socially determined themes in BASIC games, as would be highlighted by SCOT theorists. The unique French type-in BASIC program by Philippe Henri called *Cadavres Exquis*, which generates random surrealist poetry (Montfort & Savoie, 2018) is an interesting example. It was recently presented in English translation by Nick Montfort and Ariane Savoie in the *Vassar Review*, but although it was available for a brief time as an actual running program on the journal's website using an online emulator, it now seems to have been switched because of concerns about copyright to a brief animated GIF of the output of the program (Montfort, 2018).

The early Japanese type-in game *Pitman* by Yutaka Isokawa, illustrates a uniquely Japanese fascination with puzzle games combined with arcade game elements. This BASIC game was so popular that it was later ported to modern game systems under the name "Catrap" (Sketcz, 2012). It has been identified by John Szczepaniak, author of *The Untold History of Japanese Game Developers* as one of the most influential early BASIC programs (Szczepaniak, 2014a). Smartphone versions have also been made (Szczepaniak, 2012). The combination of puzzle and arcade gameplay can also be found in *In the Woods* by Ken Fujisawa, which has led Szczepaniak to speculate that this distinctly Japanese gaming style originates in the influence of these early BASIC programs (Szczepaniak, 2014b). Possibilities for discerning distinctly national themes of gameplay like this abound. Simon N. Goodwin's *Shop Steward*, for example, is a simulation of being a union organizer in a factory producing "consumer durables" in a "declining economy," which reflects the social anxieties of the United Kingdom in the early 1980s. Dominique Laroche's *Microgargouilles* (micro-gargoyles) is a type-in version of the 1983 arcade hit *Pengo*. In his unique variation, however, the premise is modified to reflect something more appropriate for the land of Joseph Lister, with a little gothic ambiance thrown in from the architecture of Notre Dame.

Conclusion

How can speculations like some of those touched on above, which can possibly illustrate various theoretical perspectives on technology, be grounded in scholarship if such programs only take the form of non-interactive magazine scans or simply as references discussed by authors from the "chronicle era" of game history? Unless there can be real engagement with these programs by subsequent scholars without the need of arcane programming knowledge, then digital humanities and software studies research on this critical early stage of computer development will be extremely difficult. In the same way that paleontology must find ways to dynamically illustrate its findings rather than simply leaving them in boxes in warehouses, those interested in the early stages of computer game development must provide ways for future researchers to interact with materials.

And just like the disciplines of archeology and paleontology had to eventually transition from relying on adventurers and plunderers to obtain their materials, those pursuing software studies and digital humanities need to begin to consider the need for professionalizing the preservation and curation of early software. This process has begun to take shape, such as with the creation of the Software Preservation Network (SPN), but it is still at an early stage. The SPN, for example, is only just moving from being a grant supported project to a self-supporting network (Software Preservation Network, 2019). Hobbyists have played and will continue to play important roles in "gathering data" and "protecting and providing access to it" as Guins notes, but they have no specific professional obligations to do so in ways that respect professional standards like those laid out by the SPN. It is the responsibility of scholars and professional archivists to ensure that this is done. This might strike some as a strange concern since such materials, in historical terms, are relatively recent, but as Michael Heim notes:

> Transitions are privileged location for philosophical reflection. By examining ourselves as we experience the computer revolution, we may be able to pass along to the inhabitants of the unknown future some fleeting glimpses into our common human destiny. (Heim, 1987, p. 10)

Those with the skills useful for the curation of materials from the historical period of the early stages of the microcomputer revolution must begin to prepare those materials to allow future researchers to participate in the analysis of that time of transition. As David Berry observes, "what remains clear… is that looking at computer code is difficult due to its ephemeral nature, the high technical skills required of the researcher and the lack of analytical or methodological tools available" (Berry, 2011, p. 5).

Much of my analysis here has risen out of my experience at creating a hobbyist website. With the demise of that website, I realized that it was time to make more systematic efforts at properly curating these programs. Although I transferred many materials from that site to my institutional file storage workspace, I have excluded the source code for any programs for which I have been unable to confirm copyright being passed to the public domain or the author's permission. An essential step for researchers and archivists is to begin seeking out the copyright holders of early BASIC games in order to ask permission to share their work. This step is daunting given the large numbers of early programs, but it could provide some solid undergraduate research work. My discussion here is a plea for greater preservation efforts to be carried out, but it is also a notification about my own preliminary efforts and a request for feedback from researchers and teachers in other fields regarding other possible applications of early BASIC programs in the classroom and in digital humanities research.

References

Adams, S., & arcadeattack. (2018). *Scott Adams (Adventure International): Interview.* https://www.arcadeattack.co.uk/scott-adams/

Ahab. (2016, May 6). Game 184: Sokoban. https://datadrivengamer.blogspot.com/2020/05/game-184-sokoban.html?showComment=1634499462013#c2710916858597715534

Ahoy. (2013). *A Brief History of Video Games.* https://youtu.be/GoyGlyrYb9c

Anderson, N. (2018). *First Encounter: COMPUTE! Magazine and its Glorious, Tedious Type-in Code.* https://arstechnica.com/staff/2018/11/first-encounter-compute-magazine-and-its-glorious-tedious-type-in-code/

Archive.today. (2007). *Star Trader Tribute Page.* http://archive.li/DEUfb; originally from www.langesite.com/startrader/

Aufderheide, P., Butler, B., Cox, K., & Jaszi, P. (2018; 2019). *Code of Best Practices in Fair Use for Software Preservation September 2018; Revised February 2019*. Association of Research Libraries; Center for Media & Social Impact, Washington College & Law.

Australasian Digital Heritage Assoc. (2020). *Our Digital Heritage.* http://ourdigitalheritage.org/index.php

Aycock, J. (2016). *Retrogame Archeology: Exploring Old Computer Games*. Springer.

Bangeman, E. (2012). *First encounter: Ohio Challenger 2P (and its 4KB of RAM).* https://arstechnica.com/staff/2012/12/first-encounter-ohio-challenger-2p-and-its-4kb-of-ram/

Barton, M. (2007). *The Story of The Wizard's Castle.* http://www.armchairarcade.com/neo/node/1381

Berry, D. M. (2011). *The Philosophy of Software: Code and Mediation in the Digital Age*. Palgrave MacMillan.

Bolingbroke, C. (2013). *Game 90: The Wizard's Castle (1980).* http://crpgaddict.blogspot.com/2013/02/game-90-wizards-castle-1980.html

Bolognesi, E. (2018). *The fascinating story of Santa Paravia and Fiumaccio.* https://gamesnostalgia.com/story/166/the-fascinating-story-of-santa-paravia-and-fiumaccio

Cadenhead, R. (2003). *I'm an Ahl-knowing Computer Book Author.* http://workbench.cadenhead.org/news/1365/im-ahl-knowing-computer-book-author

Carey, J. W. (1972). *The Politics of the Electronic Revolution: Further Notes on Marshall McLuhan*. Institute of Communications Research, University of Illinois.

Downey, S. (2014). History of the (Virtual) Worlds. *The Journal of Technology Studies, 40*(1), 54-66. 10.21061/jots.v40i2.a.1

Figueroa, L. (2014). *Retro Game Programming.* http://www.brainycode.com/downloads/RetroGameProgramming_v_12.pdf fubenalvo. (2017). *A hős lovag.* https://hungame.blog/2017/05/14/a-hos-lovag/

Gerrie, J. (2019). *Type-in Mania: An Interactive Database of Early Basic Computer Applications.* http://faculty.cbu.ca/jgerrie/Home/Type-in-Mania-Programs.html

Gerrie, J. (2003). Was Foucault a Philosopher of Technology? *Techné: Research in Philosophy and Technology, 7*(2), 14-26.

Grouvee Media LLC. (2019). *AX-2: Uchuu Yusousen Nostromo.* https://www.grouvee.com/games/26457-ax-2-uchuu-yusousen-nostromo/

Guins, R. (2014). *Game After: A Cultural Study of Video Game Afterlife*. The MIT Press.

Hague, J. (2019). *The Giant List of Classic Game Programmers.* https://dadgum.com/giantlist/

Handelman, I. (2014, 5 Nov). Internet Archive Working Toward Classic Game Preservation. *The Spokesman-Review*

Hanson, B. (2015). *How Steve Wozniak's Breakout Defined Apple's Future.* https://www.gameinformer.com/b/features/archive/2015/10/09/how-steve-wozniak-s-breakout-defined-apple-s-future.aspx

Heim, M. (1987). *Electric Language: A Philosophical Study of Word Processing*. Yale University Press.

Jankiewicz, J. (2014). *Heiankyo Alien.* http://www.hardcoregaming101.net/heiankyo-alien/

Jerz, D. G. (2007). Somewhere Nearby is Colossal Cave: Examining Will Crowther's Original 'Adventure' in Code and in Kentucky. *Digital Humanities, 1*(2), Aug 21, 2019.

joafro1. (2016). *Cavern 2160 on Sharp MZ-80A.* https://youtu.be/AmPRf4ZStS8

Law, J. (1994). Technology and Heterogeneous Engineering: The Case of Portuguese Expansion. In W. E. Bijker, T. P. Hughes & T. Pinch (Eds.), *The Social Construction of Technological Systems* (pp. 111-134). MIT.

List, J. (2018). *The Basic Issue With Retro Computers.* https://hackaday.com/2016/11/18/the-basic-issue-with-retro-computers/

Lorenzo, M. J. (2017). *Endless Loop: The History of the Basic Programming Language*. SE Books.

McCracken, H. (2014, April 29). Fifty Years of BASIC, the Programming Language That Made Computers Personal. *Time Magazine,*

McLuhan, H. M. (1964). *Understanding Media: The Extensions of Man* (2nd ed.). McGrawHill.

Montfort, N. (2018). *Exquisite Corpses are Now on Display.* https://nickm.com/post/2018/07/exquisite-corpses-are-now-on-display/

Montfort, N., & Savoie, A. (2018). *Cadavres exquis / Exquisite Corpses.* https://review.vassar.edu/issue/2018/digital/nick-montfort/

Orland, K. (2018). *ROM Sites are Falling, but a Legal Loophole Could Save Game Emulation.* https://arstechnica.com/gaming/2018/08/can-a-digital-lending-library-solve-classic-gamings-piracy-problem/

Pepe, F. (2016). *The CRPG Book Project: Sharing the History of Computer Games.* Unpublished manuscript.

Pinch, T., & Bijker, W. E. (1994). The Social Construction of Facts and Artifacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other. In W. E. Bijker, T. P. Hughes & T. Pinch (Eds.), *The Social Construction of Technological Systems* (). MIT.

Quénard, N. (2011). *Projects and new developments around the PHC-25.* http://www.phc25.com/projets.htm

Rankin, J. L. (2018). *A People's History of Computing in the United States*. Harvard University Press.

Reed, A. (2021, May 6, 2021). 1988: P.R.E.S.T.A.V.B.A. https://if50.substack.com/p/1988-prestavba

shmuplations. (2015). *Heiankyo Alien: 2002 Developer Interview.* http://shmuplations.com/heiankyoalien/

Sketcz. (2012). *Basic History of BASIC - Dr Kurtz, David Ahl, John Lutz, Steve Wozniak.* http://blog.hardcoregaming101.net/2012/09/basic-history-of-basic-dr-kurtz-david.html

Software Preservation Network. (2019). *Get Involved.* https://www.softwarepreservationnetwork.org/get-involved/

Swalwell, M., Ndalianis, A., & Stuckey, H. (Eds.). (2017). *Fans and Videogames:Histories Fandom Archives* (1st ed.). Routledge.

Szczepaniak, J. (2014a). *BASIC's 50th Anniversary Draws Near: 10 (+1) Things Touched by BASIC.* https://www.gamasutra.com/blogs/JohnSzczepaniak/20140428/216345/BASICs_50th_anniversary_draws_near.php

Szczepaniak, J. (2012). *Catrap.* http://www.hardcoregaming101.net/catrap/catrap.htm

Szczepaniak, J. (2013). *Steve Wozniak discusses early Apple years and BASIC.* https://www.gamasutra.com/blogs/JohnSzczepaniak/20130507/191813/Steve_Wozniak_discusses_early_Apple_years_and_BASIC.php

Szczepaniak, J. (2014b). *History of Japanese Computer Games.* https://www.gamasutra.com/blogs/JohnSzczepaniak/20140304/212177/History_of_Japanese_computer_games.php

Szczepaniak, J. (2017). *Stealth Game History: The 'First' Stealth Game.* https://youtu.be/qrOiXcoABVg

Target, S. (2018). *Learning BASIC Like It's 1983.* https://twobithistory.org/2018/09/02/learning-basic.html

Thorpe, N. (2016). *Cosmic Cavern 3671 On the Way.* https://www.retrogamer.net/blog_post/cosmic-cavern-3671-on-the-way/

U.S. Copyright Office. (2013). *More Information on Fair Use.* https://copyright.gov/fair-use/more-info.html

University of British Columbia. (2018). *Public Domain in Canada.* https://guides.library.ubc.ca/c.php?g=698822&p=4961440#s-lib-ctab-15622469-1

Venor, J. (2014). MZ-80 Nostalgia: Early MZ-80 Software (Part I). *Sharp Users Club, 34*(2), Aug 24, 2019-6-12.

Wall, W. (2017). *Computer Chess Pioneers.* http://billwall.phpwebhosting.com/articles/computer_chess_pioneers.htm

Wikipedia. (2020). *Santa Paravia en Fiumaccio.* https://en.wikipedia.org/w/index.php?title=Santa_Paravia_en_Fiumaccio&oldid=994539606

Wikipedia. (2021a). *Construction and Management Simulation.* https://en.wikipedia.org/w/index.php?title=Construction_and_management_simulation&oldid=1037959110

Wikipedia. (2021b). *Crush, Crumble and Chomp!* https://en.wikipedia.org/w/index.php?title=Crush,_Crumble_and_Chomp!&oldid=1047090672

Wikipedia. (2021c). *Empire (1972 video game).* https://en.wikipedia.org/w/index.php?title=Empire_(1972_video_game)&oldid=1039886217

Wikipedia. (2021d). *Heiankyo Alien.* https://en.wikipedia.org/w/index.php?title=Heiankyo_Alien&oldid=1023915009

Wikipedia. (2021e). *Interactive fiction.* https://en.wikipedia.org/w/index.php?title=Interactive_fiction&oldid=1047428061

Wikipedia. (2021f). *List of Commercial Video Games Released as Freeware.* https://en.wikipedia.org/w/index.php?title=List_of_commercial_video_games_released_as_freeware&oldid=1043138421

Wikipedia. (2021g). *Orphaned Technology.* https://en.wikipedia.org/w/index.php?title=Orphaned_technology&oldid=1045022632

Wikipedia. (2021h). *Platform game.* https://en.wikipedia.org/w/index.php?title=Platform_game&oldid=1050328012

Wikipedia. (2021i). *Public-domain Software.* https://en.wikipedia.org/wiki/Public-domain_softwarehttps://en.wikipedia.org/w/index.php?title=Public-domain_software&oldid=1047837420

Wikipedia. (2021j). *Puzzle video game.* https://en.wikipedia.org/w/index.php?title=Puzzle_video_game&oldid=1048334191

Wikipedia. (2021k). *Roguelike.* https://en.wikipedia.org/w/index.php?title=Roguelike&oldid=1050512785

Wikipedia. (2021l). *Star Trader.* https://en.wikipedia.org/w/index.php?title=Star_Trader&oldid=1049434810

Wikipedia. (2021m). *Stealth Game.* https://en.wikipedia.org/w/index.php?title=Stealth_game&oldid=1045954918

Wikipedia. (2021n). *Survival Horror.* https://en.wikipedia.org/w/index.php?title=Survival_horror&oldid=1040055644

Wikipedia. (2021o). *Tim Hartnell.* https://en.wikipedia.org/w/index.php?title=Tim_Hartnell&oldid=1039110104

Winner, L. (1980). Do Artifacts Have Politics? *Daedalus, 109*(1), 121-136.

[1] The concept of "type-in" program refers to software written in *any* language published in magazines and books that could be typed into a computer. However, the kinds of programs I am concerned with here are those written in interpreted forms of Basic during the early age of personal computing, roughly defined as the period between 1970 and the mid-1980s. Alternative terms might be "classic Basic" or "Vintage Basic" programs but those phrases, in my experience, have come to refer to a narrow category of simple early Basic programs, especially, the compilations published by David Ahl.

[2] This book is typical. A search of it produces no major reference to Basic as a significant programming language for retrogames.

[3] The only significant Basic game discussed at length in many greatest video game compendiums is "Oregon Trail," and even this is largely considered only in terms of its later commercial machine code releases.

[4] Only two people are listed on the Grouvee game database as having copies of this game. I was able to locate a copy of it in an online repository for the NEC PC-6001 at some point, but subsequent searches no longer turn up this site.