Schoppmeier, Sören. "Breaking the Habitual: 'Pony Island' as Countergaming." Modernities and Modernization in North America, edited by Ilka Brasch and Ruth Mayer, Universitätsverlag Winter, 2018, pp. 427–48.

SÖREN SCHOPPMEIER

Breaking the Habitual: *Pony Island* as Countergaming

"There is no software," Friedrich Kittler claimed in 1992. He was referring to the ways in which all software, or any kind of writing in the computer age, boils down to nothing but dynamic, physical hardware operations: "Grammatologies of the present time have to start with a rather sad statement. The bulk of written texts—including this text—do not exist anymore in perceivable time and space but in a computer memory's transistor cells" (Kittler). Such an observation, then, refutes any notion of software "exist[ing] as a machine-independent faculty" (Kittler). Yet in 2013, Lev Manovich asserted that "[t]here is only software" (*Software* 152). Manovich argues that all media production and consumption today, after "[m]edia [b]ecame [n]ew" (*Language* 21), is defined by software applications, whose configurations and workings are as significant for the user's experience of media content as the content itself (*Software* 147-57). Foregoing Kittler's insistence on the material conditions of media technology, its hardware, this essay follows Manovich's emphasis on the forms and operations of new media. Although software may not exist self-sufficiently, as Kittler would have it, its impact is, nonetheless, fundamentally real, which is why considerations of its forms and functions need to take center stage in discussions of the role of computational technology in twenty-first-century life.

In the following pages, I discuss how the computer game *Pony Island*, which was published on Steam[1] by Daniel Mullins Games in

[1] Steam is an online-only computer game distribution platform developed and operated by Valve Corporation and is the largest and most important platform for the distribution of PC games today (Grubb).

2016, engages with the primacy of software and algorithms in contemporary life by implicating players in the logics of new media through gameplay. I ultimately argue that *Pony Island* constitutes a case of what Alexander R. Galloway calls "countergaming" (*Gaming* 109), as it uses (game)play and the conventions of computer games to foreground some of the computational structures that usually remain hidden (and deliberately so), even as they define computer games in particular and large parts of contemporary life in general. I begin by delineating some central characteristics of the computer age that are essential to understanding (the meaning of) *Pony Island*. I will then introduce the game before I outline the idea of countergaming. Once all of these parts are in play, I will analyze the ludic expression of the logics of the computer age presented by *Pony Island*.

The Computer Age

Let us begin with a rather simple question: what is the computer age? Instead of providing a comprehensive history and theory, I suggest using the term in a broad sense, especially since it also works as an umbrella term for related, though not necessarily synonymous, terms frequently employed, such as 'the information age,' 'the digital age,' and, most recently, 'the algorithmic age.' Going back to the 1830s, a century before visionaries like Alan Turing and Konrad Zuse laid the groundwork for modern computing, the roots of the computer age, as Manovich shows, can be found in the inventions of Louis Daguerre's daguerreotype and Charles Babbage's Analytical Engine, the latter of which was never realized (*Language* 21-26). But the computer age really began in the second half of the twentieth century, sometime between the 1960s and 1980s, as the computer increasingly became the technology underlying all forms of production, from industrial to cultural, throughout the so-called developed world, and reached an ever-expanding state of full bloom around the turn of the millennium, not least because of the emergence and rapid expansion of the Internet during the 1990s. What matters more than determining when it began, however, is that we are living in it. For people born from roughly the mid-1980s onward, at least those growing up in postindustrial societies, there has never even been another age; their modernity is the computer age.

The computer informs and often dominates all forms of production today—the production of goods, the production of culture, the production of social life, in short, the production of reality. To live today is to interact with, to coexist with, to depend on, to be affected by computational logics and processes. This can partly be attributed to what David Golumbia, in *The Cultural Logic of Computation*, calls "*computationalism*" (2). He describes this phenomenon as "a commitment to the view that a great deal, perhaps all, of human and social experience can be explained via computational processes" (8), which can be considered the ideological basis of the ever more pervasive computerization of human life.[2] In the following, drawing largely on Manovich's *The Language of New Media* (published in 2001)—which, despite its relative age and the criticisms it has been subjected to over time, remains a key work on the forms that are exemplary for and ubiquitous in the computer age[3]—I would like to focus on three of these forms: the human-computer interface, the logic of selection, and the algorithm. While time (and, subsequently, technology) has moved on since Manovich first published his seminal work, these forms still define much of contemporary human-computer interactions. As this essay will show, they are, furthermore, central to the "*procedural rhetoric*" of *Pony Island* (Bogost ix), which

[2] Similarly, Wendy Hui Kyong Chun, in her book *Programmed Visions: Software and Memory*, argues for the role computers and software (along with fields like modern genetics) have played in advancing a "logic of programmability" (9), which she relates to a Foucauldian understanding of governmentality.

[3] Two of the most prominent critiques of Manovich's book can be found in Mark B. N. Hansen's *New Philosophy for New Media* and Galloway's *The Interface Effect*. Both, while skeptical about generalizations of essential features defining *all* new media, do not necessarily deny or refute the forms described by Manovich (at least if considered in light of the period Manovich was concerned with); rather, they are dissatisfied with the purely formalist approach itself and instead call for theorizations foregrounding new media's affective (Hansen) and social/political (Galloway) dimensions. Galloway furthermore concisely maps the main counterarguments to Manovich's position and specifically discusses Hansen's critique in his introduction before presenting his own argument, while also absolving Manovich from misguided accusations of him turning a blind eye to the political implications of computational technology (*Interface* 1-24).

makes *The Language of New Media* an invaluable resource in understanding the game and its implications.

According to Manovich, the human-computer interface has, since the 1990s, become so prevalent that most culture is now filtered through it (*Language* 64), mainly through the latest iteration of the form of the screen, the interactive screen, coupled with a physical input device—although, today, the input device is often integrated into the screen itself.[4] This kind of human-computer interface allows the user to access digital data, often organized in the form of a database, which Manovich views as the key expressive form of the computer age (*Language* 218-19). As the screen is now, and has long been, "the main means of accessing any kind of information," Manovich describes the American society as "a society of the screen" (*Language* 94), and the same, of course, holds true for all other postindustrial societies.

Beyond filtering cultural artifacts, the interactive screen can be seen as the dominant life-structuring device for many people in these societies, particularly the younger generations. They wake up in the morning, turn on their smartphones (if they turned them off at all), read the news and their social media feeds, send messages to friends and family, and so on, all interwoven with their other daily routines. As they leave their homes, they check the fastest public transport connection or maybe use Google Maps to navigate them to wherever they need to be. When they are at work, they sit down in front of yet another screen, using it to complete the task set for them. And on and on like this it goes. Put simply, for many people in these societies, to live today is to use an interactive computer screen.[5]

"The [l]ogic of [s]election" is intimately connected with the human-computer interface and is described by Manovich as "the logic of all new media" (*Language* 123, 126). In short, it refers to the ways in which

[4] In this way, I am (like Manovich) concerned with specific *forms* of the interface and not with "*interface effects*," as Galloway is in his thought-provoking study (*Interface* vii).

[5] This development, of course, contributes to the blurring of the boundary between work and leisure under the current neoliberal regime: "In this society, work and leisure activities not only increasingly involve computer use, but they also converge around the same interfaces" (Manovich, *Language* 65).

creators as well as users of new media largely act by selecting items from a predefined set of choices offered by the computer, coded in software. If the database is the defining form of the computer age, selection is its main operation. Manovich discusses this first and foremost in the context of the creation of cultural objects, and he writes that "[n]ew media objects are rarely created completely from scratch; usually, they are assembled from ready-made parts. Put differently, in computer culture, authentic creation has been replaced by selection from a menu" (*Language* 124). I would argue that one can, again, expand this notion and trace the logic of selection as defined by Manovich throughout other aspects of contemporary life. When people in postindustrial societies undertake a road trip, they no longer sit down over a printed map, trying to construct the perfect route; instead, they turn to a computerized navigation system, they are offered a selection of routes—for example, shortest and fastest—and they pick one, to which they then adhere. When they search for information about a particular matter, they google; since they cannot possibly find and assess *all* information out there, they rely on algorithms like Google's PageRank to structure them first, so that they can then select from a list of suggestions. These are just two examples that indicate how the logic of selection underlies much of modern life and its cultural production. As Galloway puts it: "In short, to live today is to know how to use menus" (*Gaming* 17). In any case, the point is not that the act of selection would be anything radically new,[6] but rather that, in computer(ized) culture, it becomes "standardized" and ingrained in the interfaces people use to access data to such an extent that it becomes a default operation underlying many others (Manovich, *Language* 130).

Finally, let me turn to the algorithm. Generally speaking, an algorithm is "a final sequence of simple operations that a computer can execute to accomplish a given task" (Manovich, *Language* 223). Regardless of the kind of data they engage, algorithms are the principal procedures underlying all forms of computation. In this sense, any software

[6] Manovich, for example, points out how the emergence of these technologies coincides with the era of postmodernism and its aesthetic styles, such as the use of pastiche, the production of which was strongly supported and facilitated by new types of software (*Language* 129-31).

can be described as a particular set of algorithms written in computer code. Even though they define the essential workings of any interaction with software, however, and contrary to both the human-computer interface and the logic of selection, algorithms usually remain concealed. This is a secrecy by design, both because the code itself would be incomprehensible for most users and because, from the user's perspective, functionality (something works toward a desired end) is generally valued more than understanding the machine (how does it work?); in this way, "software is invisibly visible" (Chun, *Programmed* 9).

Yet debates about the impact of hidden informatic processes on our reality have proliferated in recent years. In 2017, for example, news media and the public in the United States and Europe were startled by a story about a company called Cambridge Analytica, which was purported—mainly by people affiliated with it—to have played a role in both the Brexit vote and the Trump election by using big data[7] to profile and target individual voters online, mainly on Facebook, in order to influence the political climate in favor of the agendas of its clients (Confessore and Hakim). While the effectiveness of its method is highly disputed and while it is unlikely that it did play a decisive role in the two cases at hand (Taggart), the mere thought of such a machine being able to substantially influence the outcomes of real political processes worried commentators.[8]

On a more general level and with a broader perspective, academic studies like Frank Pasquale's *The Black Box Society: The Secret Algorithms That Control Money and Information*, have emerged out of growing concerns about the social effects of computational algorithms. Pasquale traces the ways in which such algorithms are increasingly impacting Americans' lives even when people are not aware that they do

[7] In the words of Mark Andrejevic, "big data denotes the moment when automated forms of pattern recognition known as *data analytics* can catch up with automated forms of data collection and storage" (1675), usually concerning extremely large and often heterogeneous sets of data.

[8] In early 2018, new revelations about Cambridge Analytica's method appeared to shift the public discourse on this matter to Facebook's failure to protect its users' data—even as monetizing this data has been tantamount to their entire business model all along—while the actual impact of the operation so far continues to remain a mystery.

or how they do so, and he delineates how these algorithms have become a major source of money and power in the twenty-first century. His critique is directed less at the idea of big data as such, which, like all technologies, is neither inherently good or bad (but also never neutral) (cf. Winner), and which certainly promises positive effects in areas like, for example, medical research. Rather, he is concerned mainly with the ways in which algorithms affect the everyday lives of Americans while, at the same time, said software largely operates in secrecy and, thus, beyond independent supervision and control. Pasquale writes: "Reputation. Search. Finance. These are the areas in which Big Data looms largest in our lives. But too often it looms invisibly, undermining the openness of our society and the fairness of our markets" (5). Asking "So why does this all matter?" he writes that "[i]t matters because authority is increasingly expressed algorithmically. Decisions that used to be based on human reflection are now made automatically" (8).

Many people are, in one way or another, already affected daily by the workings of hidden algorithms. Just think of something like Google's PageRank: whenever we look for information on Google, what we get is a ranked list of a pre-filtered selection of possible results, determined both by the comparative relevance of a page judged by which and how many other pages link to it, and our own digital footprint that we leave whenever we use the internet (Vaidhyanathan 57-67, 182-83). This effect can only be mitigated to some degree by extreme precaution and alternative web practices that are often either inconvenient for or unknown to many users. In becoming filters that automatically select a particular set of information allegedly identical with what it is that we are looking for, such algorithms increasingly impact the ways in which individuals as well as corporate and state actors perceive the world and its people. As Siva Vaidhyanathan writes in his book *The Googlization of Everything: (And Why We Should Worry)*:

> If Google is the dominant way we navigate the Internet, and thus the primary lens through which we experience both the local and the global, then it has remarkable power to set agendas and alter perceptions. Its biases (valuing popularity over accuracy, established sites over new, and rough rankings over more fluid or multidimensional models of presentation) are built into its algorithms. And those biases affect how we value things, perceive things, and navigate the worlds of culture and ideas. (7)

This dynamic, of course, lies also at the heart of recent discussions about the growing polarization of political discourse and what is largely perceived as 'filter bubbles' and 'echo chambers,' amplified by networked digital media that feed people increasingly more of what they already believe rather than exposing them to alternative views (cf. Vaidhyanathan 182-84).

Generally, all of the points outlined in the previous paragraphs speak to the ways in which interactions with computers pervasively structure and inform everyday life in the twenty-first century, while the machine and its workings largely retreat from our perception. Most of the time we are not *consciously* turning to these machines; rather, they have become a natural part of our lives, the default way to do certain things. Already in the early 1990s, computer scientists predicted that this time would eventually come. In a 1991 article titled "The Computer for the 21st Century," for example, Mark Weiser explains the significance of technologies becoming so pervasive that they become transparent:

> The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.
>
> Consider writing, perhaps the first information technology. The ability to represent spoken language symbolically for long-term storage freed information from the limits of individual memory. Today this technology is ubiquitous in industrialized countries. Not only do books, magazines and newspapers convey written information, but so do street signs, billboards, shop signs and even graffiti. Candy wrappers are covered in writing. The constant background presence of these products of "literacy technology" does not require active attention, but the information to be transmitted is ready for use at a glance. It is difficult to imagine modern life otherwise. (94)

The "[u]biquitous computing" Weiser describes has arguably materialized in a different way than he anticipated (94)—a mix of personalized, portable, multi-use devices like smartphones and stationary purpose-specific machines instead of the myriad devices picked up, used, and left anywhere by anyone anytime that he envisions in his article. Nonetheless, computers and software applications have indeed "[woven] themselves into the fabric of everyday life" in such a way that, today, "they are indistinguishable from it" (94).

In this context, Wendy Hui Kyong Chun speaks of "habitual new media" (16), which she theorizes in her latest book, *Updating to Remain the Same: Habitual New Media*, "focusing on habits—things that remain by disappearing from consciousness" (x). In Chun's terms, the use of new media—understood, in this essay, as computational (and oftentimes networked) media running on algorithms—has become habitual in the sense that these technologies are employed constantly and pervasively while the use itself is generally an entirely unconscious one. Chun writes that

> **our media matter most when they seem not to matter at all,** that is, when they have moved from the new to the habitual. Search engines are hardly new or exciting, but they have become the default mode of knowledge acquisition. Smart phones no longer amaze, but they increasingly structure and monitor the lives of their so-called owners. (*Updating* 1)

This habitual use of software technologies is a defining aspect of life in twenty-first-century postindustrial societies. As inherently computational artifacts, computer games have repeatedly been described as a cultural form particularly expressive of the algorithmic age (for example, by Galloway, Ian Bogost, and Noah Wardrip-Fruin). Combining play and computation, computer games appear to elicit fundamentally ludic engagements with our modernity. Yet although they should be particularly suited to engage central issues of the algorithmic age, their elaborate fictional worlds and the illusion of player agency within these worlds usually work to disguise the technological apparatus behind these games. This remains true even as playing a game successfully, in a sense, often means discovering "its hidden logic—in short, its algorithm" (Manovich, *Language* 222; cf. Galloway, *Gaming* 90-91).[9] Since

[9] Many contemporary mainstream games arguably complicate this notion somewhat since the emergent behaviors of their systems are far more complex than those of the games of the late 1990s and early 2000s that Manovich and Galloway refer to, which can be attributed to advances in artificial intelligence and processing power. Nevertheless, this should be seen as a difference in degree rather than quality, so Manovich's dictum still holds true for most games today.

even this moment is usually concealed by the game's semiotic layer of representation, it still resembles the ways in which algorithmic technologies remain hidden—or 'transparent'—in other spheres of life. Recent metafictional indie games such as Daniel Mullins's *Pony Island* or *else Heart.Break()* by Erik Svedäng et al., however, present a contrary move that accentuates not only the 'game-ness' of computer games but also their status as software—that is, as algorithmic cultural objects. At last, the time has come to visit *Pony Island.*

Pony Island

Pony Island was developed by Daniel Mullins during the Ludum Dare 31 game jam[10] in 2014 and published on Steam in 2016. It is a metagame in which the player interacts with a virtual arcade machine to play a game called Pony Island.[11] Immediately after loading the game, something peculiar happens, as one cannot, in fact, start the game since clicking on "start" in the menu leads to nothing but a short audiovisual feedback and an error message. This immediately undermines player expectations as it radically disrupts one of the oldest conventions in computer gaming—the start button. Bereft of other possible interactions, the player has no choice but to enter the "options and help" menu instead, where one can find the option "fix start menu." Once she checks this box, however, the player must also fix the present menu, as the "back" button needed to return to the start menu collapses to require manual reinstatement by the player. Instead of finally being able to properly start the game now, however, the "start" button gives way to a graphical

[10] A game jam "is a gathering of people for the purpose of planning, designing, and creating one or more games within a short span of time, usually ranging between 24 and 72 hours, and some even longer. Participants are generally made up of programmers, game designers, artists, writers, and others in game development-related fields" ("Game Jam"). Some game jams, like the Ludum Dare, are entirely virtual, and the developers do not meet in one place ("Game Jam").

[11] From now on, the use of italics shall differentiate between those instances in which I talk about Daniel Mullins's game (*Pony Island*) and those in which I refer to the virtual game within this game (Pony Island).

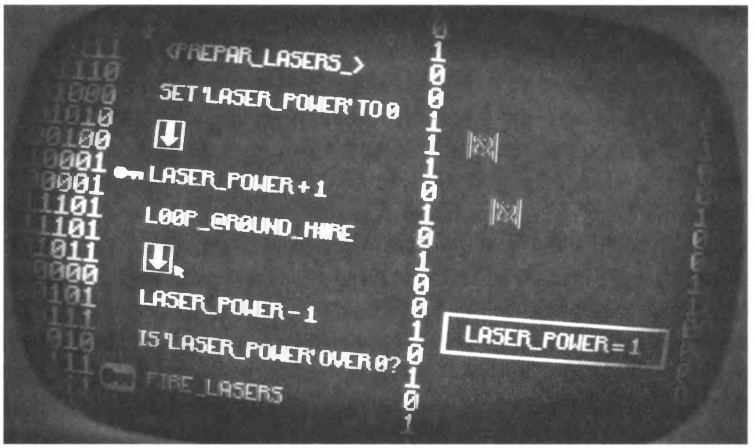

Figure 1: A piece of game 'code' in Pony Island.

glitch. Clicking on the glitch, the player finds herself confronted with what appears to be the game's source code (i.e., that of Pony Island), which is incomplete and, hence, unable to run (cf. Fig. 1). Once fixed by moving a piece of the code into the right place, loading eventually begins, only to crash, however, halfway through the process, displaying another glitch leading to another segment of code waiting to be repaired. And this is only the beginning of both the game and its unusual occurrences.

As a highly self-reflexive game about a game (possessed, by the way, by Lucifer himself, who appears to be the programmer of Pony Island), *Pony Island* throughout its course of around two hours of gameplay playfully draws attention to computer games' condition as computational artifacts, pieces of software meticulously crafted to provide engaging and often tightly controlled experiences of virtual systems and worlds that generally hide their fundamental technical composition, as the game foregrounds this layer that usually remains carefully concealed. In the following, I argue that, in doing so, *Pony Island* creates an experience of what Galloway calls "countergaming" (*Gaming* 109).

Countergaming

In the final essay of his book *Gaming: Essays on Algorithmic Culture*, Galloway develops the idea of "countergaming," drawing on Peter Wollen's analysis of Jean-Luc Godard's filmic work and on his "seven theses on countercinema" (109). For Galloway, countergaming embodies a movement against conventional forms of gaming, a movement which, through its radical break with established formal conventions, is capable of developing a critical potential lacking in mainstream forms of gaming. The ruptures he identifies and envisions would open spaces for critical reflection and intervention seldom found in commercial games. Galloway develops six theses outlining the strategies countergaming adopts as its core aesthetics in contrast to conventional gaming:

1. *Transparency versus foregrounding*. (Removing the apparatus from the image versus pure interplay of graphics apparatus or code displayed without representational imagery.)
2. *Gameplay versus aestheticism*. (Narrative gameplay based on a coherent rule set versus modernist formal experiments.)
3. *Representational modeling versus visual artifacts*. (Mimetic modeling of objects versus glitches and other unexpected products of the graphics engine.)
4. *Natural physics versus invented physics*. (Newtonian laws of motion, ray tracing, collisions, etc., versus incoherent physical laws and relationships.)
5. *Interactivity versus noncorrespondence*. (Instant, predictable linkage between controller input and gameplay versus barriers between controller input and gameplay.) [...]
6. *Gamic action versus radical action*. (Conventional gaming poetics versus alternative modes of gameplay.) (*Gaming* 124-25)

All of these points will be referenced during my analysis of Pony Island, but first, it is worth noting that at the time of Galloway's writing in 2006, he only registered existent countergaming in modifications of commercial games that essentially took all gameplay out of them. These works reverted back to noninteractive, purely audiovisual forms, with the artists merely using the games as their raw material, thus resulting in pieces that could no longer be called games by common definitions. The countergaming Galloway hoped would emerge eventually, however,

would "be a realization of countergaming *as* gaming, just as Godard was a realization of countercinema *as* cinema" (*Gaming* 126). *Pony Island*, as I will show in the following paragraphs, lives up to this vision.

Pony Island as Countergaming

Let us reenter *Pony Island* at the moment where we left it earlier and we can already see how it realizes Galloway's first thesis. By confronting the player with the code of Pony Island, *Pony Island* immediately foregrounds computer games' status as software. Without code, no software; without syntactically complete code, no process; without process no game—as most ludologists would agree (cf. Aarseth). Of course, the code players get to see (and fix) is no real code, only a highly simplified and stylized symbolic representation of it; this, in turn, only becomes possible through the trick of embedding a game within the game and, thus, positioning the player on a meta-level with an outside perspective at the same time as she is implicated in the gameplay herself. Its purely symbolic nature notwithstanding, however, the code players engage with certainly displays basic programming structures like the loop and principles like the syntax error. Turning the computer game as an algorithmic artifact inside out within its first few minutes of play and throughout much of the rest of the game experience, then, *Pony Island* draws attention to a basic truth, not only of gaming but of much of the computer age in general: essentially, what we are doing is nothing but interacting with computational algorithms, specified in code, manipulating the state of a machine.

Pony Island's visual aesthetics are worth some consideration here, too. On the one hand, its simple graphics can be partly attributed to the circumstances of its production—a game produced within an extremely short span of time during a game jam cannot but rely on somewhat basic aesthetics to be realized at all. On the other hand, the game seems to embrace these formal limitations to create a certain, perhaps nostalgic, atmosphere by consciously turning them into an easily recognizable retro look. This design choice, in turn, entails consequences for the gameplay experience. Principally, it amplifies the disruptive effects of the game because it augments the initial homely feel of the game. This looks and feels like the type of game the player has encountered count-

less times before, which makes it all the more estranging when she finds it to be *unlike* all the others, as assumed conventions, implied by the game's visual aesthetics, gradually fall apart. Familiarity is instilled only to be negated and turned into its antithesis. There is, nonetheless, some air of nostalgia that lingers in the game, especially when it references older forms or versions of certain types of software. This becomes most apparent during a passage when the player has to go through different renditions of Pony Island in classic genres like the text adventure. In relation to the code-fixing challenges, however, the notion of nostalgia carries an additional meaning. It harks back to a time, particularly the 1980s, when code was both more accessible, due its comparative simplicity, and often openly shared, especially with games, which is not usually the case today—the recently reemerging open-source movement notwithstanding. *Pony Island* thus plays out the conditions of its own historical context with a sense of longing for the bygone era referenced by its visual style.

Besides fixing the start menu in the beginning of *Pony Island*, the player repeatedly needs to navigate various other menus during the course of the game. This goes so far that, after Pony Island crashes entirely at one point, the player finds herself navigating a desktop screen (reminiscent of early versions of Microsoft's *Windows*), communicating with a character called H0peles$0ul (who appears to be trapped in the program) through an instant messaging program, and accessing and manipulating various files through shortcuts found on the desktop to help the prisoner. Set against and alternated with conventional game mechanics like side-scrolling jump-and-run (navigating the player's pony over sets of hurdles), which is the dominant mechanic of Pony Island, and puzzle (organizing elements into a functioning order of succession), these episodes constitute "formal experiments" in line with Galloway's second thesis (*Gaming* 125). *Pony Island* constantly disrupts player expectations and breaks with gameplay conventions; modes of gameplay switch unexpectedly, leaving players confused, and Pony Island breaks down, which constitutes a situation computer games would generally try to avoid at all cost. In doing so, the gameplay experience of *Pony Island* foregoes the often-desired transparency or immediacy and effectively foregrounds the human-computer interface in two of its most common forms: the desktop of a mainstream operating system and the menu of selectable options. Especially through the latter, the game

also draws attention to the logic of selection, as the gameplay in these moments revolves around selecting and sometimes rearranging elements from a set of possible options to progress in the game.

One of the most compelling moments of *Pony Island* is a good illustration of this. At some point during the first half of its gameplay, the game confronts the player with an unbeatable challenge: her pony is attacked by a group of demonic heads closing in, which cannot be evaded, thus leading to a reset to the beginning of the level and a loss of one of the pony's lives, ultimately resulting in a "game over." The only way to beat this level and advance further in Pony Island (or *Pony Island*, for that matter) is finding a secret menu that unlocks a new option. After fixing another collapsed options menu, the player needs to find her way through the successive menus of "advanced options" → "advanced gameplay" → "old prototype options" → "really useless options" → "there are better options than this" → and, finally, the mysterious "pony salvation options." In the latter, the player finds a box to activate an option called "pony lasers," which promises progress at last. However, upon moving her mouse cursor toward it—certain that, in a moment, she will blast away those annoying, grotesque faces—Lucifer shrinks the box and pulls it away from wherever the cursor is moved. Once again, the player is required to find a graphical glitch that grants her access to the code, of which she then has to manipulate several parts to finally enable the lasers that allow the pony to eliminate the demonic heads and the player to proceed to the next level.

This short section of the game alone embodies most of Galloway's theses. The apparatus is foregrounded once more in the player's engagement with the game-within-the-game's code. Experimental gameplay emerges from the impossibility of making it past the demonic heads with skillful eye-hand coordination only and the subsequent need to navigate the game's menus instead, in order to unlock a new option. Graphical glitches feature prominently both as access points to the code and in the representation of the pony's lasers, which consist of randomly changing series of zeroes and ones. And the beginning of this part, which starts when Lucifer moves a non-respondent mouse cursor across the desktop to restart the game with the newly implemented challenge, constitutes a disconcerting moment of no control for the player, while the evasive "pony lasers" box instantiates a barrier between input and gameplay. Beyond pushing against mainstream game conventions, *Pony Island*'s

countergaming also highlights a fundamental characteristic of the computer age: control is ultimately always situated in the technology and its algorithms, not the user (cf. Pias 197).[12] Only by enabling particular options, which are themselves inscribed in, and thus determined by, the program, the user is allowed, or even enabled, to act in a way that allows her to progress. The code giveth and the code taketh away, thus goes the gospel of the algorithmic age.

A moment in *Pony Island* that is particularly illustrative of this occurs when the player communicates with Lucifer via the in-game messenger program (see Fig. 2). Lucifer informs the player that he has made some changes to the game (the unavoidable demonic heads mentioned before) and invites her to try it again. Then he commands the player to type "yes master" and something peculiar happens: regardless of which letter keys the player presses, she will, in fact, always type "yes master," only to suddenly see her mouse cursor automatically move

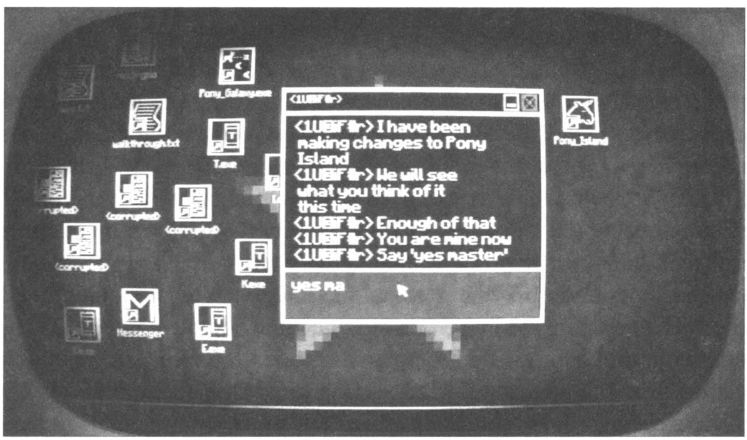

Figure 2: The player chats with Lucifer.

[12] As several new media theorists have shown—especially Chun, in *Control and Freedom: Power and Paranoia in the Age of Fiber Optics*, and Galloway, in *Protocol: How Control Exists after Decentralization*—particularly the protocols enabling and maintaining the Internet as we know it, such as HTML and TCP/IP, are essentially tantamount to control as described by Gilles Deleuze in his "Postscript on the Societies of Control."

toward the icon that restarts Pony Island (now including the demonic heads relentlessly attacking the pony). And while the game still somewhat responds to the player's mouse input, she cannot actually control and stop the cursor from clicking on the icon and restarting the game.

Occasionally, *Pony Island*, goes even further and suggests a breaking down of the line that separates what belongs to the realm of the game and what is outside of it. At one point, Asmodeus, one of the demons guarding the game's core files from deletion—which is needed to free H0peles$0ul—seemingly causes the entire game to crash, resulting in a *Windows* crash notification that only upon a closer look turns out to be fake (see Fig. 3).[13] Another time, the player receives a chat notification from one of their Steam contacts from outside the game, who comments on the player playing the game while she plays it—again, only an illusion, a clever sleight of hand. Moments like these serve to blur (or even dissolve), again and again, the boundary between in-game and out-game reality. That such a clear distinction is hardly tenable in the first place has been demonstrated repeatedly. Whether Mia Consalvo's critique "There Is no Magic Circle," refuting of one of the theoretical pillars of early game studies, Johan Huizinga's idea of the "magic circle" (Huizinga 10) that separates a game spatially and conceptually from everyday reality through its rules; or actor-network-theoretical accounts like Seth Giddings's "Playing with Non-Humans: Digital Games as Technocultural Form" and Mark Cypher and Ingrid Richardson's "An Actor-Network Approach to Games and Virtual Environments," which emphasize the (undeniably real) nonhuman, technological agency at work in computer games and other digital environments; these games are clearly not closed, self-sufficient experiences. Instead, in-game and out-game reality are always closely and dynamically entangled with one another in various ways. Computer games, then, are clearly much more than just "*half-real*" (Juul 196), and so is, without a doubt, any form of software technology; with its countergaming aesthetics, *Pony Island* makes this apparent throughout.

[13] Since the design of the notification emulates the look of *Windows 7*, users of this version are more likely to be fooled into believing *Pony Island*'s conceit than are those who use other versions or even an entirely different operating system. For Mac and Linux users, for example, this moment clearly has little disconcerting effect.

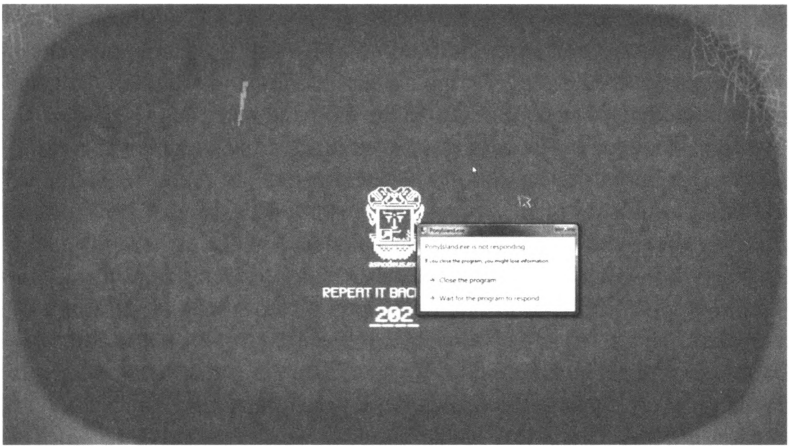

Figure 3: The game seemingly crashes.

Ultimately, *Pony Island* features all criteria of countergaming outlined by Galloway except invented physics (this island, *alas*, remains a thoroughly Newtonian one). It does not sprinkle them in occasionally and for mere effect, as do many commercial games,[14] but employs them consistently, as its guiding principle, its fundamental aesthetics, throughout its entire experience. All this amounts not only to the radical action Galloway calls for, "a critique of gameplay itself" (*Gaming* 125); in the way *Pony Island* realizes Galloway's idea of countergaming it also expresses a larger phenomenon of the modernity we live in. Relating player agency back to program code and thus coupling mastery of the game with control of its underlying mechanisms, *Pony Island* foregrounds the computational processes—which are themselves always already expressive, as, for example, Bogost and Wardrip-Fruin contend—usually concealed in other games and many other twenty-first-century human practices. As the game mediates these in a playable form, *Pony Island* succeeds in creating a moment of "countergaming *as*

[14] Consider here especially Galloway's discussion of an episode in the game *Metal Gear Solid*, when "in the most grievous violation of diegetic illusion, the player is required physically to move the game controller from port one to port two on the console in order to defeat Mantis [one of the major antagonist's in the game]" (*Gaming* 35).

gaming" as envisioned by Galloway (*Gaming* 126). More importantly, in doing so, it urges players to consider the invisible, or unperceived, but decisive informatic structures that define not only computer games but the algorithmic age in general. At a time when the use of new media, to invoke Chun again, has become "habitual" (*Updating* 1), users often only become conscious of their interactions with software technology—and, consequently, the nature of such an object—when it breaks, when it does not work as it is supposed to, thus disrupting the habit. This phenomenon, of course, is something that similarly applies to all tools and technologies used by humans; in *Sein und Zeit*, Martin Heidegger describes this as a transformation from a state of "Zuhandenheit" (readiness-to-hand) to "Unzuhandenheit" (un-readiness-to-hand), or even mere "Vorhandenheit" (presence-at-hand) (*Sein* 73).[15] Likewise, Bruno Latour addresses this logic in *Reassembling the Social: An Introduction to Actor-Network-Theory* when he discusses the occasions and strategies that render the otherwise invisible agencies that make objects do things visible again (80-81); in Pony Island, this occurs through both "*distance*" (80), at least as long as the player does not have a working knowledge of coding, and "breakdowns" (81), as I discussed in the preceding paragraphs. In terms of the Wachowskis' 1999 science-fiction film, these are the moments when we see the cracks in the Matrix and, if only fleetingly, become aware of it (and its workings). In constantly breaking the illusion, *Pony Island* urges its players to consider the nature of the computational technologies they use habitually all the time without ever perceiving them as what they are. The game disrupts the familiar and expected, and it creates moments of wonder which make the player halt and, perhaps, lead her to view the thing as what it is rather than what it invites her to imagine. In doing so, *Pony Island* turns itself inside out, urging players to consider and take seriously Manovich's claim quoted at the beginning of this essay, as if it, too, were saying that, perhaps, there is only software.

[15] The English terms are taken from the translation by John Macquarrie and Edward Robinson (Heidegger, *Being* 102-03).

Works Cited

Aarseth, Espen. "Computer Game Studies, Year One." *Game Studies*, vol. 1, no. 1, 2001. gamestudies.org/0101/editorial.html. Accessed 10 Oct. 2017.

Andrejevic, Mark. "The Big Data Divide." *International Journal of Communication*, vol. 8, 2014, pp. 1673–89. ijoc.org/index.php/ijoc/article/view/2161/1163. Accessed 1 Nov. 2017.

Bogost, Ian. *Persuasive Games: The Expressive Power of Videogames*. MIT P, 2007.

Chun, Wendy Hui Kyong. *Control and Freedom: Power and Paranoia in the Age of Fiber Optics*. MIT P, 2006.

---. *Programmed Visions: Software and Memory*. MIT P, 2011.

---. *Updating to Remain the Same: Habitual New Media*. MIT P, 2016.

Confessore, Nicholas, and Danny Hakim. "Data Firm Says 'Secret Sauce' Aided Trump; Many Scoff." *New York Times*, 6 Mar 2017, nyti.ms/2zg0QS8. Accessed 5 Sept. 2017.

Consalvo, Mia. "There Is No Magic Circle." *Games and Culture*, vol. 4, no. 4, 2009, pp. 408–17, *Sage*, DOI: 10.1177/1555412009343575, accessed 2 Oct. 2017.

Cypher, Mark, and Ingrid Richardson. "An Actor-Network Approach to Games and Virtual Environments." *CyberGames '06: Proceedings of the 2006 International Conference on Game Research and Development*, Perth, 2006, pp. 254–59, dl.acm.org/citation.cfm?id=1234382. Accessed 2 Oct. 2017.

Deleuze, Gilles. "Postscript on the Societies of Control." *October*, vol. 59, Winter, 1992, pp. 3–7, *JSTOR*, www.jstor.org/stable/778828. Accessed 12 Aug. 2016.

else Heart.Break(). Erik Svedäng et al., 2015. PC Game.

Galloway, Alexander R. *Gaming: Essays on Algorithmic Culture*. U of Minnesota P, 2006.

---. *The Interface Effect*. Polity, 2012.

---. *Protocol: How Control Exists after Decentralization*. MIT P, 2004.

"Game Jam." *Wikipedia*, 2017, en.wikipedia.org/wiki/Game_jam. Accessed 4 Oct. 2017.

Giddings, Seth. "Playing with Non-Humans: Digital Games as Techno-Cultural Form." *Proceedings of DiGRA 2005 Conference: Changing Views—Worlds in Play*, Vancouver, 2005, eprints.uwe.ac.uk/15062/. Accessed 14 Aug. 2016.

Golumbia, David. *The Cultural Logic of Computation*. Harvard UP, 2009.

Grubb, Jeff. "Valve Won't Manually Curate Steam because It Dominates PC Gaming." *VentureBeat*, 13 Feb. 2017, www.venturebeat.com/2017/02/13/valve-wont-manually-curate-steam-because-it-dominates-pc-gaming/. Accessed 5 Sept. 2017.

Hansen, Mark B. N. *New Philosophy for New Media*. MIT P, 2004.

Heidegger, Martin. *Being and Time*. 1927. Translated by John Macquarrie and Edward Robinson, Harper, 1962.

---. *Sein und Zeit*. 1927. 16th ed., Max Niemeyer, 1986.

Huizinga, Johan. *Homo Ludens: A Study of the Play-Element in Culture*. 1949. Routledge, 1980.

Juul, Jesper. *Half-Real: Video Games Between Real Rules and Fictional Worlds*. 2005. MIT P, 2011.

Kittler, Friedrich. "There Is No Software." *CTheory*, 18 Oct. 1995, www.ctheory.net/articles.aspx?id=74. Originally published in *Stanford Literature Review*, vol. 9, no. 1, 1992, pp. 81-90, accessed 4 Sept. 2017.

Latour, Bruno. *Reassembling the Social: An Introduction to Actor-Network-Theory*. Oxford UP, 2007.

Manovich, Lev. *The Language of New Media*. MIT P, 2001.

---. *Software Takes Command*. Bloomsbury, 2013.

Metal Gear Solid. Konami, 1998. PlayStation Game.

Pasquale, Frank. *The Black Box Society: The Secret Algorithms That Control Money and Information*. Harvard UP, 2015.

Pias, Claus. *Computer Game Worlds*. 2002. Translated by Valentin Pakis, U of Chicago P, 2017.

Pony Island. Daniel Mullins Games, 2016. PC Game.

Taggart, Kendall. "The Truth About the Trump Data Team That People Are Freaking Out About." *Buzzfeed News*, 16 Feb. 2017, www.buzzfeed.com/kendalltaggart/the-truth-about-the-trump-data-team-that-people-are-freaking?utm_term=.sfR1jEnDe#.eaNE7JPK3. Accessed 5 Sept. 2017.

Vaidhyanathan, Siva. *The Googlization of Everything: (And Why We Should Worry)*. U of California P, 2012.

The Wachowski Brothers. *The Matrix*. Warner, 1999.

Wardrip-Fruin, Noah. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. MIT P, 2012.

Weiser, Mark. "The Computer for the 21st Century." *Scientific American*, vol. 265, no. 3, 1991, pp. 94–105, *JSTOR*, www.jstor.org/stable/24938718. Accessed 4 Sept. 2017.

Winner, Langdon. "Do Artifacts Have Politics?" *Daedalus*, vol. 109, no. 1, 1980, pp. 121-36, *JSTOR*, www.jstor.org/stable/20024652. Accessed 9 Apr. 2017.