(Queer) Algorithmic Ecology

The Great Opening Up of Nature to All Mobs

Amanda Phillips

In 1997, Electronic Music Foundation President Joel Chadabe described electricity's contribution to composition as "the great opening up of music to all sound." This sentiment resonates with other mid–90's techno-utopian figurations of the digital as an escape from the constraints of the physical world—a medium of endless possibility. Though such optimism has waned in the intervening decades, computational representations of natural environments open up new orientations toward nature that easily read as fantasies of capitalist frontier expansionism enabled by the reduced material constraints of the digital. The digital worlds of *Minecraft* (Mojang, 2011) "open up" both literally and figuratively, with algorithmically generated and procedurally expanding environments that are theoretically infinite but bound by the operational constraints of the software. *Minecraft*'s unfolding nature is also reflected in the game's crafting system, in which resources open up into increasingly small parts of themselves: one block of wood, for example, will yield 4 wooden planks, which can be crafted into 8 sticks total, and so on. While it is true that *Minecraft*'s generated landscapes are optimized for the calculation of yield values and have spawned countless user-generated colonial fantasies, their stunning empty vistas also substitute spawning for reproduction, versioning for evolution, and drop probabilities for history in ways that defer neat categorization. Oriented away from heteronormative and capitalist temporalities, this algorithmic ecology opens nature up not only to human occupation for productivity, but to the trivial, nonproductive wanderings of automated digital inhabitants. *Minecraft* conceals within its thoroughly exploitable mathematical models queer possibilities of reproduction, temporality, and occupation of space that move beyond the purposes for which they have been coopted.

Queering Platform Studies

The emergence of platform studies as a discipline has encouraged scholars to tease out "the connection between technical specifics and culture" (Bogost and Monfort, 2009) and offers a good starting point for thinking about the queer modalities of *Minecraft*. However, rather than appeal to the existing canon of platform studies, this chapter adds some new critics to the conversation. Computational systems operate according to logics that are normative by definition in order to guarantee a seamless modular integration with other components of the system, and feminist thinkers like Tara McPherson (2012), Wendy Chun (2007, 2011), and Lisa Nakamura (2002, 2008) have traced how these normative logics emerge from and contribute to hegemonic systems of power, containing difference within structures that are easily manageable by automated systems. These scholars map out an approach to a kind of platform studies that attends to the cultural power embedded within technical systems, from fiber optic networks or computer architecture (Chun) to user interfaces on the Internet (Nakamura). McPherson (2012) charts how the modular logics of operating systems like UNIX, which emerged alongside the Civil Rights movements in the mid-twentieth century, are part of a larger institutional trend that organizes both knowledge and populations in such a way that obscures the relationship between surface and structure. She argues that "the fragmentary knowledges encouraged by many forms and experiences of the digital neatly parallel the lenticular logics which underwrite the covert racism endemic to our times" (p. 33). For McPherson, these "forms and experiences" begin at the level of code—in this example, at the moment in which modularity became a core principle of UNIX computing—and she calls for increased attention to the mutual imbrication of culture and platform.

Kara Keeling (2014) mods McPherson's project further into what she calls "Queer OS," an approach to feminist platform studies that additionally "understands *queer* as naming an orientation toward various and shifting aspects of existing reality and the social norms they govern, such that it makes available pressing questions about, eccentric and/or unexpected relationships in, and possibly alternatives to those social norms" (p. 153). This turn toward queerness as an analytic for digital and other computational media opens up new possibilities for interpretation, which some scholars are already shaping. Jacob Gaboury (2013), for example, writes a queer history of computing that explores how the sexuality of some of the foundational mathematicians and programmers in computing history might have shaped their "mode of being in the world," and as a result, their scholarship. At one point, Gaboury (2013b) suggests that since Alan Turing's concept of the computable number, a foundational principle in the development of computing, requires the excision of entire classes of numbers from a system, we might look to locate queer com-

puting beyond these constraints, in the computer's excess. While it is important to think through speculative technologies and develop new methods to open up more possibilities for computational expression, it is equally valuable to locate queerness in existing applications of computational logic. Following Gaboury's associative stitching together of the history of computing, the personal lives of some of its forefathers, and the material instantiations of their research, this exploration of a different kind of queer computing opens up with the algorithm, the formalization of which was one of Turing's contributions to computing history.

Turing, Gaboury points out, found no algorithm for creativity or intuition. And yet creative algorithms of a kind do exist, even if they operate within the bounds of calculable numbers and concrete tasks. In fact, Gaboury (2013c) includes in his queer history of computing an anecdote about Christopher Strachey, a gay programmer working at Manchester University in the 1950s, who designed a program that would algorithmically generate love letters dripping in "melodramatic Victorian overtones" and then posted them anonymously on the department's bulletin board for his colleagues to find.[1] This type of creative production hearkens back to analog forms—a relevant contemporary example would be the paper computational experiments of the Oulipo—but automating the process allows creative resources to be used in different ways. Randomized level generation in video games, for example, dates back to the 1980s and has been implemented in landmark titles such as *Rogue* (Toy & Whichman, 1980) or *Diablo II* (Blizzard, 2000). As a design strategy, it decreases the need for specific game assets to be stored, making it a desirable way to reduce the size of a game without sacrificing playable content, which partially accounts for its popularity in older titles that needed to use storage space more judiciously.

Ashmore and Nitsche (2007) suggest that the contemporary resurgence of procedural generation began as a result of increasing costs of production coupled with demand for more content, as well as high-profile games like *Spore* (Maxis, 2008) that renewed interest in the quirky combinations enabled by computerized creativity. *Spore*'s procedural animation techniques were a brief Internet sensation, with users creating bizarre and obscene creatures and posting videos of the suggestive ways that the software made them move. More recently, big-budget titles like *BioShock Infinite* (Irrational Games, 2013) utilized procedural generation during development to ease the task of building highly detailed, expansive cities (McMahon, 2013), while indie darlings like *Spelunky* (Yu & Hull, 2009) and *FTL: Faster Than Light* (Subset Games, 2012) have taken advantage of the efficiency that procedural generation provides in-game to allow for a variety of experiences across multiple replays and player experiences without requiring massive development resources and time. According to critic Claire Hoskins (2013), procedural generation might pro-

vide a new frontier for developers as the improvement curve of photorealistic graphics reaches its limits.

Some video game critics, such as *Indie Statik*'s Chris Priestman (2013), credit the recent boom in procedurally generated content to *Minecraft* itself, which is perhaps the most successful contemporary game to leverage the power and wonder of procedural level generation. A game that encourages the gamer to move outward into continuously changing territory benefits immensely from the sense of discovery and serendipity that random content creation provides. However, Priestman suggests that it is multiplayer activity that sustains *Minecraft* as a platform, that procedural generation is a mere "numbers game" providing "quantity over quality." And yet, the game's procedurally generated environments underwrite both single and multiplayer experiences and in fact provide the drive for much of its sociability: for example, in this author's experience, part of the pleasure of playing with others in *Minecraft* is in dividing up labor to find, harvest, and build with resources more efficiently in a vast world. The numbers game in *Minecraft* is, in fact, the heart of the platform, and its mixing with ecological aesthetics creates a strange form of nature—an algorithmic ecology—that is simultaneously ripe for capitalist exploitation and full of alternative queer embodiments and relations. It is in the spirit of Queer OS that we proceed under the hood to tease these out.

Procedural Generation and Algorithmic Ecology

What I call "algorithmic ecology" in *Minecraft* gestures toward the fact that automated computational processes govern nature in the game and in many ways, thanks to aesthetic design and game mechanics, subsume the ecological within the mathematical. This is not necessarily an anti-ecological gesture. In fact, Timothy Morton's (2007) suggestion to remove nature from ecological thinking might find unique application in looking at algorithmic ecology; while the aesthetic veneer of nature is certainly interesting in an analysis of *Minecraft*, focusing on its underlying processes and relationships with the user might be a way around the obscuring "view" of nature beyond which Morton wishes to move (2). This strategy, conveniently, also resonates with the methodologies of platform studies. By investigating the technological structures (or, in this case, the technical design decisions) that underwrite the aesthetics of this particular software platform, a more complex understanding of both will emerge.

The processes that fuel *Minecraft*'s terrain generation establish the foundation for an ever-expanding ecology that can be shaped by the gamer. Although math is ultimately the foundation of all digital games, there is something about *Minecraft*'s open dedication to cubes and crafting recipes that help

bring its abstract algorithms onto the surface. Aesthetically, the world is an homage to the simplest Cartesian formulation of space, invoking the graphs and grids of childhood mathematical training. Procedurally, the symbolic weight of the algorithm exists in the crafting function itself: the drive to use resources efficiently and to create ever more complex objects in the game world. Break trees into logs into planks into sticks to recombine them again as useful tools. Use tools to gather increasingly rare resources to recombine into new tools and objects. Community members reinforce this drive with exhaustive guides that show new users how to make the most of their time and materials with elaborate formulae. *Minecraft* parlance is littered with stacks and percentages and altitudes and probabilities, and the community has experimented with the game to the point that while they may not understand everything about the code under the hood (although many of them do), they certainly have calculated its numerical impact on in-game activity.

Each new *Minecraft* world is generated using calculations driven by a "world seed," an integer converted from letters, words, or numbers that can be set by the player upon beginning the game. If the player chooses not to seed the world, the game uses a seed based on the date and time of world creation.[2] The world seed ensures a significant degree of uniqueness amongst *Minecraft* worlds, but also allows users to share with one another if particularly compelling territories emerge as a result. *Minecraft* initially spawns the player and creates a finite "chunk" of land around them, populating the world with structures like caves, lakes, ravines, and others according to the world seed. Chunks are one of the organizing units in *Minecraft*: where a block is its most basic physical form, a chunk is a 16x16x256-block pillar of land that is generated around the player. Each chunk has its own biome assignment, which dictates animal and plant life, geological structures, and local weather conditions. *Minecraft* worlds are things of voxelated beauty, with complicated environmental features emerging as generated geographical structures collide, overwrite, and blend with one other.

As a player moves through space, the program continuously generates new terrain based on the seed and current generation algorithm, which has historically been updated to increase variety and smooth out the transitions between biomes. There is some debate about the actual rendering limits of a *Minecraft* world. The game's creator, Markus Persson, went on record in 2011 to clarify that there was no hard limit to the world's size, but due to rounding errors and other mathematical effects, movement, structures, objects, and other aspects of the game may become distorted and behave erratically the farther the gamer moves from the world's initial spawn point. Currently, there is an invisible wall 30,000,000 meters in any direction from the world's initial spawn location that prevents a gamer from exploring further. In fact, avatar location is a key variable in many of the game's calculations, used to determine not only

expected things like draw distances for graphics and enemy line-of-sight but also which portions of the world are loaded into memory and actively updated. "Mobs," the mobile units (mostly animals) that exist within the game, also turn off and on based on player location, and many will disappear if the player moves too far away. This operational radius is necessary for system performance in a theoretically infinite world, but it has important implications on play.

Despite the proximate relationship between gamer and environment, the game environment also operates independently of human interference. *Minecraft* isn't a persistent world that continues to function after users exit the system, but its ecology runs off of automated algorithms that do not require user input. Alexander Galloway (2006) describes a moment of gamic action he calls "pure process," in which the machine would operate on its own in the game world without input from the operator. In some games, when the operator ceases to act, the machine reaches a state of ambience, "slowly walking in place, shifting from side to side and back again to the center. It is running, playing itself, perhaps" (10). Standing still in *Minecraft* reveals these moments: animals spawn out of view and wander around, bleating and clucking in the distance. They might hunt each other or fly into a lava fall and burn up. Any crops that were planted will grow and mature. The sun rises and falls. Weather continues to change.[3]

Galloway's work also emphasizes the fact that operator and machine play the game together, something that the operational radius mechanic of

Figure 1. **A waterfall and lava fall side-by-side (Minecraft ®/TM & © 2009–2013 Mojang / Notch).**

Minecraft deftly illustrates. In most games, this mechanism is a fairly obvious tit-for-tat of combat or other continuous stream of engagement. For *Minecraft*, operator and machine share a subtle copresence based on location, an ecological simulation that underscores both the embeddedness of the human in an environmental system as well as their irrelevance to its mundane operations. In writing about the possibility for a game that models ecological principles, Alenda Chang (2011) observes some of the fundamental mistakes that video game designers make with digital environments: the first of these is that they relegate environments to the background. As the operational radius and procedural drive to craft illustrates, there is nothing about the environment of *Minecraft* that keeps it in the background. Interaction with the world is the entire point of the game in any of its player-selected modes, whether that it comes in the form of exploration in Peaceful difficulty, daring combat in Adventure Mode, or unrestricted building in Creative Mode. [4]

Chang's second video game ecological design error is the stereotyped environment. Gaming classics like *Super Mario Bros.* (Nintendo R&D4, 1985) or *The Legend of Zelda* (Nintendo R&D4, 1986) have long-held traditions of environment types that repeat in every installment of the series: grassy overworlds, dark caves, lava fields, oceans and lakes. While some of the biomes in *Minecraft* may draw on common environment types like forests, deserts, and jungles, it also includes less well-known varieties like taiga and tundra and makes nontrivial gestures at localizing weather, animal, and plant life, achieving a kind of environmental realism that Chang finds desirable. *Minecraft* bestiaries, for example, include information on the biomes that spawn particular species so that less common animals such as ocelots may be more easily found and tamed. Indeed, some *Minecraft* users share world seeds that generate particularly rare biomes at the spawn; the interest in these may range from aesthetics to rare resources, but it is clear that environmental variety and specificity in the game is an important feature.

By these introductory criteria, *Minecraft*'s algorithms seem to model ecology in a meaningful way. However, Chang's hope for environmental realism in video games is also connected to environmentalist politics. She asks, "Why must games replicate the same kind of costly obliviousness we see everyday in the non-digital world" when both culture and game design might benefit from seeing the consequences of environmental exploitation played out to extremes (61)? This leads to her third criterion for the failed ecological game: simplistic relations between gamer and environment, such as the "extraction and use of natural resources" (58). There is much to fail *Minecraft* on that score, and it is directly related to its algorithmic qualities. Every block is a standing reserve of materials that can be gathered and opened up, sometimes exponentially, for the purposes of production, and the continuously expanding environment gestures toward plentitude rather than consequences.

But that's not the only thing gamers can do with this particular numbers game, and that's certainly not what the game's engine, when left to its own devices, would do with itself. While it is not my intent to bracket important political critiques for the sake of a niche argument, it is important to consider that *Minecraft* as a platform makes it quite easy to turn away from the capitalist expansion narrative that so many find on its surface. In fact, the aesthetic beauty of the game's algorithmic ecology is found in its vast empty vistas that are populated by precious few sentient creatures and no industrial communities. There is plenty of space, literally and figuratively, for living life with minimal impact on the natural world, coexisting rather than conquering. This is not a sterile nature by any means, but it is not one built solely for the taking.

Queer Time, Queer Places

In *The Queer Art of Failure*, Judith Halberstam (2011) writes about the "Pixarvolt" genre of animated films, which deal on the surface with narratives of capitalist individualism but ultimately "[serve] as a gateway to intricate stories of collective action, anticapitalist critique, group bonding, and alternative imaginings of community, space, embodiment, and responsibility" (43–4). While the narrative content of these films clearly contribute to Halberstam's analysis, and while not all computer animated films have queer revolutionary potential, there are some structural aspects of computer animation that seem to distinguish the genre from traditional hand-drawn cartoons. Halberstam discusses, for example, what many others have observed about the content of Pixar's films: the narrative follows the technological breakthrough. Shiny surfaces generated *Toy Story*. Hair textures generated *Monsters, Inc.* In Halberstam's examples, the technological breakthrough was the animation of "numbers, groups, the multitude. Once you have an animation technique for the crowd, you need narratives about crowds, you need to animate the story line of the many and downplay the story line of the exception" (176). What follows are stories about ants, bees, and fish that Halberstam finds particularly compelling.

Minecraft's technological breakthrough is algorithmic ecology, the continuously generating, automated world that seems on the surface to exist purely for exploitation. What follows this innovation, however, is not a compelling anticapitalist narrative—or any narrative at all. Algorithmic ecology in *Minecraft* produces a very queer form of nature that proceeds according to its own nonreproductive rhythms.

Most discussions of queer ecologies begin with animal embodiments, from "homo or nonrepro queer penguins" (Halberstam 2011, 41) to intersex deer (Morton 2010, 276). Because the discourse of "natural" sexual orientation

and gender identity drives much homo- and transphobic rhetoric, activists have pointed to the existence of queer animals to counter these claims and to generate discussions about essentialist and constructivist worldviews. *Minecraft* has its own variety of queer animal life, whose primary mode of reproduction is computer spawning. Most mobs spring into the world fully formed based on the game's algorithm for creating them: first, the player must be close enough for a chunk to be active. Next, the game checks that the mob cap hasn't been reached. Next, it attempts to spawn on a particular location, and will fail or succeed depending on the conditions of the terrain it randomly selects. Then the process repeats. The rules and caps differ based on the type of mob, and some mobs only spawn from physical spawn blocks located in the environment, but the result is fairly straightforward: baby animals do not occur "naturally" in the game.[5]

This does not mean that there is no sexual reproduction, but even breeding has some very queer features. The operator can initiate breeding in any two domesticated animals of the same species by feeding them their favorite food, a rather curious design decision that associates reproduction with nonsexual desire. Moreover, while the ability to breed gestures toward a heteronormative temporality oriented toward family units and capitalist production, it is quite queer in execution, revealing something about *Minecraft*'s world that is rare for other reproductive simulations: sexual difference does not exist. One can trace the lack of sex back to the initial *Minecraft* player character, which is androgynously blocky and has a skin that can change their outer appearance at the user's will. Unlike many games, *Minecraft* neither asks the gamer to choose a gender for their avatar nor really imposes one upon them.[6] All of the game's cows and chickens produce milk and eggs, respectively, further suggesting that there is no sexual dimorphism in this algorithmic environment. Reproduction is often a limit case in ecological simulations; virtual animal breeding games from Maxis's *SimLife* (1992) to Nintendo's *Pokémon* (Game Freak, 1996) continue to utilize an individual's sex as a major mechanic of reproduction, even when its other natural features are based on fantasy physics.

And yet, animal sex in nature is only part of what constitutes a theory and practice of queer ecology. Morton (2010) suggests complicating or even retiring the concept of the animal entirely, arguing that "queer ecology would go to the end and show how beings exist precisely because they are nothing but relationality, deep down" (277). Echoing work by theorists like Donna Haraway, Morton suggests that queer theory's expertise in questions of fluidity and relations between bodies, temporalities, and spaces, makes it an "intimate" intellectual counterpart of ecology (281). In the first multi-author collection dedicated exclusively to exploring the facets of queer ecologies, Catriona Mortimer-Sandilands and Bruce Erickson (2010) outline "a new practice of ecological knowledges, spaces, and politics that places central attention on

challenging hetero-ecologies from the perspective of non-normative sexual and gender positions" (22). Their collection moves through queer animality,[7] green and pink political movements, and the queer uses of public space,[8] and it makes a compelling case for the various ways in which queer and environmentalist activisms and theories may intersect.

Algorithmic ecology provides yet another site to investigate queer environments beyond ecomimesis and queerness beyond sexual identity, as it enables plenty of nonproductive activities and is a playground to experiment with new ways of being. There is no primary life orientation in *Minecraft*; wandering, hunting, farming, building, mining, fishing, standing still, and more are all equally viable paths.[9] One may scramble to find shelter and hide to survive the first night if they desire, but the ambient electronic and piano music that accompanies the game suggests something quite different than combat for the game's primary orientation. Peaceful difficulty, which despawns hostile mobs in the game, is a simple menu switch away. This difficulty setting gestures toward something of a queer fantasy: imagine if all the hostility in the world could be turned off with the flip of a switch. The ecological utopia opened up by Peaceful difficulty offers an imaginative landscape in which one can literally pick one's own battles, in which violence does not necessarily go hand-in-hand with spatial exploration. The act of disappearing all of one's enemies can also be read as a quintessential conquering fantasy; I choose to read it in a utopian light here because the hostile mobs in *Minecraft* are themselves remnants of a fantasy of settler colonialism: the implacable Other who

***Figure 2*. Sheep, pigs, and chickens wandering on a rocky hillside (Minecraft ®/TM & © 2009–2013 Mojang / Notch).**

must be destroyed as a matter of self-preservation. By despawning (rather than destroying) this myth, which motivates so much violence in society, Peaceful difficulty metaphorically restores the world to a state of peace without fear of difference.

With hostile mobs turned off, one has an abundance of time to appreciate the world's unique rhythms. Very little happens in *Minecraft* without user input or presence after the initial world spawn. This may seem to contradict earlier claims about the game's active ambient state, but the "happening" to which I refer here deals with reproductive futurity, a topic of some interest during queer theory's turn toward temporality. Halberstam (2005) notes in the introduction to *In a Queer Time and Place* that "Queer uses of time and space develop, at least in part, in opposition to the institutions of family, heterosexuality, and reproduction.... If we try to think about queerness as an outcome of strange temporalities, imaginative life schedules, and eccentric economic practices, we detach queerness from sexual identity..." (1). Both the temporal and economic aspects of this argument are relevant here, since it is possible (and common) to play the game without engaging in any (or many) of its productive algorithms and still be a functional part of its ecology. This opening up of queer identity in terms of time might apply to any number of gaming temporalities, very few of which operate according to real world rules, but this is especially so in *Minecraft* time. It is easy to recognize the drive for ecomimesis in the proliferating biomes and weather patterns added to each subsequent *Minecraft* update, but something about the world feels a bit too purposeless to be a useful ecological simulation.

If crops aren't planted, they won't grow. If animals aren't bred manually, they won't reproduce or mature. Ice might freeze and melt, and rain storms will come and go. The sun rises and falls, but time as marked by reproductive life stands still.

And yet, the game plays. The lack of a temporal movement toward the future is connected to the game's lack of a diegetic past. Queer orientations toward time and space in *Minecraft* extend to alternative relationships to history, as well. Though environmental traces, all planted by the spawn algorithms, point toward history on both geological and cultural scales, they are completely decontextualized and give little hope of ever yielding a narrative. Temples, dungeons, and strongholds appear according to how the world seed and algorithms dictate, sometimes located where no gamer will ever find them. Halberstam (2011) discusses this same type of queer haunting—"the empty promises of utopia"—in the context of the photography of queer Spanish artist collective Cabello/Carceller, who documented abandoned spaces such as swimming pools and dance clubs in California. Interested in outlining queer mobilizations of failure to thrive and engage in capitalist futures, Halberstam notes that "the empty pools stand like ruins, abandoned and littered with

leaves and other signs of disuse, and in this ruined state they represent a perversion of desire, the decay of the commodity, the queerness of the disassociation of use from value" (111). While the ruins of *Minecraft* do represent an Orientalist *Indiana Jones* opportunity to retrieve rare items, their existence outside of time—literally as procedurally generated features of the landscape with the same temporality as the frozen animal and plant life—denies any sense of closure or conquering one might get from discovering their diegetic origins. For Halberstam, it is precisely the deferral of narrative that prompts the meditation on queer failure in the photographs of Cabello/Carceller.[10]

Queer Algorithmic Ecologies

We might return here to Priestman's (2013) warning about the tendency of procedural generation to get stuck in its own temporality, to fail where other games may succeed with hand-crafted content. *Minecraft*'s rise to success, in some ways, was in fact predicated on its early cyclical failures: using the paid-alpha/beta release model, it was in an extended period of public iteration that was both quintessentially capitalist and defined by the need for alternative funding models to sustain independent game development. Since its popularity was massive before an official launch version even hit the shelves, it is perhaps no surprise that the resulting play possibilities are quite wide open; here, we see a preview of what developing for the multitude might ultimately mean. If Halberstam saw queer possibilities following logically from computer rendering of crowds, then we might claim that *Minecraft*'s numbers game follows a similar path. A set of algorithms dictating the continual production of an expanding world is unleashed to an exponentially increasing group of users for whom the game is tailored and updated until official release, opening up a queer ecological system frozen in time that appeals to all ages.

So where does this leave us with an algorithmic ecology? We have explored how terrain and mob generation are governed by a complicated set of procedural rules modified by a world seed. We have seen how resources can be taken apart and added up into new formulations. We have even discussed the community's dedication to cracking open and exploiting the game's probabilities in the interest of maximizing profit. And, crucially, none of this is hidden underneath the veneer of realism or narrative like so many other simulations, instead reveling in an aesthetic that hearkens to early childhood math class and building blocks. While algorithmic ecology exists (with similar properties) in other virtual worlds like *Second Life* (Linden Lab, 2003) or *The Elder Scrolls* (Bethesda, 1994–2014) series, it is *Minecraft*'s dedication to space rather than narrative and voxels rather than photorealism that brings out both its ecological and algorithmic properties.

Figure 3. **A sunset in a *Minecraft* field (Minecraft ®/TM & © 2009–2013 Mojang / Notch).**

Minecraft's algorithmic ecology exists in a queer time that is cyclical rather than linear, one that deploys surface structures in place of temporal depth, and which produces queer forms of animal life and endless opportunity to walk hand-in-hand with the game's algorithms as they continuously open the world up and show users its awe-inspiring structures. Sitting still in a cave with bats fluttering around the avatar's head and chickens clucking somewhere up above, it is easy to see how algorithmic ecology is the source of the magic in this place. As a theoretically infinite world with inexhaustible soil and virtually waste-free resource conversion, *Minecraft* may not always provide a lesson in responsible attitudes toward nature. And yet it is more than an exploitative fantasy, full of exciting queer possibilities.

Notes

1. Gaboury traces the efforts of both Noah Wardrip-Fruin and Jeremy Douglass to analyze Strachey's program as a computational curiosity, privileging its form over content in reaching their final conclusions: the love-letter program "is a parody of the process of producing love letters" (Gaboury 2013c). However, he goes further to state that a queer approach must also recognize the critique of normative courtship procedures as well.

2. As the technical details of the game may change with updated versions of the software, some of this information may no longer be accurate. The current version as of this writing is 1.7.5.

3. Of course, *Minecraft* veterans will note that such quiet moments are only possible during daytime conditions on the surface of the world, or while playing in Peaceful difficulty, a point to which we will return later.

4. One can make the argument that the point of any game is to interact with its envi-

ronment, inasmuch as spatiality is a defining quality of digital media. However, Chang's contention specifically engages natural environments.

5. The one exception to this is the Villager, which spawns baby Villagers if there are a sufficient number of doors in its village. Like all animals in *Minecraft*, Villagers are not sexually dimorphic.

6. However, some mods of the game, such as the educational MinecraftEDU, do implement a gender choice in the game.

7. See also Chen (2012).

8. See also Gandy (2012).

9. Here I evoke Sara Ahmed's (2006) discussion of queer ways of being in terms of orientation away from prescribed life paths such as compulsory heterosexuality.

10. An additional reading that warrants further exploration is that these decontextualized ruins represent a disappearing of native peoples such that conquering space becomes a guilt-free activity. My choice to read queer temporality in these ruins rather than ultimate exploitation is not a refusal of this claim. Indeed, such contradictory strains of analysis are the lifeblood of critical discourse.

References

Ahmed, S. (2006). *Queer phenomenology: Orientations, objects, others*. Durham: Duke University Press.

Ashmore, C., & Nitsche, M. (2007). The quest in a generated world. *Situated Play, Proceedings of DiGRA 2007 Conference*, pp. 503–509.

Bethesda Game Studios. (1994–2014). *The Elder Scrolls* [PC game]. Bethesda Games.

Blizzard, North. (2000). *Diablo II* [PC game]. Irvine, CA: Blizzard Entertainment.

Bogost, I., & Monfort, N. (2009). Platform studies: Frequently questioned answers. Paper presented at *Digital Arts and Culture*, Irvine, CA, December. Retrieved from http://www.bogost.com/downloads/bogost_montfort_dac_2009.pdf

Chang, A. (2011). Games as environmental texts. *Qui Parle: Critical Humanities and Social Sciences*, *19* (2), 57–84.

Chen, M. (2012). *Animacies: Biopolitics, racial mattering, and queer affect*. Durham: Duke University Press.

Chun, W. H. K. (2006). *Control and freedom: Power and paranoia in the age of fiber optics*. Cambridge, MA: MIT Press.

Chun, W. H. K. (2011). *Programmed visions: Software and memory*. Cambridge, MA: MIT Press.

Gaboury, J. (2013a, February 19). A queer history of computing. *Rhizome*. Retrieved from http://rhizome.org/editorial/2013/feb/19/queer-computing-1

Gaboury, J. (2013b, March 19). A queer history of computing: Part two. *Rhizome*. Retrieved from http://rhizome.org/editorial/2013/mar/19/queer-computing-2

Gaboury, J. (2013c, April 19). A queer history of computing: Part three. *Rhizome*. Retrieved from http://rhizome.org/editorial/2013/apr/9/queer-history-computing-part-three

Gaboury, J. (2013d, May 6). A queer history of computing: Part four. *Rhizome*. Retrieved from http://rhizome.org/editorial/2013/may/6/queer-history-computing-part-four.

Gaboury, J. (2013e, June 18). A queer history of computing, part five: Messages from the unseen world. *Rhizome*. Retrieved from http://rhizome.org/editorial/2013/jun/18/queer-history-computing-part-five.

Game Freak. (1996). *Pokémon* [PC game]. Nintendo.

Galloway, A. (2006). *Gaming: Essays on algorithmic culture.* Minneapolis: University of Minnesota Press.

Gandy, M. (2012). Queer ecology: Nature, sexuality, and heterotopic alliances. *Environment and Planning D: Society and Space*, 30, 727–747. doi: 10.1068/d10511

Halberstam, J. (2005). *In a queer time and place: Transgender bodies, subcultural lives*. New York: New York University Press.

Halberstam, J. (2011). *The queer art of failure*. Durham: Duke University Press.

Hosking, C. (2013, December 10). Stop dwelling on graphics and embrace procedural generation. *Polygon*. Retrieved from http://www.polygon.com/2013/12/10/5192058/opinion-stop-dwelling-on-graphics-and-embrace-procedural-generation

Irrational Games. (2013). *BioShock Infinite* [PC game]. 2k Games.

Keeling, K. (2014). Queer OS. *Cinema Journal*, *53*(2), 152–7. doi: 10.13353/cj.2014.0004

Linden Lab. (2003). *Second Life* [PC game]. San Francisco, CA: Linden Lab Entertainment.

Maxis. (1992). *SimLife* [PC game]. Maxis Software.

Maxis. (2008). *Spore* [PC game]. Electronic Arts.

McMahon, C. (2013, April 11). *BioShock Infinite*: A city in the sky. *3D Artist Online*. Retrieved from http://www.3dartistonline.com/news/2013/04/bioshock-infinite-a-city-in-the-sky/

McPherson, T. (2012). U.S. operating systems at mid-century: The intertwining of race and UNIX. In L. Nakamura and P. A. Chow-White (Eds.), *Race after the Internet*, 21–37. New York: Routledge.

Mojang. (2011). *Minecraft* [PC game]. Stockholm, Sweden.

Mortimer-Sandilands, C., & Erickson, B. (2010). Introduction: A genealogy of queer ecologies. In C. Mortimer-Sandilands and B. Erickson (Eds.), *Queer ecologies: Sex, nature, politics, desire*, 1–47. Bloomington: Indiana University Press.

Morton, T. (2007). *Ecology without nature: Rethinking environmental aesthetics*. Cambridge, MA: Harvard University Press.

Morton, T. (2010). Queer ecology. *PMLA*, *125*(2), 273–82.

Nakamura, L. (2002). *Cybertypes: Race, ethnicity, and identity on the Internet*. New York: Routledge.

Nakamura, L. (2008). *Digitizing Race: Visual cultures of the Internet*. Minneapolis: University of Minnesota Press.

Nintendo R&D4. (1985). *Super Mario Bros.* [NES]. Nintendo.

Nintendo R&D4. (1986). *The Legend of Zelda* [NES]. Nintendo.

Persson, M. [notch]. (2011, March 9). Terrain generation, Part 1 [Tumblr]. Retrieved April 2, 2014 from http://notch.tumblr.com/post/3746989361/terrain-generation-part-1

Priestman, C. (2013, December 8). Why it's best to be cautious around procedurally generated indie games. *Indie Statik*. Retrieved from http://indiestatik.com/2013/12/08/procedural-generation

Subset Games. (2012). *FTL: Faster Than Light* [PC game].

Toy, M., & Wichman, G. (1980). *Rogue* [PC game].

Yu, D., & Hull, A. (2009). *Spelunky* [PC game]. Mossmouth.