Loading… The Journal of the Canadian Game Studies Association
Vol 10(16): 75-86
http://loading.gamestudies.ca

Procedural Elaboration: How Players Decode Minecraft

Nic Watson
Concordia University
nicwatson@nicwatson.net

Abstract

Minecraft play practices reveal a type of analytic play in which significant work is invested in discovering esoteric details about the game, without necessarily providing practical prescriptions for optimizing play. This paper proposes the term "procedural elaboration" to describe such activities and the knowledge thereby produced. In contrast to the existing concept of theorycrafting, the products of procedural elaboration are primarily descriptive rather than prescriptive. However, this knowledge is far from trivial or banal. I argue that these knowledge-making activities can be explained through two functions of procedural elaboration. First, it provides players with a tool for dealing with the threatening inscrutability of some procedural game systems. Second, it acts as a ritual form of communication that helps to solidify a coherent *Minecraft* player community, while also establishing a social order within that community. Subsequently, I consider why players persist in using specifically experimental methods in procedural elaboration, even though the online availability of decompiled *Minecraft* source code means that the rules are not fully hidden as they are in most other games. I argue that the experimental method persists for these reasons: because it does not require specialized programming skills; because the gameplay already casts scientific experimentation as play; and because the iterative nature of *Minecraft's* development has produced source code that is structured in a way that resists direct deciphering.

Author Keywords

Minecraft; procedural elaboration; procedural literacy; procedural hostility; experimental play; tactics; ritual communication; cultural capital; gamer capital

Introduction: Procedural Elaboration and its Functions

MrSideliner is a video game record-setter. Specifically, he holds the record for the slowest possible movement speed in *Minecraft*, as documented in his YouTube video (MrSideliner, 2013). His accomplishment has less to do with the finesse or precise movement one usually associates with record-setting game play, and more to do with engineering skill. Improving on the methods of other slow-walkers before him, he determined that if he built an apparatus that layered cobwebs atop still water, above a layer of soul sand, with ice blocks underneath it all, and then walked through backwards, under the effects of a Potion of Slowness, while sneaking and drawing a bowstring, he could achieve constant motion at a whopping *18 metres per hour*.

A normal walking speed in *Minecraft* is 860 times faster than MrSideliner's record, clocking in at around 4.3 metres per second (where one voxel "block" equals one metre), according to a table compiled by players on the Minecraft Wiki ("Transportation," n.d.). The same table contains calculated average speeds for 19 other combinations of transportation method and conditions, all determined empirically through player testing: minecarts on powered track or on slopes; boats travelling on downstream rapids or over flat water, with or without a running start; sneaking, sprinting, swimming on the water's surface, swimming underwater (in both flowing and still water); with or without speed potions; and other combinations besides.

This table tells us two things. First, it indicates the fastest way to get around (without using teleportation or the flying abilities offered by the game's "Creative Mode") is to ride a horse, for an average speed of 9.675 m/s. Second, it shows that *Minecraft* players will invest hours of work in setting up controlled experiments to empirically derive vast amounts of quantitative trivia. According to the article's associated Talk page, experimenters built giant ramps and flat tracks, and measured the time it took to travel 100 metres in two timed trials. The original wiki article, before it was edited for conciseness, reported the individual trial times themselves.

The experiments, their results tables, and accompanying wiki discussion, together with the in-game experimental constructions used to carry out the investigations, constitute a mode of analytical play practice that I propose to call *procedural elaboration*. This term was chosen to capture how players trace, rework, and redirect procedural game rules in order to discover something about their nature. The word "elaboration", having the same root as "labour" and "laboratory", implies a process of *work* through which something is produced, while also suggesting the notion of extending or adding to a core concept for the sake of explication (as in the phrase, "Please elaborate on your statement"). Procedural elaboration does precisely this: it explicitly investigates, documents, and renders visible the (sometimes unexpected) results of procedure that are only implied as potentials in the game code itself.

Through procedural elaboration, *Minecraft* players invest significant work in order to discover what look like esoteric details regarding the underlying rules of the game engine. Although in some ways similar to theorycrafting as it is described by Paul (2011)[1], procedural elaboration differs significantly in that it is primarily *descriptive* (whereas theorycrafting is *prescriptive*). Procedural elaboration is not directly associated with power-gaming or strategies for optimizing play. It may therefore be tempting to dismiss the fruits of procedural elaboration as trivial curiosities, but on the contrary, these activities play central roles in the articulation of *Minecraft* gameplay. Adapting de Certeau's (1984) concept of *tactics*, I will show how procedural elaboration enables players to productively inhabit and navigate the procedurally-hostile *Minecraft* world. Furthermore, drawing on James Carey's (2009) work on the ritual function of communication, I argue that procedural elaboration and the circulation of its products serve as rituals through which social order is enacted and maintained within the player community. Subsequently, I consider why in-game experimental methods of procedural elaboration persist in the case of *Minecraft*, despite the general availability of its source code for direct reading.

Background

Briefly, *Minecraft* is an open-world sandbox game, released by Mojang AB in 2011, in which

players can explore, modify, and build within a procedurally-generated world. It is often described—imperfectly, according to Duncan (2011, p. 11)—as an expansive, inhabitable virtual Lego set.

To date, most extant literature on *Minecraft* is concerned with its potential educational uses (e.g. Bayliss, 2012; Short, 2012; Schifter & Cipollone, 2013). Other scholarship from the field of game studies does exist. Goetz (2012) describes the key pleasure of *Minecraft* gameplay as a "tether and accretions" fantasy, while Keogh (2013) has explored how the threat of permanent player death in a *Minecraft* world can provide a narrative-producing tension. Lastowka (2011) argues that *Minecraft* has benefited from a Web-2.0-like strategy of harnessing amateur creativity, while Duncan (2011) suggests that *Minecraft* has a "platform" nature, pointing to how the iterative and agile nature of *Minecraft* development, coupled with customer collaboration, results in "many Minecrafts" (p. 2)—the existence of many different types of gameplay that are all called "Minecraft."

***Optimizing Minecraft*?**

It is necessary to acknowledge that instrumentalism and play optimization—the traditional aims of theorycrafting—are present in *Minecraft* as motivations for empirically investigating hidden game mechanics, if only to highlight how procedural elaboration goes beyond these uses. With a simple combat system and relatively few varieties of gear compared to typical role-playing games, *Minecraft* demands less statistical optimization from players than games like *World of Warcraft* do. Most optimization prescriptions concern the combinatorial use of resources. The *Minecraft* Wiki article on "Smelting", for example, provides tips on how to optimally convert wood logs into usable furnace fuel (charcoal), listing the seconds of smelting time per log-to-charcoal conversion for each of the other wood-derived fuels ("Smelting", n.d.). A table on the same page suggests selecting fuels based on the number of items to be smelted in a batch, to minimize waste. Another important area of optimization is in item and book enchanting, in which the player can maximize the possibility of acquiring a specific enchantment by selecting the right enchantable equipment and spending the right number of experience points[2].

The knowledge of how to move very, very slowly does not seem to align with these kinds of instrumental goals. Yet this, and other similar forms of knowledge, serve as important tools for players dealing with the everyday facts of *Minecraft* life.

A Tactic Against Intractability

Procedural hostility is one such fact of *Minecraft* life. This does not refer to visible, diegetic hostility in the form of starvation, lava pits, treacherous cliffs, or the zombies and creepers that emerge at night. Procedural hostility instead resides in the difficulty that the player has in making sense of the operations of the game engine. It makes itself known whenever the game seems to behave in a glitchy and unpredictable way, or when it presents an unexpected obstacle to the player's plans. It appears also in the inability of the player's mind to grasp the complex and inscrutably rapid operations of the machine that runs the game. Even when set to the monster-less "Peaceful" mode, *Minecraft* remains procedurally hostile.

The case of the glitchy portal

Anyone who has tried to set up a fast travel system in *Minecraft's* Nether dimension will have at least a passing familiarity with this hostility. The Nether is a hellish dimension that exists apart from the normal *Minecraft* world (the Overworld). The two spaces can be linked together by means of magical teleportation portals. The Nether is spatially shrunk relative to the Overworld, such that two portals that are close together in the Nether may link the player to corresponding portals that are much further apart in the Overworld. A common player project, therefore, is to link distant Overworld locations via easy highway routes through the Nether. The engineering of such routes is complicated by the sometimes bizarre ways in which Nether and Overworld ("normal" world) teleportation portals are paired up. Multiple Overworld portals can become linked to a single Nether destination, while a Nether portal can unexpectedly link to an Overworld portal other than the one the player intended to reach. To make matters worse, if a portal in a pair is blocked or deactivated, then attempting to travel through what used to be its mate can lead *Minecraft* to create an entirely new portal, either in a different (and unexpected) location, or placed awkwardly atop the existing portal.

A 2010 thread on MinecraftForum.net, titled "THEORY: Why your portals are 'glitched'" (Mitch7656, 2010), attempts to figure out the reasons behind this erratic behaviour by making educated guesses as to how the game algorithm would go about selecting or creating a destination portal upon teleport. The original poster, Mitch7656, created a series of cartoon images to help explain how the spatial relationship between Overworld and Nether plays into this algorithm. In this early era of *Minecraft* development (open alpha testing), part of the purpose of these investigations appears to have been to determine if the observed portal behaviours should be considered a bug or a feature. In either case, these players were largely at the mercy of the developers and their update cycle for patches. In the meantime, a working knowledge of the procedural rules governing Nether portals would help to take the edge off of the existential threat posed by their strange behaviour, by implying a kind of mastery over the game engine, even in the absence of actual control. At the very least, knowing when and how things will break helps to counteract the disorientation caused by unexpected glitches.

This act of procedural elaboration did not result in a full-fledged 'Theory Of The Nether Portal', but rather an ad-hoc theory of why the Nether portal breaks. Although it implies a possible solution (place your portals sufficiently far apart from one another so that they do not interfere, and put them in protected locations to prevent deactivation by monsters), the primary purpose of the exercise appears to be descriptive rather than prescriptive.

This function of procedural elaboration constitutes a practice of everyday *Minecraft* life, one that counters the logistical problems posed by the game's everyday procedural hostility. It is what de Certeau (1984) calls a *tactic*. In *The Practice of Everyday Life*, de Certeau describes media texts metaphorically as landscapes over which media producers claim ownership. These producers employ planned, top-down strategies in order to direct and constrain the ways in which their works will be interpreted and used. Audiences, however, employ the tactics of nomadism and poaching to carve out their own inhabitable interpretive spaces, and navigate their own reading-paths through the text. In the case of *Minecraft*, the concept of the text maps onto the game code as an intentionally designed artifact: the producer made decisions about how *Minecraft* would behave based on strategies that balance game design goals, business and development models,

programming conventions, and the capabilities of target machines. Although these strategic decisions may leave players perplexed and frustrated, they can resort to procedural elaboration tactics to help them navigate through the untamed procedural frontier of an early-alpha game. They can thus assert a sense of control, mastery, and belongingness therein, and ultimately turn what (at player-level) seems chaotic and threatening into something familiar and domestic.

The problem of motion

Motion through space (a concept distinct from teleportation and travel) poses additional procedural puzzles for players to deal with. While the ability to move fast is an obvious convenience, what is at stake in the deliberations over the details of speed (described in the introduction, above) is a deeper understanding of how the *Minecraft* code actually builds motion as a phenomenon. MrSideliner's slow-motion machine demonstrates which slowness-causing conditions are additive, and provides an opportunity for players to reflect on how these conditions can combine to produce emergent procedural extremes of behaviour. At its limits, slowness itself becomes an intractable problem. The inability to achieve very slow speeds under normal conditions leads to a lack of precision: players have difficulty getting their avatars into the exact positions and orientations desired when, for instance, posing for a third-person screenshot, or trying to approach a ledge as closely as possible without falling off. At normal walking or sneaking speeds, it is too easy to over-shoot the mark. While experiencing slowness through MrSideliner's device does not provide a solution to this problem, it does help make sense of it. More philosophically, it is a means for wrapping one's head around the perplexing notion of near-imperceptible-yet-inexorable slow motion and the startling effect of multiplicative modifiers, and as such, has precedents in the art world. There is a striking similarity between MrSideliner's apparatus and Arthur Ganson's kinetic sculpture, *Machine With Concrete*, housed at the Exploratorium museum in San Francisco. *Machine With Concrete* consists of a motor spinning at a quite visible 212 RPM (approximately one revolution every 0.28 seconds), connected to a series of 12 worm gears. Each worm gear drives the next axle in the chain at one-fiftieth of its own frequency. After only 12 such speed reductions, the final gear is spinning so slowly that it makes one revolution every 2.3 trillion years—slow enough that it can be safely encased in a concrete block.

Procedural elaboration through sculpture

In another instance in which the products of procedural elaboration look more like descriptive art than prescriptive strategy, a group of experimenters set out to investigate the details about how players are respawned in a *Minecraft* world after death. The results of this experiment are showcased on the *Minecraft* Wiki's page about respawning ("Spawn/Multiplayer details", n.d.). The players built an obsidian pillar at the precise nominal "spawn coordinates" (a specific block location, as set using administrator commands), on a flat bend of sand. One player repeatedly died and respawned, digging a hole in the sand on the exact point where the player reappeared each time. After several dozen iterations, the resulting distribution of holes made clear that respawning players would be placed randomly within a 20-by-20-metre square centered on the nominal spawn point. This knowledge does have some practical use—making it possible for multiplayer server administrators to build "spawn jails" to hold new or respawned players—but the wiki page provides far more information than is practically necessary. Instead of simply announcing the experiment's conclusion, the page provides an explanation of the method and a screenshot of the test setup, showcasing the process of discovery itself. What is most compelling about the experimental apparatus is that it works as a sculpture that is a scatterplot of its own data set, literally

graphed directly onto the terrain. The creation of this sculpture can be read as an instance of data art practice, which Whitelaw (2008) describes as "a concrete exploration of what data is, does, and can do" (Introduction section, para. 3). In this instance, however, the art is concerned with a specific understanding of what data "can do", namely in terms of using physically inscribed data to trace human-programmed, black-boxed procedures.

To summarize, the tactics of procedural elaboration are ways of *dealing with* problems posed by difficult-to-comprehend aspects of the game world. Importantly, "dealing with" is not the same thing as *solving*. Sometimes, procedural elaboration can do little more than comment on the natural order of the (procedural) universe. This leads to the second function of procedural elaboration: as a form of ritual communication that underpins the social life of the *Minecraft* player community.

Procedural Elaboration as Ritual Communication

Reflecting the natural order

James Carey argues that newspapers can be understood "less as sending or gaining information and more as attending a mass, a situation in which nothing new is learned but in which a particular view of the world is portrayed and confirmed" (2009, p. 16-17). Seen this way, the newspaper becomes "a presentation of reality that gives life an overall form, order, and tone" (p. 17). The acts of procedural elaboration discussed above can also be understood in this way. What is most striking about the examples given is that the product of the investigations, as presented to the world, is more than just a piece of knowledge or an answer to a research question—that is, more than what we would expect from thinking of these activities in terms of meaningful statements to be transmitted from knowledgeable experimenters to other players. The centrepiece of MrSideliner's work is not the verbal answer to the question, "What is the slowest possible speed and how is it achieved?", but rather the machine he built to arrive at the answer. The respawning experimenters similarly provided documentation of their process and screenshots of their apparatus. The player community seems less interested in the answers to questions about the game engine than they are in the process of how such answers were obtained. This makes sense if, taking a cue from Carey, we see the communication of procedural elaboration less as attempt to disseminate instrumental or prescriptive information about how to play *Minecraft*, and more as a way of dramatically depicting the experience of life in the procedurally-constituted *Minecraft* world.

The production and sharing of these depictions is key to the social construction of a *Minecraft* player community on multiple levels. The experimental activities of procedural elaboration themselves invite collaboration between multiple players—a point which is discussed further in the section on "Empirical versus generative methods of decoding" below. Furthermore, much as the mass circulation of print media gave rise to the "imagined communities" of Anderson's theory (1991)[3], the depictions of acts of procedural elaboration for a broader audience, via YouTube or the *Minecraft* Wiki, creates a sense of common experience for *Minecraft* players who will never actually meet in the same *Minecraft* worlds.

Procedural elaboration, paratext, and gamer capital

Procedural elaboration does not just reinforce social cohesion, it also plays a role in establishing social *order* by being associated with prestige and what Consalvo (2007) calls "gamer capital". It has this in common in theorycrafting. However, where the link between prestige and disseminating prescriptions for optimal play and prestige is intuitively easy to see, the notion that prestige also adheres to those who practice descriptive procedural elaboration is less obvious, and is worth explaining here.

Paul (2011) has argued for understanding theorycrafting, and the discourse that surrounds it, as game paratexts which, in Consalvo's words, "work to shape the gameplay experience in particular ways" (2007, p. 9). The same analysis applies to procedural elaboration, which is communicated through paratexts such as wiki articles, YouTube videos, and forum postings. For Consalvo, the ability to speak authoritatively about games—that is, to mobilize these paratexts—is tied to "gamer capital," a subtype of Bourdieu's (1986) cultural capital. The ability to discover, share, and debate detailed knowledge about hidden game engine behaviour through paratexts thus serves an important function in the circulation of cultural capital within a gaming community. It marks some people as experts who are procedurally literate (Mateas, 2005), even if the knowledge they have seems of little immediate use from the point of view of someone who is only looking to optimize his or her gameplay. Just as there is cultural capital associated with knowing all manner of trivia—and thereby performing well at bar trivia or *Trivial Pursuit*, there is cultural capital associated with having specialized knowledge of the details of *Minecraft's* game engine. For example, players often install mods to change their *Minecraft* gameplay experience, and a particular choice of which mods to play with is a defining characteristic of one's play style. Being able to speak authoritatively about balance, overpowered/underpowered features of various mod combinations, or about the realism of a particular mod's method of representing electrical energy, can help define one's own personal mod choices as rational and legitimate in public discourse.

Debate is not limited to occurring between players, either. Procedural elaboration also plays an important role in dialogue between players and developers. Several contributors to the aforementioned forum discussion on Nether portals used the discussion as an opportunity to push the position that there is something wrong that the developer needs to "fix." Crucially, procedural literacy must be demonstrated for a successful critique of a developer's actions. The ability to perform such critique not only renders procedure familiar and domestic (as was previously discussed), but also positions the writer of the critique as a knowledgeable person with sufficient status to participate legitimately in the player community. After all, if one has sufficient standing to go toe-to-toe with the almighty developer (whether or not the developer is actually paying attention), then one must surely have sufficient standing to engage with other players.

This section has explored how descriptive procedural elaboration serves important social functions for the *Minecraft* player community. In the following section, I consider an unusual problem posed by the case of *Minecraft*: why experimental procedural elaboration persists as a vibrant practice even when source code containing all the answers is available.

Empirical Versus Generative Methods of De-coding

In most games, players who are interested in gaining a deeper understanding of the game's operations must attempt, through repeated empirical study and statistical analysis, to triangulate on the hidden rules of the game engine. *Minecraft* players often use a similar approach, but they also have an unusual alternative option available to them. With the release of the fan-made Minecraft Coder Pack in 2011, it became relatively simple for anyone with some knowledge of the Java programming language to decompile the game into de-obfuscated, readable source code. Rather than trying to guess the underlying rules through trial and error, players can now read those rules directly, which I call a "generative" method. This arguably yields more precise results. Yet as the *Minecraft* Wiki's table of *Minecraft* movement speeds demonstrates, both methods persist.

Obstacles to decoding

The easiest explanation as to why empirical methods persist is that not everybody has the means to read the Java code. The barriers to entry into the world of empirical experimentation are much lower than the barriers to deciphering source code. We might still expect that code-reading would supplant empirical methods by positioning itself as a more precise—and therefore more legitimate—discourse, but curiously this does not appear to have happened on the *Minecraft* Wiki. A partial explanation may be that empirical methods hold their own because of the creative twists they introduce, such as showing off the cleverness of the experimental apparatus, or building in-world data visualizations.

Furthermore, looking inside the black box of code does not necessarily clarify anything. Sometimes, after a peek inside the code, even the experienced programmer will decide that it would be easier to triangulate the outside of the black box, rather than untangle the inside. The code is not structured and partitioned rationally like an apartment building, but is messy and organic like a wasp's nest. This is the natural result of *Minecraft's* iterative development and just-in-time addition of new features. Duncan (2011) suggests that there are "many Minecrafts"—meaning that there are many radically different ways to articulate gameplay within *Minecraft*, but also more literally that there are many different publicly-released versions of the software. There are dozens of official "vanilla" (non-modded) releases alone, from early development snapshots, through alpha and beta stages, to the current "release" version that Mojang has continued to iterate upon, despite it having been ostensibly "finished" years ago. The differences in game mechanics between versions can substantially alter the experience of gameplay[4]. More to the point, the code to add new features is often shoehorned awkwardly into the existing code-base, in a way that is not always easy to puzzle out. The "agile" nature of *Minecraft* development that Duncan (2011) identifies as key to its success becomes an obstacle to those who would study the code instead of experimenting in game.

For instance, the algorithm for determining walking speed has a series of checks for conditions such as sprinting, jumping, moving through water, moving on ice, attempting to use an item, being under the influence of a potion, and more. These checks are not all performed in an orderly fashion in one part of the program, but are instead spread haphazardly across multiple subroutines. One of the major tasks of the player-movement code is translating between movement as a forward speed and strafing speed (i.e. the local axes, relative to the player), to movement as a displacement along the global X and Z axes. The various checks and modifiers on movement speed occupy awkward

positions before, in-between, and after these translation steps. Additionally, the choice of multipliers for normal walking speed seem to be largely arbitrary. A contributor to the *Minecraft* Wiki's talk page for the Transportation article has actually puzzled out the entire algorithm for calculating walking speed by reading source code, and has re-written it in a simplified form ("Transportation", n.d.). In this explanation, the contributor notes that the "internal movement speed," a variable number that conveys little meaningful information on its own, can be translated into metres per second by multiplying by 4274/99. It is probable that these values were themselves tweaked by developers through trial and error. Thus, this portion of the code itself was likely written, tested, and modified through empirical experimentation, rather than by generative planning.

Given these obstacles to mapping out game rules by studying code, it is no wonder that experimentation remains a comparatively effective method for explicating the game's behaviour. The structure of the code inherently resists the big-picture strategy of top-down, line-by-line explication, so players resort to the ground-level tactic of experimentation instead.

"I'm going to try Science!"

A certain fascination with the thematic representation of the scientific method may also be responsible for the popularity of empirical methods among *Minecraft* players. "Science" (often spelled with a capital 'S' in the experimental play I have observed) becomes a ritual mode through which both playfulness and expertise are represented. Furthermore, it is a mode that invites participation, collaboration, and the showing-off of skillfully-constructed Scientific apparatus. Where studying source code isolates investigators from each other, in-game experimentation brings them together.

Furthermore, it is not surprising that *Minecraft* players would be predisposed to enjoy experimentation, since the game already invites such activity from the outset—for instance, in the initial discovery of crafting recipes (before a preponderance of how-to guides began to appear on the internet). Popular mods like Thaumcraft make experimental research explicit. In Thaumcraft, the player must use instruments to study objects in the environment in order to determine their elemental composition. New elements are discovered through a combination of finding them in nature, and a trial-and-error process of attempting to combine elements in a laboratory setting. Empirical science is already framed and experienced as a play activity. Thus, the experimental mode of procedural elaboration is more readily experienced as play than is code-reading, especially since it actually involves running the game program instead of just reading it.

In the *World of Warcraft* theorycrafting community, there is some concern that access to a game's code would mean the end of theorycrafting as a vibrant practice, and would render the game as a whole more shallow (Paul, 2011, Theorycraft: Why Players Use It section, para. 7). Whether or not this would be the case for *WoW*, the case of *Minecraft* demonstrates that such an outcome is not generally inevitable, especially if we look beyond the purely instrumental view of experimentation. The structure of the game code, combined with the entrenched appeal of ritual modes of communication enabled by procedural elaboration, help to maintain the use of experimental methods even when source code is available.

Conclusion

The existing theorycraft-based explanation of empirical investigation of game mechanics has largely explained such activities in terms of prescriptions for optimization and power-gaming. This paper has argued for the need to take a more expansive view of these activities. The concept of procedural elaboration helps to illustrate what instrumental theories of gameplay fail to capture: how the production of descriptive procedural knowledge through experimentation is a gameplay end in itself. Players use the tactics of procedural elaboration to render procedurally hostile spaces safe and domestic, assert mastery over the game space, and position themselves as knowledgeable experts, rich in gamer capital. These tactics stand in contrast to the strategies of theorycraft, wherein some players feel that prescriptivism threatens to dictate one "right" way to play the game (Paul, 2011, Theorycraft: Why Players Resist section, para. 9). Without a prescriptive message to transmit, the communication of acts of procedural elaboration nevertheless serves a ritual function, constructing and reinforcing social bonds within the player community.

There is a need to investigate whether procedural elaboration practices of the kind seen in *Minecraft* are also found in other games. Moreover, procedural elaboration can benefit from being integrated with larger theories of emergent play. Newman (2008) has drawn attention to the notion of playing and experimenting with games as an alternative mode of play that reframes the game as a toy. Parker's (2008) concept of "expansive play" provides opportunities for interpretation through a procedural-elaboration framework. With expansive play, players synthesize their own game experiences out of a combination of procedural rules already embedded in the game program and self-imposed rules. Knowledge produced via procedural elaboration can be applied towards creating and refining the kinds of player-made games-within-games discussed by Parker—games-within-games like Jeep Tag may themselves be examples of procedural elaboration. Furthermore, it may be productive to consider procedural elaboration itself as an instance of expansive play. This would point to an investigation of whether there are player-imposed rules for carrying out procedural elaboration, and how these rules come about.

Finally, an emic[5] perspective on the experience of, and motivations for, procedural elaboration is needed. A logical future step would therefore be to directly interview players engaging in these practices.

References

Anderson, B. (1991). *Imagined communities: Reflections on the origin and spread of nationalism* (Revised Edition). London, UK: Verso.

Bayliss, J. D. (2012, September). Teaching game AI through Minecraft mods. In *Games Innovation Conference (IGIC)*, 2012 IEEE International (pp. 1-4). IEEE.

Bourdieu, P. (1986). The forms of capital. In J. Richardson (Ed.), *Handbook of theory and research for the sociology of education* (pp. 241-258). New York, NY: Greenwood.

Carey, J. W. (2009). A cultural approach to communication. In J. W. Carey (Ed.), *Communication as culture: Essays on media and society* (Revised Edition) (pp. 11-28). New York: Routledge

Consalvo, M. (2007). *Cheating: Gaining advantage in videogames* [Amazon Kindle Edition]. Retrieved from www.amazon.com/Cheating-Advantage-Videogames-Mia-Consalvo-ebook/dp/B001D221M4/

de Certeau, M. (1984). *The practice of everyday life* (S. Rendell, Trans.). Berkeley, CA: University of California Press.

Duncan, S. C. (2011). Minecraft, beyond construction and survival. *Well Played: A Journal on Video Games, Value and Meaning*, 1(1), 1-22. Retrieved from http://dl.acm.org/citation.cfm?id=2207097

Goetz, C. (2012). Tether and accretions: Fantasy as form in videogames. *Games and Culture,* 7(6), 419-440.

Keogh, B. (2013, September). When game over means game over: Using permanent death to craft living stories in Minecraft. In *Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death*. ACM.

Lastowka, G. (2011). *Minecraft as web 2.0: Amateur creativity & digital games*. Available at SSRN 1939241.

Mateas, M. (2008). Procedural literacy: Educating the new media practitioner. In D. Davidson (Ed.), *Beyond fun* (pp. 67–83). Pittsburgh, PA: ETC Press. Retrieved from http://dl.acm.org/citation.cfm?id=1594207.1594217

Mitch7656. (2010, October 31). THEORY: Why your Nether portals are “glitched”. Message posted to http://www.minecraftforum.net/forums/archive/alpha/alpha-minecraft-halloween-update/838694-theory-why-your-nether-portals-are-glitched

MrSideliner. (2013, January 20). Slowest you can walk in Minecraft: 18 blocks per hour! [Video file]. Retrieved from https://www.youtube.com/watch?v=Q8b5xehp3sg

Newman, J. (2008). *Playing with videogames*. New York, NY: Routledge.

Parker, F. (2008). The significance of Jeep Tag: On player-imposed rules in video games. *Loading…*, 2(3). Retrieved from http://journals.sfu.ca/loading/index.php/loading/article/viewArticle/44

Paul, C. A. (2011). Optimizing play: How theorycraft changes gameplay and design. *Game Studies*, 11(2). Retrieved from http://gamestudies.org/1102/articles/paul

Schifter, C., & Cipollone, M. (2013, March). Minecraft as a teaching tool: One case study. In *Proceedings of Society for Information Technology & Teacher Education International Conference* (pp. 2951-2955).

Short, D. (2012). Teaching scientific concepts using a virtual world – Minecraft. *Teaching Science*, 58(3), 55-58.

Smelting. (n.d.). In *Minecraft Wiki*. Retrieved February 12, 2015, from http://minecraft.gamepedia.com/Smelting

Spawn/Multiplayer details. (n.d.). In *Minecraft Wiki*. Retrieved February 12, 2015, from http://minecraft.gamepedia.com/Spawn/Multiplayer_details

Transportation. (n.d.). In *Minecraft Wiki*. Retrieved February 12, 2015, from http://minecraft.gamepedia.com/Transportation

Whitelaw, M. (2008). Art against information: Case studies in data practice. *The Fibreculture Journal*, 11. Retrieved from http://eleven.fibreculturejournal.org/fcj-067-art-against-information-case-studies-in-data-practice/

---

[1] Paul (2011) describes the emergence of theorycrafting as a player practice in *World of Warcraft*. Through controlled, repetitive experiments and statistical analysis, theorycrafters attempt to triangulate on hidden rules of the game engine, such as how exactly a given weapon's attack rolls are calculated, or the probability of a monster dropping a rare item. The end goal is usually to come up with strategies for optimizing one's gameplay—for instance, determining the best equipment set for fighting in a certain area at a certain level.

[2] Players can use the fan-made, web-based *Minecraft Enchantment Calculator* for this task: http://www.minecraftenchantmentcalculator.com/rev6/

[3] This comparison takes a more expansive view of the concept of imagined communities than Anderson does in his book *Imagined Communities* (1991), extending it beyond an explanation of nationalism to describe more generally a sense of community that can form around media artifacts shared in common, even without direct communication between members.

[4] For example, the addition of beds in Beta Version 1.3, which allowed players to skip through nights, substantially changed the prior rhythm of differing daytime and nighttime activities described by Goetz (2012, p. 425).

[5] That is, an internal, local perspective that considers the cultural meanings espoused by the practitioners of procedural elaboration themselves.