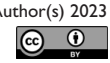
Article

new media & society
2024, Vol. 26(11) 6205–6219



DOI: 10.1177/14614448221144989
journals.sagepub.com/home/nms


From Indy to ubiquity: Minecraft as platform and infrastructure

David Murphy
Staffordshire University, UK

Abstract

This article is about digital games, their evolving connections with platforms and infrastructures, and the influence that a decade of *Minecraft*'s development is having on this process. It begins with a discussion of previously disparate but increasingly convergent methodologies and literatures, including platform studies, media archeology, game studies, and cultural anthropological approaches to the study of infrastructure. Then, it applies points of convergence within these literatures to a political economic analysis of *Minecraft* that attributes its decade of growth to the systemic and metaphoric merging of platforms and infrastructures. Finally, it provides an ethnographic analysis of computers, made in *Minecraft*, which show how the game is not only taking on characteristics of platforms and infrastructures, but also affording a means of programming, visualizing, and experiencing platforms as infrastructure.

Keywords

platform studies, game studies, infrastructure, media studies, Minecraft

Introduction

Imagine spending an evening inside of your computer observing data as it flows. Imagine a processor, memory, and bus working in concert; imagine a file streaming bit by bit; imagine a graphic appearing pixel by pixel. In theory, these components and events need each other to work, but they work too fast for the eye to see, so people abstract them and take them for granted, ignoring process and favoring speed. Now imagine a platform that slows down these processes—a platform that renders them visible. Due to the micro

Corresponding author:
David Murphy, 331 Leek Road, Stoke-on-Trent, ST4 2BU, UK.
Email: David.Murphy@staffs.ac.uk

nature of micro-processing, most contemporary platforms are limited with respect to their ability to aestheticize computational infrastructure, but computers made in *Minecraft* are different. These computers are made from a redstone: a magical, programmable substance which is combined to create redstone circuits, resulting in a unique aesthetic that privileges computational form and process over speed and functionality.

Redstone added the element of programmability to *Minecraft*, transforming an interactive form of media that was initially conceived as a game into something that is not only being valued as a platform but is also taking on the characteristics of infrastructure. For this reason, *Minecraft* is a unique case that not only carries implications for the future of game production, but also the future of media production in general, as it takes on aesthetic and systemic metagaming (Boluk and Lemieux, 2017) characteristics—which are not traditionally attributed to screen based media—that bridge technologies, epistemologies, and systems. To expand on this point, this paper will begin with a discussion of previously disparate but increasingly convergent methodologies and literatures, including platform studies (Poell et al., 2022; Gillespie, 2010; Bogost and Montfort, 2007, 2009), operative media archeology (Parikka, 2011), game studies and cultural anthropological approaches to the study of infrastructure (Larkin, 2013). Then, it will apply points of convergence within these literatures to a brief history of *Minecraft* that attributes its decade of growth to the systemic merging of platforms and infrastructures (Plantin et al., 2018). Finally, it will provide an aesthetic and ethnographic analysis of computers, made in *Minecraft*, which show how the game is not only taking on characteristics of platforms and infrastructures, but also affording a means of programming, visualizing, and experiencing platforms as infrastructure. In doing so, my intention is not to condemn or celebrate *Minecraft*, but rather to read recent scholarship on the convergence, obfuscation, visualization, and regulation of platforms and infrastructures through *Minecraft* by way of a macroeconomic and micro-ethnographic analytic approach.

The politics and poetics of platforms and infrastructures

Platform studies (Bogost and Montfort, 2007) is a relatively new media studies methodology which examines the material and cultural factors informing the creative production of digital works. In Montfort's (2006) "Combat in Context," for example, the approach can be best described as a "bottom up" analysis of the unique layering of hardware, software, and firmware that is affiliated with the Atari Video Computer System. Put another way, a platform study investigates layered information systems that support creative work, with creative work being interpreted broadly to include digital games, computational art, and applications software. The methodology is materially and empirically grounded, from this perspective, but a platform is still an abstract entity in and of itself which is created and shaped by heterogeneous epistemologies. In its most common iteration, a platform study excavates affordances and constraints which influence creative design decisions that other methodologies miss. Montfort and Bogost (2009), for example, describe incompatibilities created by the decision to interface with television signals that limited the Atari Video Computer System's graphical capabilities, and how developers exceeded this limitation by hacking the platforms graphic registers.

Platform studies has added technical insight to the study of game history, but the approach has not been free from criticism. It shares the material and historical concerns of media archeology; however, it has also been criticized for granting the platform "a degree of stability and consistency as a technical object" (Apperley and Parikka, 2018: 353). This perceived stability, which is attributed to the process of writing platform histories, has also been associated with academic work on the history of infrastructure. According to Larkin (2013), a significant portion of infrastructure studies ignores the political and metaphorical use of infrastructure, or the fact that infrastructures can ''also exist as forms separate from their purely technical functioning" (p. 329).

Fortunately, an expanding subset of scholars are engaging with platform and infrastructure studies in a variety of critical contexts while applying insights from a variety of disciplines and literatures. On the platform side, Langlois and Elmer (2013) apply a political economics of communications approach to the study of corporate social media platforms, which are conceptualized as digital objects that sell users attention to advertisers (p. 4), and Poell et al. (2022) are developing a growing body of critical scholarship on the platformization of cultural production. On the infrastructure side, Parks and Starosielski apply Karan Barad's feminist science and technology studies work on the entanglement of matter and meaning to the study of media infrastructures, describing infrastructures as "material forms as well as discursive constructions" (p. 5). Significant to both approaches is attention to the correlations and contradictions that occur when platforms and infrastructures operate on technical/material and communicative/discursive levels, and efforts to account for these interactions by way of mixed methodological enquires.

Recognizing these shared concerns, scholars engaged in the historic and contemporary study of platforms are discovering points of mutual interest and concern with scholars engaged in the historic and contemporary study of infrastructure. Why are these conversations converging? Scholars from multiple disciplines provide a variety of reasons. Plantin et al. (2018) point to decades of neoliberal policies which favor the replacement of services that were previously performed by publicly controlled infrastructures with services performed by privately controlled platforms. Constantinides et al. (2018) cite corporate interests in platform business models that can be "created and cultivated on top of infrastructures," and Nieborg and Poell (2018) describe ''the penetration of economic, governmental, and infrastructural extensions of digital platforms into the cultural industries" (p. 1).

How does *Minecraft* fit into this equation? How does a game, which came to be valued as a platform, take on the characteristics of infrastructure? To answer this question, this paper will combine two forms of analysis, beginning with a macroeconomic historical analysis of a free-to-play game (which became a platform) that is being monetized and scaled into infrastructure, followed by a micro-ethnographic analysis of computers (made from redstone) that provide a means of visualizing and experiencing platforms as infrastructure. This approach, which demonstrates how theories and methods applied to the study of Google and Facebook can also be applied to *Minecraft*, not only builds on recent macroeconomic and micro-ethnographic studies of media infrastructure (Parks and Starosielski, 2015) and research on the platformization of cultural production (Poell et al., 2022), it also connects the convergence of platforms and infrastructures to contemporary

game production and the obfuscation and representation of infrastructures and platforms in screened entertainment.

As multiple scholars have noted, Bogost and Montfort (2007) refused to provide a technical definition of a platform, preferring to encourage speculative and creative interpretations of the concept. The early literature written about *Minecraft* reflects this perspective, with multiple authors describing *Minecraft* (Duncan, 2012; Lastowka, 2012; Leavitt, 2013) as a platform without defining the concept. Implicit in this literature is a social understanding of the game, which emphasizes the importance of its player community. In other words, *Minecraft* is a platform that facilities user-generated content because players use it as such.

The initial literature on *Minecraft* was written at a time when user-production was predominantly associated with amateur fan activity, but, as subsequent analysis will show, *Minecraft*'s status as a platform would shift, moving from an experimental and surprisingly social game that was used like a platform into a socio-technical game-as-platform business model. This shift corresponds with recent literature on platform capitalism (Srnicek, 2017) in which platforms are understood as a social-technical intermediaries designed to create two-sided market structures which "mobilize code and data analytics to compose immanent infrastructures" (Langley and Leyshon, 2016: 2).

It is important to call attention to the fact that 'infrastructure' is a loosely defined relational term in the literature on platform capitalism (and the platformization of cultural production), with Poell et al. (2022) offering a "platforms as components-based data infrustructures approach" (p. 60). The literature on media infrastructure, on the other hand, makes use of a frequently cited history of modernity and electricity, in which Hughes (1993) defines infrastructure as a technical system which comes into being once it has reached a particular scale. This scaling, which involves technical, scientific, economic, political, and organizational fields of human activity, can involve the convergence of previously independent technical networks, which requires the development of standards and gateways, or it can involve the dominance of one technical system over its competition (Hughes, 1993: 2).

Recent research on the convergence of platforms and infrastructures makes use of Hughes definition, as well, applying it to Google and Facebook—two recent case studies that demonstrate the infrastructuralization of platforms and the platformization of infrastructures. This shift, which corresponds with the rise of ubiquitous networked computing and changing political sentiment, has created an environment in which platforms can reach unprecedented scales that compete, coexist, and supplant previously existing infrastructures (Plantin et al., 2018: 301). Google and Facebook, for example, have captured a massive amount of web traffic that now runs through closed applications cut off from the systems, networks, and protocols which constitute the open web (Anderson and Wolff, 2010).

Like Google and Facebook, *Minecraft* has developed a user base that is unprecedented for a digital game, averaging 141 million monthly active players in 2021, according to Statista (Clement, 2022), but unlike Google and Facebook the literature on the convergence of platforms and infrastructures has yet to account for *Minecraft*. This oversight is significant because an unprecedented user base is also where Minecraft's infrastructural value lies, and monthly user numbers are now cited more regularly than copies

of the game sold, which is surprising given that Minecraft has sold over 200 million copies (Clement, 2022) making it one of the highest selling digital games of all time. Integral to this shift is the success of the free-to-play Chinese version of the game which has surpassed 500 million registered players (as cited in Kulevska, 2022), the majority of whom are blocked not only from access to the open web, but also from platforms like Facebook, Twitter, and YouTube.

Minecraft's intersection with infrastructure is intrinsically related to its concurrent monthly user base, and like most large-scale technical systems that became infrastructures the expansion of this user base was disorganized and uneven, featuring instances of failure, appropriation, monetization, and disruption. Integral to this journey is the fact that *Minecraft* was supposed to be an independent fantasy game, not a platform that is reaching the scale of infrastructure, when it was released as a free-to-play demo in May 2009.

Open development?

For a game that would grow into a billion-dollar franchise, *Minecraft*'s launch was inconspicuously brief, with the first alpha version appearing for free in a TIGSource forum (Persson, 2009b, May 17). The demo was quite different from the game that Persson intended to make, and the game that exists at the time of this writing, but the creativity it inspired was apparent immediately. Within 24 hours, players were already sharing photos and videos of their creations, which primarily consisted of bridges, castles, and towers. People were returning to the game, as well, despite frequent crashes that erased their work.

After the demo was met with positive feedback, Persson (2009a, June 10) announced plans for a final retail version via his blog. Players willing to pre-purchase were promised a steady stream of updates, while players already content with the prototypes were permitted to keep playing for free. Taking a cue from free/open source software production, Persson (2009a, June 10) also created his own terms of service agreement: "Once you've bought the game, it's yours. Do whatever you want with it!" (para. 5).

Persson's terms of service agreement was a symbolic act which reflected an independent co-productive spirit, on the one hand, while obscuring the degree of control which he would retain over the game's algorithmic structure, on the other. This discrepancy is important because it gave Persson a considerable degree of power that was amplified by his specific design approach. A lot of contemporary games, especially games that are designed independently, are created in object-oriented software environments that make use of game engines which reuse prewritten code. In contrast, the first version of *Minecraft* was written from scratch using algorithms that produced content procedurally, which made the game updatable to the point where it could be altered repeatedly. Procedural content creation refers to the "algorithmic creation of game content with limited or indirect user input" (Shaker et al., 2016: 1), which is a technique that carries several advantages, including the fact that assets in the game environment do not have to be created manually. There have been no *Minecraft* sequels, as a result, because there do not need to be: the game's algorithms are updated, instead, and with each update capital is created algorithmically, which is an attractive model not only for producing a game but also for scaling and monetizing infrastructure.

In addition to using algorithms to generate in game content, *Minecraft* was split into two programs: a client application running on the player's computer and a server application hosting the world created. This type of design facilitates large-scale collaborations hosted on unofficial servers allowing players to connect from all over the world. Subscription-based MMOs are programmed in a similar manner, but server applications are closely guarded by the companies developing the games.

By letting players host their own servers, Persson nurtured the emergence of a vibrant modding culture, but the strategy was fraught with tensions. Persson was initially afraid of a modified version becoming more popular than his official Mojang release, so he updated his terms of service agreement. Server customization was permitted, but only if players agreed not to distribute modified versions, effectively limiting all client and server downloads to official files (Persson, 2009c, July 8). Mods written to customize appearances were encouraged, and mods altering key game components were frowned upon, although Persson did little to enforce his rules, citing his belief "in the user's right to [. . .] modify games they've bought" (as cited in Handy, 2010: para. 34).

Implicit in Persson's social contract was a desire to maintain control of the project while still incorporating input from the game's community, but as the game continued to grow this balance proved difficult to manage. After selling a significant amount of pre-orders, Persson was able to quit his job and begin working on *Minecraft* full time. The process necessitated the founding of a production company, called Mojang, which naturally led to development delays. Most fans were understanding of Persson's hectic schedule, but others were not, and a Distributed Denial of Service (DDOS) attack was launched on 20 October 2010. The attack, which dropped Minecraft's servers, was attributed to several angry fans that were upset over a perceived lack of promised updates (Crecente, 2010).

While the DDOS attack passed quickly, other problems were brewing. As *Minecraft* grew in popularity, user-created modifications increased the number of players that severs could host. Hosting hundreds of players required a lot of time and effort, as servers were growing quickly, prompting popular hosts to solicit donations from regular users. Some servers, which developed independently of Mojang, implemented free-to-play sales models and transformed their communities into for-profit businesses.

As a sales strategy, free-to-play models are ostensibly free because they rely on micro-transactions to generate revenue. Instead of purchasing the game, users are given the option of purchasing virtual assets which perform a variety of in-game functions, ranging from functions that have no impact on gameplay to functions that provide competitive advantages. Most players ignore in game purchases, but a small minority overspends, which raises complex ethical issues if the overspending players are children. Writing on his blog, Persson (2011, October 24) expressed distaste for the process, criticizing developers for exploitative design that results in games more akin to "a slot machine than Halflife 2" (para. 2).

While Persson initially ignored the development of for-profit severs, Mojang began to receive complaints about money being exchanged for in-game items. In response, Mojang updated its terms of service agreement, specifying what owners were and were not allowed to charge for. Instead of restricting all sales, players were permitted to receive payment for access and cosmetic items, as long as purchases had no effect on play (Owen, 2014, June 12, para. 8). Unsurprisingly, server owners protested loudly,

defending existing sales models as the only means of ensuring economic viability. According to Gregory Bylos (2014, n.d.), Mojang's new terms of service were "killing gameplay innovation in the multiplayer space" (para. 2).

Central to this debate was a dispute over updates to the terms of the original end user license agreement. For Persson (2014, June 14), making money was always frowned upon, except in the case of YouTube videos. Bylos (2014) countered this position, citing an old terms of service agreement that granted permission to create and sell plug-ins (para. 4). As multiple server owners threatened to shut down their communities, Mojang was accused of selling out, prompting Persson (2014) to tweet his frustrations: "Anyone want to buy my share of Mojang so I can move on with my life? Getting hate for trying to do the right thing is not my gig" (10:11 PM, 16 Jun 2014). Three months later, Microsoft acquired Mojang for 2.5 billion dollars, and *Minecraft* became the property of one of the biggest software companies in the world.

Standards and gateways

The purchase of *Minecraft* by Microsoft was a clear indication of the fact that *Minecraft* was not only a game, but also a platform with the potential to be scaled further. When providing an explanation for the sale, Persson insisted that the game had reached a point where the people who created it could no longer manage it effectively, requiring a company like Microsoft to take it over. Infrastructures often develop in this way, from a historical perspective, because they "are seldom built by system builders from scratch" (Appel et al., 2018: 12). They come into being, instead, "revealing fragile and often violent relations between people, things, and the institutions that govern or provision them" (Appel et al., 2018: 3).

Following the Mojang sale, *Minecraft* would be managed in a different manner. Infrastructures evolve by developing standards and gateways that connect heterogeneous systems, according to Hughes (1993). Platforms, in contrast, can discourage interoperability because platforms achieve their success by getting independent developers and users to contribute to the platform's ecology, creating a lock-in effect that accumulates profits from both sides (Plantin et al., 2018: 9). In the case of *Minecraft*, however, this lock-in effect was not occurring in precisely same way that it did in the cases of other platforms: for-profit servers were contributing to *Minecraft*'s ecology, but they were not sharing their profits with Mojang, and the company was receiving complaints about the ways in which these severs were operating. For this reason, *Minecraft* was sold to Microsoft under the assumption that the company would be cracking down on unofficial servers, but outside of a few exceptions Microsoft was more interested in connecting servers—through the development of standards and gateways—to a single, centralized marketplace system that is under their control.

How is Microsoft pursuing an expansion strategy that borrows from the development of infrastructure? When *Minecraft* was in its initial development stage, it could only be played online through a web browser, but under Microsoft's management Persson's Java version has been replaced by a standard application version that runs on multiple platforms. At the time of this writing, versions of Minecraft have been released for platforms controlled by Microsoft, Nintendo, and Sony. In addition, Microsoft

released editions for the Raspberry Pi, Samsung's Virtual Reality headset, and the Apple and Amazon Television streaming services.

From a platform perspective, the releasing of *Minecraft* on competing platforms makes little short-term business sense for Microsoft given that they own and operate the Xbox. Historically, video game platforms are not developed with interoperability in mind because platform business models emphasize the importance of exclusive content, but when viewed from an infrastructure perspective a different long-term strategy emerges. Beginning on 20 September 2017 (Davies, 2017), Microsoft released a series of updates that transformed platform specific versions of the game (including versions on the Xbox One and the Nintendo Switch) into a single bedrock version. These updates effectively function as gateways that allow players to collaborate across platforms connected to the new *Minecraft* marketplace, which is an online marketplace that gives users the opportunity to purchase and sell in game content, provided that they agree to split their profits with Microsoft.

At the time of this writing, players running most bedrock editions can play together if they install the latest version of the game, which represents an upscaling of the game-as-service (Whitson, 2013; Sotamaa and Karppi, 2010) platform model into a game-as-service infrastructure model. There are no *Minecraft* sequels, as a result, because there do not need to be: the new standard version has not only replaced Persson's version, it has also platformized Persson's "do what whatever you want with it open" development model, transforming a game that players treated like a platform into a platform that is being monetized and managed as if it were infrastructure.

"Infrastructural changes often occur quickly and without notice, short circuiting citizen-users ability to participate in systems development," according to Parks and Starosielski (2015). This invisible nature of infrastructure, which helps explain why most changes made to Minecraft have gone unnoticed, is not only a major theme in infrastructure studies, it is also a concern shared by Emerson (2014), who identifies a ideological shift "from the 1970's philosophy of open hardware/software to the mid 1980's ideology of the user friendly via closed hardware/software" (p. 2). Given these points, one can view the convergence of platforms and infrastructures as a figurative black box that is unnoticed by most citizens. But from a cultural anthropology perspective, the invisibility of platforms and infrastructures is only a partial truth. "Invisibility is certainly one aspect of infrastructure, according to Larkin, but it is only one and at the extreme edge of visibilities that move from unseen to grand spectacles and everything in between" (p. 336). Via this logic, the convergence of platforms and infrastructures is not necessarily invisible, which is a point that a cultural ethnographic analysis of computers built in Minecraft will show.

Oh wow, logic gates!

Before Minecraft was sold to Microsoft, input from its player community played a central role in the game's evolution, following Persson's self-described open approach to development which he explained in a postmortem published in Game Developer:

> From the start, I was very open about Minecraft's development. I talked about it on forums, primarily those of TIGSource, and told people what I was doing and where I wanted to take the

> game. Fairly soon, we set up an IRC channel (live interactive text messaging) for Minecraft for more rapid discussion, and after a while, I set up a Tumblr blog to get information out to more people more easily. Discussing with the players and listening to suggestions, I learned a lot about how the game could be played and what directions were more interesting to others. Usually, people played it in a completely different way than I did. (p. 26)

In the first few months after the prototype's release, some interesting projects were created, but building was limited by the in-game materials available. Then, in January 2010, matthew102000 (2010) started "The logic gate Minecraft computer thread," an online conversation about projects which featured the basic building blocks of computation. When linked together, logic gates can form integrated circuits, like the computational infrastructures etched onto microchips, but when the thread began *Minecraft*'s limitations became evident. Connecting logic gates was cumbersome and carrying an output into an input proved to be exceptionally difficult. Snowman (2010, January 21) suggested a diode and CanadianOverlord (2010, January 21) proposed a transistor, but both required extensive terrain to function. Recognizing the setback, Jacob (2010, January 19) hypothesized a possible solution, imagining a wire connecting outputs to inputs (para. 2). Wires did not exist inside *Minecraft*'s cliché fantasy realm, but Persson (2010a, January 20) read the thread and became inspired by the problem. Nearly 6 months later, the hypothesized "wire" became a reality, after an update inviting players to "dig deep" (Persson, 2010b, July 3, para. 2).

Upon its introduction, redstone expanded the possibilities of what could be built in the game, but its addition was also controversial. Minecraft was supposed to be a fantasy role-playing game that fit into the neo-rogue subgenre (Garda, 2013), but circuits and signals were often omitted from cliché fantasy worlds. Writing on his blog, Persson (2010c, July 7) defended the changes by alluding to his fascination with the study of cellular automata. According to cellular automatists, every material and immaterial system stems from a comparatively sophisticated computation (Wolfram, 2002: 4), so instead of being entangled in complex socio-technical assemblages, programming literally creates all known systems. Code is subsequently represented as a pseudo-biological force distinguished from the signals and circuits that computational processes rely on.

One of the consequences associated with depicting an algorithm as a life-giving force is the way in which the discourse represents important ingredients in the computational process as a substitute for the process as a whole. This mystification is not explicitly political, but it implicitly coincides with the neoliberal expansion of intellectual property and the automatic granting of property rights to the programmers of software—a process which ignores the contributions of engineers and scientists who designed the infrastructure that software relies on. According to Chun (2013), depicting source code as magic effectively erases the processes of execution, and the broader institutional and technical "structures needed to ensure the coincidence of source code and its execution" (p. 21). Programming with redstone, on the other hand, demystifies the messy, complicated processes that high-level programming languages conceal by forcing players to build their own computational infrastructure. Programming is not magic, as a result, nor is it limited to the typing of code; it is an educational, mindful practice that can be appreciated in and of itself, not unlike the do-it-yourself electronics kits that inspired redstones' creation.

Instead of accelerating computational processes, computers made from redstone slow them down, aestheticizing circuits and patterns of information and creating a unique visual experience that material electronics are unable to afford. These experiences are not unlike the experiences that infrastructure can provide in instances where it is rendered symbolic, metaphorical, and visible, when "the poetics of infrastructure allows us to understand how the political can be constituted through different means" (Larkin, 2013: 329).

How does redstone engage with the politics and poetics of platforms and infrastructures? Redstone gives players the ability to program, visualize, and experiencing platforms as infrastructure because it was inspired by players interested in electronic circuitry, an interest that captured the imaginations of electronics hobbyists after the Second World War. These hobbyists were fueled by utopian visions of a more rational, technologically enhanced home, and mail-order businesses sprang up to support them, giving non-engineers access to the wires, tubes, and transistors that were powering a new wave of invention. Magazines, like Popular Mechanics, published advertisements for do-it-yourself electronics kits, which included early microcomputers (Swalwell, 2012) that had to be assembled and programmed manually, just like the hypothetical Minecraft computer that matthew102000's thread envisioned. *Minecraft* computers are not computers in a strictly material sense, from this perspective, but like the electronics kits that came before them they still traffic in signals, existing not only as a visual and mathematical iteration of a platform, but also as an interactive means of visualizing demonstrating and experiencing platforms as infrastructure. To expand on this point, it is useful to consider the example of Ben Craddock's arithmetic logic unit: one of the first computers made in *Minecraft*.

Slow programming

On 28 September 2010, Ben Craddock uploaded a video to YouTube of a project built in *Minecraft*. In under 10 minutes, the clip tours an arithmetic logic unit (ALU), which is a basic building block of computation found in all microprocessors (Craddock, 2010). Due to their micro nature, arithmetic logic units cannot be demonstrated visually, but Craddock's logic unit was different. Built using redstone, its simplicity, sluggishness, and size combine to create a unique aesthetic that privileges the contemplation of infrastructure over speed and ease of use, which is an approach to design that Hallnäs and Redström (2001) describe as slow technology.

Craddock's ALU is not only visually impressive; it also provides a unique opportunity to witness the process of computation in action. Humbly demonstrating simple addition, Craddock calculates 2 + 1 in binary notation (0:55–2:16) by manually walking through his ALU. Performing the operation is cumbersome, requiring the placing of multiple torches, but instead of compressing time, the delay adds it, providing Craddock with an opportunity to showcase the processor's bus.

Further demonstrating the educational advantages of slow programming, Craddock proceeds to reveal the aesthetic wonders of ripple carry adding (4:18–7:35), a process that contemporary micro-processors render invisible. Once again, numbers are entered by walking through the ALU and placing torches manually. Upon reaching the output, a

result is already visible, but Craddock thinks quickly, negating one of his control bits. Seconds pass and the redstone lights up, visually demonstrating an answer rippling down.

Craddock's ALU eschews efficiency as a means of aesthetically privileging the contemplation of computational infrastructure. Contradicting an industrial push toward smaller, faster, devices, it visualizes platforms as infrastructures, creating the time needed to aestheticize the patterns, circuits, and relations that material ALUs obscure. Instead of disappearing from sight when code is compiled, programming is subsequently transformed into a socio-technical, ecological relation with a rich history that is interconnected with hardware. For Craddock,

> there is not just a disconnect between regular people. Even programmers just usually take it on faith that they click a button and now all the stuff they wrote can be run. [. . .]. There's definitely this class of machines we have now that are tantamount to magic [. . .] and that's just where it stops and you don't say . . . ok . . . how does this work. What are we really doing? (as cited in Owens, 2012: 1:01:25–1:02:40)

The demonstration of Craddock's ALU was a revelation not only in the *Minecraft* community, but also in the broader ecosystem of user-generated content. Its YouTube video slowly garnered millions of views, after its release, alongside a video detailing a full-scale recreation of the starship enterprise. Soon, *Minecraft* videos were regularly listed among the most watched content on YouTube, popularizing both the game and projects created inside it.

Mods, machines, and drives

Craddock's video ignited imaginations, but it was the mere beginning of a working processor. Fully functioning redstone computers appeared, subsequently, as computer building spread through the redstone community. Hans Lemurson's (2011) Minecraft in Minecraft project hardcoded a two-dimensional fan-made version of *Minecraft* on a dedicated redstone circuit, reusing a strategy used to create arcade games and early dedicated home gaming consoles. Skipitup (2012) created a fully functioning general purpose computer, named Bluestone, that can draw a line between two x and y coordinates. Laurens Weyn (2014) produced a series of popular computers, including the Commandore 32 which runs Bresenham's algorithm:[1] one of the first-line algorithms still used in contemporary computer graphics. And n00b_asaurus (2016) created Deep Thought, an elegant 4-bit machine using an architecture inspired by the Intel 4004, which was the first commercial microprocessor released in 1971.[2] All of these projects afford a means of programming, visualizing, and experiencing platforms as infrastructure by way of redstone circuits that create new interactive experiences.

In addition to creating simple computers, players have been experimenting with more elaborate designs, but the bigger the project, the more time that it takes for redstone signals to travel. Codey Littley's hard drive (Spainhour, 2014), for example, can store 1 kilobyte of data, if the user is willing to wait between 6 and 7 minutes for retrieval, while Sethbling's Atari 2600 emulator (2016) takes 3 to 4 hours to output the 60-frames needed to form a 1-second video signal. This emulator—which is running on top of an emulator

that is running on top of another emulator—is slow to the extent that games are unplayable. But instead of rendering the platform useless, the delay extends the platforms loading time, allowing players to observe and even modify individual pixels as each game loads. This experience not only represents infrastructural change; it also embodies it affectively, providing an empowering means of experiencing an infrastructure, which was previously obfuscated, that can be restored and repurposed.

Conclusion: from commodities and services to platforms and infrastructures

What are the long-term implications of *Minecraft*'s journey from an independent game into something that is not only being valued as a platform but is also taking on the characteristics of infrastructure? *Minecraft* is a unique case which demonstrates how platforms, infrastructures, and games are being managed and experienced in interrelated ways, and how invisibility is only one part of this interrelation, which involves an emerging spectrum of visibilities and experiences. From an ethnographic perspective, redstone is giving players the time and space needed to appreciate the infrastructure that information flows through when computations occur, so a generation that is growing up playing *Minecraft* may be more attuned to the politics and poetics of platforms and infrastructures in their various technical, systemic, metaphoric, and material forms. But from a macroeconomic perspective, games (which were previously described as AAA games) are trying to replicate *Minecraft*'s platform/infrastructure model, and attention in a platform economy is a precious economic resource. At the time of this writing, *Minecraft* is continuing to scale to unprecedented heights, but it could just as easily be abandoned—like plenty of infrastructures which used to define the modern world. What is clear, following an unprecedented decade of growth, is that *Minecraft*'s status as platform and infrastructure is political, economic, social and technical, and at the same time it is symbolic, metaphoric, and speculative.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

David Murphy https://orcid.org/0000-0001-7478-0335

Notes

1. See Jack Elton Bresenham (1965).
2. See William Aspray (1997).

References

Anderson C and Wolff M (2010) The Web is dead: long live the Internet. *Wired Magazine*, 17 August. Available at: https://www.wired.com/2010/08/ff-webrip/

Appel H, Anand N and Gupta A (2018) *The Promise of Infrastructure*. Durham, NC: Duke University Press.

Apperley T and Parikka J (2018) Platform studies' epistemic threshold. *Games and Culture* 13(4): 349–369.

Aspray W (1997) The Intel 4004 microprocessor: what constituted invention? *IEEE Annals of the History of Computing* 19(3): 4–15.

Bogost I and Montfort N (2007) New media as material constraint: an introduction to platform studies. In: *Proceedings of the first international HASTAC conference on electronic technotonics: thinking at the interface*, pp. 176–193. Available at: http://bogost.com/downloads/bogost%20montfort%20hastac.pdf

Bogost I and Montfort N (2009) Platform studies: frequently questioned answers. Presented at the after media: embodiment and context digital arts and culture 2009, arts computation engineering, University of California, Irvine, CA. Available at: http://escholarship.org/uc/item/01r0k9br

Boluk S and Lemieux P (2017) *Metagaming: Playing Competing, Spectating, Cheating, Trading, Making, and Breaking Videogames*. Minneapolis, MN: University of Minnesota Press.

Bresenham JE (1965) Algorithm for computer control of a digital plotter. *IBM Systems Journal* 4(1): 25–30.

Bylos G (2014) An open letter to Notch. sterlingplays.com, n.d. Available at: https://web.archive.org/web/20140617230724/http://sterlingplays.com/

CanadianOverlord (2010) The logic gate and Minecraft computer thread (Online forum comment). 21 January. Available at: http://www.minecraftforum.net/forums/minecraft-discussion/creative-mode/classic-creative-mode/369819-the-logic-gate-and-minecraft-computer-thread (accessed 22 December 2017).

Chun WHK (2013) *Programmed Visions: Software and Memory*. Cambridge, MA: MIT Press.

Clement J (2022) Minecraft active player count worldwide 2016-2021. statista, 22 August. Available at: https://www.statista.com/statistics/680139/minecraft-active-players-worldwide/

Constantinides P, Henfridsson O and Parker G (2018) Introduction: platforms and infrastructures in the digital age. *Information Systems Research* 29(2): 381–400.

Craddock B (2010) 16 bit ALU in Minecraft [video file]. 28 September. Available at: https://www.youtube.com/watch?v=LGkkyKZVzug (accessed 22 December 2017).

Crecente B (2010) Minecraft apparently under siege by angry addicts demanding more updates. Kotaku, 21 October. Available at: https://kotaku.com/minecraft-apparently-under-siege-by-angry-addicts-deman-5669766

Davies M (2017) The better together updated is here [Web log comment]. Available at: https://www.minecraft.net/en-us/article/better-together-update-here

Duncan S (2012) Minecraft, beyond construction and survival. *Well Played* 1(1): 1–22.

Emerson L (2014) *Reading Writing Interfaces: From the Digital to the Bookbound*. Minneapolis, MN: University of Minnesota Press.

Garda MB (2013) Neo-rogue and the essence of roguelikeness. *Homo Ludens* 1(5): 59–72.

Gillespie T (2010) The politics of "platforms." *New Media & Society* 12(3): 347–364.

Hallnäs L and Redström J (2001) Slow technology–designing for reflection. *Personal and Ubiquitous Computing* 5(3): 201–212.

Handy A (2010) Interview: Markus "Notch" Persson talks making Minecraft. *Gamasutra.com*, 23 March. Available at: https://www.gamasutra.com/view/news/118662/Interview_Markus_Notch_Person_Talks_Making_Minecraft.php (accessed 22 December 2017).

Hughes TP (1993) *Networks of Power: Electrification in Western Society, 1880–1930*. Baltimore, MD: The John Hopkins University Press.

Jacob (2010) The logic gate and Minecraft computer thread. [Online forum comment]. 19 January. Available at: https://www.minecraftforum.net/forums/archive/classic-creative-mode/369819-the-logic-gate-and-minecraft-computer-thread

Kulevska A (2022) How many people play Minecraft? cybercrew, 22 June. Available at: https://cybercrew.uk/blog/how-many-people-play-minecraft/#:~:text=Minecraft%20has%20141%20million%20monthly,million%20registered%20players%20in%20China

Langley P and Leyshon A (2016) Platform capitalism: the intermediation and capitalization of digital economic circulation. *Finance and Society* 3(1): 11–31.

Langlois G and Elmer G (2013) The research politics of social media platforms. *Culture Machine* 14: 1–17.

Larkin B (2013) The politics and poetics of infrastructure. *Annual Review of Anthropology* 42: 327–343.

Lastowka G (2012) Minecraft as Web 2.0: amateur creativity and digital games. In: Hunter D, Lobato R and Richardson M (eds) *Amateur Media: Social, Cultural, and Legal Perspectives*. Florence: Routledge, pp. 153–169.

Leavitt A (2013) Crafting Minecraft: negotiating creative produsage-driven participation in an evolving cultural artifact (Selected papers of Internet research). Available at: http://spir.aoir.org/index.php/spir/article/view/798 (accessed 22 December 2017).

Lemurson H (2011) Minecraft in Minecraft [video file]. Available at: https://www.youtube.com/watch?v=gdnO1QZJ–M (accessed 22 December 2017).

matthew102000 (2010) The logic gate and Minecraft computer thread [Online forum comment]. 18 January. Available at: http://www.minecraftforum.net/forums/minecraft-discussion/creative-mode/classic-creative-mode/369819-the-logic-gate-and-minecraft-computer-thread (accessed 22 December 2017).

Montfort N (2006) Combat in context. *Game Studies* 6(1). Available at: http://gamestudies.org/0601/articles/montfort

Montfort N and Bogost I (2009) *Racing the Beam: The Atari Video Computer System*. Cambridge, MA: MIT Press.

Nieborg DB and Poell T (2018) The platformization of cultural production: theorizing the contingent cultural commodity. *New Media & Society* 20(11): 4275–4292.

n00b_asaurus (2016) Deep Thought Minecraft Computer [Video file]. Available at: https://www.youtube.com/watch?v=AiMUI9Ty6no

Owen (2014) Let's talk server monetisation! [Web log message (12 June)]. Available at: https://mojang.com/2014/06/lets-talk-server-monetisation/ (accessed 22 December 2017).

Owens P (Director) (2012) *Minecraft: The Story of Mojang* [Motion Picture]. Sweden: 2 Player Productions.

Parikka J (2011) Operative media archaeology: Wolfgang Ernst's materialist media diagrammatics. *Theory, Culture & Society* 28(5): 52–74.

Parks L and Starosielski N (2015) *Signal Traffic: Critical Studies of Media Infrastructures*. Chicago, IL: Illinois University of Illinois Press.

Persson M (2009a) In the spirit of the new economy! *Web Log Message*, 10 June. Available at: http://notch.tumblr.com/post/121044061/in-the-spirit-of-the-new-economy (accessed 22 December 2017).

Persson M (2009b) Minecraft alpha (Online Discussion Group), 17 May. *TIGSource.com*, 1 December. Available at: http://forums.tigsource.com/index.php?topic=6273.0

Persson M (2009c) What your allowed to do with Minecraft (Web log message), 8 July. Available at: http://notch.tumblr.com/post/137683043/what-youre-allowed-to-do with-minecraft

Persson M (2010a) Oh wow: logic gates! (Web log message), 20 January. Available at: http://notch.tumblr.com/post/344181442/oh-wow-logic-gates (accessed 22 December 2017).

Persson M (2010b) Seecret Friday update the third! OooooOOOoo! (Web log message), 3 July. Available at: http://notch.tumblr.com/post/763147353/seecret-friday-update-the-third-ooooo ooooo (accessed 22 December 2017).
Persson M (2010c) The rationale for redstone (Web log comment), 7 July. Available at: http://notch.tumblr.com/post/779956568/the-rationale-for-redstone (accessed 22 December 2017).
Persson M (2011) Postmortem: Minecraft alpha. *Game Developer* 18(2): 24–30.
Persson M (2014) Anyone want to buy my share of Mojang so I can move on with my life? Getting hate for trying to do the right thing is not my gig (Tweet). @notch, 16 June.
Plantin JC, Lagoze C, Edwards PN, et al. (2018) Infrastructure studies meet platform studies in the age of Google and Facebook. *New Media & Society* 20(1): 293–310.
Poell T, Nieborg D and Duffy BE (2022) *Platforms and Cultural Production*. Cambridge, MA: Polity Press.
Sethbling (2016) Atari 2600 emulator in Minecraft [video file]. Available at: https://www.youtube.com/watch?v=5nViIUfDMJg (accessed 22 December 2017).
Shaker N, Togelius J and Nelson M (2016) *Procedural Content Generation in Games*. Cham: Springer.
Skipitup (2012) Programming bluestone: a Redstone computer in Minecraft, part 2 [Video file]. Available at: https://www.youtube.com/watch?v=gFoNFWjIy_k, accessed 22 December 2017
Snowman (2010) The logic gate and Minecraft computer thread [Online forum comment]. Available at: http://www.minecraftforum.net/forums/minecraft-discussion/creative-mode/classic-creative-mode/369819-the-logic-gate-and-minecraft-computer-thread (accessed 22 December 2017).
Sotamaa O and Karppi T (2010) Games as services-final report. TRIM research reports 2. Available at: https://tampub.uta.fi/bitstream/handle/10024/65772/978-951-44-8167-3.pdf?sequence=1&isAllowed=y
Spainhour B (2014) Cody Littley's Minecraft 1K Hard Drive on snapshot [Video log file]. Available at: https://www.youtube.com/watch?v=q7clz1TPK8o
Srnicek N (2017) *Platform Capitalism*. Cambridge: Polity Press.
Swalwell M (2012) The early micro user: games writing, hardware hacking, and the will to mod. In: *Proceedings of DiGRA Nordic 2012 conference: local and global—games in culture and society*. Available at: http://www.digra.org/digital-library/publications/the-early-micro-user-games-writing-hardware-hacking-and-the-will-to-mod/
Weyn L (2014) Bresenham's algorithm working! [Video file]. Available at: https://www.youtube.com/watch?v=zMNNCPUcgIE (accessed 22 December 2017).
Whitson JR (2013) *Game design by numbers: instrumental play and the quantitative shift in the digital game industry*. Doctoral Dissertation, Carleton University, Ottawa, ON, Canada.
Wolfram S (2002) *A New Kind of Science* (Vol.5). Champaign: Wolfram media.

Author biography

David Murphy is a Senior Lecturer in the Department of Games Development at Staffordshire University and a former co-chair of the Society for Cinema and Media Studies Video Game Studies Scholarly Interest Group. His scholarly interests include game studies, communication and cultural studies, digital cultural heritage and community management and his writing has appeared in *Games and Culture, The International Journal of Canadian Studies*, and *Game Studies: The International Journal of Computer Game Research*.