Wrong Warping, Sequence Breaking, and Running through Code  
*Systemic Contiguity and Narrative Architecture in The Legend of Zelda: Ocarina of Time Any% Speedrun*

James Newman

**Abstract**

This article focuses on Nintendo’s influential and much-celebrated 1998 videogame *The Legend of Zelda: Ocarina of Time* (hereafter *OoT*). In particular, it explores the often unexpectedly creative and wholly transformative ways in the game is played by ‘speedrunners.’ Seeking to race through the game as quickly as possible, speedrunners play in a distinctive way that combines mastery of performance execution with a deep knowledge of the game’s operation and how its systems may be exploited. Speedrunning performances are creative because they involve astonishingly detailed investigations of the minutiae of game behaviors. They are transformative because they disrupt and even invert much-vaunted aspects of the game as sequences that slow down progress are circumvented. As such, what would otherwise class as crucial moments in the storyline, key character development and even locales, are not only raced through but are actively skipped. Compared with the game’s narrative, the connectivity of its spaces, and the complex but clearly mapped passage of time as set out in Nintendo’s officially-endorsed ‘Strategy Guides’ (see Buchanan et al 1998; Hollinger et al 1998; Loe and Guess 1998), the *OoT* speedrun constructs an altogether different game with vastly altered priorities. And so, the offer to, ‘Join legendary hero Link as he journeys across Hyrule, and even through time, to thwart the plans of Ganondorf.’ (Nintendo 2017), is recast as a breakneck dash to the closing credits sequence evading and avoiding all but the essential moments of gameplay.

**Prologue: A Tale of Two Games—Or Rather, Two Tales of One Game**

**Version 1.**

“A long time ago...
Before life began, before the world had form, three golden goddesses descended upon the chaotic land of Hyrule. They were Din, the goddess of power. Nayru, the goddess of wisdom and Farore, the goddess of courage. Din, with her strong flaming arms, cultivated the land to create the Earth. Nayru poured her wisdom onto the Earth to give the spirit of law to the world. Farore’s rich soul created all life forms who would uphold the law. These three great goddesses returned to the Heavens, leaving behind the golden sacred Triforce. Since then, the Triforce has become the basis for Hyrule’s providence. Where the Triforce stood became sacred land. In the vast, deep forest of Hyrule, the great Deku Tree served as the guardian spirit. The children of the forest, the Kokiri, lived with the great Deku Tree. Each Kokiri had his or her own guardian fairy. Except one. His name was Link” (Nintendo 1998, 5-6). “Join legendary hero Link as he journeys across Hyrule, and even through time, to thwart the plans of Ganondorf. Wield incredible weapons and items, battle ferocious bosses, and solve brain-teasing puzzles in this fan-favourite chapter from the series” (Nintendo 2019)

Version 2.
A childlike character dressed in green and armed only with a wooden stick and an empty bottle runs and glides backwards, sometimes striking at walls and rebounding off them and sometimes passing through them as though they weren’t there. A short battle with a monster ensues after which the child steps into a pool of blue light and is warped to a collapsing castle through which they effortlessly glide (again, backwards), before dispatching a towering enemy with their trusty stick. Roll credits.

The first version is as told in the instruction manual for Nintendo’s The Legend of Zelda: The Ocarina of Time game and on its corporate website. This is an epic tale of heroism, coming of age and time travel played out over complex and expansive landscape and taking more than 60 hours of gameplay to complete (Edge 2001, 70). The second version of the story is as told by the Any% speedrunner and involves no ocarinas, no time travel, no Triforce, and a sword only in the final seconds (which is knocked out of the player-character’s hand by the final enemy even though it was not collected during the game). This second version plays out in just over 17 minutes of gameplay.

Introduction
This article focuses on Nintendo’s influential and much-celebrated 1998 videogame The Legend of Zelda: Ocarina of Time (hereafter OoT). In particular, it explores the often
unexpectedly creative and wholly transformative ways in which the game is played by “speedrunners.” Seeking to race through the game as quickly as possible, speedrunners play in a distinctive way that combines mastery of performance execution with a deep knowledge of the game’s operation and how its systems may be exploited. Speedrunning performances are creative because they involve astonishingly detailed investigations of the minutiae of game behaviors. They are transformative because they disrupt and even invert much-vaunted aspects of the game as sequences that slow down progress are circumvented. As such, what would otherwise be classed as crucial moments in the storyline, key character development, and even locales, are not only raced through but are actively skipped. Compared with the game’s narrative, the connectivity of its spaces, and the complex but clearly mapped passage of time as set out in Nintendo’s officially-endorsed “Strategy Guides” (see Buchanan, Leung, and Shined 1998; Hollinger, Ratkos, and Tica 1998; Loe and Guess 1998), the OoT speedrun constructs an altogether different game with vastly altered priorities. And so, the offer to “[j]oin legendary hero Link as he journeys across Hyrule, and even through time, to thwart the plans of Ganondorf” (Nintendo 2019) is recast as a breakneck dash to the closing credits sequence, evading and avoiding all but the essential moments of gameplay.

Nowhere is this reconfiguration of the game more apparent than in the “Any%” speedruns that are the focus of this article. Any% running permits the use of any technique available in the game, prohibiting only external hardware hacking/cheating devices (such as those noted by Consalvo 2007) or modifications to the joypads used for play (e.g., adding automation to negate, augment, or override the player’s inputs). Any% performances comprise finely tuned techniques expertly executed with frame-perfect precision. The extraordinary levels of virtuoso skill on show will be evident even to those unfamiliar with the games being played and ensure that these performances fall squarely into the category of “YouTube-worthy gameplay” that Postigo (2014, 342) notes. This is gameplay so comprehensively expert in its execution that it warrants watching not merely as a tutorial but as entertainment in its own right. As such, it is no surprise to find that Any% speedruns have become widely spectated online, with events such as Games Done Quick (https://gamesdonequick.com) consolidating and giving yet more visibility to an already increasingly mainstream practice.

The Any% speedrun is by far the quickest of any category and reduces what contemporary reviewers (e.g., Edge 1998) calculated at over 60 hours of gameplay (not
including detours and sidequests) to considerably less than 20 minutes. This systematic—and systemic—exploitation of the game’s operation gives rise to quite astounding reconfigurations of the game that see OoT’s much-celebrated narrative and structure comprehensively transformed, as plot sequences are skipped and spatialities are deconstructed and reconstructed through highly considered research and skillfully manipulative play that modifies the very code of the game in real time. While there is self-evident refinement and precision in the performances and the way the game is consumed, such a contraction is only possible by virtue of an extensive reworking of the fundamental operation of the game and the ways in which narrative, space, and time are (re)produced.

However, there is more to speedrunning, and Any% speedrunning in particular, than playing quickly. In addition to being honed examples of expert play, these performances actively and extensively exploit inconsistencies or glitches in the game’s code. Any% speedrunning is not simply a matter of memorizing a route through the game’s space and masterfully executing the techniques and maneuvers that that game’s system makes available in order to progress. Rather, the practice involves mining the ludic potential of the game to reveal new behaviors and affordances unanticipated by the game’s developers and recalling the inherent imprecision of technologies created by humans, as noted by Cascone (2000). This facet of speedrunning also summons Virilio’s discussion of the immanence of the error or accident in reminding us that every technology implies and presages its own accident. To invent the ship is simultaneously to invent the shipwreck; to invent electricity is to bring about the existence and inevitability of electrocution (Lotringer and Virilio 2005). The practice of Any% speedrunning, then, is one that exploits the inevitability of the glitch and the crash, and yet it does not represent a fetishization of failure. Any% runs do not simply foreground the revelation of coding inconsistencies for their own sake. The recordings of Any% speedrunning performances are far from video-captured complications of bugs and glitches in games. Instead, the operational “failures” of the code are analyzed, marshaled, brought under control, and transformed into new forms of artistic and ludic practice. These are radical reworkings of games whose newly defined contours are contingent on techniques and opportunities made available through the creative exploitation of the materiality of the code and whose trajectories emerge from errors, inconsistencies, and quirks in the ways data are stored and flow through the system.

By investigating the development and execution of Any% speedrunning performances and code-mining techniques, I hope to show how OoT’s narrative
architecture and tempo-spatiality are overridden by the harnessed and manipulated logic of the game’s system and simulation model. Through these forms of play, I argue that OoT’s ludic/narrative space is transformed to a point where the coherence, navigability, and, vitally, the contiguity of OoT’s code become utterly preeminent. Broken apart and reassembled, doors that once opened into a given room now warp the player across the time and space of the game’s mythical storyworld. In doing this, previously unthinkable and putatively discontiguous connections not only are formed, but become the dominant traversable route.

It is essential to bear in mind here that, while they subvert the game’s space and time as set out in its official guidebook, these new tempo-spatial constructions are produced by the very same code, albeit influenced and manipulated in creative ways. Ultimately, I argue that in order to move from the start to the finish of the game in as speedy a manner as possible, the Any% speedrunner is better thought of not as charting a route through the representational space produced by the game’s engine, or along its narrative arc, but rather as traversing and pathfinding through the game’s code and data structures. In playing with the game in this manner, the Any% runner short-circuits the existing narrative and representational logic of the game and creates new connections, new stories, and new spaces, all of which are available in the original game but which are only revealed through the calculated and deliberate manipulation of code and data.

**Rationale**

Despite a near-unanimously positive critical reception since its release in 1998 for the Nintendo 64 console platform, it is surprising to note that there have been few scholarly studies of OoT. Consalvo’s (2003) early work on “Zelda 64” fandom, Cuddy’s (2008) philosophical analysis of the Zelda series, and Sivak’s (2009) discussion of the way OoT’s world and myth are reinforced by its gameplay mechanics are notable exceptions. The game is one of what remains only a small handful of titles to be awarded maximum review scores by specialist videogame magazines such as *Edge* (UK) and *Famitsu* (Japan). Indeed, such are the critical plaudits, a review from the aggregation service Metacritic notes that

> In the history of Metacritic, only one game has ever achieved a score of 99: this fifth *Legend of Zelda* release, which was the first to appear on the Nintendo 64 (as well as the first to feature 3D graphics). *Ocarina of Time* routinely appears on (if not at the very top of) every all-time best games list, and seems to resurface on each new
Nintendo platform (including the 3DS, where a 3D remake is the highest-scoring release to date). (Dietz 2017)

Since its release, the game has continued to enjoy consistent critical and popular favor and routinely tops lists of influential games. Indeed, such are its innovations in storytelling, level design, and control method that GameTrailers (2006, 18:51) labeled it a “walking patent office.” Without doubt, this is a popular and critically lauded game and, given the amount of working and reworking undertaken by players over nearly decades, it is clear that scholars have much to learn from studying the title as well as the activities and cultures supporting and sustaining it.

In a similar vein, while the speedrunning that shapes these investigations has been a prevalent part of gaming culture since the very earliest days of the medium (see Newman 2008, 123-148), this “practiced practice,” as Scully-Blaker (2014) puts it, has received comparatively little sustained scholarly analysis. Notable exceptions include Lowood’s (e.g., 2006) work on high-performance play and “Quake Done Quick” as an “experimental physics” playground; Parker’s (2008, 35) analysis of Halo “Jeep Tag” exploring the ways players design and impose new rulesets on games to extend playable opportunity; and Newman’s (2005; 2008, 124) on speedrunning as a constitutive or “configurative” (following Eskelinen 2001) act of “superplay” combining performance prowess and reverse engineering. More recently, in addition to work on the codification of speedrunning technique (e.g., Newman 2016, 2017), Scully-Blaker (2014) has offered some initial typological and terminological distinctions to aid in the differentiation and analysis of the practice. As such, at least part of the drive to undertake this research is to contribute to the further development of scholarship on OoT and on speedrunning as a practice involving expert play, detailed research, and the production and consumption of new forms of gameplay. Beyond this more general ambition, however, the rationale for the focus on OoT Any% speedrunning as a distinctive category of superplay is twofold.

First, in the online forums and discussion boards of SpeedDemosArchive (hereafter SDA) and ZeldaSpeedRuns (hereafter ZSR), OoT has an extraordinarily rich and active community. These remain active today and are the engine room of discovery and research on the game’s ludic opportunity. It is crucial to note that while a speedrun is typically performed by an identifiable individual, the work is typically the product of a network of contributors each supporting the endeavor in different ways (see also Jenkins 2006 on collective intelligence and Morris’s [2003] discussion of the extent of
community-created tools). While ongoing, there are noticeable spikes of activity following the discovery of new OoT techniques that reveal the workflows of speedrunning. As Wright, a key OoT speedrunner notes,

In 2006, I started frequenting the Speed Demos Archive forums—a place where people interested in speedrunning would gather and discuss strategy. The single largest game thread on SDA belonged to *Ocarina of Time*. Discussion mainly revolved around experimenting with the game. In fact, very few speedruns were actually completed back then. It felt more like we were doing science than anything else. This was spurred on by a user going by the name of Kazooie posting videos of some clever jumps that weren’t thought to be possible before. These jumps allowed us to access certain areas in the game sooner than intended. (Wright 2014, 0:44)

Importantly, as we have suggested above, the knowledge generated by these “Zelda Scientists” does not simply recount observably successful ways of playing but digs deep into the operation of the game to apprehend and exploit it at a systemic level. Most crucially, the insight and collective intelligence of the Zelda speedrunning community is well documented, with SDA and ZSR hosting myriad materials detailing the specificities and systemic operation of techniques, charting the ongoing analyses and debates that give rise to and refine them, and showcasing video-captured performances of the top speedruns across all categories of gameplay (and across other Legend of Zelda titles, hereafter LoZ).

Second, by zooming in on a particular category of speedrun, I hope to add to the scholarly discussion of the practice more broadly. Even a cursory examination of ZSR’s leaderboards or the information on the twelve different “routes” through the game immediately reveals that “speedrunning” is not a singular practice or undertaking. Rather, there are numerous related but discrete categories, each making use of and, just as importantly, disallowing specific techniques and exploits as well as requiring (or not) the completion of certain sequences of gameplay. It is essential to note here the way in which the spirit of exploration that pushes at the boundaries of gameplay potential, and which is unambiguously celebrated in these sites, is simultaneously tempered by a regulatory tone that strictly defines the rules of engagement. While this might be read as an attempt to provide a level playing field upon which competitive speedrunning can operate, there is nonetheless a concerted attempt to police the categories. Of course, such definitions are debated within the speedrunning community. The 100% category
might seem straightforward enough in requiring the player to complete not only the game’s main quest but also each and every additional mini-game, detour and side-quest. However, the recent acceptance of another category of 100% run (known as “100% No Source Requirement”) speaks to the fluidity of classifications. Rather than demanding that each item or object be acquired from their original location (or ‘source’), this new ‘100% NSR’ run allows the use of bugs, glitches and techniques not permissible in the previous 100% category (ZFGa 2016).

The difficulty of satisfactorily defining a “glitch,” for instance, creates obvious issues for those attempting to delimit the contours of a “glitchless” category of run or deciding which glitches will be permitted. However, it also problematizes the application of even more nuanced scholarly distinctions such as those proposed by Scully-Blaker (2014) between “finesse runs” and “deconstructive runs,” as so much of what we see, in the Any% run and many other categories, is contingent on both. It is essential to note here that “the glitch” represents far more than an aesthetic alteration or aberration. Rather, any given glitch might foundationally affect the operation of the game and the performative potentiality of gameplay. That being the case, I wish to suggest an alternative way of analyzing the speedrun based on three categories of activity: “hidden affordances”; “exploiting inconsistencies”; and “manipulation and reconstruction.” Each category implies different research, development, and performance techniques and involves different ways of engaging with the game as a ludic, representational, and systemic entity. However, in keeping with the regulations of the OoT Any% speedrunning competition, each category of activity relies on skillful and creative manipulation of the code and data structures rather than their modification or alteration. These are performance practices rather than “mods” (software modifications that add new code to expand and alter the underlying functionality of the game system). As such, while different in terms of the methods and outcomes employed, each category of Any% speedrun activity, no matter how transformative and disruptive, can be seen as part of a process of revelation and discovery or of playing with the game (Newman 2008).

Some notes on methodology
This research forms part of a larger-scale analytical and methodological project concerned with investigating videogames as objects of both design and play. By demonstrating the ways games were played and played with, these materials hosted at SDA and ZSR offer a fascinating and instructive basis for comparison with design
documentation and officially sanctioned accounts of game(play). “Designerly intent” is, obviously, difficult to assert. However, I seek to reveal it through its codification in instruction manuals and “authorized” or “official” publications such as Strategy Guides which chart “preferred” routes through the game. With claims such as “All Hidden Items Revealed” emblazoned on their covers, these materials define “preferred” routes through the game and legitimize specific ways of play (see Hollinger, Ratkos, and Tica 1998; Loe and Guess 1998). Scrutiny of the materials at ZSR and SDA shows how games can be played, how their intended secrets have been revealed and exploited, and how new secrets are revealed and added to OoT’s ludic roster. As well as extending the scope of “All Hidden Items,” the insight and performances we see here reconfigure the very textuality of the game through transformative acts of research-informed play.

Given the importance of systems and data in developing the Any% run, it is important to consider the ways in which players encounter and engage with the videogame code created by developers and to conceive this as a key site for interaction, negotiation, and the construction of meaning and ludic potential. My interest here is not simply thinking about the way code operates as the framework within which gameplay is constrained and contained but rather to consider how it is rendered as malleable material with which to reshape and even create new gameplay opportunities and experiences. Importantly, the way in which code and data are used by players varies considerably and the manipulations and applications may involve sophisticated analyses born of the scrutinization of listings and decompiled data. Alternatively, but just as significantly, it may involve equally insightful explorations of systemic operation and ludic potential enacted through play and conducted with a joypad/controller rather than a keyboard and text editor. Thus, the player practices that are the focus of this project are not necessarily the analogue of scholarly code studies (though some practices share much in common with it) but it is certainly the case that the practices we focus on in this article involve players, fans, and researchers in the most complex analyses of the operational effect of codes, systems, and algorithms along with the creative manipulation of data.

Additionally, analysis of the later 3D re-release of OoT for Nintendo’s handheld 3DS platform offers further insight as the patching (fixing) of certain exploitable in-game behaviors in this new game helps us assert their status as unintended glitches in the original. Most significantly, this research makes extensive use of recordings of Wright’s live Any% performance at “Awesome Game Done Quick 2013” (Wright 2013) and the “18:10” Any% speedrun (Wright 2014). Performed on 20 April 2014, the
18:10 run is over a minute slower than the current Any% record time of 17:01 set by Torje in March 2019 (Torje 2019) and necessarily does not include some of the most recently discovered techniques. However, I draw on Wright’s runs partly in recognition of their significance within and beyond the OoT speedrunning community (having garnered specialist and mainstream news coverage: see CNN 2014), and because the original 18:10 video has been augmented with an explanatory audio commentary overdubbed by Wright. This narration offers extraordinary insight into the planning, research, and ultimate execution of the run. Here, I both offer a critical reading of this distinctive performance and use the recording(s) and commentary as a codification of a particular subset of techniques developed by the community of “Zelda scientists.”

**Why speedrun?**

Before proceeding with the detailed discussion of Wright’s run(s), it is useful to consider the motivations for such an extreme and intensive form of superplay. The introduction to the 18:10 commentary offers one reading.

Have you ever played a game so much that you feel like you’ve run out of replay value? Some people choose to come up with additional challenges, such as beating a game as fast as possible, scoring as many points as possible, or pushing a multiplayer game to its competitive limit. Ocarina of Time has no scoring system. It’s not multiplayer. It is a single-player journey through an epic world. So, how fast can someone beat it? (Wright 2014, 0:07)

Immediately obvious here is the way that completion of the game, which it must be remembered is a not inconsequential task (hence the availability of Strategy Guides), is taken as read. Speedrunning, then, is unambiguously coded as an activity predicated on and requiring a comprehensive mastery of the game and these explanations certainly speak to the degree of labor involved in attaining such a status. However, while the desire to eke out every last drop of available gameplay is consistent with activities we see elsewhere in gaming culture, such as in the production of walkthroughs or fanfiction (see Burn 2006), it rather underplays the depth of understanding and knowledge that underpins the practice and that it offers back to the community of players and analysts. The introduction to the OoT speedrun wiki gives a good account of the extent of the research and the desire to explore the game.
We’ve discovered endless glitches, tricks, sequence breaks, exploits and more leading to OoT being beautifully broken in ways no other game is. There’s a running joke among OoT runners that says anything is possible in this game. As we discover new things, this has proven to be true. We can kill enemies by idly standing near them, we can clip through walls, float on thin air, warp from the first dungeon to the last, duplicate items, obtain light arrows in the deku tree, even literally write ones and zeros to our inventory to give ourselves new items. That barely scratches the surface. (OoT Speedruns Wikia n.d.)

Taking on board the transformativity of these actions, I suggest that this practice can also be understood as a critical reexamination and remaking of the game that fundamentally apprehends and performs it in ways not only distinctive and refined but wholly unrecognizable from the preferred playing codified in Nintendo’s materials.

Reading the Any% Speedrun: Hidden Affordances

One of the most immediately striking aspects of Wright’s run is that it appears to be performed backwards. That is, rather than adopting the usual viewpoint in which the game’s camera is placed behind Link (the player’s onscreen character) and the action is performed running “into” the screen, here Link faces forward with the camera directed towards the vanishing point as usual, but instead of purposefully striding forward he runs backward “toward” the screen, apparently stuck in reverse gear. As if controlled by Marshall McLuhan, Link marches “backwards into the future” (McLuhan, Quentin, and Agel 1967, p. 75). In fact, scrutiny of other Ocarina of Time speedruns across all categories reveals that this technique is near-universally employed.

The effect is certainly an unusual one and makes for an initially difficult viewing experience. Without the same intimate knowledge of the routes and pathways through the starting point of Kokiri Forest, the journey charted is a disorienting one. Gone are the sprawling vistas stretching out ahead. That which is in front of Link is space already consumed. In this playing of the game, Hyrule’s space is only made visible once it has been successfully traversed. Given the difficulties such movement and camera positioning bring for both player and spectator, we might reasonably ask what its purpose is. Is this simply showboating, or a demonstration of the considerable feat of memory and manifest mastery of the game that does away with the need to survey its contents? Certainly, those who have seen the myriad “blindfold speedrun” performances at AGDQ will have some sympathy for such a reading. Perhaps it is a particularly
sophisticated spatial reference to the game’s time-traveling theme that substitutes future for past, or maybe the continual backwards motion arises from an additional rule superimposed on the game to ramp up the challenge still further? Each of these explanations carries some weight and, viewed through the lens of duGay’s (2007) notion of “personhood,” we can see a conspicuous desire in the speedrunner to perform as an expert. However, once again, there is an altogether more functional explanation born of an intimate knowledge of the operation of the game’s fundamental systems.

| Movement Type                      | Movement Value | Notes                                                   |
|------------------------------------|----------------|---------------------------------------------------------|
| Supersliding/Hyper Extended Superslide | 27             | This is the fastest known method of travel               |
| Epona: Sprinting                   | 20             | Decreases linearly to normal over 5 seconds.             |
| Extended Superslide                | 14             | ---                                                     |
| Kokiri Boots: Backwalking/Rolling  | 13.5           | This is the fastest normal method of movement as an adult.|
| Epona                              | 12-14          | Cycles between speeds over a period of time.             |
| Sidehopping                        | 12.75          | Value is same regardless of boots or age.               |
| Child Link: Backwalking/Rolling    | 12.375         | ---                                                     |
| Hover Boots: Backwalking/Rolling   | 12.375         | Larger acceleration delays make this a non-preferred method of travel. |

Table 1: (“Movement Speeds” [excerpt], 2016)

Quite simply, owing to a quirk in the way the game’s movement routines are calculated and implemented, running in reverse (or “backwalking” as OoT speedrunners call it) is quicker than running forward. As ZFG (2016b) explains during a live OoT Master Quest 100% run, the specific use depends on a number of factors. Where distances can
be traveled in four or fewer rolls, backwalking is optimal, and beyond that it is preferred even accounting for the extra time taken to turn Link around.

In fact, neither backwalking nor rolling is the quickest way to move across Hyrule’s polygonal 3D spaces, and for those wishing to explore the full range of motions available within the game, ZSR hosts a detailed table analyzing every possible option. Attributed to CHAOSV1 (“Movement Speeds” 2016), the table lists 19 different movement types with notes on uses, limitations, and speed values as observable in the game. These include the “Hyper Extended Superslide” (the fastest means of movement in a straight line, which is offset by limited maneuverability) and consider the effect on movement of equipping different boots or Link’s Child/Adult status. The following table is an excerpt of ZSR’s movement speed data analysis, with detailed notes including recognition of the acceleration delay within the game’s engine (making these the maximum speeds attainable) and the game system’s internal misreporting of the speeds as 2/3 slower (which is assumed by Zelda scientists to be a function of OoT’s use of a modified Super Mario 64 game engine that ran at 30 frames per second rather than OoT’s 20 frames per second).

Scrutiny of these data reveals an intimate knowledge of the game’s operation born of close scrutiny and comparison as well as fine-grained commentary on how the technique might be utilized in the game. ZSR’s discussion of the Extended Superslide (attributed to Swordlesslink) is a case in point.

[ESS] is slightly slower than the full Superslide, but has the advantage of being able to turn in any direction. Another advantage is the ability to pull out bombs or bombchus while you are doing an ESS. This allows for a quick set up to another superslide or another trick. The direction of the ESS can be determined based on the direction you tilt the joystick. It also acts like the Superslide and Ground Jump, in that it will skip certain cutscenes such as the owl texts. (“Extended Superslide” 2018)

Wright’s 18:10 run ably gives expression to this abstract knowledge as Link chains together combinations of sidehop, backflip, forward roll, and backwalk using each only where they are the most efficient choice. These choreographed performances that traverse the terrain and bring Link to specific points at calculated angles of attack are called “setups” in speedrunning parlance and are essential for success. Key to this is making use of OoT’s “z-targeting” system so widely noted by reviewers as innovating
3D combat. When used outside combat, this allows Link’s facing direction to be locked relative to the scenery, thereby defining a precise orientation. Performing often lengthy sequences of jumps, rolls, and sidehops while setting and resetting the z-targeting, complex movement comes under precise control. The key here is that each movement, whether side-hop, roll, or backflip, covers a predictable amount of ground. When combined with the rotational locking that z-triggering enables, Link can be maneuvered into the specific, pixel-perfect positions often required for subsequent actions.

At the bottom here, I actually jumpslash into the loading zone because it sets Link’s speed to zero in the next area which saves me a roll. By jumpslassing this wall in Hyrule Field I get a lot of backwards speed in the water. If I hold the control stick slightly off center, I can keep this backwards speed on the land using the z-button to either lock the angle or rotate Link. During this slide, Link is considered “busy” so that Gaebora’s annoying text is skipped en route to Kakariko. (Wright 2014, 5:47)

Clearly, such setups and routes are not arrived at simply. Wright’s introduction to the commentated 18:10 run notes that, even once the core techniques and strategies were known, it would take a further two years’ worth of refinement, optimization, and route planning to assemble the final run. Interestingly, not all of this planning involves performing breathless dashes, hops, and leaps, as there are yet faster ways to travel across Hyrule than even an Extended Superslide.

Just under halfway through the speedrun, having completed a particularly frenetic sequence of gameplay set in the location of Kakariko Village and involving the collection of chickens (a signature motif in LoZ titles), Wright’s performance takes a quite unexpected turn as the gameplay is frozen and the action restarts. It is as though the game has crashed and reset. Knowing that we are witnessing the then-world-record speedrun makes this unlikely, as runs are restarted routinely for performance imperfections far slighter than this. In fact, this apparent interruption to the otherwise breakneck speed of the speedrun is quite deliberate and just as much a part of the carefully planned strategy as any other maneuver.

I save and reset the game. This takes me back to Link’s house in the forest, which is faster than traveling back across Hyrule Field. (Wright 2014, 8:30)
And so, what appears at first glance to be a pause in the otherwise speedy gameplay is, in fact, a creative traversal of the game’s space. The “savewarp” (and the “deathwarp” variant), as speedrunners call it, takes advantage of the machinery of the game’s system to literally reset the player’s position to a given point. While this point cannot be directly manipulated, as the reset point is selected by the game system dependent on the position of the player at the time of saving (typically repositioning at the entry point to a specific location on the game’s map such as the start of a “dungeon”), these “respawning” points are entirely predictable, and thus can be used to the player’s advantage. As such, while the obvious (and literally, graphically represented) path from Kakariko to Kokiri Forest is across Hyrule Field, it is simply not as quick a route as via the save screen (even running backward!). In fact, the OoT’s Pause feature has a number of uses for speedrunners and allows the “pre-inputting” of commands which are executed at the precise moment the game is unpaused, thereby aiding the frame-accurate performance of certain techniques.

Here, then, we get an important glimpse into the way in which these practices of superplay deconstruction eschew the “journey of Link set in the epic world of Hyrule” in favor of the creation of a new journey through the system of OoT that exploits the particular characteristics of its game system. Even at the risk of rendering moot their own sobriquet, if it is speedier to warp via the save screen than to move through the 3D polygonal space of the gameworld, the speedrunner does not run.

Again, if we wish for further evidence of the meticulousness of the research, the importance of every last fraction of a second, and even the impact of the specific version of OoT used in the performance (reminding us of the “instability” and “extreme fragmentation” of videogame texts: see Newman 2012, 123–24; Giordano 2011), Wright’s discussion of the analysis and comparison of savewarping as a means of spatial traversal can surely leave us in no doubt.

This reset speed is rather slow because I’m on the Chinese version. Though the Chinese version has slow resets, it makes up for it in lag and text length. I actually spent many hours timing the fastest versions of the game to figure out which was optimal. (Wright 2014, 8:37)

Thus far, the techniques we have encountered, while perhaps counterintuitive in their utility and creatively harnessed, arise from the design of the game’s engine and the coding decisions informing its implementation. And so, while it might not have been
anticipated by *OoT*’s designers and developers that the comparatively high velocity of backwalking or side-jumping would see them utilized in favor of walking forward (and certainly, there are no images or guidance in authorized Strategy Guides or instruction manuals even hinting at this), such movement is entirely consistent with the game mechanics enshrined in *OoT*’s engine. Even the use of the save function to effectively move across space is born of a (highly imaginative) use of the available tools.

As Squire and Jenkins (2002) remind us, in one sense, nothing exists in any gameworld that was not in some way placed there, or whose placement is not in some way enabled, through design.

> Game worlds are totally constructed environments. Everything there was put on the screen for some purpose—shaping the game play or contributing to the mood and atmosphere or encouraging performance, playfulness, competition, or collaboration. (Squire and Jenkins 2002)

Of course, representationally, backwalking is an almost absurd sight and is an extremely eloquent and immediately readable indicator of this very particular approach to gameplay. Similarly, save- and death-warping not only appear unintuitive techniques but positively reconfigure the boundaries of in- and out-of-game. Indeed, in citing the work of Hans (1981), who draws attention to the way in which play pushes at the boundaries of the rules that appear to confine it, Salen (2002) encourages us to think of play as “transformative” in precisely this way. In doing so, we usefully begin to chart a path between the apparent formality of the design process that, as Squire and Jenkins (2002) note, gives rise to the system and its rules, implementations, and representations, and the player as an agent working both within and at the margins. While there surely was a “purpose” that governed the inclusion of every aspect of the simulation and that placed every item on the screen, the transformative nature of play recasts this, making it malleable material whose affordances support myriad uses that need not reference nor be concerned with any originating intention.

Transformatively and enabling though backwalking or savewarping may be, however, as they do not interfere with but rather reveal aspects of *OoT*’s simulation that are always already present (no matter how unobvious in their functionality), I categorize such techniques as “hidden affordances.” By this, I mean to invoke Murray’s (2011) and Norman’s (1999) explorations of the relationships between the attributes of a (designed) object and the actors making use of it. In particular, I wish to argue that what we see in
the performances of backwalking and savewarping is both a dissociation of gameplay and representational systems, and an apprehension of the game system as a disassembled collection of opportunities rather than a definitive or stable specification. That these techniques are not promoted or codified into the paratextual framing of OoT’s gameplay quite demonstrably has no bearing on their status as affordances of the game’s engine. The research, comparison, and planning of implementations for specific sequences of gameplay are all, I suggest, processes of evaluating and revealing hidden affordances. Such work speaks of an approach that encounters all input control as malleable material for play that is unbound by concerns about representational or narrative consistency but, rather, is guided by the primacy of the speedrunning objective and enabled by the game system’s underpinning simulation model.

However, where the revelation of these hidden affordances of control inputs may represent a re-framing of design “purpose,” there is another distinct category of technique we see deployed in speedrunning that seeks not only to explore the margins of possibilities but to push far out of bounds.

**Reading the Any% Speedrun: Exploiting Inconsistencies**

If savewarping plays with the margins of what is inside and outside the game by incorporating save/restart functionality into real-time gameplay, the practice of “clipping” extends this exploration by rendering permeable the putatively solid structures of OoT’s 3D polygonal space. In short, as ZSR puts it, “Clipping allows one to escape the normal playing bounds of the game” (“Clipping” 2018). Because of the way 3D gameworlds are constructed, clipping through an apparently solid wall or door does not lead the player to immediately lose a life and have their position reset in-bounds. Instead, players may find themselves in spaces never intended to be seen, spaces that connect those seemingly far apart in the gameworld, or perhaps constructed solely by the act of being in them (see also Newman 2017 on the procedural generation of “The Minus World” in Nintendo’s Super Mario Bros). There are a number of ways to clip through spaces in OoT, as the advice at ZSR indicates. One method involves the precision-timing of movements we have noted above.

For nearly any angle less than 90 degrees, a jumpslash will be able to push you through one side of the wall. For angles a jumpslash can’t get through, a superslide usually can. Other ways to clip through acute angles include recoil boosting and
Hyper Extended Supersliding, Superslides can get through some 90 degree angles, but only if one of the two walls is an actor. (“Clipping” 2018)

Regardless of the means of execution, each of the techniques used for passing through otherwise impassable spatial dividers relies on the specificities of real-time 3D graphics engines and, crucially, the peculiarities of the N64’s graphics hardware and OoT’s graphics routines. As the discussion above notes, it is the seams between adjacent polygons that are the key. And through these vulnerable vertices that sit between the polygons comprising OoT’s world, Link can pass. In representational terms, the gaps are not big enough to allow Link in either Child or Adult form to “authentically” pass through. However, by approaching and moving at these virtual seams in precise manners, Link can breach them and thereby move into the unrendered but traversable liminal zones that should be protected from view and access. It is as though the whole game is a virtual film set constructed from a series of flats between which exist gaps almost imperceptible to the eye but large enough to allow passage. Of course, that a player can peer behind the scenery does not tell us why a speedrunner would wish to. As we have seen, with so much of speedrunning dedicated to reducing the amount of space consumed, why produce yet more?

One answer is found in the “Forest Escape” that allows Wright to exit the first locale in OoT significantly sooner than the game’s linearity ordinarily allows, thereby avoiding many challenges, trials, and cutscenes. What is important about this is that the technique relies on an astonishingly intimate knowledge of OoT’s game engine and the relationship between different structures as they are represented in code and made visible in polygons (see Bainbridge and Bainbridge 2007, 64–65, on the distinction between software’s “display” and “world” models).

Drawing on the extraordinary detail presented at ZSR, we learn that what we think of as the locales in OoT are actually comprised of a number of key elements: Scenes, Rooms, and Actors that interact in specific ways. By way of example, Kokiri Forest, where Link’s adventure begins, is actually one “Scene” comprising three “Rooms” (“the village”; “the maze”; “the Great Deku Tree meadow”) that the game dynamically switches among depending on which Room Link is “in.” Importantly, Scenes also contain a “collision mesh” which specifies the entry and exit points, and defines the solidity/permeability of objects for every Room within the Scene. Crucially, this mesh is always loaded in its entirety regardless of which Room is currently displayed on screen. Therefore, the exit points for all Rooms are actually loaded at all
times but, by virtue of strategically positioned objects or characters (Actors), only those relating to the currently loaded and visible Room are intended to be accessible. However, by clipping through gaps in the visible scenery or past the obstructive Actor in one Room, it is possible to access points on the collision mesh of another unloaded Room. As ZSR puts it,

For example, if Kokiri Forest’s “village” room is loaded, walking into where the Deku Tree’s mouth would be will still transport you to the Inside the Deku Tree scene.

Exits can be thought of as being always “on”; only geography and actor placements are used to block access to them. (“Zelda 64 Engine” 2014)

Wright’s discussion of the research leading to the revelation and refinement of these techniques demonstrates a move beyond the observation of gameplay and a more systemic interrogation of the game’s operation.

To figure out those exact angles for the Forest Escape, I had to use memory-hacking software to read Link’s exact facing direction and X Y Z position, then gather a lot of data to find the absolute ranges in which a sidehop jumpslash actually goes through the wall. And from there come up with a way to get Link into that range quickly. (Wright 2014, 5:03)

Techniques that create “out-of-bounds” clips are widely used and highly prized by speedrunners as, like savewarping, they offer opportunities that transcend the imposed limitations of the game’s representational space and begin to chart routes behind, beneath, and beyond. In performing such actions, space that is elsewhere lauded for the way it combines “…virtually any kind of geographical feature you could imagine” (Edge 1998, 84) is immediately rendered consistent only by virtue of the player’s ability to pass through it. The linkages newly created in code give contiguity between locations hitherto unconnected on the game’s map of its representational space.

A key feature of techniques such as clipping is that they engage with the game rather differently from, for example, backwalking and savewarping. Where those techniques rethought the use of techniques always available and documented in OoT’s instruction manual, here we see the exploitation of inconsistencies in the game’s code—the exploitation of unintended glitches. Taking as our example the “Infinite Sword
Glitch” (ISG), another mainstay of OoT speedrunning which allows the player’s sword (or wooden “Deku Stick” in the case of the “swordless” Any% runs) to be constantly active, thereby maximizing the damage that can be inflicted on enemies, we can say with confidence that this behavior is unintended as it was removed in the 3DS remake of OoT. (In fact, although the original methods have been patched, inventive players have found a 3DS-specific technique.)

Because they actively seek to take advantage of aberrations and inconsistencies in the code, I argue that techniques such as clipping (and ISG), while in the service of a similar goal to backwalking and pausewarping, belong to a different category. These involve developing playable strategies and setups that exploit programming bugs and systemic errors in the code’s design and execution. As we see with Wright’s discussion of using memory-hacking software to inspect the flows of data through the code, we see a further shift here that opens spaces outside the game in which it is both played and played with.

**Reading the Any% Speedrun: Manipulation and Reconstruction**

Perhaps the ultimate expression of this fusion of OoT gameplay, performed within OoT itself and outside in the text editors and Hex viewers, is to be found in the final category of technique. Where the first two categories center on revealing that which is in the game, either by reconceiving uses or revealing errors, the third describes techniques that involve arbitrarily altering the operation of the program and reassembling its data structures. Importantly, this is not achieved by disassembly or editing the code outside of the game, but rather through real-time play. In describing the “Ocarina Items” exploit, for example, Wright gives a sense of the complexity of the operations.

I drop the bugs and recatch them so that the bottle turns blue in Link’s hand. This will let me do a glitch called “Ocarina Items” in which I play a fake ocarina using my Deku stick. I perform Ocarina Items on the same frame that I touch the blue warp which allows me to regain control of Link while the warp tries to carry Link away. (Wright 2014, 12:58)

Although it has uses elsewhere, here Ocarina Items is actually just part of the most transformative part of the OoT Any% speedrun. As its name suggests, the “Ganondoor” (also known as the “Wrong Warp”) manipulates an existing warp point within the game to send Link to a newly specified location. It is a remarkably complex technique that
involves operations on a large number of interrelated functions in the game’s system. It also requires frame-perfect precision in execution, with particular inputs having to coincide with in-game timers and triggers. As Wright (2014, 12:41) notes, “This is the glitch that took over 13 years to discover and still astounds me that it’s even possible.” Once control of Link is regained using Ocarina Items,

…an invisible timer in the game is ticking up each frame to 101 at which point it will add the Cutscene Offset to the next area to load. If I open the door on this exact frame… voilà! the game loads the Deku Tree Basement on the same frame that it adds this Cutscene Offset. As Deku Tree Basement has no cutscene, the game ends up transporting me to Ganon’s Tower Collapse Interior instead of the Kokiri Emerald cutscene. This bypasses the Ganondorf fight and allows Child Link access to the endgame. (Wright 2014, 13:16)

The outcome of the technique is the creation of a connection in code that is subsequently rendered in polygonal space between regions in the game’s spatial and narrative architecture that are as palpably unintended by OoT’s developers as they are disruptive and apparently injurious to the integrity of the game’s myth. In short, Child Link is able to pass directly to the final battle with Ganon without completing the trials, interim battles, and quests that comprise the game’s narrative space. Importantly, the Wrong Warp achieves this through a meticulously optimized Setup play that manipulates data registers and (over)writes code and data into the program in order to cause it to (mal)function in a desired but unintended manner.

Scrutinizing the documentation at ZSR, we learn that the position at which Link is spawned in Hyrule when traveling through one of the many entrances/exits is governed by a number of variables. The combination of “Base Entrance” and “Scene Setup” indices govern at what point in Hyrule Link (re)appears, which Scene is to be loaded and, by way of various offsets that are dynamically added, whether the game’s clock should be set to day or night, whether Link is an Adult or Child, and whether an introductory movie sequence should play before the action begins. The game’s Entrance Table contains 1556 different entries.

Typically, these variables are written with data that is defined by player actions, but only insofar as they reflect in-game conditions including current location, time, and other aspects of gameplay progress. As these values in these Indexes have the ability to affect where Link will spawn in the map and in time, they are obviously a source of
great interest to speedrunners who have studied their operation and the structures of data used in *OoT* fastidiously. The operation of Ganondoor is contingent on a wide variety of glitches coming together that constitute a Setup so complex that it makes the Forest Escape look like child’s (or at least, Child Link’s) play.

| Index | Scene                | Entrance Description                                      |
|-------|----------------------|----------------------------------------------------------|
| 0242  | Death Mountain Trail | From Dodongo's Cavern                                    |
| 0246  | Death Mountain Crater| From Goron City                                           |
| 024A  | Death Mountain Crater| From Fire Temple                                          |
| 024E  | Forest Temple        | Outside Boss Door Forest Temple Boss Secret Map #0       |
| 0252  | Deku Tree            | From Deku Tree Boss                                      |
| 0256  | Tower Collapse Interior| Burning Rocks Tower Collapse Interior Lower               |
| 025A  | Market - Child Day   | From Castle                                               |
| 025B  | Market - Child Night | From Castle                                               |
| 025C  | Market - Adult       | From Ganon's Castle                                       |
| 025D  | Market - Adult       | From Ganon's Castle                                       |
| 025E  | Market - Child Day   | From Temple of Time                                       |

Table 2: (“Entrance Table” [excerpt], 2018)

By using a sequence of glitches including “Ocarina Items,” canceling from supposedly uncancellable sequences so as to gain control of Link when he is supposed to be under the control of the game system, and managing the in-game timer and movement through the space so as to execute a clip with an accuracy of a single frame of gameplay, the data written to Base Entrance and Scene Setup Index registers can also be directly manipulated by the player. ZSR details the technical operation of the glitch:
…if you open the door on the right frame, you will overwrite the “next entrance” variable with the base value of the entrance back into the Deku Tree (0x0252) rather than the value set by the blue warp, but with the “next cutscene number” set to 0xFFF1 (cutscene 1). This causes you to end up at 0x0257, which loads the Inside of the Tower Collapse scene…

The Inside the Tower Collapse scene does not have the 0x18 header command, so it instead attempts to play whatever is being pointed to by the cutscene pointer. (“Wrong Warp Explained” 2017)

Examining the “Entrance Table” data excerpt above, it is evident that Base Entrance value of the “Tower Collapse” is actually 0256 rather than the 0257 dynamically generated by the Any% technique. However, the OoT game engine rounds down entrance values, meaning the connection is made from the Deku Tree to the Tower rather than to Hyrule Market at 025A, which is the next logical Entrance in the list. What is particularly important about the Ganondoor is partly a consequence of its staggering complexity and the range of techniques involved, but is also found in the number of collaborators involved in its discovery, development, and execution.

In February 2012, a user named ChristianF23 found a way to skip many of the post-dungeon cutscenes. If performed in the Fire Temple, it not only skipped the cutscene but warped Link into the Forest Temple. The entire OoT community was astonished and the Zelda Scientists went to work…. Amongst all the craziness that was going on, a clever user by the name of r0bd0g was poring over the game’s internal entrance lists and had a startling realization. The door to Deku Tree basement from Gohma’s room on the internal entrance list was adjacent to part of the Castle Collapse Interior. Nobody had thought of this before because the stone slab that blocks the way made it not-obvious there was another option there. Sockfolder tested r0bd0g’s theory and it worked. For the first time in history, OoT could be completed as Child Link alone. This was faster than a Credits warp. There was an explosion of activity. Many people did many speedruns trying to optimize this new route including myself. (Wright 2014, 2:27)

It is clear from the Ganondoor exploit that navigating the reworked code makes for a faster speedrun than even the most efficient movement or clip through the game’s polygonal or narrative spaces. Indeed, such is the transformative potency of the
Wrong Warping, Sequence Breaking, and Running through Code
Systemic Contiguity and Narrative Architecture in The Legend of Zelda: Ocarina of Time Any% Speedrun

Ganondorf that _OoT_’s credits scene may be reached without Link’s iconic transformation from Child to Adult. In fact, it is reached without even obtaining the iconic Master Sword (although it is shown being swiped from Link’s hand in the final cinematic scene of the game, which not unreasonably assumes that Link must possess it to even be in this position). Thus, I suggest that the gameplay we see enabled by the Ganondorf exploit (and more broadly across the gamut of techniques employed by Any% speedrunner) does not reinforce _OoT_’s world or myth, as Sivak might suggest, but instead fundamentally subverts and rewrites them. The story of Any% _OoT_ is the story of a child who, using nothing but a wooden stick, defeats the Evil Ganon who was always already waiting at the top of the Castle. The detours, the coming-of-age, the time travel, the gatekeeping enemies, the loyal retainers and helpers, the weapons, and even Princess Zelda are all utterly immaterial and, in most cases, not even present.

According to the narrative frame espoused and codified in the official Strategy Guides and the pages of publications such as the _The Hyrule Historia_ (Gombos et al. 2013), the Any% _OoT_ could almost be read as disrespectful in its complete lack of regard for plot points, character arcs, temporality, and spatiality. Viewed through this lens, perhaps Wright’s 18:10 performance need not qualify as a performance of _OoT_ at all. Certainly, there are plenty of detractors of Any% speedrunning who would claim too great a gulf exists to treat these as performances of the same game (see the comments on Whitehead’s [2015] article on A Link to the Past glitches). But, according to the logic of code, the Any% story makes absolute sense. The tempo-spatiality of Hyrule is confirmed and made contiguous by the linkages that are (re)constructed through exploits, glitches, and the revelation of (un)intended affordances—all completed in less than 20 minutes. These are, indeed, performances of the same and different games simultaneously.

**Summary**

By utilizing a variety of techniques that I have sought to categorize, which differently exploit and manipulate aspects of the game’s intended and unintended behaviors, this article has shown how the _OoT_ Any% speedrunner reconfigures the relationship between the game system and its much-celebrated world and myth. Because it is possible to complete “the game” without completing _OoT_’s battle-against-the-odds storyline, I suggest that what we see in these distinctive (super)playing practices is the logic of the game’s underpinning system being exploited to override its narrative and representational architecture.
If the narrative architecture constructs an experience in which pleasure is derived from the circuitous traversal of space and time where certain activities and pathways are even coded as extraneous “sidequests,” the Any% speedrun eschews such detours and charts the quickest route from the initial moment that player-control is assumed to the point where it is relinquished as the final blow is struck. I argue that it does this not by mapping a route through the representational space produced by the game’s engine, but through the code itself. To speedrun OoT at Any% is to harness the ability to clip through vertices, and traverse maps, scenes, and eras in ways that are systemically permissible in code but which confound and reconfigure the game’s narrative and the tempo-spatiality of its canonical storyline. However, the traversal of the code results in an internally coherent contiguity, rendering entirely logical the “Wrong Warp” from the Deku Tree to the interior of Ganon’s collapsing castle—from the very beginning of the game’s putative narrative and space to the final moments and location of its denouement—just as deliberately pausing or even dying at a specific frame, invoking the Game Over screen, and resetting the player’s position allows traversal of Hyrule far more quickly than even backwalking allows.

Through careful analysis and deliberate manipulation of the flows of data through the game system, the Zelda Scientists and Any% Speedrunners are engaged not only in a creative exploration of ludic and performance possibilities, but also in a process of authoring new versions of the story of OoT. While these new versions eschew the narrative architecture and arcs of Nintendo’s OoT and are almost unrecognizable in their concision, connections, and tempo-spatial continguities, they are nonetheless viscerally experienced OoT narratives. Importantly, these new constructions of OoT are not told by virtuoso performers simply showcasing their skills by backwalking but rather depend on the execution and exploitation of ludic features, facets, and affordances revealed only by backwalking through the game’s representational spaces and traversing its code and data structures.

Finally, in drawing on a wide variety of (para)textual materials including Strategy Guides, reviews, commented gameplay footage, and fan discussions, I hope that this article contributes to advancing methodologies for game studies and broadens the resources and sites for ongoing research into player practice, design, and interactions with data and code.
Wrong Warping, Sequence Breaking, and Running through Code

Systemic Contiguity and Narrative Architecture in The Legend of Zelda: Ocarina of Time Any%
Speedrun

References

Bainbridge, Wilma Alice, and William Sims Bainbridge. 2007. “Creative Uses of Software Errors Glitches and Cheats.” Social Science Computer Review 25 (1): 61–77. doi:10.1177/0894439306289510.

Buchanan, Levi, Jason Leung, and Paul Shined. 1998. The Legend of Zelda, Ocarina of Time: Official Nintendo Player’s Guide. Redmond, Washington: Nintendo of America.

Burn, Andrew. 2006. “Reworking the Text: Online Fandom.” In Computer Games: Text, Narrative and Play, edited by Diane Carr, David Buckingham, Andrew Burn, and Gareth Schott, 82–102. Cambridge: Polity Press.

Cascone, Kim. 2000. “The Aesthetics of Failure: ‘Post-Digital’ Tendencies in Contemporary Computer Music.” Computer Music Journal 24 (4): 12–18. doi:10.1162/014892600559489.

“Clipping.” 2018. ZeldaSpeedRuns. Last updated October 22, 2018. http://zelda.speedrun.com/oot/tech/clipping.

CNN. 2014. “Game Faces: Cosmo Wright.” CNN International Edition. CNN Business video, 1:36. http://edition.cnn.com/videos/tech/2014/12/12/ns-natpkg-game-faces-cosmo-wright.cnn/video/playlists/video-games/.

Consalvo, Mia. 2003. “Zelda 64 and Video Game Fans: A Walkthrough of Games, Intertextuality, and Narrative.” Television & New Media 4 (3): 321–34. doi:10.1177/1527476403253993.

Consalvo, Mia. 2007. Cheating: Gaining Advantage in Videogames. Cambridge, MA: MIT Press.

Cuddy, Luke, ed. 2008. The Legend of Zelda and Philosophy: I Link Therefore I Am. Peru, IL: Open Court Publishing.

Dietz, Jason. 2017. “The All-Time Best Games on Every Platform.” No. 19, “The Legend of Zelda: Ocarina of Time.” Metacritic. March 28. Accessed December 22, 2018. http://www.metacritic.com/pictures/all-time-best-games-on-every-platform/19.

du Gay, Paul. 2007. Organizing Identity: Persons and Organizations “after Theory”. London: Sage.
Wrong Warping, Sequence Breaking, and Running through Code
Systemic Contiguity and Narrative Architecture in The Legend of Zelda: Ocarina of Time Any% Speedrun

Edge. 1998. “Testscreen – The Legend of Zelda: Ocarina of Time.” Edge 66 (Christmas): 84–87.

Edge. 2001. “The Legend of Zelda: Ocarina of Time.” Edge: The Most Significant Reviews from the First 100 Issues: 70-71.

Eskelinen, Markku. 2001. “The Gaming Situation”. Game Studies: The International Journal of Computer Game Research 1 (1).
http://www.gamestudies.org/0101/eskelinen/

“Entrance Table.” 2018. ZeldaSpeedRuns. Last updated February 6, 2018. 
http://zelda.speedrun.com/oot/wrongwarp/entrance-table.

“Extended Superslide.” 2018. ZeldaSpeedRuns. Last updated February 2, 2018. 
http://zelda.speedrun.com/oot/tech/extended-superslide.

GameTrailers. 2006. “Top 10 Best and Worst Games of All Time - GT Countdown.”
GameTrailersArchive [YouTube channel]. March 18, 2016 (originally published November 17, 2006). Accessed December 22, 2018.
https://www.youtube.com/watch?v=RASMwD60Zg0.

Giordano, Federico. 2011. “Almost the Same Game: Text and Gaming Experience between Continuity of Forms and Changing Context.” Dead Media Beat, Wired, April 21. https://www.wired.com/2011/04/dead-media-beat-federico-giordano-almost-the-same-game/.

Gombos, Michael, Takahiro Moriki, Heidi Plechl, Kumar Sivasubramanian, Aria Tanner, and John Thomas, translators. 2013. The Legend of Zelda: Hyrule Historia. Milwaukie, OR: Dark Horse Books.

Hans, James S. 1981. The Play of the World. Amherst: University of Massachusetts Press.

Hollinger, Elizabeth, James Ratkos, and Don Tica. 1998. The Legend of Zelda, Ocarina of Time: Prima’s Official Strategy Guide. Rocklin, CA: Prima Publishing.

Jenkins, Henry. 2006. Convergence Culture: Where Old and New Media Collide. New York: New York University Press.

Loe, Casey, and Gerald Guess. 1998. The Legend of Zelda: Ocarina of Time: Official Perfect Guide. Sunnyvale, CA: Versus Books.
Lotringer, Sylvère, and Paul Virilio. 2005. *The Accident of Art*. New York: Semiotext(e).

Lowood, Henry. 2006. “High-Performance Play: The Making of Machinima.” *The Journal of Media Practice* 7 (1): 25–42. doi:10.1386/jmpr.7.1.25/1.

McLuhan, Marshall, Fiore Quentin, and Joel Agel. 1967. *The Medium is the Massage*. New York: Bantam Books.

Morris, Sue. 2003. “WADs, Bots and Mods: Multiplayer FPS Games as Co-creative Media.” In *DiGRA ’03—Proceedings of DiGRA International Conference: Level Up*, edited by Marinka Copier and Joost Raessens. N.p.: University of Utrecht. http://www.digra.org/digital-library/publications/wads-bots-and-mods-multiplayer-fps-games-as-co-creative-media/.

“Movement Speeds.” 2016. ZeldaSpeedRuns. Accessed December 23, 2018. http://zelda.speedrun.com/oot/generalknowledge/movement-speeds.

Murray, Janet H. 2011. *Inventing the Medium: Principles of Interaction Design as a Cultural Practice*. Cambridge, MA: MIT Press.

Newman, James. 2005. “Playing (with) Videogames.” *Convergence: The International Journal of Research into New Media Technologies* 11 (1): 48–67. doi:10.1177/135485650501100105.

Newman, James. 2008. *Playing with Videogames*. London: Routledge.

Newman, James. 2012. *Best Before: Videogames, Supersession and Obsolescence*. London: Routledge.

Newman, James. 2016. “Mazes, Monsters and Multicursality: Mastering Pac-Man 1980–2016.” *Cogent Arts & Humanities* 3: 1190439. doi:10.1080/23311983.2016.1190439.

Newman, James. 2017. “World -1: Glitching, Codemining and Procedural Level Creation in Super Mario Bros.” In *Fans and Videogames: Histories, Fandom, Archives*, edited by Melanie Swalwell, Helen Stuckey, and Angela Ndalianis, 146–62. New York: Routledge.

Nintendo. 1998. “The Legend of Zelda: Ocarina of Time: Instruction Booklet.” Accessed December 18, 2018.
Wrong Warping, Sequence Breaking, and Running through Code
Systemic Contiguity and Narrative Architecture in The Legend of Zelda: Ocarina of Time Any% Speedrun

https://cdn02.nintendo-europe.com/media/downloads/games_8/emanuals/nintendo_8/Manual_Nintendo64_TheLegendOfZeldaOcarinaOfTime_EN.pdf

Nintendo. 2019. “The Legend of Zelda: Ocarina of Time.” Accessed June 4, 2019. https://www.nintendo.co.uk/Games/Nintendo-64/The-Legend-of-Zelda-Ocarina-of-Time-269536.html.

Norman, Donald A. 1999. The Design of Everyday Things. New York: Basic Books. First published 1988 as The Psychology of Everyday Things.

OoT Speedruns Wikia. n.d. Accessed December 18, 2018. http://oot-speedruns.wikia.com/wiki/OoT_Speedruns_Wikia.

Parker, Felan. 2008. “The Significance of Jeep Tag: On Player-Imposed Rules in Video Games.” Loading... 2 (3).

http://journals.sfu.ca/loading/index.php/loading/article/view/44.

Postigo, Hector. 2014. “The Socio-technical Architecture of Digital Labor: Converting Play into YouTube Money.” New Media and Society 18 (2): 332–49. doi:10.1177/1461444814541527.

Salen, Katie. 2002. “Quake! DOOM! Sims! Transforming Play: Family Albums and Monster Movies.” Walker Art Center, Minneapolis, MN, October 19, 2002. Accessed May 14, 2018. http://www.walkerart.org/gallery9/qds.

Scully-Blaker, Rainforest. 2014. “A Practiced Practice: Speedrunning through Space with de Certeau and Virilio.” Game Studies: The International Journal of Computer Game Research 14 (1). http://gamestudies.org/1401/articles/scullyblaker.

Sivak, Seth. 2009. “Each Link in the Chain is a Journey: An Analysis of The Legend of Zelda: Ocarina of Time.” In Well Played 1.0: Video Games, Value and Meaning, edited by Drew Davidson, 130–47. Pittsburgh: ETC Press.

http://press.etc.cmu.edu/content/legend-zelda-ocarina-time-seth-sivak.

Squire, Kurt, and Henry Jenkins. 2002. “The Art of Contested Spaces.” Accessed December 23, 2018. http://web.mit.edu/~21fms/People/henry3/contestedspaces.html. Also published in Game On, edited by Lucien King, 64–75. New York: Universe, 2002.
Wrong Warping, Sequence Breaking, and Running through Code
*Systemic Contiguity and Narrative Architecture in The Legend of Zelda: Ocarina of Time Any% Speedrun*

Torje. 2019. “Ocarina of Time Any% Speedrun in 17:01 by Torje.” ZeldaSpeedRuns. Accessed June 4, 2019. http://www.zeldaspeedruns.com/speedruns/115049.

Whitehead, Thomas. 2015. “Video: Learn How to Clear The Legend of Zelda: A Link to the Past in Under Two Minutes.” NintendoLife, October 2. http://www.nintendolife.com/news/2015/10/video_learn_how_to_clear_the_legend_of_zelda_a_link_to_the_past_in_under_two_minutes.

Wright, Cosmo. 2013. “Legend of Zelda: Ocarina of Time Speed Run in 0:26:34 by Cosmo #SGDQ 2013 [iQue].” SpeedDemosArchive [YouTube channel]. October 23. Accessed December 26, 2018. https://www.youtube.com/watch?v=_0N1lh1csGQ.

Wright, Cosmo. 2014. “Zelda Ocarina of Time Speedrun in 18:10 by Cosmo [WR] [commentated].” Archive.org (originally uploaded to YouTube). Accessed December 26, 2018. https://archive.org/details/ReuploadZeldaOcarinaOfTimeSpeedrunIn1810ByCosmoWRcommentated.

“Wrong Warp Explained.” 2017. ZeldaSpeedRuns. Last updated January 16, 2017. http://zelda.speedrun.com/oot/wrongwarp/wrong-warp-explained.

“Zelda 64 Engine.” 2014. ZeldaSpeedRuns. Last updated May 29, 2014. http://zelda.speedruns.com/oot/generalknowledge/zelda-64-engine.

ZFG. 2016a. “Ocarina of Time "100% No Source Requirement" Speedrun in 3:21:51.” ZFG [YouTube channel]. September 10. Accessed December 26, 2018. https://www.youtube.com/watch?v=IaRSSDiqUmc.

ZFG. 2016b. “AGDQ 2016 - Zelda: Ocarina of Time Master Quest - All Dungeons Speedrun in 1:40:25 by ZFG.” ZFG [YouTube channel]. January 12. Accessed December 26, 2018. https://www.youtube.com/watch?v=kCNcxhdHzBQ.