# 26 GAME GLITCH

*Peter Krapp*

According to the *Oxford Dictionary of Slang*, *glitch* is a word "applied to a sudden brief irregularity or malfunction of equipment, originally especially in spacecraft"; however, despite dating it to 1962, the dictionary marks its origin as unknown (Ayto 1998, 418).[1] Probably derived from the German verb *glitschen*, meaning "to slip or slide" (or the related adjective *glitschig*, meaning "slippery"), *glitch* in English has over the past four decades become associated particularly with unpredictable consequences of software errors (Nunes 2010). By contradistinction, the fears of global computing trouble at the turn of the millennium (referred to as "Y2K") helped further popularize the term *software bug*—if the trouble is anticipated, it is thus not classified as a glitch.[2] *Glitch* remains "a precariously vague term, which however captures some of the slipperiness of digital media" (Vanhanen 2001). The word has become particularly common in computer game circles, where it is sometimes synonymous with sudden disruptions—graphic display or sound mistakes, communications errors, failed collision detection, games freezing or crashing, or other malfunctions, often prompted by a particular input combination. Glitches in games can halt gameplay, but in some cases they can lend an unforeseen advantage to players who know how to exploit the situation. Intriguingly, some players seek to re-create the conditions of possibility of a glitch in order to repeat it, sometimes in the hope that the re-created glitch may become an exploit or at least something to brag about.

How does a computing setup that fundamentally does not encourage anything random become the platform for a lively pursuit of the random? What makes a glitch interesting, first of all, is that it simply stands out in a landscape of tightly regulated interactions between software and hardware. If necessity and chance are complementary, then one might expect not only games of chance but also games of necessity. Yet we can see that a fully rationalized game—such as a state-operated lottery that allows for play, minimizes risks, and benefits the commons with profits—is simply a fiscal and moral calculation that recuperates contingency into a legally sanctioned form of gambling that fills the state coffers. Jorge Luis Borges is surely

not alone in protesting that the lottery hardly lives up to its promise of an "intensification of chance" because in reality it is a tightly controlled process of finite drawings and final decisions, almost "immune to hazard" (1962, 69–70). It is worth noting that Gilles Deleuze picks up on Borges's suggestion, albeit without making a connection between that digital potential and an ideal game "in which skill and chance are no longer distinguishable" (1990, 60).

Moreover, the common understanding of the glitch is that it has little or nothing to do with operator errors. Computing culture is predicated on communication and control to an extent that can obscure the domain of the error. Between a hermetically rule-bound realm of programmed necessity and efficient management of the totality of the possible, we find a realm of contingency: distortions in the strictest signal-to-noise ratio, glitches and accidents, or moments where the social hierarchy of computer knowledge condescendingly ascribes to "user error" what does not compute. As our digital culture oscillates between the sovereign omnipotence of computing systems and the despairing agency panic of the user, glitches become aestheticized, recuperating mistakes and accidents under the conditions of signal processing: "Glitches can be claimed to be a manifestation of genuine software aesthetics" (Goriunova and Shulgin 2008, 111).

Some game glitches render the situation inoperable not only if they freeze the screen or disconnect from further input but also when they drop the gameplay into bad infinity, as in the example of the *Pokemon* bad egg (in generation III, IV, and V—*Pokemon* 2002 et seq.)—an egg that keeps wanting to hatch with every step the player takes or that hatches further bad eggs. Interestingly, a bad egg can be removed out of one's game with a cheatcode. Not so with the *Pokemon* glitch "MissingNo." wherein a certain sequence of actions in the Nintendo game, as documented in 1999, results in disruption of the game's graphics and potentially in erased games. Players could flee from, fight, or capture MissingNo. yet, despite Nintendo's warnings, several publications provided instructions on how to *arrange* such an encounter (Bainbridge 2007). Any such glitch will break the illusion of an immersive game world and remind the player of the game's software and hardware conditions of possibility.

Due to this extradiegetic dimension, the glitch plays a pivotal role also in alternative play or counterplay—that is, in exploring ways of playing *with* the game rather than just playing the game. As Tom Apperley has it, counterplay explores the limits of digital games, including the cataloging and exploiting of bugs and glitches, but without having to code or hack—it does not interfere with software or hardware at any level except that of the user interface. One example Apperley offers is that "in the original release of *Sid Meier's Civilization* [MicroProse, 1991] a game-winning tactic was quickly developed that involved building large numbers of cities close to each other so that their zones of influence overlapped and then swamping the opposing civilizations with large numbers of low powered units. The use of this strategy

meant that the game could easily be over, with the globe dominated by one culture by 400BC" (2010, 142).

This glitch from 1991 is of course no longer a feature of the later installments of the *Civilization* series. Yet there is hardly a game that has not provided a glitch worthy of recording, replicating, and exploiting to its regular players. In the first installment of *Halo* (Bungie, 2001), players were able to make the Covenant Elites and Jackals bleed when they were wounded or killed—even after killing these enemy avatars, players were able to keep hitting them to produce more blue-purple blood. A glitch in the game appeared when the blood did not fade away: "If you kept hitting the body of one dead alien over and over, you could paint the ground in layers and layers of alien blood until the framerate slowed to a crawl as the Xbox tried to deal with all these extra layers of blood" (Keogh 2012, 110). Of course, to discover such a software glitch that actually affects the hardware, the player must first indulge in a cruel excess of alien bashing to accidentally discover the material impact such actions can have on the machine. Like the *Civilization* glitch, this *Halo* glitch raises interesting questions about these games' premise and some basic assumptions held by their players.

Game studies often discusses games in terms of a risky, lawless world versus a safe, riskless world. The fact that there is a veritable gaming culture around the glitch illustrates this tension in game design—what is attractive about the glitch is that it is a risk players incur but also something they seek to harness. Players have found a number of ways to recuperate such moments: "glitching" websites aggregate discoveries in various games and advise how to re-create situations that were discovered accidentally, thus turning exceptions into exploits. Deliberately inducing a glitch can be risky in live play, especially in competitive online situations, but it can also lend an edge to players who successfully arrange it to work in their favor. "When you learn a glitch that can be used in actual gameplay," one glitch website advises, "you may be tempted not to attempt that particular glitch during gameplay (someone might kill you while you're trying to accomplish that glitch.) But if you learn to incorporate that glitch during gameplay I can't tell you how effective that will be."[3]

For example, in *Metroid* (Nintendo, 1986), players aware of a particular glitch were able to access unused map data Nintendo never intended them to see by getting stuck in a blue door, which allowed them to walk through walls and access what some thought were secret worlds inside the game. Similarly, in the Wii game *Mario Kart* (2008), the PlayStation 3 version of *Tom Clancy's Rainbow Six: Vegas* (2007), and the anniversary edition of *Lara Croft Tombraider* (Crystal Dynamics, 2007) a player could turn rapidly to catch a glimpse of what is on the other side of a wall before the in-game camera adjusted its distance behind the player's avatar—a glitch that allowed the player to see through walls.

A more complicated manner of exploiting a glitch is exemplified by the ghost mode in *Call of Duty 3* (Activision, 2006), where a player can have a member of their team go over from the

Allies to the Axis side, get on a motorcycle, meet them by a tank, join them on the motorcycle, and then have a third team member in the tank shoot the motorcycle. If the player has "team damage" turned off, the teammate playing an Axis soldier is killed, but the other player not—they can walk around invisible to their teammates (though not to the enemy). Multiplayer online games must carefully calibrate and balance labor with appeal, both in time sinks and money sinks, to regulate their virtual economy. A classic online game, Lucasfilm's *Habitat* (1986), provided a system of avatar currency (tokens) and stores where items could be purchased. When a glitch in the program produced a virtual money machine, a small group of *Habitat* avatars became virtual millionaires overnight. Over protests from the "poor," the *Habitat* programmers decided to let the finders keep the virtual loot (Dibbell 1998, 70–72). Another early multiplayer example helps to highlight how the glitch can incite new modes of gameplay not foreseen by the game's developers and administrators. During the beta test of *Ultima Online* (Origin Systems, 1997), the game's creator Richard Garriott held an assembly where his avatar, Lord British, addressed a group of players, but one of the players attacked and killed Lord British despite the fact that he was the supreme ruler of the fictional game world who should have been indestructible. A glitch had rendered him vulnerable—a server reboot just before the event (Nitsche 2008, 27)

Strictly speaking, there cannot be a glitch-free computer game or, more generally, a glitch-free computing environment. One of the more interesting insights offered by pioneering computer engineer Severo Ornstein is that synchronization becomes more problematic the faster processing cycles become. Ornstein (who worked at Lincoln Labs, Bolt Beranek & Newman, and Xerox PARC during some of their most influential phases) resolved to debunk any claims about the possibility of a glitch-free circuit and started to propagate the notion that the efforts to remove the glitch should in all honesty just be called "move-the-glitch": "In every case we were able to determine to where, in the design, the ingenious designer had moved the glitch. In each case, the glitch persisted, and the presenter was brought into alignment with this new correct view of the universe. Our fervor approached that of religious zealots. It was our mission to stamp out anti-glitch apostasy wherever we could find it" (2002, 136)

The problem of the glitch, the unavoidable chance factor that introduces mistakes and noise, is thus only too well known among gamers. The interesting question for the history of computing is why—despite steadily rising storage, better and quicker processors, and all kinds of technical advances in software and hardware—a veritable cult of the glitch developed nonetheless, including the cult of 8-bit game sounds. When and how does visual, acoustic, and audiovisual raw material that is way below the technically possible become an aesthetic choice?

It did not take long for the glitch to become aestheticized (Hernandez 2012). Chicago saw several annual GLI.TC/H events,[4] and Oslo mounted the Motherboard Glitch Symposium in 2002. Glitch art, often though not exclusively inspired by computer and game culture, has become acceptable in galleries and museums and has found entry into videos and installations. Early examples include the manipulation of a Bally Astrocade game console for the art video *Digital TV Dinner* (1979) by Raul Zaritsky, Jamie Fenton, and Dick Ainsworth; more recent work by Cory Arcangel and Eddo Stern could also be mentioned in this context, but that would require another whole entry for this book. The machinima clip *Untitled (for David Gatten)* by Phil Solomon and Mark LaPore (2005) exploits the frozen screens, data corruption, and other glitches Solomon and LaPore found in *Lara Croft: Tombraider* to create a sequence of "dripping, elongated textures and blotches of green hurtling towards the screen" (Sicinski 2007). The play of aesthetics shows itself in these examples as a reduction of contingency through form, harnessing adventure through aleatory or stochastic management of the event and the surprise. Another example is jodi's game modification *SOD*, a mod of the classic game *Castle Wolfenstein* (Muse Software, 1981) that retains the soundtrack but scrambles the visuals so as to render the screen a black-and-white abstract pattern of glitchy, nonrepresentational patterns, appearing to make the user's computer run amok.[5] A related example is the art game *ROM CHECK FAIL* (2008), an "independent" game predicated on the glitch as an aesthetic tool, shuffling around the elements of a number of classic two-dimensional computer games.[6] As Jarrad Woods describes *ROM CHECK FAIL*, "Every few seconds, there is a glitch internal computer noise, and everything in the current level stays in the same position but changes into something else" (qtd. in Rose 2011, 48). Random switching to alternative rules apparently complicates the game mechanic and allows *ROM CHECK FAIL* to recycle an oddball assortment of arcade and console classics, to be navigated with the arrow and space keys. Of course, this kind of 1980s retro-game generator cashes in on nostalgia and aestheticizes the glitch. Going one step further in that direction is the commercial release *Mega Man 9* (Capcom/Inti Creates, 2008). Though developed for recent consoles including the Nintendo Wii, *Mega Man 9* uses graphics and sounds harking back to the 8-bit era of the original Nintendo Entertainment System. A "legacy mode" emulates the lower frame rates of the game's ancestors, only partially renders sprites, and causes them to flicker when they crowd the screen; this degradation feature is achieved not via an emulator but using a dedicated engine that simulates an outmoded technology. By the same token, it is evident that any glitch becomes more palpable with the advent of higher expectations from audiovisual resolution.

Thus, it follows that "glitch" should also have become a music genre. Clicks and glitches come to undermine the tightly controlled, layered loops of electronic music (Krapp 2011). As Chris Salter observes, "Another movement within the audiovisual coding scene amplified its processual characteristics through the revealing and subsequent use of errors and glitches"

(2010, 178). But in a recursive system where differentiations between data and programs are obsolete, it comes as no surprise that there should also be a Glitch plug-in for audio software that chops up audio files and applies a variety of effects, depending on how much you tweak the controls. With unremarked irony, the website for that Virtual Studio Technology software plug-in warns, "This version of *Glitch* is still just a prototype and contains a few bugs—obviously I am working towards fixing everything as soon as possible."[7] Thus, it becomes clear that even with a software routine that can be directed to apply randomized effects, human–computer interaction remains predicated on an interpretation of control that tends to bar artists from finally taking the place of contingency: "It is from the 'failure' of digital technology that this new work has emerged: glitches, bugs, application errors, system crashes, clipping, aliasing, distortion, quantization noise, and even the noise floor of computer sound cards are the raw materials composers seek to incorporate in their music" (Cascone 2004, 393). Even and especially in the electronic music inspired by game technology, digital tropes of perfect sound copies are abandoned in favor of errors, glitches become aestheticized, and mistakes and accidents are recuperated for art under the conditions of signal processing. The glitch has been almost completely domesticated (Menkman 2011). There is even a game called *Glitch* (2011), a web-based massively multiplayer online puzzle developed by Tiny Speck (led by Flickr cofounder Stewart Butterfield) that sends players into the past to inhabit the minds of 11 giants (Terdiman 2010).

Yet if we allow ourselves to conceive of human–computer interaction as organized around the glitch, it is not to smuggle a covert humanism in through the back door of technological determinism; rather, it is to emphasize what Alexander Galloway calls "the cultural or technical importance of any code that runs counter to the perceived mandates of machinic execution, such as the computer glitch or the software exploit, simply to highlight the fundamentally functional nature of all software (glitch and exploit included)" (2006, 326). One might conclude, however provisionally, that gaming glitches are part of the art form in the same way that brushstrokes are part of painting. Game developers may be tempted to regret this comparison as little consolation to a user who just had a program crash or indeed to a programmer who is trying to debug the system. However, the glitch may be that hairline fissure that can widen on to new vistas in gameplay and game studies—and thus on to better mistakes.

## Notes

1. *A Dictionary of Astronomy* defines *glitch* as "an abrupt disturbance in the regular train of pulses from a pulsar, appearing as a sudden change of the pulsar's period and spin-down rate. Glitches tend to occur in younger pulsars whose rate of spin is slowing rapidly, notably the

Crab Pulsar and Vela Pulsar. They are believed to be caused by the sudden release of stress energy either in the crust (a starquake) or between the crust and the superfluid interior" (Ridpath 1997, 191). It is plausible that John Glenn, who used the word *glitch* in the sense of random system error in 1962 (in the book *Into Orbit*, an account of Project Mercury, the United States' first human spaceflight program), would have been aware of the astronomical meaning of the term.

2. Y2K fears anticipated what might happen when year fields switched over from 99 to 00 because memory conservation in COBOL (common business-oriented language) had limited years to two-digit representation. However, the majority of Y2K issues in finance and insurance software were fixed during 1999, and the new millennium began without any major software errors attributable to the Y2K bug.

3. Glitching advice from the now defunct site http://glitchmania.com/ is partially archived by the Internet Archive at https://web.archive.org/web/20110129003547/http://glitchmania.com/Home.html; another repository, also defunct, of game glitch exploits was http://glitchblog.com, partially extant at the Internet Archive at https://web.archive.org/web/20111021025055/http://glitchblog.com/.

4. The archives of three glitch festivals (2010, 2011, and 2012) are found at http://gli.tc/h/.

5. For *SOD*, go to http://sod.jodi.org/.

6. For more on *ROM CHECK FAIL*, seehttp://www.farbs.org/games.html. Also see Joan Leandre's art projects *R/C* and *NostalG* at http://www.retroyou.org and http://runme.org/project/+SOFTSFRAGILE/.

7. The Glitch 2 plugin for VST is available from http://illformed.org/plugins/. The original Illformed glitch plugin is ranked #5 among 100 best free VST plugins; see https://edmranks.com/top-100-free-vst-plugins-for-electronic-music-producers/.

## Works Cited

Apperley, Tom. 2010. *Gaming Rhythms: Play and Counterplay from the Situated to the Global*. Amsterdam: Institute of Network Cultures.

Ayto, John. 1998. *Oxford Dictionary of Slang*. New York: Oxford University Press.

Bainbridge, William Sims. 2007. "Creative Uses of Software Errors: Glitches and Cheats." *Social Science Computer Review* 25 (July): 61–77.

Borges, Jorge Luis. 1962. *Ficciones*. New York: Grove.

Cascone, Kim. 2004. "The Aesthetics of Failure: 'Post-digital' Tendencies in Contemporary Computer Music." In *Audio Culture*, ed. Christoph Cox and Daniel Warner, 392–398. London: Continuum.

Deleuze, Gilles. 1990. "Tenth Series of the Ideal Game." In *The Logic of Sense*, 58–65. London: Athlone.

Dibbell, Julian. 1998. *My Tiny Life: Crime and Passion in a Virtual World*. New York: Henry Holt.

Galloway, Alexander. 2006. "Language Wants to Be Overlooked: On Software and Ideology." *Journal of Visual Culture* 5 (3): 315–332.

Glenn, John. 1962. *Into Orbit.* London: Cassell.

Goriunova, Olga, and Alexei Shulgin. 2008. "Glitch." In *Software Studies: A Lexicon*, ed. Matthew Fuller, 110–119. Cambridge, MA: MIT Press.

Hernandez, Patricia. 2012. "It's Not a Glitch, It's a Feature. It's Art. It's Beautiful." *Kotaku,* August 10. http://kotaku.com/5955722/its-not-a-glitch-its-a-feature-its-art-its-beautiful.

Keogh, Brendan. 2012. *Killing is Harmless: A Critical Reading of "Spec Ops: The Line."* Marden, Australia: Stolen Projects.

Krapp, Peter. 2011. *Noise Channels: Glitch and Error in Digital Culture*. Minneapolis: University of Minnesota Press.

Menkman, Rosa. 2011. *The Glitch Moment(um)*. Amsterdam: Institute of Network Cultures.

Nitsche, Michael. 2008. *Video Game Spaces: Image Play and Structure in 3D Worlds*. Cambridge, MA: MIT Press.

Nunes, Mark, ed. 2010. *Error: Glitch, Noise, and Jam in New Media Cultures*. New York: Continuum.

Ornstein, Severo. 2002. *Computing in the Middle Ages. A View from the Trenches 1955–1983*. New York: Authorhouse.

Ridpath, Ian. 1997. *A Dictionary of Astronomy*. New York: Oxford University Press.

Rose, Mike. 2011. *250 Indie Games You Must Play*. New York: CRC Press.

Salter, Chris. 2010. *Entangled: Technology and the Transformation of Performance*. Cambridge, MA: MIT Press.

Sicinski, Michael. 2007. "Phil Solomon Visits San Andreas and Escapes, Not Unscathed: Notes on Two Recent Works." *CinemaScope Magazine* 30. http://www.academichack.net/solomon.htm.

Terdiman, Daniel. 2010. "In Depth with Tiny Speck's Glitch." *CNET News*, February 9. http://news.cnet.com/8301-13772_3-10449721-52.html.

Vanhanen, Janne. 2001. "Loving the Ghost in the Machine: Aesthetics of Interruption." *CTheory* a099. http://www.ctheory.net/articles.aspx?id=312.