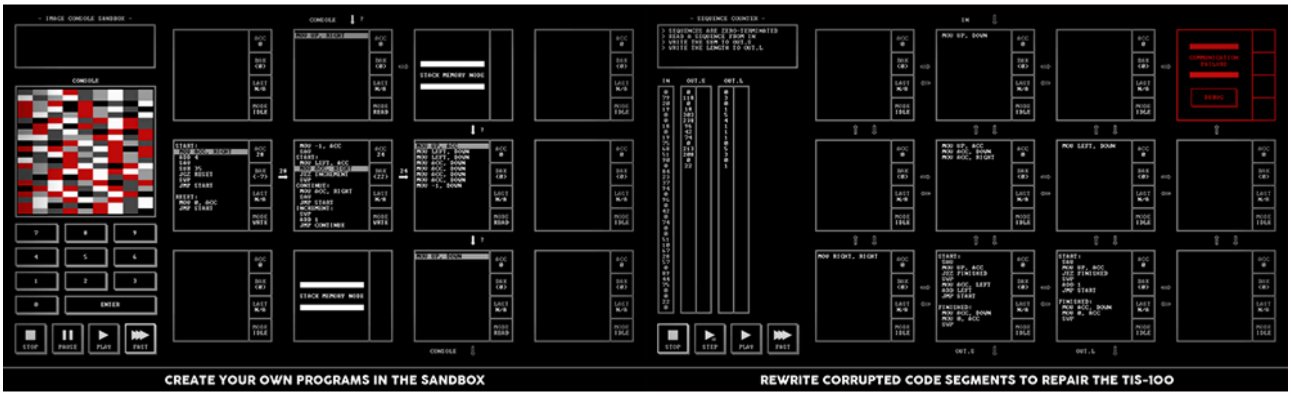
“If You Can See Something for its True Essence”: Exploring the Origins of the Personal Computer in TIS-100

Gregory Phipps
gregory.phipps@mail.mcgill.ca
University of Iceland



CREATE YOUR OWN PROGRAMS IN THE SANDBOX

REWRITE CORRUPTED CODE SEGMENTS TO REPAIR THE TIS-100

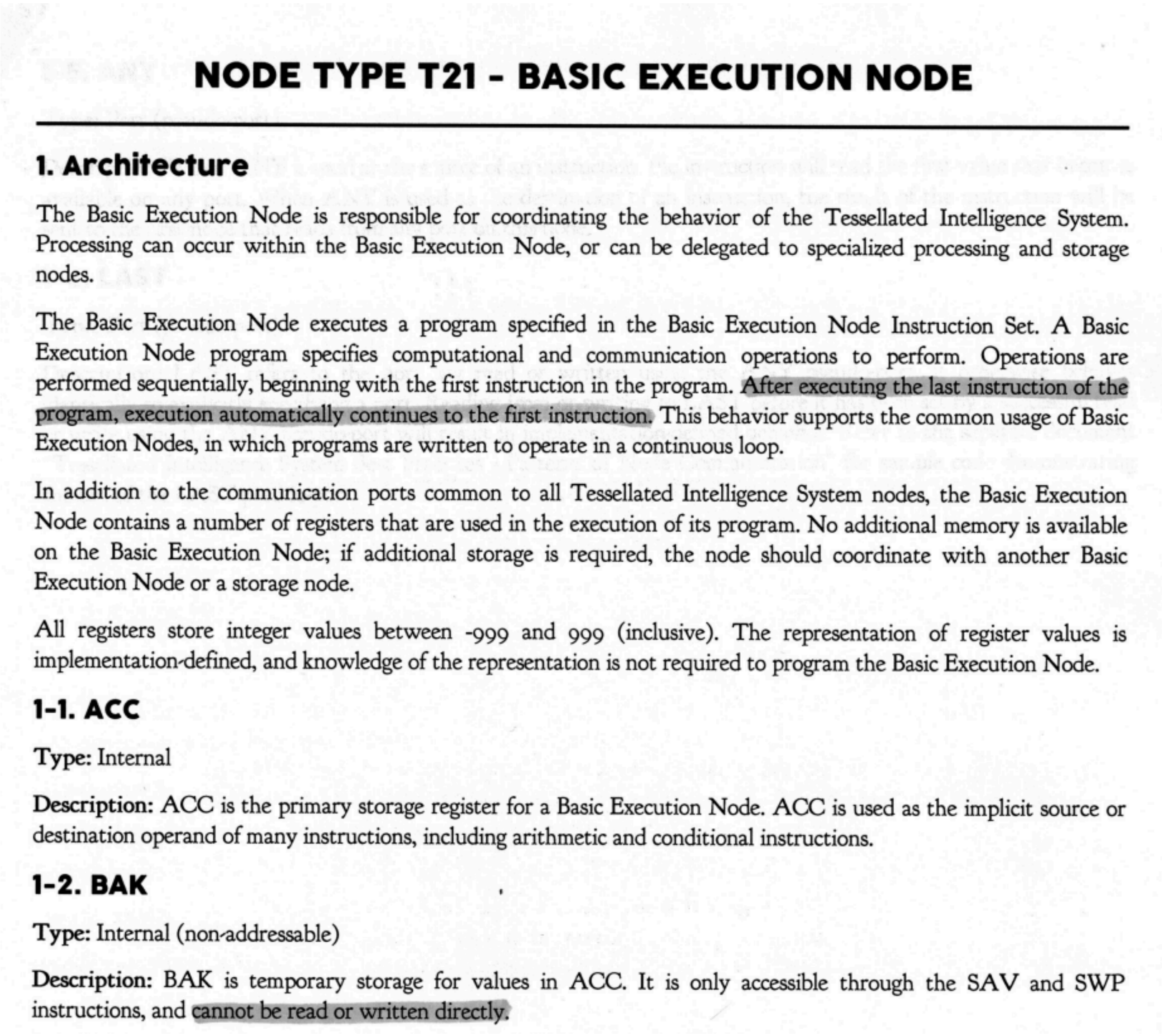
Abstract: TIS-100 is a minimalist game centered on assembly programming, evoking the cultural atmosphere of 1970s computing. Lacking visual spectacle, it instead explores deep questions about the individual’s relationship to computers. Through its narrative and retro interface, the game revisits a time when programmers sought creative autonomy on institution-owned machines. The story follows a user decoding a mysterious computer system, leading to a philosophical reflection on human–machine interaction. TIS-100 highlights how even the most basic computational tasks can become a space for creativity and community, suggesting that the essence of computing lies in enabling self-expression and collective connection.

Tags: Game Study, Personal Computers

1 Introduction

Released by Zachtronics in 2015, TIS-100 *(Zachtronics, 2015)* is a minimalist and punishingly difficult computer game about assembly language programming. Zachtronics has gained recognition for indie puzzle games such as SpaceChem *(Zachtronics, 2011)*, Infinifactory *(Zachtronics, 2015a)*, Shenzhen I/O *(Zachtronics, 2016)*, and Opus Magnum *(Zachtronics, 2017)*, but TIS-100 is arguably the company’s most grueling game to date, not least because it offers almost nothing in the way of either a user-friendly interface or visual and auditory stimuli. In TIS-100, the player is tasked with writing programming commands into a series of rectangles that generate sequences of numbers, with little guidance beyond a jargonistic instruction manual. In an online presentation for Google, the founder of Zachtronics, Zach Barth, reveals that he nearly quit working on the project numerous times, and that, even when it was complete, he was reluctant to introduce it to others because he assumed that it would be universally unappealing *(at Google, 2017)*.

TIS-100 has nonetheless proven to be not only one of Zachtronics most popular titles, but also, I would argue, one of its more multilayered games. Speaking generally, TIS-100 inspires nostalgia for the early days of computers while directing attention to the origins of programming. Its gameplay is structured around the mechanics of assembly language programming, allowing users to hone their skills in this area while also recalling a period when interactions among individuals and computers centred on the most basic forms of coding. In creating an interface reminiscent of early digital computing, TIS-100 presents a somewhat barren setting for gameplay. However, it also invokes the origins of programming by representing the interrelationship among individual users, communities of programmers and the bureaucratic institutions that typically owned and operated computers during the 1970s. In particular, the narrative of TIS-100 tells the story of an individual programmer who gains unlimited access to a computer that ordinarily would be restricted to an institutional context. However, rather than merely celebrating the programmer's freedom to experiment with a classified device, the narrative also details his obsessive identification with it, which gradually reveals a desire on his part to locate an underlying essence to the relationship between individuals and computers.



NODE TYPE T21 - BASIC EXECUTION NODE

1. Architecture

The Basic Execution Node is responsible for coordinating the behavior of the Tessellated Intelligence System. Processing can occur within the Basic Execution Node, or can be delegated to specialized processing and storage nodes.

The Basic Execution Node executes a program specified in the Basic Execution Node Instruction Set. A Basic Execution Node program specifies computational and communication operations to perform. Operations are performed sequentially, beginning with the first instruction in the program. After executing the last instruction of the program, execution automatically continues to the first instruction. This behavior supports the common usage of Basic Execution Nodes, in which programs are written to operate in a continuous loop.

In addition to the communication ports common to all Tessellated Intelligence System nodes, the Basic Execution Node contains a number of registers that are used in the execution of its program. No additional memory is available on the Basic Execution Node; if additional storage is required, the node should coordinate with another Basic Execution Node or a storage node.

All registers store integer values between -999 and 999 (inclusive). The representation of register values is implementation-defined, and knowledge of the representation is not required to program the Basic Execution Node.

1-1. ACC

Type: Internal

Description: ACC is the primary storage register for a Basic Execution Node. ACC is used as the implicit source or destination operand of many instructions, including arithmetic and conditional instructions.

1-2. BAK

Type: Internal (non-addressable)

Description: BAK is temporary storage for values in ACC. It is only accessible through the SAV and SWP instructions, and cannot be read or written directly.

Figure 1: TIS-100 Manual: A sample of the jargonistic instruction manual.

The possibility that such an essence exists builds tensions and contradictions that stand at the heart of TIS-100, helping frame links between its gameplay, setting and narrative. Focusing on the ways these tensions and contradictions unfold within the world of TIS-100, this article explores how the game dramatizes both the cultural milieu of early programmers and philosophical questions about the purposes of computers that persist to this day. Ultimately, the article argues that, insofar as TIS-100 does reveal an essence to the interactions between individuals and computers, it centres on the way computers can bind together communities of people by reflecting not only an individual's personal intentions and specific modes of expression, but also their links to other people.

Structurally, the article contains an integrated analysis of TIS-100, though it also falls broadly into four parts. The first part introduces TIS-100 in the context of scholarly debates about the multifaceted connections between computer games and the real world before discussing its relevance to real-life programming. The second section considers how the gameplay, setting, and narrative of TIS-100 reflect contemporary issues and historical concerns related to programming, with a particular focus on the cultural environment that fostered conflicts between individual programmers and institutions in the 1970s. The third part focuses primarily on the narrative of TIS-100, which describes a single user's obsessive desire to locate the "true essence" of the relationship between individuals and computers. The final part reworks the assumption that such an essence exists. I argue that, in the world of TIS-100, this essence only emerges through the personal creativity of the individual and their links to a community of like-minded people.

TIS-100 itch.io page link

2 TIS-100: The Beginning and Purpose of Programming

Barth remarks that his first objective when designing a computer game is to "create a world for the game to exist in," further clarifying, "a lot of games pretend that they're movies when they do story stuff, and they think about it as this linear thing. [By contrast, we] try to build world and then plug everything into it" *(GDC, 2018)*. As Bissell *(Bissell, 2010)* asserts, some games are "more about the world in which the game takes place than the story concocted to govern one's progress through it" (p.11). Barth's statements suggest that his games fall under this umbrella, but this classification raises questions about the broader status of fictional worlds in computer games. For example, critics continue to scrutinize the relationship between computer games and the real world. Farrow and Iacovides *(Farrow & Iacovides, 2014)* argue that the fictional worlds of many games are not designed to "emulate real life"; rather, they work to "re-create shared cinematic, literary or televisual worlds which have no physical correlate" (p. 224). As such, these games do not try to duplicate a player's interactions with the real world, but instead try to immerse the player in a fantasy world that is akin to those found in literature and film. On the other hand, Rusch *(Rusch, 2017)* points out that, even if games create the same types of immersive experiences as literature and film, they are crucially different in that they demand from players an active and embodied involvement in their worlds. For Rusch, games are "comparable to how we experience real life" because they evoke the "actual experience of real-life experiential gestalts through quasi-bodily enactment" (pp. 73-74). Together, these competing ideas gesture toward the complexities involved in characterizing the fictional worlds of computer games.

In the context of this dialogue, part of what defines the games of Zachtronics is the way their fictional realms usually do hinge on direct correlates to the so-called real world. If, as Rusch describes, games evoke real-life experiences through processes of enactment, then Zachtronics games tend to evoke quite specific modes of experience that are centred on individual forms of labour, whether they involve chemistry, electronics, engineering, or programming. Commenting on this aspect of his games, Barth criticizes the "fake" progression mechanics of many Role-playing games (RPGs), stating that his company tries to create games that are centred on "actual skills" *(GDC, 2018)*. To be sure, the process of developing skills that are transferable and/or derived from concrete experience in the real world stands at the core of most Zachtronics games. For instance, the company's breakthrough game, SpaceChem, focuses on the production of chemical molecules.

Drawing heavily on real chemistry and the mechanics of industrial automation, SpaceChem takes place in an austere fictional world that is largely devoid of cinematic or literary flourishes. Zachtronics's more recent game Shenzhen I/O carries this theme forward, as its instruction manual is, as Barth states, full of "real electronics things," such as application notes, data sheets, and technical information *(GDC, 2018)*.

Themed on assembly language programming, TIS-100 arguably captures this aspect of Zachtronics games more explicitly than any other title. Assembly language programming is a diverse concept that typically refers to a form of coding that allows users to communicate with the fundamental architecture of a computer without having to slog through unwieldy machine code. In this sense, assembly language programming is one of the most basic ways that an individual can interact with the framework of a computer. In accurately representing the mechanics of assembly language programming, TIS-100 has gained a reputation as a game that can teach a player how to program in real life. Sayer *(Sayer, 2016)* declaims, "For the programmer-in-training, TIS-100 is the final hurdle; if you can survive its stiff challenge, you're ready to code." So too have computer scientists utilized TIS-100 to demonstrate the results of new programming methods *(Rosin, 2019)*. However, probably the best evidence of the connection between TIS-100 and the real world of programming is found in online forums devoted to the game, which are stocked with programmers who offer advice to novices and compete with one another to achieve optimal results. Thus, TIS-100 maintains a correlate to the real world insofar as it permits the player to progress only if their skills in programming are improving within the framework of the game.

TIS-100's relevance to the real world of programming hinges on its gameplay as well as its setting, narrative and general atmosphere. Beginning with the former, gameplay in TIS-100 consists of solving a series of puzzles, each of which is contained in variations of a black-and-white screen filled with squares and rectangles. In any given puzzle, the player must perform a task with a string of random numbers in order to generate a precise sequence of numbers. For instance, in the level "Signal Edge Detector," the player must calculate the differences in value between one number and the next in a random chain. If the value of a number changes by more than 10 from its predecessor, the player outputs a 1; if it changes by less than 10, they output a 0. Thus, for the sequence 0, 32, 30, 27, the player must output a 1 (for the change from 0 to 32), then a 0 (for the change from 32 to 30), and then another 0 (for the change from 30 to 27). In this level, the set of numbers that the player outputs is analogous to a binary code (i.e., 1001001 . . .). Alternatively, in a different level entitled "Sequence Sorter," the player is tasked with arranging strings of numbers in order from lowest to highest. Other tasks involve dividing numbers, reversing the orders of sequences, and using numbers to graph images.

None of these tasks are complicated in and of themselves, but the challenges of the game arise through the limited number of commands that the player has at their disposal. Such commands -- which the player writes manually into squares (or nodes, as the manual calls them) that communicate with each other -- enable the player to add and subtract numbers, move them between the nodes, store them one at a time, and compare them to each other, within strict limits. Adhering to the dynamics of assembly language programming, the commands consist of simple abbreviations (mov up acc, sub left, jez, swp, etc.). The player must arrange these commands into a loop that will allow them to continue outputting the correct numbers indefinitely. In other words, for a level like Sequence Sorter, the player must establish a framework of commands that will allow them to sort any string of numbers from lowest to highest. The primary objective of the game is to create these frameworks successfully, but, because there are many possible solutions for any given puzzle, the second goal is to optimize said frameworks by reducing the number of instructions, nodes, and cycles within them. Many of the commentaries among programmers in online forums revolve around strategies for optimization.

TIS-100 can be conceived of as a game concerned solely with the challenges of assembly language programming because it appears at first to be devoid of literary and cinematic flourishes. Lacking eye candy, the game is set entirely in different versions of the same black-and-white screen. Gameplay occurs through text input, and the only text that the player can enter are the programming commands outlined in the instruction manual. The manual itself reads like a document written by a computer scientist who assumes that the player will already know the lingo. The game has no music, and the sound effects are one-dimensional. It is impossible to imagine a film or literary equivalent of TIS-100 unless it happened to be an avant-garde prank.

On the other hand, the brutal minimalism of TIS-100 and the technicality of the manual contribute to the creation of a distinct setting while also serving as a conduit for the game's narrative. Significantly, the setting and narrative provide another connection to the real world of programming by invoking an earlier era of computers. Thus, if the gameplay is grounded on the mechanics of the most basic forms of coding, the setting and narrative invoke a period when basic coding defined the types of interactions that users had with computers. The cultural context of TIS-100 is contemporary, but the dates of the TIS-100 (Tessellated Intelligence System-100) computer itself are listed in the start-up screen as 1972 to 1975, indicating that it was built in the days when the "enabling technology for the personal computer" was still in development *(Campbell-Kelly et al., 2013)*. The TIS-100 predates the commercial sale of personal computers, hearkening back to a time when computers were often owned and run by corporate and military institutions. Computers during this era were large, ungainly and prohibitively expensive devices that only select individuals could access for short intervals of time. In this sense, TIS-100 tells the story of a person who gains unlimited access to a computer that would ordinarily be available to individuals only for brief stretches of time in circumscribed environments.

In telling this story, TIS-100 alludes to a cultural struggle that pitted early computer programmers against corporations and military apparatuses. Scholars have detailed different versions of this struggle, but the connecting thread in their accounts centres on the conflict between, on one side, bureaucratic institutions and, on the other side, individuals and close-knit communities. Isaacson *(Isaacson, 2015)* recounts the early development of the computer during the mid-twentieth century: "Instead of becoming personal tools and memory banks for individuals to use, [computers] became hulking industrial and military colossi that researchers could time-share but the average person could not touch" (p. 264). By the 1970s, computers the size of small refrigerators had emerged, but companies continued to dismiss the "idea that there would be a market for desktop models that could be owned and operated by ordinary folk" (p. 264). As Isaacson and also Levy *(Levy, 2010)* describe, these early computers could be described as Orwellian constructs because they were associated with the dehumanizing military and corporate institutions that operated them. Levy writes that, as late as 1973, computers were "generally regarded as inhuman, unyielding, warmongering . . . Orwellian monsters" (p. 151). Programmers could use them at select times, but the standards and rules of the parent institution regulated their activity.

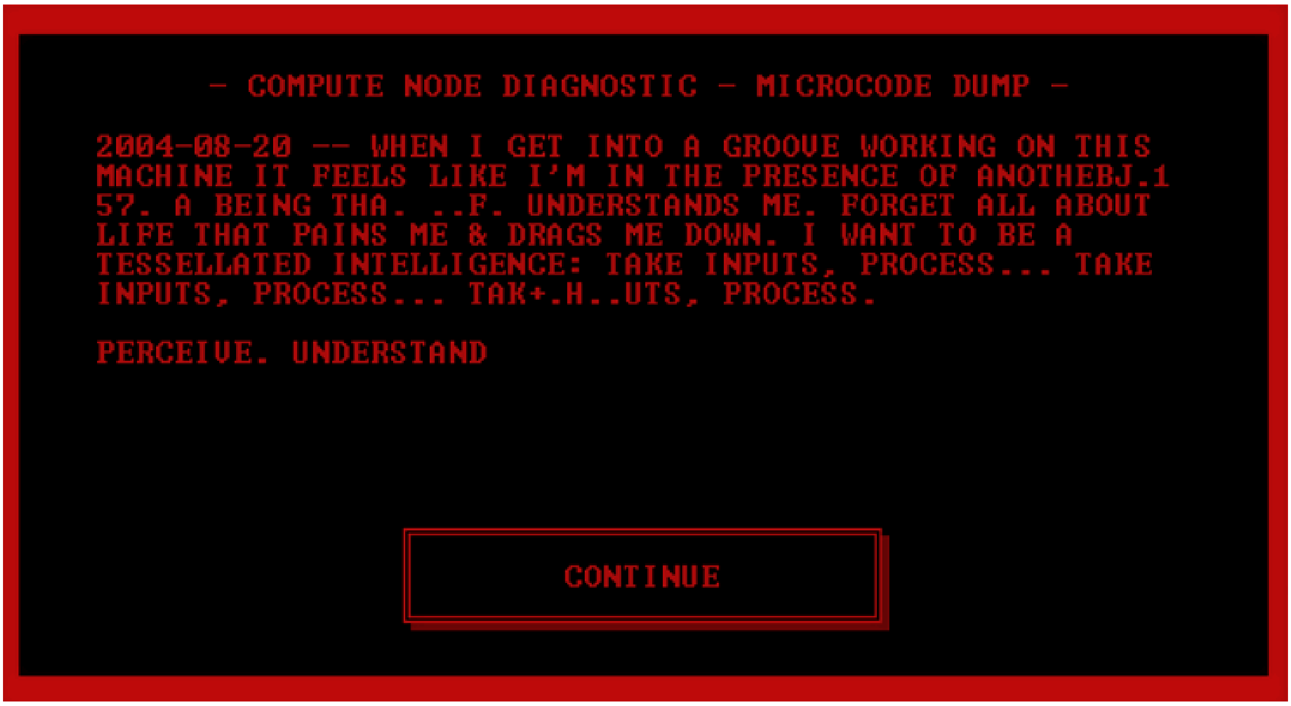
Nonetheless, during the late 1960s and early 1970s, a practical need for programmers who could work on these computers instigated both conflicts and new interconnections between institutions and small groups of individuals. A cultural mystique coalesced around the latter, as the earliest programmers were often unkempt people who performed tasks that many regarded as a "black art" due to their opacity *(Campbell-Kelly et al., 2013)*. Far from adhering to images of corporate or military conformism, these programmers were typically seen as individualistic and eccentric people: "By the end of the 1960s, the stereotype of the socially awkward computer nerd--long haired, bearded, sandal wearing, and 'slightly neurotic'--had been well established in both the industry literature and popular culture" *(Campbell-Kelly et al., 2013)*. By extension, some programmers became linked to countercultural ideals of individualism, forming communities that were invested in liberating the computer from stultifying institutions and corporations, thereby turning it into a device for personal use and expression. Working cooperatively, they not only fuelled a conflict between establishment institutions and communities that opposed them, but also helped instigate intersections between these two sides that hinged, in part, on the flexibility of computers and their growing potential to become personalized devices.

The narrative of TIS-100 raises this conflict, gesturing toward the early cultural context of programming. This narrative unfolds through the diary entries of the player's Uncle Randy. Near the beginning of these entries, Randy speculates that the TIS-100 is a top-secret device, such as "something the CIA cooked up to comb through Soviet radio signals" *(Zachtronics, 2015a)*. The introductory screen to the game's secret level advances this theme, presenting an image of an eagle holding a shield next to a block of text that reads, "Tampering with the memory contents of this device is a violation of the law and must be promptly reported to the state security bureau. This device is registered to: >Department of Defense >Quantum Information Science Lab" *(Zachtronics, 2015a)*. This screen confirms that the TIS-100 is not a personal computer, but a machine that once belonged in the laboratory of a military complex. Importantly, Randy's diary commences following his purchase of the computer in 1979, indicating that he occupies the role of an early programmer, though he seems to obtain a privileged position given that he is able to work on the TIS-100 in private over several decades. The player, meanwhile, inhabits the vicarious role of an early programmer, reliving Randy's struggles to understand the purpose of the computer.

The narrative does not detail Randy's practical efforts to understand how the TIS-100 functions, though the gameplay is probably meant to mirror his endeavours. In the context of this narrative, the gameplay reflects a dynamic conflict between the individual programmer (in this case, the player) and the faceless institution. As mentioned earlier, the player's task involves completing a series of menial assignments involving sequences of numbers. As in the case of a programmer working on a classified military project, there is no explanation as to why the player needs to complete any of these tasks. Presumably, they are contributing to the functions of the TIS-100, which are listed in the instruction manual as “automated financial trading, bulk data collection, and civilian behavioral analysis,” but the player never learns the nature of these contributions *(Zachtronics, 2015a)*. However, like the first programmers, the player does have an opportunity to work in an independent and creative manner since there are multiple solutions to each assignment. Thus, the player is capable of leaving an individual mark on the world of TIS-100, mainly because the simplicity of the assembly language programming console works in concert with the diverse number of arrangements and circuits that it allows. Indeed, the ability to approach the puzzles creatively and to design solutions that reflect a personal set of strategies is central to the value of TIS-100 as a game, a point I return to below. Not only that, but the growth of an online community has, in its own way, brought up another point of connection to the early days of computers, recalling the communal interconnectivity that defined (and still defines) programming as a cooperative endeavour. Case in point, no player ever completes a puzzle in isolation since the model that they craft is automatically measured against other players’ solutions, which are ranked according to optimization. Furthermore, the second set of puzzles in the game, which is called TIS-NET, features levels designed by other players. In this respect, the individual is able to experience diverse manifestations of an early programming culture that pushed back against institutional control, as the player combines personal creativity with involvement in a close-knit community of other TIS-100 fans (many of them professional programmers).

Then again, part of what is noteworthy about the setting and narrative of TIS-100 is that they work against positive associations of early programming culture by presenting a stark world overlaid with impressions of solitude and paranoia. On one side, this atmosphere reflects the status of the TIS-100 as a product of a secretive military operation that involves civilian behavioral analysis. On the other side, the atmosphere also speaks to Uncle Randy’s narrative about his increasing isolation and identification with the computer. Randy’s story commences with his speculations about the purpose of the TIS-100, but his entries become more introspective as he loses the people in his life as well as his job. His descent into isolation is linked to his growing fixation on the TIS-100. In the level entitled Sequence Generator, Randy remarks that his girlfriend encourages him to get rid of the microcomputers that he has collected over the years, but he resolves to keep the TIS-100, mainly because he is worried that he could face repercussions if anybody discovers that he owns “something I wasn’t supposed to have” *(Zachtronics, 2015a)*. In the next level, Sequence Counter, his attention shifts from his girlfriend to the computer, and his tone becomes both resentful and delusional: “No, she won’t understand. Of course she doesn’t. Why would you expect her to? Just keep working . . . I feel like I’m learning about the people behind this and their design decisions. The thoughts that went into figuring out how things shoul work. Fr$ those people feel very close to me now” *(Zachtronics, 2015a)*. Just as an imagined intimacy with the designers of the TIS-100 supplants the relationships in his life, his obsessive work on the TIS-100 also replaces his professional life. In Signal Pattern Detector, he indicates that his unemployment has enabled (and in fact has probably been caused by) his continuing work on the TIS-100: “Now that I have plenty of free time, I might as well get back to this. Keep my skills sharp while I look for a new job” *(Zachtronics, 2015a)*. In outlining Randy’s progression toward solitude and obsession, the narrative offers a sketch of an individual who has become entangled in the single-minded task of discovering the purpose of the TIS-100 computer.

This progression works in unison with Randy's deepening identification with the TIS-100. The computer functions not only as a replacement for loved ones and a job, but also as an extension of Randy's mind. He hints at this identification in levels such as Sequence Peak Detector, commenting that, like the TIS-100, he is still as "good now as [he] ever was" despite being "old" and "washed up" *(Zachtronics, 2015a)*. He proceeds to make this connection explicit in Sequence Reverser: "When I get into a groove working on this machine it feels like I'm in the presence of "another being. A being that" understands me. Forget all about life that pains me & drags me down. I want to be a tessellated intelligence: take inputs, process" *(Zachtronics, 2015a)*. In Signal Divider, he further states "To perceive is to take in. . . Seeing something takes it inside of you," indicating that the console of the TIS-100 computer has infiltrated his mind, leading him to believe that he is indeed transforming into a personification of a Tessellated Intelligence System *(Zachtronics, 2015a)*. In the context of TIS-100's gameplay, this transformation seems to entail the reduction of a human mind to the performance of simultaneously menial and yet complex tasks involving sequences of numbers.



```
- COMPUTE NODE DIAGNOSTIC - MICROCODE DUMP -

2004-08-20 -- WHEN I GET INTO A GROOVE WORKING ON THIS
MACHINE IT FEELS LIKE I'M IN THE PRESENCE OF ANOTHEBJ.1
57. A BEING THA. ..F. UNDERSTANDS ME. FORGET ALL ABOUT
LIFE THAT PAINS ME & DRAGS ME DOWN. I WANT TO BE A
TESSELLATED INTELLIGENCE: TAKE INPUTS, PROCESS... TAKE
INPUTS, PROCESS... TAK+.H..UTS, PROCESS.

PERCEIVE. UNDERSTAND
```

Figure 2: TIS-100 Reverser: The narrative textbox in Sequence Reverser.

TIS-100 therefore dramatizes a conflict between a representation of early programming as a creative and cooperative endeavour conducted among people with shared interests and a more sinister depiction of an individual's descent into paranoid isolation. From one perspective, this conflict speaks to a dichotomy between the individual and the institution. In particular, Randy's increasingly obsessive identification with the TIS-100 implies that this top-secret device is designed not only to analyze but also to modify civilian behaviour. This possibility brings another angle to the portrayal of a conflict between the individual programmer and the dehumanizing institution. It could be the case that Randy becomes enmeshed in a covert military operation that is intended to strip away his identity. This analysis suggests that, for Randy, the institutional purpose of the computer trumps the positive opportunities it affords for cooperation and creativity. One implication of this reading could be the notion that the player has an opportunity to experience the more affirmative aspects of programming while escaping the institutional control that entangles Randy.

However, this reading neglects Randy's larger, idealistic intentions with the computer, which display a measure of agency that works in concert with the gameplay of TIS-100. His identification with the TIS-100 computer is structured, in part, around his belief that it can access some of kind of essential truth. Suggestions of this idea surface in Image Test Pattern 2, where he describes a "dream about a tremendous circle of eyes pointed inward at the 1st.1.cteur of the world" *(Zachtronics, 2015a)*. This statement buttresses the recurring subject of perception in the narrative, while the scrambled rendition of "first centre," which includes a c and an anagram for the word "true," is symptomatic of a desire on Randy's part to locate an originary locus to the world. This thread continues in Histogram Viewer, with Randy envisaging a "large number of perceiving devices" operating together to collapse the "wave function" of a single, undefined object, thereby making it "real" *(Zachtronics, 2015a)*. Given the TIS-100's association with a "Quantum Information Science Lab," the perceiving devices are apparently meant to signify the computer itself. Thus, Randy's identification with the TIS-100 seems to be based on the notion that it might have the capacity to reveal an inner essence to the world that exists on a quantum level. At the same time, the image of the TIS-100's "eyes" pointing inward suggests that this essence may also exist within Randy himself, a connection strengthened by his assessment that he is akin to a Tessellated Intelligence System that has internalized the workings of the computer. His final, truncated diary entry in Sequence Sorter reads, "Seeing is a form of control. And if you can see something for its true essence, you will be . . .," a statement that brings these threads together while also terminating on an inconclusive note *(Zachtronics, 2015a)*. The final cutscene of the game (which I consider below) and the abrupt conclusion of the diary imply that Randy and the computer collapse together at the moment that a "true essence" is revealed.

Randy's search for a true essence brings an extra layer to the relationship between a nascent programming culture and the world of TIS-100. In addition to representing the trials of a solitary programmer, the game also unfolds as a retroactive exploration of the origins of personal computers, further solidifying the place that the early programmer occupies within this world. TIS-100's portrayal of these origins hints that the so-called true essence Randy covets may involve a fundamental basis to the relationship between the individual and the computer. As such, the 1970s programmer fits into this world insofar as he is apparently qualified to understand this basis -- not just because of his programming skills, but also because he belongs to the era when the origins of programming first materialized. In fact, Randy does develop one interpretation of an essence, identifying the individual as an intelligence that receives inputs through perception and then processes them. In this respect, the basis of a relationship between the individual and the computer hinges on the way that they are both structured systems, the chief aims of which are to receive and process data.

Expanding on these points, one could say that the essential correspondence that Randy seeks between the individual and the computer takes shape through the gameplay of TIS-100. From this perspective, this correspondence, such as it is, is grounded on exchanges of information involving rudimentary commands, numbers and geometric shapes. As remarked, the main purpose of each puzzle is to create a loop among the nodes that will enable the player to run sequences of numbers indefinitely. TIS-100 therefore boils interactions among the individual and the computer down to a framework consisting of numbers that flow between squares in the shapes of endless loops. Speaking symbolically, the game implies that this framework captures a distilled version of the relationship between the individual and the computer. If both an individual and a computer can be categorized as structured systems that receive and process data, then the gameplay of TIS-100 paints an image of how these systems operate and interact with each other on their most fundamental levels. In other words, arrangements of numbers that move in circuits represent an elementary picture of how individuals and computers process data together.

At the same time, Randy's attempt to identify an essential basis clashes with the observation that computers are (and always have been) such flexible and adaptable devices that it becomes difficult to frame them according to any "essence." Indeed, if TIS-100 offers a vision of the most basic interactions between the individual and the computer, it does so by revealing the form of these interactions, not their essential content. Ted Nelson's landmark 1974 work Computer Lib/Dream Machines speaks to this distinction, outlining a philosophical understanding of computers from the perspective of an early programmer. Nelson alludes to the conflict between programmers and institutions when he refers to "computer people" as members of a "priesthood" who have obtained a "stranglehold" on knowledge about the workings of computers (p. 3). As Nelson sees it, these programmers have achieved this stature because, according to the consensus view, they are mysterious if not "frightening" individuals who appear to be "completely preoccupied with unfathomable concerns and seemingly indifferent to normal humanity" (p. 46). For Nelson, though, the irony is that the computer is the "most general machine man has ever developed" (p. 10). Functioning as all-purpose devices, "computers have no nature and no character, save that which has been put into them by whoever is creating the program for a particular purpose." He further emphasizes, "Computers are, unlike any other piece of equipment, perfectly BLANK. And that is how we have projected on it so many different faces" (pp. 10-11). He does identify something akin to the essence of programming when he describes the "basic, central magical interior device of the computer" as a "program follower," the main purpose of which is to read instructions in order, conduct tests on symbols, and perform tasks in continuous loops. Fittingly, this description of assembly language programming does mesh with the gameplay of TIS-100, but, in outlining this "central, magical interior," Nelson only underscores the "power of such a machine to do almost anything" (pp. 10-11). On these terms, the core operations of a computer do not reflect a universal "truth" about the relationship between the individual and the machine; rather, these operations offer a blank scaffolding that is designed merely to accommodate any specific user's intentions. Much as a typewriter only produces what the typist wants to write, a computer only reflects the information that a user wants to process. If there is an essence to programming, it only emerges as a reflection of the user's intentions, which they express by inputting information.

These points help reveal the full ramifications of TIS-100's representation of the origins and essence of programming, which hinge on interconnections between the individual user and the wider community of players/programmers. It is tempting to read Randy's isolation and paranoia as a product of the machine's growing control over him, but, following Nelson's analysis, his degeneration may simply be a sign of his descent into addiction. Notwithstanding the TIS-100's background as a top-secret device, it may have "no nature and no character" in and of itself, from which it follows that Randy's obsessive identification with it reflects nothing more than his own downward spiral into madness. Furthermore, this decline could be read as a warning of the dangers that might afflict the solitary gamer. From this perspective, the narrative offers a (perhaps satirical) commentary on the addictive qualities of TIS-100, suggesting that it is analogous to a massively multiplayer online role-playing game (MMORPG) in that it may drive a gamer to neglect their personal and professional life as they get caught up in completing the assignments and optimizing their solutions. The only cutscene in the game, which appears when the player finishes the final level, is a grainy image of what appears to be an emaciated face wearing an expression of terror and horror. The face spreads across the screen before the latter abruptly switches off, affording an image of a person trapped inside the console. The image evidently provides a summation of what happens to Uncle Randy at the moment that he locates a "true essence," but it also serves as a wry specular image of what a player may have become while ploughing through the levels of TIS-100. In short, the image presents the face of an addict who has been sucked inside the game.

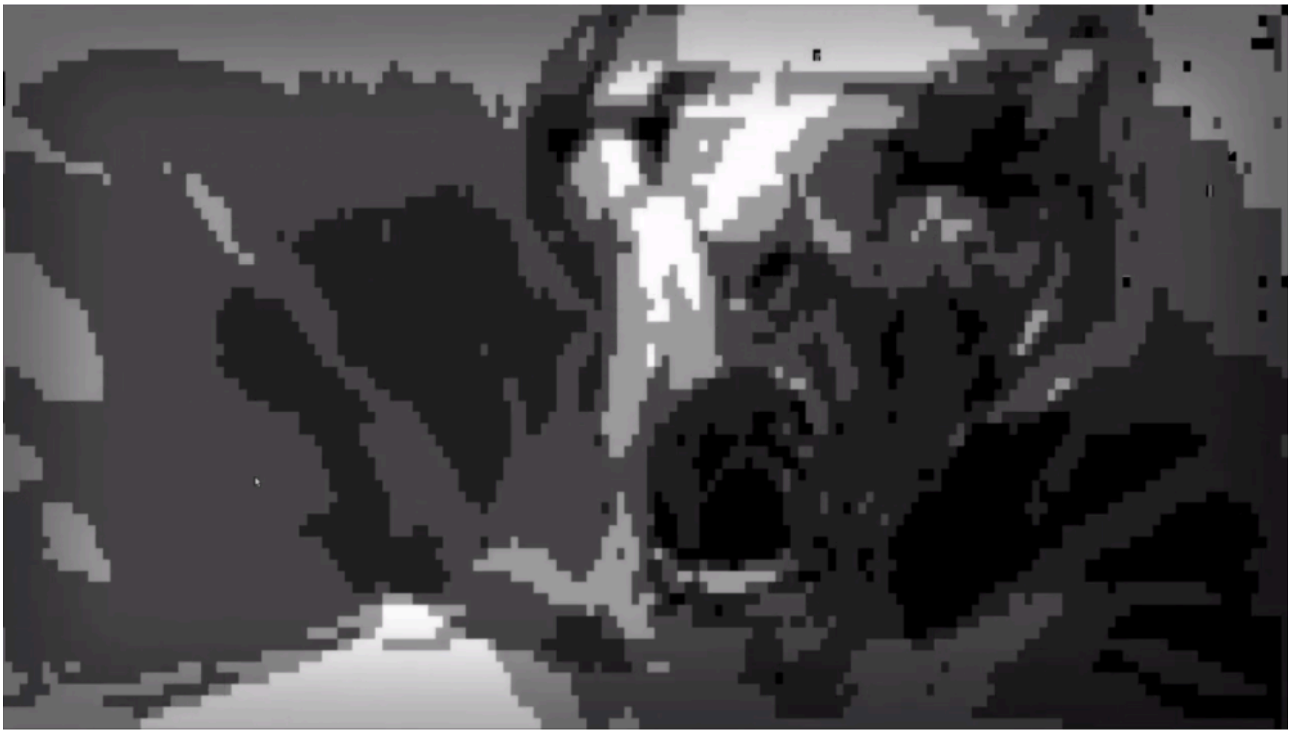
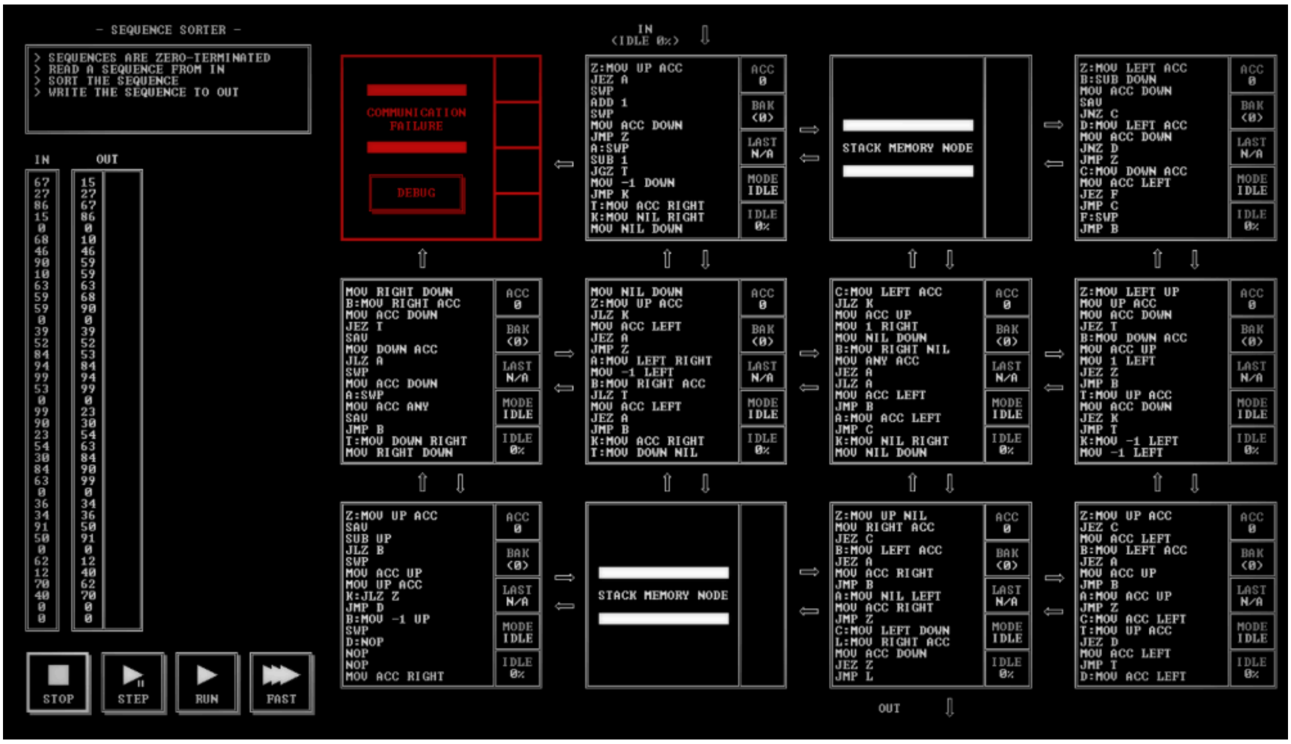


Figure 3: TIS-100 Conclusion: The final cutscene in the game.

Uncle Randy's deterioration evidently works in tandem with his self-imposed removal from any communal network. This disconnection occurs in part because he works on an outdated computer over several decades, but it also happens due to his implicit belief that he alone can access the essential groundwork of the relationship between the individual and the computer. The more he focuses on locating the purported "true essence" of this relationship, the more he reveals his own alienation from others. In the context of TIS-100's gameplay, the repeating loops of commands and numbers that fill this world reveal Randy's own obsession and isolation -- not because this is what the world of TIS-100 does represent, but because it is capable of representing anything that the user brings to it.

The player who tackles TIS-100 in isolation is apt to leave a personal mark on this world as they work their way through the different levels, but, unlike Randy, they have the ability to engage with a robust community of players. As the player progresses, they will invariably adopt a strategy that generates recurring patterns in the combinations of commands that they assemble. Given the sheer diversity of solutions that are available, these patterns will be quite distinct, reflecting the player's level of skill, but also, perhaps, their general approach to confronting complex puzzles. A professional programmer's repertoire of solutions will likely exhibit a streamlined approach that demonstrates ingenuity, while a novice's solutions will be more cluttered and experimental. For somebody who belongs in the second group, the tension and furious strategizing that come with squeezing commands into a limited number of nodes will be on full display in the more taxing levels, such as Sequence Sorter. A player who completes the game and revisits the levels in order will probably discover a two-dimensional sculpture of sorts that exhibits their strategies and mentality in the stark but expressive diction of assembly language programming.


- SEQUENCE SORTER -

```
> SEQUENCES ARE ZERO-TERMINATED
> READ A SEQUENCE FROM IN
> SORT THE SEQUENCE
> WRITE THE SEQUENCE TO OUT
```

```
IN   OUT
67   15
27   27
86   67
15   86
 0    0
68   10
46   46
90   59
10   59
63   63
59   68
59   90
 0    0
39   39
52   52
84   53
94   84
99   94
53   99
 0    0
99   23
90   30
23   54
54   63
30   84
84   90
63   99
 0    0
36   34
34   36
91   50
50   91
 0    0
62   12
12   40
70   62
40   70
 0    0
 0    0
```

```
Z:MOV UP ACC
JEZ A
SWP
ADD 1
SWP
MOV ACC DOWN
JMP Z
A:SWP
SUB 1
JGZ T
MOV -1 DOWN
JMP K
T:MOV ACC RIGHT
K:MOV NIL RIGHT
MOV NIL DOWN
```

```
Z:MOV LEFT ACC
B:SUB DOWN
MOV ACC DOWN
SAV
JNZ C
D:MOV LEFT ACC
MOV ACC DOWN
JNZ D
JMP Z
C:MOV DOWN ACC
MOV ACC LEFT
JEZ F
JMP C
F:SWP
JMP B
```

```
MOV RIGHT DOWN
B:MOV RIGHT ACC
MOV ACC DOWN
JEZ T
SAV
MOV DOWN ACC
JLZ A
SWP
MOV ACC DOWN
A:SWP
MOV ACC ANY
SAV
JMP B
T:MOV DOWN RIGHT
MOV RIGHT DOWN
```

```
MOV NIL DOWN
Z:MOV UP ACC
JLZ K
MOV ACC LEFT
JEZ A
JMP Z
A:MOV LEFT RIGHT
MOV -1 LEFT
B:MOV RIGHT ACC
JLZ T
MOV ACC LEFT
JEZ A
JMP B
K:MOV ACC RIGHT
T:MOV DOWN NIL
```

```
C:MOV LEFT ACC
JLZ K
MOV ACC UP
MOV 1 RIGHT
MOV NIL DOWN
B:MOV RIGHT NIL
MOV ANY ACC
JEZ A
JLZ A
MOV ACC LEFT
JMP B
A:MOV ACC LEFT
JMP C
K:MOV NIL RIGHT
MOV NIL DOWN
```

```
Z:MOV LEFT UP
MOV UP ACC
MOV ACC DOWN
JEZ T
B:MOV DOWN ACC
MOV ACC UP
MOV 1 LEFT
JEZ Z
JMP B
T:MOV UP ACC
MOV ACC DOWN
JEZ K
JMP T
K:MOV -1 LEFT
MOV -1 LEFT
```

```
Z:MOV UP ACC
SAV
SUB UP
JLZ B
SWP
MOV ACC UP
MOV UP ACC
K:JLZ Z
JMP D
B:MOV -1 UP
SWP
D:NOP
NOP
NOP
MOV ACC RIGHT
```

```
Z:MOV UP NIL
MOV RIGHT ACC
JEZ C
B:MOV LEFT ACC
JEZ A
MOV ACC RIGHT
JMP B
A:MOV NIL LEFT
MOV ACC RIGHT
JMP Z
C:MOV LEFT DOWN
L:MOV RIGHT ACC
MOV ACC DOWN
JEZ Z
JMP L
```

```
Z:MOV UP ACC
JEZ C
MOV ACC LEFT
B:MOV LEFT ACC
JEZ A
MOV ACC UP
JMP B
A:MOV ACC UP
JMP Z
C:MOV ACC LEFT
T:MOV UP ACC
JEZ D
MOV ACC LEFT
JMP T
D:MOV ACC LEFT
```

Figure 4: TIS-100 Sequence: One possible solution for Sequence Sorter.

At the same time, the player will also be aware that their personal creation fits within a wider community in more ways than one. First, as described, every solution that they have generated will have been measured against other solutions. Second, they will have access to the community of players who have created these alternative solutions via online discussion boards. Such forums afford an opportunity for players to expand upon their strategies and to explain their thinking in crafting specific solutions. Finally, a player will also have the opportunity to engage in more direct ways with the community by taking up the second part of the game, TIS-NET, which consists of puzzles designed by other players. TIS-NET is a testament to the potential the game offers for programmers to bring their own approach to utilizing, designing and reworking the interface of the game. Similar to mainstream games that attract large fan bases, TIS-100 has developed into a cooperative endeavour, despite its quite niche appeal.

Therefore, TIS-100 can be conceptualized as a game that takes the player back to the early days of programming on multiple levels. With its story about a top-secret device that drives a player to complete menial tasks for a cryptic objective, the game recalls a period when bureaucratic institutions attempted to dictate the use and purpose of computers. Then again, the unidimensional console and assembly language programming that define the game also invite a player to reflect on the most basic interactions between the individual and the computer. Yet what the player tends to discover is that the heart of these interactions consists simply of themselves -- the approaches that they have adopted and the patterns that they have generated while progressing through the game. Most of all, though, TIS-100 inspires players to look beyond their own experiences with the game to engage with how other people have immersed themselves in the same world.

References

at Google, T. (2017). *Zachtronics: 10 Years of Terrible Games*. YouTube video. https://www.youtube.com/watch?v=Df9pz_EmKhA

Bissell, T. (2010). *Extra Lives: Why Video Games Matter*. Pantheon Books.

Campbell-Kelly, M., Aspray, W., Ensmenger, N., & Yost, J. R. (2013). *Computer: A History of the Information Machine* (3rd ed.). Hachette UK (Westview Press). https://doi.org/10.4324/9780429495373

Farrow, R., & Iacovides, I. (2014). Gaming and the Limits of Digital Embodiment. *Philosophy & Technology*, *27*(2), 221–233.

GDC. (2018). *Inside the Indie Mind: Zach Barth (Zachtronics)*. YouTube video. https://www.youtube.com/watch?v=iAfcJB1WaDU

Isaacson, W. (2015). *The Innovators: How a Group of Hackers, Geniuses, and Geeks Created the Digital Revolution*. Simon & Schuster.

Levy, S. (2010). *Hackers: Heroes of the Computer Revolution*. O'Reilly.

Rosin, C. (2019). Stepping Stones to Inductive Synthesis of Low-level Looping Programs. *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, *33*, 2362–2370. https://doi.org/10.1609/aaai.v33i01.33012362

Rusch, D. (2017). *Making Deep Games: Designing Games with Meaning and Purpose*. CRC Press.

Sayer, M. (2016). *Can Videogames Teach You Programming?* Rock Paper Shotgun blog post. https://www.rockpapershotgun.com/can-video-games-teach-you-programming

Zachtronics. (2011). *SpaceChem*. Zachtronics.

Zachtronics. (2015a). *Infinifactory*. Zachtronics.

Zachtronics. (2015b). *TIS-100*. Zachtronics.

Zachtronics. (2016). *Shenzhen I/O*. Zachtronics.

Zachtronics. (2017). *Opus Magnum*. Zachtronics.