THE VIDEOGAME TEXT: TYPOGRAPHY AND TEXTUALITY

By

ZACHARY NATHAN WHALEN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2008

To my wife, Stacy; and to our dog Teddy;
who both listened patiently to all my ideas.

ACKNOWLEDGMENTS

First and foremost, this project could not have been envisioned without the unwavering support and encouragement of my wife, Stacy, who continued to be my strongest supporter through everything. Several other individuals have contributed nuggets of their time or knowledge toward the successful completion of this project. My committee members, Donald Ault, Terry Harpold, Marsha Bryant, and Brian Slawson each provided crucial feedback in the early stages and were willing to work with my accelerated schedule for completing this project on time as well as my late changes to the organization of chapters. Additionally, although not officially part of my dissertation committee, Paul Fishwick supplied an important perspective on my work in its developmental stages, and Gregory Ulmer was kind enough to write a letter of recommendation on my behalf.

Matthew G. Kirschenbaum provided feedback and encouragement, and his book *Mechanisms* proved to be an important influence on the development of my thinking about digital media. Steven Jones kindly offered a pre-publication copy of his book, *The Meaning of Videogames*, which has been similarly helpful in enhancing my ideas about how to study videogames with a textual approach.

My friends and colleagues at the University of Florida, particularly the Digital Assembly and the Editorial Staff of *ImageTexT,* have sharpened my focus in numerous ways, mainly by fostering an active community of Digital Media scholarship through collaborative projects and annual conferences. Similarly, the wider community of Gameology.org and its readership provided an important audience and sounding board for discussing ideas. In particular, my co-editor Laurie N. Taylor read early drafts of chapters, offering numerous helpful suggestions.

Several individuals offered their time or knowledge to help answer key questions on historical matters. Luc Devroye and Grant Hutchinson pointed out or provided sources of information on the Moore Computer typeface, and Simon Daniels of Microsoft Typography supplied background information about Westminster. Sean Riddle and Fredric Blåholtz answered my questions about the Channel F console and its Easter Eggs, and Warren Robinett kindly provided his recollection of the implementation of a character set in the *Basic Programming* cartridge. Finally, Gio Wiederhold and Ted Panofsky provided valuable insight into the function and programming of *Galaxy Game*.

I would also like to thank the faculty of the Department of English, Linguistics and Speech at the University of Mary Washington who will accept me as their colleague in Fall 2008.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF OBJECTS

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

THE VIDEOGAME TEXT:
TYPOGRAPHY AND TEXTUALITY

By

Zachary Nathan Whalen

August 2008

Chair: Donald Ault
Chair: Terry Harpold
Major: English

This study asks how the design and configuration of text in videogames contributes to their textuality. I argue that videogames are texts in the sense that they consist of material artifacts generating meaningful content when engaged by users. Videogames are complex and expressive digital artifacts worthy of critical analysis, but much of the existing scholarship on games emphasizes their formal elements like narrativity, genre, or interactivity, without giving enough attention to their specific technological constitution. As is the case in any aesthetic medium, such as verbal text, film, or still images, videogames are subject to the affordances of their raw materials, and like these other media, videogames communicate in ways that incorporate the traits of those materials, where "materialis" include physical structures like console hardware and display screens and logical logical like bitmap graphics. *The Videogame Text* argues that alphanumeric characters shown on the videogame screen (including score display, character dialog, user interfaces, title screens, etc.) reveal discursive patterns of materiality embedded in these structures.

I orient this argument by beginning with the origins of videogame typography, but not in order to claim that earlier forms are archetypes which newer forms invoke. Rather, since both typography and material affordances are traits designed to be taken for granted or made invisible,

videogames of a sufficient historical remove illustrate more obvious and visually apparent evidence of these constraints. Typographic echoes of those constraints appear in other media and graphic designs as a way of invoking a video game context, so a study of videogame typography must account for this diffuse set of forms much like the textual studies approach to literature that treats the text as a multifaceted paratextual entity. Drawing a parallel between the textual studies approach to literature (exemplified by the work of Jerome McGann, Johanna Drucker, and others) and a trend toward artifactual analysis in new media and game studies (exemplified by the work of Ian Bogost, Nick Montfort, Steven Jones, and Matthew G. Kirschenbaum). In short, I argue that the textuality of videogames consists of differently and densely layered modalities of representation, which include formal structures such as game design, physical structures like the television or monitor display, and logical structures like programming code. By unpacking the inner workings of these modalities, we may better understand the impact of videogames as cultural artifacts. We may also move beyond prevailing theories of videogame analysis, which too often emphasize formal taxonomies as well as conceptual and disciplinary boundaries.

CHAPTER 1
VECTORS TOWARD A DISCOURSE ON VIDEOGAME TYPOGRAPHY

The title of this study, *The Videogame Text*, comprises two distinct but related assertions. The first, is that videogames contain text: alphanumeric signifiers that communicate verbal or semantic data to the games' player. Therefore, the videogame text is that specific kind of text appearing in the context of videogames. The second assertion is that videogames are texts: they are materially distinct digital artifacts which give structure to a specific communication act through the manipulation of symbols. These symbols may be significant verbally, textually, culturally, or ludically (in the sense that their symbology serves the functions of play), but their coinvolvement in the production of meaning creates a characteristic textuality of videogames. By invoking this dual sense of the term "text," this study argues that videogames depend on typographic expressions of textuality because the design of alphanumeric symbols in games depend on the constraints of the game's platform. Typographic symbols exhibit a textuality that can demonstrate constraint without necessarily impacting a glyph's ability to reliably function in a symbolic context. In other words, a highly constrained "3" is still as much a three (and as legible) as an highly ornate "3," and as such, the aesthetics of typographic expression are not tied as closely to questions of representation as is the case in other symbolic systems.

Furthermore, text plays an expressive role within games, determining how they produce meaning and express their unique aesthetics. Analyzing how these alphanumeric forms participate in and give shape to videogame textuality invites an analysis that goes to videogaming's programmatic core as a medium. In this way, the major thrust of this study is that videogame typography exemplifies the discursive mediality of videogames. This places videogames alongside other forms of electronic textuality such as interactive fiction because I find the visual articulations of their status as digital artifacts to be relevant contexts for analysis.

This project is also an effort to bring together two lines of inquiry which have previously had little contact: textual studies and videogame studies. The benefits of this convergence will be made apparent through the course of this study, but I propose typography as a logical entry point. Standing astride the textual and graphical regimes of signification, typography is an invitation to explore possibilities of meaning-making in multiple, simultaneous modalities. So-called expressive typography communicates through visual means as much as or more than the verbal or numeric content being rendered, and videogames routinely combine visual, verbal, auditory and even haptic sensory information into a coherent expression of a game space. The typographic form of the printed word has proven a valuable avenue for critical analysis in literary studies, so it is an important addition to the nascent field of videogame studies. Put another way, videogames have been one phenomenon that critical studies of typography and textual studies have overlooked; conversely the growing field of academic videogame studies has yet to incorporate an approach to video game typography within its emerging critical vocabulary. Videogame typography does, however, exhibit aesthetically unique properties and demonstrates through its gestures and forms a textual ontology that depends for its expression on the constraints of its material environment in addition to its semantic content.

In order to provide the appropriate context for this argument, this first chapter situates the question of videogame textuality within the present academic conversation around videogames. Game Studies as such casts a wide disciplinary net, and as a result it is difficult to faithfully summarize even the most dominant ideas that inform its current state of what Jesper Juul has described as "productive chaos."[1] The fact that Ian Bogost less optimistically uses the phrase "functionalist separatism" (*Unit Operations* 52) to describe the same situation indicates the degree to which conflict is at the heart of the shaky interdisciplinarity of the collective field. Because

1 Jesper Juul uses this term in his introduction to *Game Studies* 5.1 ("Where the Action Is"). Though Juul is being optimistic, his attitude implies that there are serious, intractable disagreements among those who study videogames seriously.

videogames are still a relatively recent cultural genre, it is not surprising that game studies has yet to engender a fully autonomous academic discipline focused on their study.[2] Given the wide variety of themes and genres within videogames, perhaps this is not a bad thing. Those who write on and research video games, therefore, usually do so as a function of a primary field (Media Studies, Sociology, English, etc.), though a growing number of scholars do identify themselves as primarily video game theorists. Both Juul's and Bogost's statements hint at the way interdisciplinary tension comes to bear upon conversations within the nascent field, but this may be a sign of the field's robust potential for growth. Researchers have been studying videogames for at least 20 years now, but foundational principles such as what constitutes genre in games have yet to reach a consensus view.[3] As a way of orienting what is at stake and what is assumed in the various conflicts, a brief discussion of the basic terminology around the object of study may illuminate some of the ideas certain parties bring to the conversation.

Terminology: Video Games, Videogames, and Gaming

Although there are differences, competing terms like "video game," "videogame," "computer game," "electronic game," "digital game" seem at first glance to be more or less synonymous. Considering that each construction attaches a technologically specific descriptor to "game," it is worth noting how little restriction these modifiers seem to place on the way these terms are actually used. Some of the objects commonly referred to as "video games" may not actually use video technology at all. In fact, one of the earliest games that can make a case for being the first ever videogame was displayed on an oscilloscope.[4] Furthermore, many games that

2 Several schools do offer majors in game design, and there are programs like the one at the Georgia Institute of Technology, which encompass the broader field of Digital Media. To my knowledge, however, there is not yet an autonomous university department of game studies on the order of similar film studies departments.

3 Genre in videogames is complicated by the way generic labels describe different qualities in games than in other media. For example, while "Western" and "science fiction" are valid genres of film that describe something about the film's setting, videogame genres such as "first-person shooter" or "real-time strategy" describe something about how the game is played. A first-person shooter game may be set in a far future context, but it will not be labeled a "science fiction"game. I attempt to address the problem of game genre in an earlier essay (Whalen, "Game/Genre").

4 William Higginbotham programmed *Tennis for Two* in 1958 for display on an oscilloscope (Burnham 28). *Spacewar!*, developed on the PDP-1 computer in 1961 used vector graphics on a Type-30 CRT display (Graetz

possess electronic components, such as pinball machines or board games like *Operation*, are not normally included in conversations about videogames. Therefore, "electronic game" is not specific enough, and "electronic" may potentially describe such a diverse group of games (all those relying on electricity) that it will be difficult to form a coherent concept of a medium. "Computer games," like "arcade games" and "TV games," suggests a particular platform for gaming at the exclusion of alternative platforms, although it is true at some level all of what we think of as videogames require some kind of computation. In common usage, however, those using the term "computer games" often mean to make a distinction from "console games" even though current generation game consoles contain more computing power than average PCs. David Buckingham acknowledges this distinction, but concludes that, at least in the United Kingdom, "games are called 'computer games' irrespective of whether they are played on a PC or on a dedicated games console such as a Playstation or an Xbox" (5).

Other terms are simply unsatisfactory for the task of limiting the definition to a core of ludic media. More general terms like "entertainment software" and "interactive entertainment" are simply too vague because they remove the technological descriptor as well as the crucial "game," without which one may plausibly include Microsoft Excel among "entertainment software" if it used for purposes one finds entertaining. Some of the differences among these terms appear trivial, but a conscious choice to use any of the above may cordon off certain critical assumptions while foregrounding others. Using "video," for example, emphasizes the visual and graphical elements of the game object, while "computer" emphasizes the programming structure or, perhaps, the rules that generate the game. Because my present study does focus on the visual elements of games, "video" does reflect the appropriate emphasis. There is, however, another matter to consider.

If we accept the term "video" to mean any visual display of moving images, as opposed to a specific technology involved in the production of video imaging, like VHS, PAL, or NTSC, there

46).

is still the question of how the “video” prefix performs a modification of “game.” No major dictionaries yet include the portmanteau, “videogame,” but some writers use the term consciously as a way of declaring that these artifacts are not simply a gaming alternative co-equal to parlor games, pinball games, or sports games. As Ian Bogost explains, “But I use the term ‘videogame’ for rhetorical reasons. Separating the words, in my opinion, suggests that videogames are merely games with some video screen or computer attached. But, I believe that videogames are fundamentally a computational medium ... I think that closing the space, in part, helps consolidate this concept” (Bogost, “Videogames or Video games”) So-called “videogames” are, therefore, fundamentally unique phenomena that carry with it concepts that are not clearly evoked by either “video” or “game” alone. From this point of view, videogames still depend on visual technology and borrow practices from prior gaming forms, but their significance lies elsewhere than either their status as games or their display technology. For the purposes of this paper, I have chosen the term *videogame* in part because it is the more familiar term, but also because the unique technological constraints of videogames, including the specifics of video display technology, play an important part in the analysis that they engender. Furthermore, the argument that there is a coherent videogame medium is assisted by employing the neologism, *videogame.*

A further idiomatic subtlety calls attention to an additional wrinkle in the academic studies of video games: the use of the gerund *gaming* (or *videogaming, computer gaming*, etc.) to invoke what is being studied. Using a verb form as opposed to the noun, game, implies that the object of study is the act of playing, and this starting point leads to different theoretical conclusions. Accordingly, some contemporary theorists who look to classic discussions of play such as Johan Huizinga’s anthropological text of 1947, *Homo Ludens,* or Roger Caillois’s *Man, Play, and Games* tend to use the verb form. Markku Eskelinen, for example, devotes an essay to what he calls the “gaming situation,” a term he uses in order to disrupt the narratological argument stemming from assumptions that treat the game as a textual object ("The Gaming Situation"). Huizinga and Caillois are both worth noting because theirs are among the first scholarly studies of the broad concept of play in culture and they are each useful for placing videogames in a cultural context. But both focus on play as a cultural idea, necessarily including all kinds of games (that is, not videogames) as well as instances of unstructured or even metaphorical play. By placing their emphases outside of specific instances of games, Huizinga and, to some extent, Caillois, provide logical starting points for approaches to videogame theory that do not focus on the material or textual aspects of the experience. Rather, a game is simply what one uses to structure play, so to refer to the cultural idea of *gaming* as opposed to the specific case of any one *game* bypasses the question of materiality altogether.

In a different sense, the title of Alexander Galloway's book, *Gaming: Essays on Algorithmic Culture*, does seem to correspond to his emphasis on the act of play which is manifest in his thesis that "video games are actions" (Galloway 2). Galloway clarifies that video games are an active medium, and in his book he explicitly takes into account the actual artifact of the videogame. But by approaching the subject originally through the titular designation *gaming*, Galloway subtly and specifically sets up his argument about what kind of thing videogames are and how they should be studied.

Finally, there is a grammatical ambiguity regarding the word *game* in the English language such that it is correct both to speak of checkers as a *game*, as well as to refer to a *game* of checkers. In the first sense, *game* refers to the general category of experience under which checkers may be appropriately cataloged, and in the second sense, *game* refers to a unique instance of playing checkers that is distinct from other instances of playing checkers. This ambiguity also presents an opportunity for theorists to extrapolate positions regarding videogame textuality, such that *game,* in the sense of labeling what checkers is, can refer to a text: a system consisting of signs and operations intended for reception by an audience of game players. Accordingly, the term videogame as it is used throughout this study is intended to invoke that sense of textual determinacy.

Ludology and Narratology

The most prominent disagreement about how to study video games resides in what has come to be known as the "ludology vs. narratology" debate. Though the debate as such seems to have cooled,[5] and some suggest that it never really occurred,[6] the disagreement does reveal some very different ideas about how videogame studies should be carried. Part of the reason for the escalation of rhetoric in this debate is surely that the field is young, with much of the disciplinary territory remaining unclaimed. The territorial attitude is directly encouraged by Espen Aarseth's position statement to launch the journal *Game Studies*:

> The greatest challenge to computer game studies will no doubt come from within the academic world. Making room for a new field usually means reducing the resources of the existing ones, and the existing fields will also often respond by trying to contain the new area as a subfield. Games are not a kind of cinema, or literature, but colonising attempts from both these fields have already happened, and no doubt will happen again. (Aarseth, "Computer Game Studies, Year One")

In this formulation, the work of game studies as such is first and foremost that of staking a claim of priority and warding off theoretical conquistadors from other fields of study. While Aarseth's attitude is perhaps understandable in the context of introducing a journal, his often-repeated colonial metaphor is indicative of a surprising hostility in what really should be a congenial discussion.

The terms of the debate are allegedly simple: so-called ludologists are supposed to argue that videogames' formal properties such as rules define gaming's core characteristic and require an appropriately rule-oriented criticism, whereas so-called narratologists are supposed to argue that

5 Julian Kücklich's *DiGRA Hardcore* column, for example, addresses the debate as having already concluded. In his estimation, the ludologists won ("Game Studies 2.0")

6 See, for example, Gonzalo Frasca's presentation at the 2003 Level Up conference, "Ludologists love stories, too: notes from a debate that never took place."

videogames are simply a new form of storytelling much like film or hypertext fiction. But both of these characterizations are inaccurate and promote flawed arguments if taken in the isolation that some more vocal scholars have advocated.[7] The term "narratologist" is also particularly problematic since those who do study games with an eye toward narrative are not necessarily following the methods of the narratology that has its roots in Vladimir Propp's morphologies of folktales. That is, someone accused of narratologism may not necessarily be at all interested in incorporating the ideas of Gérard Genette, Tzvetan Todorov or Seymour Chatman into a study of games. In fact, so-called ludologists are more likely to draw on Structuralism in general or Russian Formalism in particular in developing their own typologies of game structure.[8] Michael Mateas and Gonzalo Frasca have used the term "narrativist" to more accurately portray non-ludologists as those simply interested in story elements, but even that term carries its own baggage as it also used within discussions of table-top role-playing games to refer to a particular game strategy.[9]

Though much of this debate does appear to be territorial, there is more at stake in these disagreements than one type of discourse simpling holding privilege over another. Videogames are a powerful and pervasive medium, and the ultimate objective of any study of gaming, whatever its disciplinary heritage, should be to increase or improve our understanding of these artifacts. Thinking critically about and with videogames can only come about through an open and intellectually honest discourse that involves the best thinking of *all* relevant fields.

As a way around the ludology versus narrativism discussion, some writers have either sought to strike a compromise between the two positions or, recognizing that both extremes are untenable, sought entirely new approaches. Jesper Juul's *Half-Real* attempts a compromise route, arguing that games are always both rules and fiction. The title of his book puns on the games *Half-Life* and

7 The otherwise excellent volume *First Person: New Media as Story Performance and Game* contains some of the shrillest examples of this kind of exchange.
8 Aarseth's taxonomy of the varieties of cybertext is particularly indebted to this variety of category formation.
9 Cf. Edwards, Ron, *GNS and Other Matters of Role-playing Theory.*

Unreal and proposes that the half-reality of games consists in their rules being real, and their "worlds" being fictional. While Juul's conciliatory approach is encouraging, the organization of his book betrays his belief that rules are the fundamental characteristic of games and that the fictional worlds are superimposed arbitrarily on the underlying rule structure. Certainly, there would be no game without rules in some sense, but the so-called "fiction" of games (which must include the graphical representation of game entities as well as aesthetic and narrative features) exists only in service to those rules. The risk of separating these two functions of gameplay is that a videogame's potential for expression, including elements like narrative, aesthetics, and cultural context, is relegated to the game's non-essential or arbitrary components. In other words, the fiction of games is not a fundamental component, and any expression we identify in gaming is something that could also potentially appear somewhere else. This bifurcated view is tempting from the perspective of a designer, since some games like the endless Space Invaders clones available as Flash games on the web do attempt to inscribe meaning onto a universal game template by exchanging the symbols for the aliens or the gun turret.[10] The transformation of Space Invaders into Tax Invaders (Figure 1-1), Pepsi Invaders, or countless others occurs at Juul's fictional level, and therefore, the most important message of these games is occurring at a symbolic level.

Juul actually demonstrates this in two ludically identical games offered on his website: the comic "Puls in Space" (after a Danish TV Show) and the satiric "Game Liberation" (Figure 1-2) which replaces the invading aliens with invading academic fields (narratology, psychology, film studies, pathology) threatening to destroy ludology's position of privilege.[11] Furthermore, while this oppositional structure is appealing as a universal template for the production of meaning in games, it sacrifices too much of the experience by relegating it to a phenomenological autonomous

10 "John Kerry Tax Invaders" (archived at <http://web.archive.org/web/20040517093809/http://www.gop.com/taxinvaders/>) is a particularly ineffective example of this type of game.

11 Game Liberation is available on Juul's website: <http://www.jesperjuul.net/gameliberation/>.

space in which its meaning-making need not be actually game-derived. In other words, in the "Game Liberation" example, following Juul's formula for analysis, we gain the entire meaning from a description of the game (that ludology shoots down other theories) without having to play it at all.

Furthermore, it is interesting that Game Liberation is a piece of software, and as such is subject to non-explicit affordances which nevertheless influence how the game may be interpreted. By considering the formally material conditions of the game as it is deployed in a web browser, we can arrive at contrary interpretations of the game text that rely on the limitations of the game's fiction. Game Liberation is a piece of software written in the programming language, Java. The game runs in a Java applet within a web browser, so in that sense, its primary platform may be considered the Java engine running on the user's computer, while its secondary platform is the user's specific web browser. In some cases, the applet or section of the web page containing the Java engine, fails to render the game correctly, resulting in alterations of the experience, and if we are meant to understand the rhetoric of the game by interpreting the appearance and behavior of the symbolic elements within it, then the variations in those elements must also be considered significant. For example, in one memorable instance of playing the game, an error in my browser caused the graphic for the player-controlled gunship to disappear. I discovered, however, that the game was otherwise running correctly, and because the game encodes the player's ship sprite in a way such that its ability to fire bullets does not depend on its actual visual presence within the game space, I found I could fire at will without any risk of my ship being destroyed. Reading this in the context of the argument Juul is making both with the game and with *Half-Real*, it seems that the meaning that my broken version of the game generated communicates the idea that ludologists attack competing theories without asserting their own or exposing themselves to counterattack. This may be an effective tactic, but it is a unfair one, only coming into play when the rules of the game had already been broken. This counter-reading of Game Liberation relies on an accidental intrusion of the apparatus, but it exposes the limitations both of the template game as a rhetorical form and the adversarial rhetoric implied by its counter-reading.[12]

These technological and material elements of the apparatus remain external to the “real” structure of the game and seem to be elided in Juul’s discussion: “the material support needed to play a game (like the projector and the screen in cinema) is *immaterial*, since games are not tied to a specific set of material devices, but to the processing or rules” (Juul, *Half-Real* 53, emphasis in original). By this, Juul does not necessarily mean that material support is irrelevant, but statements like this betray the positivist assumptions behind Juul’s argument and foreclose certain kinds of analysis in the same way that a good deal of the more general conversations around computing treat electronic text and image as entirely separate kinds of objects. The technological reasons for this separation are practical, but the intellectual fallout is debatable and, I would argue, is at the core of the hyperbole surrounding some discussions of computers within Humanities. In other words, by granting a separate ontology to words and image, theorists encourage thinking of text (code) as a sort of ideal form of presence for an image. In the same way, rules form the source code for Juul’s discussion about the structure of games, and his analysis similarly favors semiology at the expense of a more robust, materialist approach to the medium.

A different response to the ludology and narratology divide identifies each extreme as a consequence of similar same functionalist impulses. In *Unit Operations: An Approach to Videogame Criticism*, Ian Bogost outlines an approach based on understanding a videogame or potentially any text as “a configurative system, an arrangement of discrete, interlocking units of expressive meaning” (*Unit Operations* ix). The key to this idea is that it defines games as configurative systems whose meaning is determined by the relationships among constituent

12 I understand, of course, that Game Liberation is intended facetiously, but the way it presents the relationship among schools of thought is literally (and ludically) confrontational.

elements, but Bogost makes an important point to distinguish his approach from earlier videogame theory.

For Bogost, these units can include programming structures (specifically, object-oriented programming), story elements, or player interaction, and their significance is in the way they mutually interlock to generate the expressive content of the game. Among these elements, the graphic appearance of game content also contributes to the game's total expression. In a typical first-person shooter (FPS) game, for example, the three-dimensional environments of the game are generated through a software engine that projects three-dimensional planes and decorates these with image files known as textures. Together, these textures and spaces generate the cumulative effect of a game's spatiality, and that spatiality can be as expressive and aesthetic on its own terms as architecture is in the real world. In other words, the labyrinths, cityscapes, and forests that provide the setting for three-dimensional video games express their particular sense of spatiality through the interaction of graphical and architectural elements. But moreover, the goals and pathways set by the rules of the game involve the player in a performance that, along with the rendered spaces of the game's environment, expresses its spatial "feel."

Along with these surfaces, an additional surface that is often taken for granted stands between the player and the game world into which text intervenes. This surface, known as the Heads-Up Display or HUD, usually provides information to the player that corresponds to her goals and status in the game. The information acts as a kind of glue connecting the elements of the game world that constitute its peculiar spatiality with the actions and intentions of the player in a way that completes the total, unit-operational system of the game. At first glance, the HUD seems to provide a textual context for the visual, spatial world of the game, and if considered this way, one might argue that this demonstrates a hierarchical relationship between words and images in which images provide content and words provide context. But this hints at the same semiotic tension underlying Juul's definition of games. In the FPS game, *Call of Duty*, for example, alphanumeric text and informatic images at the corners and top of the screen provide constant feedback indicating the values of various statistics like ammunition quantities and health, as shown in figure 1-3.

Unit analysis also provides a reasonable means for including typographic form within the expressive domain of videogames. Among the units in operation that potentially contribute to the game's overall aesthetic, the appearance of text and the shapes of letterforms contribute to the total expression of the game experience. This is especially important when considering the role the HUD plays in making sense of the player's interface with the game diegesis, and it is logical to think of this text as a image when it is narrowly informational or numeric. That is, since text within the HUD supplies little semantic content of its own and is most often plainly numeric, its primary function is fulfilled in being glanced at as opposed to being read. Also, although the numerical values do correspond to discrete values assigned by the programmatic conditions of the game as it progresses, the HUD communicates those values in a particular way, and the aesthetic appearance of that text depends on the graphical context of the game world and works together with it to produce an overall effect. Therefore, even though text and images in computing are stored and retrieved through fundamentally different means (at least for modern games), the possibility of expression, particularly in video games, depends on their interlocking operation. In other words, with unit operations in play, a game like *Half-Life 2* can be said to rely on images and text in such a way that visual information (e.g., spatiality) is employed toward semantic ends and textual information in turn contributes to the visual spatiality of the total game apparatus.

The recurring question of narrative and whether games should be considered narratives is something of a MacGuffin for game studies. Markku Eskelinen, for example, uses the specter of narratology as a wedge for excluding certain individuals or disciplines from the valid study of videogames. Marie-Laure Ryan, an acknowledged narratologist, addresses this by taking the question of narrative as one of degrees. In this way, and through a list of potential qualities, Ryan elaborates a context in which videogames may be acknowledged as having some degree of narrativity, which is a different type of conclusion than the "games are narratives" argument Eskelinen selects to disagree with. This is relevant not only to the question of narrative in games, but also other areas of narrativistic controversy such as instrumental music. As Ryan writes, the question of intent and recognition is also significant to the narrativity question:

> The property of "being a narrative" can be predicated of any semiotic object produced with the intent to evoke a story to the mind of the audience. To be more precise, it is the receiver's recognition of this intent that leads to the judgment: this text is a narrative, though we can never be sure that sender and receiver have the same story in mind. "Having narrativity," on the other hand, means being able to evoke such a script, whether or not there is a text, and if there is one, whether or not the author intended to convey a specific story. (Ryan 10 - 11)

Later, Ryan refers to the relationship between computer games and narrative as one of elective affinity, rather than "necessary union," indicating a productive way forward in the debate (183).

Ryan does not mention expressive typography specifically, but it is clear that this elective affinity could strike parallel balance in these and other forms of semiotic text. As an example of this, Johanna Drucker writes in *The Alphabetic Labyrinth* of many attempts in the 19th century to divine the hidden meaning of alphabetic symbols. Paraphrasing one such writer, Luther Marsh, Drucker writes, "Nicely articulating the crucial stimulus to fascination with the history of the alphabet, Marsh stressed that the alphabet itself was the repository of history, not only its instrument or means" (*Alphabetic Labyrinth* 278). In this way, the tension between content and form in typography, and the interpretation of form as expressive content, is in some ways similar to the conflict between narrative and interactivity. Both invite the reader to engage the text on more than one simultaneous and (possibly) mutually exclusive semiotic levels.

Digital Textual Studies

Typography combines verbal and graphical expression, so its importance within the overall expression of a videogame is consistent with the modality of the videogame medium. But the specific ontologies of text and image in computing media invite humanistic scholarship that takes this separation for granted. For example, all too common conflation that words and images in computing are both reducible to binary code creates the impression that the binary code is the ideal, abstracted form of any text without acknowledging the differences inherent in the successive layers of abstraction intervening between the act of digital inscription and of reading. Mathew G. Kirschenbaum has skillfully unpacked these layers, arguing that a digital artifact exhibits both forensic and formal materiality through the interventions between these layers (Kirschenbaum, *Mechanisms* 9 - 11). Furthermore, as Jerome McGann argues in *Radiant Textuality: Literature after the World Wide Web*, the problems inherent in electronically archiving literary works foreground latent problems and assumptions in traditional, semiotics-influenced textual studies. In this context, the main challenge for the bibliographer lies in capturing visual and textual components of a work in a way that allows electronic search and retrieval (the main advantage of a computer archive) but also remains faithful to the original work itself. The works of William Blake provide clear examples of the difficulties inherent in this challenge. An archivist can simply transcribe the apparent text in a plate of *Urizen* and store it as a searchable text file, but if a researcher is cross-referencing instances of, say, words with the capital letter *O* when they appear near images of orbs, even combining a text file with a scanned image of the corresponding plate is inadequate to retrieve the relationship the researcher is investigating. This hypothetical conflict between the affordances of an archive and the aims of a researcher can be seen as an outgrowth of the Romantic ideology of intentionalism to which McGann responds in his *Critique of Modern Textual Criticism*. The alternative approach to textual studies that McGann develops in conjunction with what D. F. McKenzie terms "the social text," necessarily includes a robust sense of textual materiality, and it is interesting to juxtapose McGann's sense of materiality with the fictional worlds of games which Jesper Juul identified as immaterial in the quote mentioned above. Like the eclectic text that is the goal of the New Bibliographer, the immaterial videogame envisioned by Juul and others can only lead to similar forms of scholarly crisis – whether that is considered a "productive chaos" or a "functionalist separatism."

Juul seems to use the term, *immaterial*, in the sense of, *irrelevant*, but at a fundamental level, Juul's approach bears the same kind of semiotic logic that McGann critiques. Juul's formula pits fiction against rules in a manner that imagines rules as the transcendental form of the game toward which artifacts of the fictional world are arbitrary or at best autonomous. In other words, videogame fictions are important only because of the means by which they communicate the transcendental reality of the rules. This basic assumption forecloses the procedural, material-based criticism that McGann suggests and that Bogost, I would argue, opens up for videogames. The alternative approach to videogames that I am arguing for here follows this trajectory, recently articulated by Kirschenbaum, and begins with the medial and material conditions of the game as the foundation for expression, aesthetics, and interpretation of typography.

Johanna Drucker has written about some of these issues in relation to electronic textuality, and her work therefore provides an interesting bridge between game studies and textual studies. Discussing the problems inherent in the electronic archive, Drucker begins with the basic entity of the letter, and by considering the form of the letter as it initiates the production of meaning, she provides a context for discussing typographic form as an object available for textual criticism. In her essay, "Intimations of Immateriality," Drucker poses the central problem of electronic textuality as a pair of identities for the bodies of letters – letters either have an inherent essence so that recognition proceeds from correspondence to that form, or letters derive identity from their relationship to a system of signs. She describes the first identity as phenomenological and the second as semiotic, but it is important to note that neither of these ideas involves expression or aesthetics as a necessary feature. Accordingly, Drucker proposes a new approach which is surprisingly harmonious with Bogost's: "Therefore, I suggest that in addition to logical and natural language, we consider *configured* language (that is, language in documents where format, graphical organization, or other structural relations contribute substantively to textuality) in the electronic context" (Drucker, "Intimations of Immateriality" 55, emphasis in original). She further suggests that this approach corrects a rift between form and meaning in text that is unique to the electronic environment, but it is interesting that by employing the term, "configurative," Drucker echoes Espen Aarseth's definition of one type of cybertext that is in turn echoed by Bogost's definition of unit operations.

In his influential work, *Cybertext: Perspectives on Ergodic Literature*, Espen Aarseth makes a case for a new kind of text, one that is *ergodic* in the sense that it demands non-trivial input from the user.[13] Aarseth does not spend much time in this work dealing directly with videogames, but he develops a typology of cybertexts that contrasts ergodic with linear texts. From this it is clear that videogames are among those texts in which configuration is the dominant activity of engagement. Markku Eskelinen argues that a criticism based on this understanding should itself be configurative rather than interpretive; this requires including "a combination of ends, means, rules, equipment, and manipulative action" within the gaming situation (Eskelinen, "Towards Computer Game Studies" 38). But it is not clear from Eskelinen's essay how one practices configurative criticism unless one takes an approach similar to Bogost's unit analysis. For Bogost, unit operations define a system that is itself configurative, so a criticism that first identifies a system's constituent units and then proceeds by rearranging those units into a logical meaning also configures meaning in a

13 Turning a page is trivial effort, choosing a hyperlink to click on is non-trivial because it requires the user to make a decision.

similar way. As an example of this, Bogost analyses the Tom Hanks film *The Terminal* by identifying and juxtaposing units that, when taken together, configure the entire film as a meditation on different kinds of waiting. In this way, the work of criticism lies in recognizing and arranging the elements that generate productive meaning, not in responding correctly to the film's inherent meaning or transcendent ontology.

For Drucker, the configuring that takes place in configured language appears to originate in authorship, but preserving an appreciation for it through the act of criticism bears some resemblance Bogost's procedural criticism. Simply put, both attempt to move beyond the dominion of structuralist criticism by calling into question its semiotic basis. Drucker moves this kind of criticism, derived from McGann's initial critique, into purely electronic contexts (as opposed to transcribed or electronically archived print works), and Bogost moves from the programming logic of video games to produce a similar kind of criticism that accounts for their unique ontology. Bogost does not discuss typography or textuality specifically, but since Drucker does, this coming together provides an important critical framework for developing a critical approach to videogame typography.

Text in Videogames

Whereas textual studies has the benefit of decades of textual scholarship to respond to, videogames themselves do not yet have a standard historical narrative to account for the development of their characteristic typefaces and letter designs. One initial goal of a theory of videogame typography should be, therefore, to discover and document that history. While the chapters that follow provide a first step in this historical direction, this is a history mostly taken for granted by both academic and hobbyist communities. What I set out to discover through this project are the technical, material, and aesthetic influences that bear upon the design of type and arrangement of text within early videogames.

If the lesson of textual studies is that a materialist approach to criticism tends toward increasing (rather than collapsing) the availability of plausible interpretations of a text, even at the risk of supplying incommensurable textual singularities, one task of the archivist in service of this approach seems to be to increase the granularity of the textual object. In other words, as finer detail and greater variation appear within available materials, a logical result is an approach like the one mentioned above. For example, since the Blake Archive (www.blakearchive.org) allows easier access to the multiple versions of Blake's work, and it improves the detail available in its proprietary image notation system, critical interpretations of Blake's corpus logically proliferate in more directions. Similarly, classic studies of typography like Robert Bringhurst's elegant volume *The Elements of Typographic Style* illuminate minute particulars of type form that are practically indiscernible to the uninitiated reader to harness productive readings of typographic form. For example, these readings of technical elements like stroke, axis, and weight, provide evidence for conclusions about the prevailing type forms of different historical eras. Two recurring themes in Bringhurst's discussion compare type forms by the way they relate to the human body and architecture. Early typefaces mimicked handwriting for their forms, so their leaning axis is an echo of the slant of the calligrapher's stroke. In this way, the form of the type contains an element traceable to the human body that is distinct from the so-called rational vertical axis evident in neo-classical type forms. Typefaces of this variety such as Baskerville, contain an idealization of form that is also evident in neo-classical architecture, so the shape of the letter contains within it an element reminiscent of a particular spatiality. Both the humanist and spatial properties of type are relevant to videogame typography for the way game text sutures the game world to the players experience while at the same time interlocking with the three-dimensional spatial forms within the game diegesis to produce an overall aesthetic effect. Bringhurst, not surprisingly, does not discuss videogame typography, but the closeness of his reading provides an elegant inspiration for the kinds of insights that can be drawn from similarly particular or technically minute features of typographic form in games.

Another matter for consideration is the ontological status of the text within the game world – both text that appears as part of a HUD as well as that which appears inside the spatiality of the game. In early games for systems like the Atari VCS that lacked native text rendering or character sets, programmers had to create every letter or number required as a graphical bitmap. For the VCS and most console systems, these bitmaps are formed by essentially switching bits on and off in a stack of 8-bit bytes to essentially draw the character on an approximation of a section of the raster grid. In this way, text is literally an image from the perspective of the machine, and its unusual ontology is such that letterforms are legible within the assembly code itself. In other words, videogame software can be accessed in a way that views its programming code as a literal list of bytes (bytes in this case are a sequence of 8 binary values, 1 or 0). By aligning the bytes vertically, all letterforms that appear in the game are visible. By contrast, more advanced textual encoding methods store numeric values from 0 to 255 in the binary code and access a given character based on its assigned ASCII value and render the character based on the visual instructions supplied by a separate font file. The technical differences between these two formats for the retrieval of typographic information supply an interesting complication to the previous discussion about semiotics as a basis for thinking about video games. With the logic of a unit operational approach, both forms of text signify their semantic content in the same way, but their contradictory formal conditions open up possibilities for divergent interpretations of their expression as interlocking units of an overall system. This difference can also manifest in shaping the player's experience of the game.

The constraints imposed by the underlying technology provide another avenue for discussion because they come to bear strongly on the forms available to typefaces in video games. This creates an interesting link between cultural and artistic movements concerned with the effect of constraint upon creative expression. Again, the early videogames, particular home console games, are especially relevant to this discussion because their visual forms bear a surprising affinity with minimalist graphic design popular in the 1950s and 60s. They also specifically echo type forms produced by the Dutch De Stijl group in the 1930s. What is significant here is the way that constraint by necessity relates to constraint by choice in the production of aesthetics, an idea that is developed in the works of the Oulipo and their relationship to New Media and hypertext literature.[14] Experimental type designer Wim Crouwel, writing in 1970, characterized the future of type design as one of constraint and configuration. He bases this projection on his analogy of the "cell" as the fundamental unit of computing (Crouwel seems to mean both the pixel and the bit), and he argues that the new logic of type design recognizes this fundamental unit and uses it to build "nuclei" (letters) which form "units" (words or concepts) that together assemble to form communication (Crouwel 57). His use of the term "unit" as well as the configurative logic of this philosophy seems even more appropriate for the videogame context in light of the unit operations approach, so it is interesting to compare Crouwel's own type design that reflected this philosophy with videogame type. This is a relationship I explore in the second chapter of this study.

Finally, just as the problems of electronic textuality are brought to the foreground by the challenge of creating an archive, the specific aesthetic forms of videogame type and its dispersal throughout culture betray a belief about their ideal forms. In other words, there is a kind of restorative nostalgia for early videogame graphics that frequently "cleans up" the material traces of the display technology. Although the appearance of so-called "jaggy" fonts has come to

14 It is appropriate, therefore, that *The New Media Reader*, edited by Nick Montfort and Noah Wardrip-Fruin, includes a selection of Oulipean writing. The authors explain this connection in away that appeals especially to a favorable comparison videogames: "The potential that lies within such an understanding of interactive experiences is a reconfiguration of the relationship between reader, author, and text. The playful construction within constraints that the Oulipo defined as the role of the author can become an activity extended to readers, who can take part in the interpretation, configuration, and construction of texts" (Wardrip-Fruin & Montfort 148).

characterize computer generated type, the low resolution (particularly of early TV console systems) meant that the supposedly harsh, “stair-stepped” curves of letterforms were actually more fuzzy than jaggy. The fact that the hard-edged, jaggy form prevails in so-called “retrogaming” and related invocations of videogame nostalgia suggests that the material form of the original game situation is being lost as it has been replaced with the supposedly ideal, hard-edged form of the text image. In other words, hard-edged forms promote a nostalgia for an imaginary aesthetic, but in the critical context, this revisionary imagination seems to conform to the fiction of the game object as transcendental form. The challenge for the scholar, therefore, is to find ways to recreate or simulate the original material conditions of the videogames’ design. This is relevant to the study of typographic forms because programmers of early games were certainly aware of the peculiar optical effects of the CRT display, so successful type design in the context had to take that distortion into consideration, often by taking advantage of its peculiarities. For example, the fact that the first numeric forms for keeping score in early games like *Pong* contain only straight lines and right angles is partly a way of dealing with the blurring effects of the television raster display. An approach to the materiality of videogame type should include the television or monitor screen among the interferential layers of production standing between the encoded typographic object and its reception in the space of a videogame.

Videogame as Text

Returning to the question of what a videogame is, the question of narrativity in videogames has a parallel in the question of videogame textuality. Though this study focuses on text literally, the idea of the videogame itself as a text is a foundational assumption. This concept has not garnered the same controversy as the narrative question, but it is nevertheless one that Diane Carr notes has been implicitly objected to in the theme of the 2007 DiGRA conference, “Situated Play.”

Drawing on Roland Barthes, Carr notes a contrast between structural approaches to videogame criticism and textual analysis of videogames, concluding that the latter has untapped potential:

> When adapted for digital games, Barthes' work (1977) suggests that structural analysis would involve looking at the units in the game-as-system, and these units' relative (and shifting) values, organisation, placement, mobility, relationships, as well as the scope for manipulation afforded by these units. ... Textual analysis, then, goes beyond content description, it's not limited to the 'static' or the linear aspects of a game, and it does not involve seeing a game as an isolated, static object. It looks to the game-as-played, to games in culture, and to culture in games. For these reasons textual analysis offers one approach to questions of meaning. (Carr)

She has also developed this idea in the collaborative essay collection, *Computer Games: Texts, Narrative and Play*, which was itself the product of a research project titled "The Textuality of Video Games" (Carr et al. Acknowledgments). Similarly, though it lacks such a specific defense of textuality, Geoff King and Tanya Krzywinska's *Tomb Raiders & Space Invaders: Videogame Forms & Contexts* takes a definite textual approach to unpacking videogame systems of meaning. Calling attention to contexts in the title implies there is a core of textuality in relation to which there is a contextuality.[15] In one example, *Doom* is described as a communicating a specifical formal modality by way of its paratextual materials: "Cover and interface artwork immediately establishes a fantasy framework twice distanced from the material of external reality, a blend of neo-gothic and science fiction imagery" (King & Krzywinska 20). In this way, the "distance" between the external and internal reality is mediated through a specifically associational paratext that establishes the means through which players make sense of the "in-game-world."

In a more explicit treatment, Barry Atkins in his book *More than a Game* does not hesitate to recognize the computer game not only as a kind of text, but also a full-fledged form of fiction. He even eschews his initial term *computer game* in favor of *game-fiction*, insisting that a game-fiction "presents a fictional text that rewards close critical scrutiny" and that that scrutiny "is not intended to be a work of theoretical enquiry, but a work of close textual criticism" (10). The nature of this

15 In their actual use of the concept, it appears that what King and Krzywinska refer to as *context* might better be described as *paratext* (after Gérard Genette) as Steven Jones has and as I will throughout this study.

scrutiny, however, is textual to a fault, to the exclusion of visual textuality. He eschews the term *videogame,* for example, because it "overstresses sight with no reference to cognitive understanding" (Atkins 20). In contrast to this, my goal in this study is to deal directly with the kinds of meaning produced through visual textuality in videogames.

James Newman appears to be in at least implicit agreement. Although he acknowledges the objections of ludologists – "for ludologists, it makes no sense to talk of the videogame text, in part because it cannot be seen to be constituted without the activity and action of the player" (*Videogames* 95) – it is clear that Newman is comfortable with applying the term *text* to refer to videogame content. Throughout both *Videogames* and his coauthored *100 Videogames*, Newman offers qualitative analyses of specific games, and frequently invokes the term *text* in reference to the object of study.

One other sense of videogame text bears mentioning in this context: the genre of computer which is completely text-based in its content. Nick Montfort has written what is perhaps the definitive scholarly work on so-called interactive fiction, *Twisty Little Passages*, and has also contributed an excellent piece, first presented at the 2005 Word and Image conference, on the role of text and image within gaming environments. In "How Stella Got her Text Back," Montfort concludes with a call to what can only be considered a textual studies of videogames: "To understand the place of text and image within computing, we should bring humanistic concern for history, social and cultural contexts, the computational equivalent of intertextuality, and all the rest - and we should bring an appropriate technical understanding of the systems we are considering, and a grasp of the basic nature of computing" (Montfort, "How Stella Got Her Text Back"). Clearly, text adventure games are worthy of visual analysis in terms of their presentation and arrangement of text, but this statement from Montfort highlights how the study of digital artifacts, including videogames and text adventures, benefits from a thorough understanding of the material condition of production and reception which constitute their textual identity. For example, Dennis Jerz has created one such comprehensive textual criticism of the seminal "Colossal Cave Adventure," the goal of which analysis was to perform a close reading of the game's sources – both its recently recovered source code and the actual cave in Kentucky which inspired the game (Jerz). Significantly, Jerz's goal in this article is not to recover the original text, but to inform our understanding of its origins and evolution. As such, this work presents a fascinating example of a criticism focusing on the social text of a digital artifact.

With a more explicit acknowledgment of the field and techniques of textual studies, Steven Jones orients his approach in *The Meaning of Videogames* on the question of the social text of videogames. In this work, Jones defines paratext in a way that will be useful in chapter 2 of this study. Specifically, the concept of paratext is a useful way of organizing my approach to videogame typography because it establishes a framework through which to build an understanding of interiority and exteriority for the videogame text.

Borrowing terminology from botanical nomenclature, I define typographic signifiers in these internally and externally textual situations as *holotype* and *paratype*, respectively. Chapter 2 begins this analysis with a focus on paratype, which is followed in chapter 3 with a discussion of holotype. I chose this organization because the textuality of typography works in both ways – establishing the signification obtained by text in videogames and signifying the textual conditions of videogames in other situations. In other words, the textually distinct meaning production inherent in videogames exists in a dialectical relationship with other situations of (possibly) related textuality, and the semantic content carried through typographic expression exert meaning across and through the permeable barriers of material technology separating videogame textuality from other media. Chapter 4 continues this analysis by focus on a particular dimension of expressivity – the competing or complementary aesthetics of "jagginess" and "fuzziness" which imply different ontologies for the videogame artifact. Chapter 5 concludes by offering a close typographic reading of the various versions of the game *Berzerk*.

In addition, two appendices provide additional information relevant to this study of videogame typography. Appendix A outlines a piece of software, ROMscrape, which I created in order to assist my qualitative analysis of letter design in Atari VCS games. As a flexible and Web-deployed tool, it is hoped that this software and the core search algorithm it uses can be useful for other researchers in the digital humanities. Appendix B catalogs various character sets and typefaces discussed throughout.

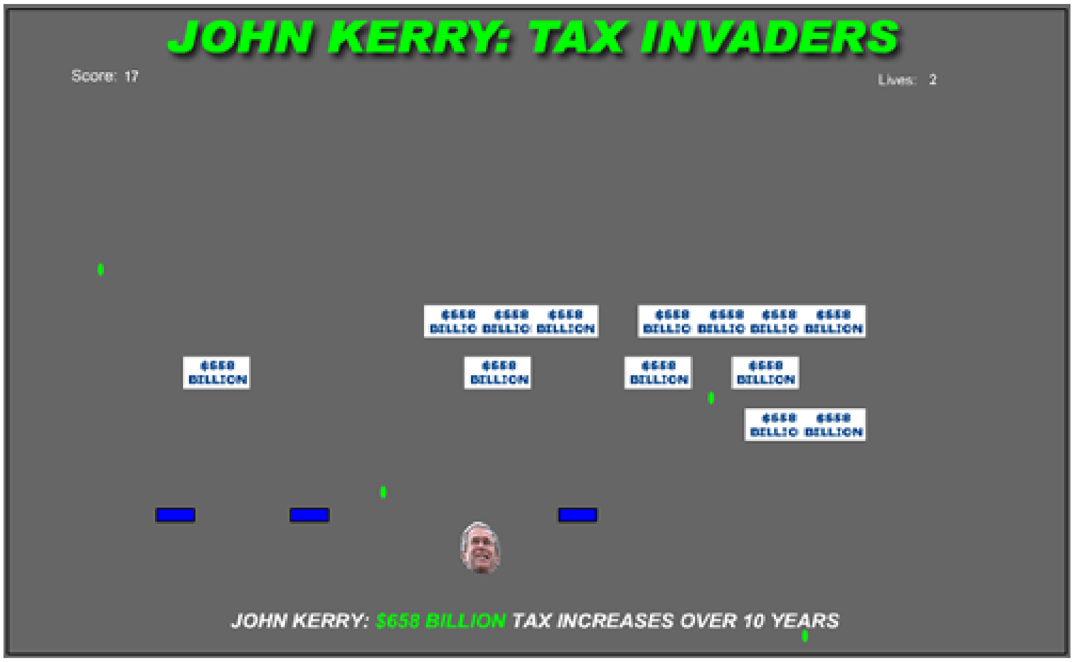


Figure 1-1. Screenshot of "John Kerry: Tax Invaders!" The game is programmed in Flash and builds on a *Space Invaders* template. Available at <http://web.archive.org/web/20040517093809/http://www.gop.com/taxinvaders/>.

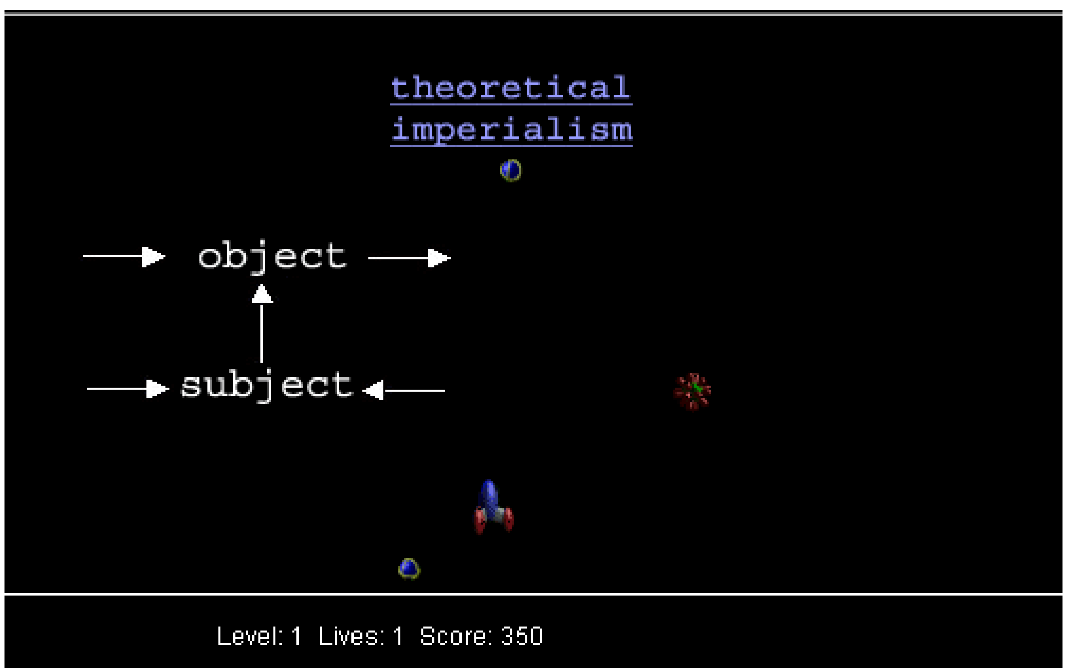


Figure 1-2. Screenshot of Jesper Juul's "Game Liberation." Available at <http://www.jesperjuul.net/gameliberation/>.

Figure 1-3. Screenshot of *Call of Duty*. The heads-up display reveals information about the current state of the game, including the player's location relative to other players and the fact that he just "fragged" the player Blue Dragon.

CHAPTER 2
PARATYPES AND PREDECESSORS

> "Great care must be used in speaking of types, as definitions are very precise." (Wikipedia contributors & 83.117.23.63)

Introduction

The epigraph for this chapter appears in the Wikipedia article for the biological nomenclature term, "holotype." The text itself offers a rather technical and thorough definition of the concept of type as it applies to biological taxonomy, but it concludes with this strangely aphoristic and ominous phrase, contributed by an anonymous user at the IP address 83.117.23.63. Although this user is speaking of type of a different sort (that is, a "nomenclatural type"), he or she supplies a succinct, if ironic, summation of the chief concern of this chapter and the next, the classification and description of typography as it relates to video games. Chapter 1 dealt with the issues at stake in a typographic approach based in textual studies and laid the conceptual groundwork for proceeding to study the textuality of videogames. In order to advance a general typographic approach that encompasses textuality, this chapter and the one that follows seek to describe and analyze typographic forms and draw conclusions about their discursive functions within videogame expression and in the broader field of cultural signification surrounding videogames. I begin by considering some issues relevant to type classification and propose two broad categories or dispositions, adapted from botanical nomenclature, which describe competing typographic categories or gestures: *paratype* and *holotype*. I explore paratypical videogame typography by examining the historical origins of some prominent typefaces commonly used in paratextual relation to videogame texts. Those origins, in turn, illuminate what it is at stake in unpacking the discursive plane of videogame typography, namely, the medially situated, technologically constrained mode of expression unique to videogame textuality.

Since videogames are a pervasive part of culture and any expansive conception of videogame textuality includes a continuum of artifacts that will be indistinguishable from other textual domains, typography that is external to videogames yet still warrants an association with gaming culture provides a valuable starting point for approaching the unique and definitive aspects of videogame texts. If one begins with the truism that videogame typography is that subset of typographic form and function which in some way relates to videogames, then the range of phenomena earning the label *paratype* is that segment of videogame typography which is at the edges (phenomenal, discursive, or logical) of that overarching set. By contrast, *holotype* describes those instances of typography which occur at the center of the set (typically, as a part of or immanently contiguous with actual videogameplay) and which define the paratypical thresholds by comparison. In this way, holotype necessarily includes any instance of alphabetic writing or numeric characters within videogames, and paratype includes alphanumeric characters used in gaming paraphernalia or other contexts that have something to do with gaming. Examples of videogame holotype include the numbers used to indicate a player's score, the start screen, menus, heads-up display (HUD) interface, and objects in the game's virtual world which contain text (e.g., a billboard or a magazine cover). Examples of videogame paratype include that the typography of arcade cabinets, instruction manuals, t-shirts, or the sign in front of the arcade. Whereas any instance of videogame holotype is always an expression of a specific game, paratype need not be, and as the examples in this chapter demonstrate, a paratypical relationship with videogames may be entirely anachronistic or a-contextual.

The boundaries of both holotype and paratype are necessarily imprecise, since different observers may disagree whether a given in-game specimen is indeed alphabetic or whether an external example legitimately recalls videogames. However, this imprecision does not mount an impassable hurdle. Instead it underscores the significance of thresholds and boundaries for the unique textuality of the videogame medium. Furthermore, the mathematical concept of fuzzy set theory provides a logical basis for proceeding with the sort of classification at hand, and because it allows multiple planes of simultaneous similarity, a fuzzy set approach to classifying videogame typography, both holotype and paratype, justifies a clear focus on the videogames produced in the so-called "Golden Age" of the late 1970s and early 80s. Put simply, this is the period in which both formal and empirical definitions of videogame typography are at their most reliable. I will return to the notion of "fuzzy" as an aesthetic mode and theoretical paradigm in a later chapter, but the difference between a typological and a taxonomical approach to classification is worth exploring as a basis in which to discuss the significance of videogame paratype.

In a thorough discussion of classification theories for social sciences, Kenneth Bailey makes an important distinction between typology and taxonomy as similar but differently-oriented synonyms for "classification." Whereas a typology depends on "concepts" and proceeds by introducing "criteria types," (K. D. Bailey 5) a taxonomy is concerned with "empirical entities" of specific physical qualities (6). The difference is important because each suggests a different approach to classifying videogame type and separating it from the larger set of all typography. What is interesting about this classification is that by using two separate criteria, empirical context and ideal form, draws our attention toward the same historical period. In other words, whether one defines videogame type as that which most faithfully complies with an established set of principles (for example, employing a low-resolution grid or bitmap for its composition) or one defines videogame type as that which appears most frequently in or around games, both organizing principles are at their most distinct when applied to the Golden Age.

Tentatively, therefore, I offer the following two criteria for determining the boundaries of the set of typographic phenomena logically relevant to the subject of videogame typography. Following Bailey's distinction, the first is typological, the second taxonomical.

> 1) Videogame typography is the semantic use of alphanumeric characters, the **form** of which exhibit any of the following properties in sufficient measure: geometric precision, rectilinear constraint, and raster granularity.
>
> 2) Videogame typography is the semantic use of alphanumeric characters in the **context** of an actual videogame or in paraphernalia or other material that directly augments or supports a videogame.

Each of these definitions contain a number of unqualified assumptions, which the remainder of this chapter attempts to address, but they provide some traction in establishing what is at stake in arriving at the various nuances and in the present discussion. As the first definition is not historically bounded, many of the paratypical samples following this formal definition occur in sources well before videogames and in contexts that appear to be entirely unrelated, but the value of aligning their aesthetic features with similar features in videogame holotypes underscores the significance of videogame textuality and provides a basis for its cultural importance. The risk associated with definitions, and especially essays that begin with definitions, is that sometimes the only way forward a definition offers is to fulfill its own prophecies. These two criteria are, therefore, meant only as signposts with which to describe the diverse coalescences in the examples that organize the discussion which follows.

It is worth noting also that the relationship of form to function is an important problem in other systems of classifying typography. In fact, Lewis Blackwell uses technologically derived fonts to undermine all typographic classification. Noting that faces like Barry Deck's Template Gothic, Zuzana Licko's Lo-Res fonts for Emigre, Wim Crouwel's New Alphabet, and OCR-A have forms heavily influenced by their technology, Blackwell concludes,

> …the faces clearly draw on technological contexts, and are in histories. But it is apparent that to map the location of a typeface, to pin it down in history and in formal properties, requires a system of multiple criteria, plotting the nature of a design on more than one axis. … There is no rule book, only a series of possible readings to be made of each new font and from which its coordinates may be plotted. (Blackwell 183)

Other authors have been more optimistic, and a number of typographic classification systems do exist. The most common uses of these organizational schemas are within the pages of font catalogs or type specimen books. For example, *The Concise Guide to Type Classification* includes the following taxonomic chapter titles: “Old Face,” “Transitional,” “Modern Face,” “Slab Serif,” “Sans Serif,” “Decorative and Display,” “Script and Brush,” “Black Letter” and “Broken” (Apicella, Pomeranz, & Wiatt 2). And *Homage to the Alphabet* employs the following categories: “even weighted sans serifs, thick and thin sans serifs, bracketed serifs, ruled serifs, spur serifs, soft serifs, square serifs, extra light, ultra bold, condensed, italics, scripts, inline/outline/contour/shaded, ornate, rustic” (Phil’s Photo, inc C1 - C2). Though many of these categories describe differences and similarities that experienced typographers will recognize and find useful, it is clear even among these two examples that different evaluations of sameness and difference can result in very different groupings. The *Concise Guide*, for example, mixes form and function, logically producing broad overlaps between categories like Sans Serif and Display (many of which lack serifs). The *Homage* is more precise, but many typefaces may exemplify more than one category. Benjamin Bauermeister has proposed a more rigorous, scientific method for typographic typology which relies on 7- or 10-digit values for a typeface where each digit represents a different criterion. This method, the PANOSE system, has its limits, however, and Bauermeister acknowledges that there are a number of criteria combinations which are possible within PANOSE but which would be impractical or impossible in any one typeface (Bauermeister 4). As these methods demonstrate, strict typological classification of typography is made impossible by the lack of non-exclusive qualifiers. In this way, a typeface may only exhibit qualities to certain degrees; therefore, any successful classification system must be based on a core of fuzzy logic.

In its most well-known formulation, fuzzy logic is defined by L.A. Zadeh as follows: "A fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership (characteristic) function which assigns to each object a grade of membership ranging between zero and one" (29). The benefit of fuzzy logic for typographic classification is not only that it allows for degrees of membership to a set, but that a set's well-formedness or (in this case) its ability to provide insight into textual analysis is not a function of the sharpness of the set's boundaries. Rather, it depends on the unique qualities of the entities with high membership relative to entities with zero membership. This is important because, when considering paratypical specimens, those typographic qualities most likely to garner an association with videogames are most similar to historical holotypes from the early 1980s. The fact that these associations need not indicate a specific historical contact with videogames suggests how important the formal classification is. Since that form depends on specific material constraints of videogame technology, the associations these constraints accrete form the context of a textual reading. Specifically, other modes of constraint which influence similar forms bring with them aesthetic, philosophical, or cultural narratives of their own which then become part of the overall picture of videogame textuality.

Paratypical Videogame Typography

In both botanical and zoological systems of nomenclature, the terms holotype and paratype have important meanings related to the designation of names for biological taxa. The *International Code of Botanical Nomenclature* defines "holotype" in its article 9, paragraph 1 as follows: "A holotype of a name of a species or infraspecific taxon is the one specimen or illustration … used by the author, or designated by the author as the nomenclatural type" (International Botanical Congress 12). By contrast, a paratype is "a specimen cited in the protologue [all elements given in the assignment of a name to a taxon] that is neither the holotype nor an isotype [a duplicate of the holotype], nor one of the syntypes [types representing a species which lacks a holotype] if two or more specimens were simultaneously designated as types" (International Botanical Congress 13).

The significance of the prefix *hol-* will be treated more thoroughly in chapters 3 and 6, but *para-* deserves consideration at this juncture for its significance to a number of relevant textual theories. Perhaps the most obvious, *paratext*, is commonly used in reference to videogames to identify supporting materials. Laurie N. Taylor, for example, uses the term in reference to videogame concept art: "Concept art as a metastructure is pivotal to understanding video games because it shows how the culture of video gaming interprets and defines the medium and it shows the significance of seemingly minor paratextual elements to the actual video game text" (227). The key principle for Taylor's argument is that a paratextual element like concept art prefigures any interpretation of in-game art, so in this sense, a textual theory of game content must include concept art as a crucial apparatus of its reception and interpretation. Steven Jones offers a similar application, emphasizing the sense in which paratexts contain an opposition of surfaces – an insideness which addresses an outsideness:

> This helps to account for what every serious gamer knows: the fact that the full potential of video games is most fully realized by the kind of dedicated, meaning-making, community-based players who call themselves fans. And this potential always extends outward from the game itself into the real social world, the "media ecology," where technologies or expressions combine with corporate interests and audience demands, and the constructed "universe" of a game, including its paratextual materials (packaging, game guides, collectible objects, online stats), narrative elements, story and back-story, and imagined game world. (Jones 10)

The original term and concept are from Gérard Genette's *Paratexts: Thresholds of Interpretation*, where he defines a book's paratext as "what enables a text to become a book and to be offered as such to its readers and, more generally, to the public" (Genette 1). Accordingly, a paratext is that which expresses the mediality of a book. Genette continues, "More than a boundary or a sealed border, the paratext is, rather, a threshold, or – a word Borges used apropos of a preface – a 'vestibule' that offers the world at large the possibility of either stepping inside or turning back"

(2). It is important to note in these three concepts, Taylor's, Jones's and Genette's original proposal, the different ways in which the units at the boundaries of the paratextual domain are deployed. Genette's definition emphasizes access, situating paratext as a way into a text. Taylor's argument also relies on this reading-in, but in Jones's usage, the textual content of the game world radiates outward through the intermedial layers and channels in which it exists.

The etymology of the prefix adds a further perspective. The Oxford English Dictionary provides two different histories and, thus, two different meanings for the construction *para-*. The most common definition is "Forming miscellaneous terms in the sense 'analogous or parallel to, but separate from or going beyond, what is denoted by the root word,'" and it derives from the Greek παρα, "by the side of, beside" ("para-, prefix1"). An alternative descends from the imperative of the classical Latin root, parare, "to prepare". This definition signifies, "Forming words with the sense 'protection from _____'" ("para-, prefix2"). This is used in such words as *parasol* (protection from the sun) and *parapet* (derived from Italian where *petto* refers to the breast; thus, "protect the breast"). It is interesting that, though Genette's discussion suggests that he intends the first definition, both senses are in play when applied to actual instances of mediality. It is not difficult to imagine a threshold or vestibule that protects an interior against intrusion from the outside world.

Paratype in the botanical sense is most clearly relying on the first definition, those types which are alongside but not identical to the holotype, but I mention the broader, dual sense of *para-* as a property of the paratext concept in order to invoke that sense of boundary or threshold involved in videogame paratype. If we take a strict definition of the videogame text, all videogame typography is ultimately paratextual, but I do not take this position. While some videogame type is indeed phenomenological exterior to specific games, I distinguish paratype as a term within the paratextual field that contains the full liminal energy of the overall concept but specifically points inward. Taken as paratext, some of the forms and examples which follow show the distribution and continuity of videogame culture as an aesthetic pattern within broader cultural fields. Since the influence of that domain is limited to actual, historical reference to videogames, I offer the examples that follow as cases of paratype because they direct a critical assessment inward toward the game text itself, but also past the holotypical text to the inner, mechanical workings of the technological apparatuses of gaming.

MICResque

When one examines artifacts of videogame culture from the 1970s and early 80s, a few distinct typographic patterns emerge. One of the most unique and striking patterns appears in the images shown in figures 2-1, 2-2 and 2-3. The distinctively blocky forms of the letters in these and many other game-related logotypes helped give videogames a distinct visual culture. The mostly rectilinear forms of the letters and the inclusions of extraneous rectangular slabs[1] create a sense of technological intrusion, but the ancestor of these designs – a font known as E-13B, developed for use in Magnetic Ink Character Recognition (MICR) – makes even more explicit the connections between human and machines as textual agents.

Two key developments in automating computer input led to the creation of fonts which continue to exert an influence over videogame typography. The core technologies of Optical Character Recognition (OCR) and Magnetic Ink Character Recognition (MICR) began taking shape in the 1930s, but the typographic identities unique to each technology emerged in the 1950s as they each saw broader application in information management (Schantz 7). Although the fonts used for OCR and MICR were designed to solve specific technological problems, they also found expanded uses within and in reference to videogames, as well as science fiction more generally,

1 One commenter on the forums at Typophile.com refers to these unnecessary forms as “digital tumors” (aluminum).

through the 1970s.[2] Just as alphanumeric character designs employed in videogames can later reference their origins in videogames by exaggerating the constraints of their digital origins, so too did fonts designed for OCR and MICR develop an association with technology that depended on repurposing or exaggerating the awkward intrusions of technological necessity for aesthetic devices. In this way, although it would be years before a videogame could render a font as complex as those based on MICR, MICR-based fonts began appearing on videogame consoles and arcade cabinets almost as soon as they became viable commercial products. Studying these fonts and the route by way of which technology for machine-reading came to develop an association with videogameplay reveals the technological discourse embedded within videogame images. Furthermore, contemporary fonts designed in the late-90s and early 00s which resemble MICR fonts bear descriptive tags such as “retro,” or “futuristic,” thus underscoring the discursive referentiality of the type design. One popular example of this, Time Machine™ by Lloyd Springer, closely resembles Moore Computer – a typeface designed in the late 60s in order to mimic E-13B – and is tagged with keywords like “data,” “ocr,” “retro,” “coding,” “futuristic,” and “avant garde.” In this way, Time Machine, employs technique of pastiche to exaggerate explicitly the associations already implicit in its forebear.

2 I am grateful to Terry Harpold for alerting me to one such font, Moore Cmputer, in use in the opening titles of *Colossus: The Forbin Project.*

Optical Character Recognition (OCR)

Early OCR technology such as David Shepard's "GISMO," a "Robotic Reader-Writer" built in an attic and unveiled to the public in 1951, focused on tasks like reading for the blind and text duplication (Schantz 8). It was not until *Reader's Digest* purchased and implemented a large-scale OCR machine for managing its database of subscribers that OCR realized its potential for streamlining data entry (Schantz 9). In this way, utility and efficiency became the driving forces of OCR innovation as numerous corporate, government, and financial institutions purchased or developed recognition technology for managing large amounts of information.

In order for any of these tools to work efficiently, a reliable input pattern must be achieved. The noted OCR developer and prolific inventor Jacob Rabinow writes of the importance of this input in developing his pattern-matching technique after working with Vannevar Bush on his Rapid Selector:[3]

> This was a 35 mm film processor where data, recorded on film, had to be identified by a dot code accompanying each record. The dot code had to be recognized 'on-the-fly' at about 300 frames per second … In working on the recognition of the dot pattern, it occurred to me that recognizing a pattern of dots is basically no different from recognizing a character … (Rabinow, qtd. in Schantz 10)

Thus, the typographic challenged facing OCR developers was to develop a font as reliable and uniform as a "pattern of dots" that yet remained legible to human readers. Emphasizing the benefit of strict control for minimizing costs, Rabinow later wrote, "Now, how can we get this control? The answer is 'Standardize!' Standardize the type of paper, standardize the size of paper, standardize the quality of printing, standardize the quality of printing, standardize the format, and standardize the font" (Rabinow 40). This drive for standardization had culminated three years prior to Rabinow's writing in a standard document issued in 1966 by the United States of America

3 The 'Rapid Selector' was a device based on microfilm which retrieved and displayed documents when requested by a user. This was effectively a forerunner to Bush's hypothetical "Memex," a device outlined in his article "As We May Think" which many see as anticipating the later development of hypertext technology.

Standards Institute (USASI).[4] This document, X3.17-1966, presented a recommendation for a standard set of alphanumeric character shapes for OCR, including 10 numerals, 26 letters (capital), 17 symbols, and 4 abstract symbols ("USASI X3.17-1966" 9 - 10).

Like the characters designed for videogames, these OCR forms depend on satisfying technological constraints. Therefore, they exhibit a high degree of stylization (see figures 2-4 and 2-5). Specifically, the angular shape of each character is based on a requirement to make each form as unique as possible so that the pattern of positive and negative space creates a distinct pattern as the OCR machine head scans across the page. But as Rabinow's call for standardization illustrates, the sense in which OCR text is constrained extends beyond the actual form of the characters and the technical apparatus includes every conceivable material aspect of the document. This move demonstrates how an aesthetic form emanating from such a thorough constraint can come to signify the entire material situation of that constraint, which is the same process of transferral occurring when videogame typography takes on a unique aesthetic identity. Videogames enforce their own standards (most notably the binary storage method of any given platform), so when any type design adopts the aesthetic properties of either OCR-A or videogame text, that design recalls the technical apparatus that constrained the original. However, the independent value of OCR-A was not immediately apparent, and the fact that some resisted it on aesthetic grounds is important for contextualizing its later adoption in videogames and elsewhere as a signifier for technological precision.

Shortly after the design and publication of OCR-A, the European Computer Manufacturers Association (ECMA) sponsored an alternative character set, eventually released by the ISO as "ISO-B" or OCR-B (Frutiger, "OCR-B" 137). Though its significance is less obvious for videogame typography, OCR-B is important because of an interesting and revealing conversation

4 Prior to 1966, USASI was known as American Standard Association. In 1969, the group changed its name to American National Standards Institute (ANSI), which it remains today.

which emerged around the time of its creation and dissemination. In an article describing the logic behind the character set, Adrian Frutiger, an influential Swiss typographer, implies that OCR-A is "offensive to human taste" and proposes that OCR-B can provide a "decipherable" alternative (Frutiger, "OCR-B" 137, 142). OCR-B's forms are indeed less stylized and are less obtrusive than OCR-A's (see figure 2-6), but as OCR technology rapidly improved and the need for either OCR-A or OCR-B decreased, OCR-A maintained the most lasting aesthetic influence. Kathleen Spangler, writing in 1971, notes that "the OCR faces are appearing on posters, advertisements, and other pieces of display type where their unconventional forms are not purely functional … It is possible that the OCR types, with their non-decorative characters, will become associated with accuracy of fact, and therefore, set a new standard in advertising type" (46). Rabinow seems to anticipate this possibility even as he dismisses the need for OCR-B on both technical and aesthetic grounds. From the examples he chooses, it appears that Rabinow's article (published in 1969) is responding to Frutiger's (1967), and as Rabinow insists on the utilitarian reliability of OCR-A, he makes an interesting argument about the origin and potential influence of its aesthetics:

> Our experience with thousands of users is that the stylizing of the A font doesn't create any human problem. … I know of no case where anyone had to go to a doctor for eye treatment or a psychiatrist because of the font. I don't even know anyone who raised even mild objections. … The esthetics of characters vary with time and place in history. The serifs which we know today are based on something which happened in Roman times due, some believe, to the problems of chiseling in stone. In any case, the Roman serifs were copied in our printing. (Rabinow 42)

Rabinow elaborates an example of reproduction technology influencing aesthetics, explaining that at the turn of the 20th century, photographers using Graflex cameras captured distorted images of race cars traveling at high speed. Because these cameras expose an inverted image from the top down, the bottoms of the cars were captured first with the rest of the car progressively later, creating an impression that the car is leaning forward. When cartoonists drew pictures of racecars, they tended to mimic this optical distortion, so a forward-slanting line became a signifier of speed that ultimately influenced the design of actual vehicles (Rabinow 42).[5] A similar progression of influence and signification occurs with videogame type where rigidly constrained typographic forms in early games are retained in later games where the constraints are no longer necessary, partly as a way of retaining their geometric aestheticism and partly as a reference to the earlier technology.

It is through this avenue of referencing technological determinism that OCR-A has been used in relation to videogames. OCR-A fonts often appear in contexts where they are meant to invoke technology, often associating that technology with alienation such as the cover for the *Cyber Crime Investigator's Field Guide* (Figure 2-7). Uses of OCR-A in or in relation to videogames are not necessarily this dystopic. For example, the covers for Ian Bogost's *Persuasive Games* and McKenzie Wark's *Gamer Theory* both use OCR-A (Figures 2-9 and 2-10), as does the cover for *Halting State*, Charlie Stross's post-cyberpunk novel revolving around a Massively Multiplayer Online Role-Playing Game or MMORPG (Figure 2-11).

Significantly, OCR-A is a font of choice for much of the *Matrix* universe created by Andy and Larry Wachowski, particularly bringing the font's referential technological constraint to its logical extreme. Like the forward-slanting lines of bus windows which associate speed with an optical defect of specific technology, OCR-A in *The Matrix* vindicates the otherwise false association between OCR-A and modern computing. Not only is OCR-A the typeface of choice for the Agents when they are in the Matrix (see Figure 2-12), it also appears in the humans' interfaces with the Matrix, shown in figure 2-13 as it used in the game *Enter the Matrix.*

The 2008 television series, *Terminator: The Sarah Connor Chronicles*, also makes frequent use of OCR-A both in its advertising and within the show itself. The association it creates in these

5 Curiously, Frutiger supplies exactly this sort of cartoon image in article published in 1970. ("Letterforms in Phototypography"). The illustration reinforces an analogy Frutiger is making between fast driving and smooth reading (careful typography is to legibility what smooth paving and wide shoulders are to driving), but its timing and placement hint that Frutiger may be responding to Rabinow.

contexts (see figures 2-13 and 2-14) is similar to the one employed by *The Matrix* – a sense of contact, overlap, or contamination bridging the gap between human and machine intelligence. In both *Terminator* and *Matrix* universes, humankind is pitted against superior machines in a fight for survival. Therefore, OCR-A in these contexts reflects a sense of dystopian anxiety or technological determinism regarding the prospect of machine intelligence, specifically the relationship between the visual senses of humans and machines.

Besides this associative or expressive use, *The Sarah Connor Chronicles* also uses OCR-A in a more subtle, discursive way. Figure 2-14 is an image from the pilot episode where the main protagonists are shown on a security camera. The information displayed on the camera uses OCR-A, associating ubiquitous surveillance technology with the apocalyptic authority of SkyNet. In other words, as a declarative statement, this use of OCR-A seems to say, “SkyNet is already watching.”

One final example of OCR-A used for expressive purposes illustrates a less dystopian, though far more articulate use of the typeface, which further underscores the ability for technological artifacts within typeface design to have meanings and applications within culture. Figure 2-15 shows an advertisement which appeared in the same issue of *Datamation* as Rabinow’s column quoted above. The ad is interesting because it uses typefaces to demonstrate a progression from human-friendly reading to machine-friendly reading. Depicting the device itself as a robotic Janus head calls attention to the dual-nature of typefaces designed for mechanical character recognition technology, the sense in which these typefaces must be legible to both machines and humans. The fact that this ad shows this relationship through a typographic progress narrative is related to the temporal gestures of Springer’s Time Machine typeface and, by implication, the historically situated, holotypical typefaces and letterings used within videogames.

As a videogame paratype, OCR-A images the same relationship among technological context and expressive form that characterize videogame typography. Furthermore, OCR technology led the way for the technology of Magnetic Ink Character Recognition (MICR) and its signature typeface, E13-B, which would have a more direct influence on and application within the broad field of videogame culture. In this way, OCR-A is a typeface which exhibits the same logic of textuality as videogame expression more generally. Therefore, identifying the potentially dystopian science-fiction narratives embedded in OCR-As form suggests one cultural context in which to situate videogames and demonstrates one means by which a cultural pattern of anxiety can find expression in typographic form.

Magnetic Ink Character Recognition (MICR) and MICResque Type Design

The technology necessary for MICR developed roughly parallel with OCR and addressed a similar need: inputting large amounts of information into a computer system using characters which could be read and verified by humans. In the 1950s, the growing demand of check processing demanded that a mechanized, automated solution replace the tedious methods of hand sorting, routing, and processing all personal checks (McKenney 61). A Technical Subcommittee of the American Banker's Association convened in 1954 to address the problem, and after a series of consultations with banks, manufacturers and the Federal Reserve Bank, the committee developed a recommendation and standard for a common machine language for check processing, which was published in its final form as Document 147 of the Bank Management Commission, first published in 1959 and still in use today (McKenney 75). The committee's use of the term *language* here is significant because the standards and specifications set forth in their recommendations encompass the numeric font itself, the location of MICR information on the check face, the ink quality, the system for encoding routing, transit, and account numbers, and the equipment required to process it. James McKenney's detailed narrative of the technical subcommittee is careful to note that the use of *language* in this context is strictly metaphorical (61), but the sense in which it describes the entire system strongly resembles Ferdinand de Saussure's use of *langue* (the complete semiologic system of any language) as a field that is distinct from *parole* (the singular expression of a specific language act). This analogy between *langue* and MICR as a "common machine language" also will help explain the means by which re-appropriated MICR fonts express associations similar to those identified with OCR-A.

The tone of document 147 is far less figurative than McKenney's, and in laying the detailed process which produced MICR technology, emphasizes the great degree of effort, innovation and cooperation which went into its development (Bank Management Commission, Technical Subcommittee 12). The clear benefit of this technology lay in its commonality (that is, its near-universal, near-simultaneous adoption), but its significance for aesthetic purposes was that it depended on the uniformity of mechanized input and processing.

MICR works by a recognition process similar to OCR, except that in MICR, the ink is magnetized and it is read by a magnetic tape head rather than an optical scanner. This prevents stray marks and paper degradation from interfering with reading, both of which were important problems the Technical Subcommittee had to solve. The typeface ultimately selected by the committee, E-13B, consists of simple, geometric forms adorned with asymmetrical rectangular slabs (Figure 2-16). This design conforms to the technical requirements of the MICR input devices, and the variability of the slab location among individual letterforms ensures that even degraded type will yield a sufficiently distinct magnetic waveform in order to be properly read.

This material durability has also given rise to E-13B's lasting aesthetic influence. Like OCR-A, E-13B's forms were originally determined by technological considerations but are now retained as aesthetic signifiers of those material circumstances constituting the original design constraint. Because E-13B's characters are even less natural,[6] their visual legacy has been more pronounced

6 In his article on OCR-B, Adrian Frutiger also dismisses E-13B as "almost unbearable" (Frutiger, "OCR-B" 139).

and has had a lasting association with videogames since the early 70s. However, like OCR-A, the designs tended toward dystopian associations before being applied more generally to science fiction contexts, including videogames. Mark Owens and David Reinfurt discuss the influence of E-13B on type design and how it "quickly became a typographic signifier of the emergent human/computer interface and the intersection of money and technology … The awkward, technically derived forms of E13B came to represent the 'pure data' of information networks, pointing towards a post-industrial future that lay just on the horizon." As evidence for this argument, Owens and Reinfurt offer that one of the first artistic uses of E-13B was in a 1967 work by Ed Ruscha which depicts simply the date 1984 set in E-13B, apparently in reference to Orwell's novel. Like OCR-A in *The Matrix*, E-13B had become the lingua franca of dystopia. It is, therefore, interesting to explore several typefaces based on E-13B that were designed in the 1970s and that made frequent enough appearances in relation to videogames that their associative qualities today still suggest 1970s videogames.

Moore Computer

The minimal aesthetic properties of E-13B saw extended influence in a number of type designs created in the late sixties and early seventies, and many of these MICR-based faces saw extensive use in relation to videogames. The typesetting and printing industries were undergoing rapid and dramatic changes during this period, adapting to new technologies like photo- and CRT-based compositors, so a number of companies and design studies were going out of business or changing hands. In addition, the decorative typefaces echoing the style of E-13B were often seen as novelty products, so records about several of these typefaces and fonts are cursory may be unreliable.[7] Nevertheless, the evidence indicates that the first full alphabet based on E-13B was a

7 The best record of typefaces from this period is a "Typeface Namebase," compiled but never completed, by Tim Ryan, who passed it on to Grant Hutchinson. The work consists of a cross referenced database of type names, years of origin, and foundries. Unfortunately, the information lacks illustrations of the designs and corroborating references on their origins. Still, I am assuming that its information is reliable.

font called Moore Computer (Figure 2-18), published by the Visual Graphics Corporation (VGC), possibly as early as 1968.

The question of precedence is significant when considering the evolution of forms, since of the typefaces under discussion, Moore Computer exhibits the least subjectivity or elaboration in its adoption of E-13B's numeral style to letterforms. The earliest extant, published reference to the font by that name appears in the 1970 edition of W. Pincus Jaspert's *The Encyclopedia of Typefaces*.[8] The encyclopedia does not include an image of the font, but references it significantly in relation to E-13B. Jaspert describes E-13B as "A type face designed to meet the needs of magnetic character recognition in automatic cheque and document reading equipment. MOORE COMPUTER complements the numerals" (213, emphasis in original). This phrasing strengthens Moore Computers case for priority, and indeed the alphabetic characters of Moore Computer do complement the numeric characters of E-13B since Moore Computer's numbers are identical to E-13B's. Other early reference sources mentioning Moore Computer include a *Visual Graphics Alphabet Library* from 1976 and an undated VGC catalog from the mid-70s. These two sources provide the further information that the face was created as a VGC original, but unfortunately do not name its designer. The fact that the *Alphabet Library* includes two other faces prefixed with "Moore,"[9] suggest that he was an in-house designer, and an un-sourced forum posting states that his first name is David ('graphic6'). Whether it was indeed the first, Moore Computer was the most rugged of the designs which followed and is the most faithful to its source. It is also the first MICR-based font to appear in advertising for a videogame. Figure 2-19 shows a slightly modified Moore Computer in use as the logotype for the Magnavox Odyssey, the first-generation videogame console first released in 1972. Moore Computer also appeared on arcade games (Figure 2-20) and

8 I am grateful to Grant Hutchinson for locating this reference.

9 These are "Moore Combo," a thick, rounded design exhibiting Bauhaus influence and including several variants for each character, and "Moore Liberty," a whimsical design imprinting stars and stripes on each character.

on some models of Coleco's Telstar console (Figure 2-21), and it is currently featured as the main title font on VintageComputing.com (Figure 2-22).

Other uses of Moore Computer stress the human/machine relationship in more dystopic or paranoid ways. Figure 2-23 shows a variant of Moore Computer (the floating slabs in M, N and W have been removed) in use as the end credits for *WarGames*, the 1983 Disney film about a world-destroying computer game. In an example of a more ironic use, figure 2-24 shows Moore Computer within a cyborg Scratchy's heads-up display in an episode of *The Simpsons*. This is clearly meant to parody similar shots from *Terminator*, so it is significant that the animators for this *Simpsons* episode selected a typeface sharing a related material history with OCR-A.

In general, the forms of Moore Computer's characters closely follow E-13B, except that each alphabetic character Moore Computer includes at least one exaggerated slab, whereas the asymmetrical enlargements mainly appear on the 1, 3, 4 and 8 of E-13B. Since Moore Computer is most commonly used in titling, one possible reason for its proliferation of slabs is to ensure that any use of Moore Computer included its distinctive characteristic. In any case, the slab seems to take on an identity of its own and even appears outside the body of the character in a few letters (Figure 2-25). The slab floats within the letters' open counters, but is not connected to the rest of the shape. Also, the alphabetic characters in Moore Computer move the location of the slab as dictated by the needs of the character. For example, the W contains an open counter similar to M, so it seems logical to place the slab in a similar position relative to the rest of the character. Other changes are dictated by the needs of a full alphanumeric character set. For example, whereas E-13B's 0 (zero) is a simple rectangular shape, Moore Computer inserts a floating slab to distinguish it from O and D.

Data Seventy

"Data 70" (or "Data Seventy") is another typeface which some claim is the first full alphabet based on the MICR font E-13B (Figure 2-26). Fortunately, the origins of Data 70 are better documented, so its claims to originality can be evaluated in more detail. Owens and Reinfurt, in their article on the influence of E-13B, only discuss Data 70, but they claim that, like Jaspert does of Moore Computer, that Data 70 "expanded the look of the E13B numeral set into a full upper and lowercase alphabet" (Owens & Reinfurt 147). Since Moore Computer is an uppercase-only typeface, it is true that Data 70 is, in one sense, a further expansion than Moore Computer. However, the fact that Data 70 includes Arabic numerals bearing little to no resemblance to E-13B (see figure 2-27) suggests instead an indirect line of influence. Bob Newman designed Data 70 in 1970 for Letraset, a manufacturer of rub-on lettering and other graphics (Owens & Reinfurt 147). Its characters adopt the extraneous slabs of E-13B, but Data 70's are more rounded and more judicious and consistent in their distribution. In general, the typeface reflects a specific stylization that evokes the sense of E-13B while emblematizing the original. This can be seen most clearly when comparing the characters 1, 4, and 0 (Figure 2-28).

Data 70 is essentially a sans-serif font, which is clearly visible in its version of the Arabic numeral 1. While E-13B's 1 prominently features an abnormally large slab serif, Newman has replaced this with the uniform slab present in each Data 70 character. By placing this on the left side of the vertical stroke and removing the upper spur, Newman has created a sans-serif 1 that is distinguishable from his lowercase l, which places its slab on the right side of its stroke. The transformation of the 4 in Data 70 is more dramatic, enclosing the open counter. Like Moore, Newman must distinguish the 0 (zero) from O with the strategic placement of a slab. However, since Data 70's forms are more consistent, the slab placement is also the only distinguishing characteristic among 0,O, capital D, and capital Q, as shown in figure 2-29.

Data 70 makes frequent appearances on Arcade Fliers and game operating manuals. It was the logotype for the 1972 arcade game “Star Trek” (a copy of Nolan Bushnell’s “Computer Space,” figure 2-30), “Computer Space Ball” (also from 1972, figure 2-31), and in “BiSci Blood Type Challenge” (a contemporary educational game that is a “send-up” of 1980s arcade games, 2-32). Data 70 is also the title font for Walter Buchsbaum and Walter Mauro’s 1979 textbook, *Electronic Games: Design, Programming, and Troubleshooting* (Figure 2-33).

Orbit-B

Orbit-B, designed in 1973 by Stan Biggendon, was another VGC original like Moore Computer. It adopts a similar slab distribution pattern to Moore Computer (see figure 2-34), but the slabs themselves are slightly narrower and connect to the body of the letter with a steep diagonal rather than a right angle. Its strokes are also proportionally narrower than Moore Computer or Data 70 while its counters are noticeably wider. The numerals in Orbit-B are still quite different from those E-13B, but show more in common than Data 70. Figure 2-35 shows Orbit-B and E-13B side-by-side.[10]

Orbit-B is less common than either Moore Computer or Data 70, possibly because its MICR influence is more subtle and less arbitrarily intrusive, but it still appears frequently in and around videogames and in contexts where some intimacy is suggested between humans and computers. For example, Orbit-B appears as display text for a security scanner in the movie *Johnny Mnemonic* (Figure 2-36), yet another dystopian science fiction narrative expressing a thinly veiled anxiety about the overlap or competition between human and machine intelligence. Orbit-B also makes a significant appearance on the cover of *Galactic Empire*, a 1979 strategy game which has the distinction of being the first game released the noted game development company, Brøderbund

10 I have not been able to determine the origin of the “B” in “Orbit-B,” but it may be a reference to E-13B. I have not been able to confirm this, however, nor have I yet found an “Orbit-A.”

(Falk). Figure 2-37 shows the 1982 release of the game, with Orbit-B clearly augmenting and invoking an outer-space theme.

Westminster

The origins of Westminster (Figure 2-38) are somewhat unclear, but it emerged at least as early as 1971 as figure 2-39 demonstrates. It is also the most pervasive and common of the MICR fonts because it has been distributed freely with Windows operating systems since Windows 98. Its distribution of slabs has much in common with Orbit-B, but Westminster's proportions are much narrower. Microsoft's typography website does provide a few tantalizing details about Westminster, but does not identify the designer or exact year of origin. Printed below is the full text of Microsoft's information about Westminster, which comes included with the font software:

> In the mid-1960s after banks began printing machine-readable account numbers on checks, a British font designer made an entire typeface along the same lines. No one took this typeface seriously, however, until Photoscript produced it, naming the typeface after the bank that helped Photoscript fund the font's production. Westminster was an instant hit, and the very font makers who had previously rejected the idea rushed out to commission alternative designs. This is the first of those designs, and it's the best. Although you're welcome to use only the numbers (perhaps you run a bank), the rest of the face can provide a number of interesting uses at both large and small sizes. (Microsoft)

It is difficult to verify the claim that Westminster is the "first of those [MICR-based] designs," though its uses in 1971 place it among that original group.

According to Simon Daniels of Microsoft's Typography division, the text embedded in the Westminster font file would have been composed by the late Robert Norton, head of Microsoft Typography during the mid-1990s (Daniels). Daniels believes it is possible that Norton originally designed Westminster himself, and the little available evidence does support this possibility. The font description mentions Photoscript, a phototypesetting company Norton founded in 1970 (Macmillan 141); the unnamed designer is identified as British, as was Norton; and the choice to focus the font's description on the story of a willful designer who is ultimately vindicated seems consistent with Norton's sense of humor[11] and habit of self-deprecation (Berry).

As to Microsoft's claim that Westminster is "the best," it appears that many designers agree since the font can be found in many different uses where it is meant to convey some relationship between humans and computing. Its forms are somewhat similar to Orbit-B's in that they both employ diagonally angled slabs, but Westminster's strokes are significantly wider than Orbit-B's. Westminster's counters are also generally narrower, giving the glyphs a more compact look. The numeral characters are significant because, despite Microsoft's implication that one could run a bank using Westminster, its Arabic numerals are quite different from E-13B's (see figure 2-40). In fact, Westminster's numerals appear more similar to Data 70's than to E-13B, suggesting that one may have influenced the other.

One interesting, recent use is in the web cartoon *Homestarrunner*, created by Matt and Mike Chapman, where Westminster is depicted (in a role typical for MICR-based fonts) as the lingua franca of 1980s computing.[12] In figure 2-41, the character Strongbad checks his email (a weekly feature on the website) on his "Tandy 400," which uses Westminster as its primary display font. In a later Strongbad email segment, Westminster is used as the embodiment of "characters" from the parodic text adventure game "Thy Dungeonman" (Matt Chapman & Mike Chapman, "Web Comics"). In the e-mail feature, the premise of the joke is that a Saturday morning cartoon show has been made based on "Thy Dungeonman," but since the game lacks graphics, its only visual components are the input text (Figure 2-42).

This group of typefaces – E-13B, Moore Computer, Data 70, Orbit-B and Westminster – all share many common features, the most obvious of which are the angular, geometric glyphs and the presence of superfluous slabs adorning many of the forms. As a vestige of a technology for

11 See, for example, Norton's *Microsoft Typography: A Disagreeably Facetious Type Glossary* (1995).

12 *Homestarrunner* is a Flash-based website available at <www.homestarrunner.com>. Strongbad checks his e-mail at <www.homestarrunner.com/sbemail.html>.

machine-reading, the presence of these slabs for aesthetic reasons signifies an association with computing and machinery that can be either reinforce a positive association (in the sense of “futuristic” and “new”) or negative association (in the sense of “dystopia”). Interestingly, these slabs also serve as reliable indicators of which of these rather similar-looking fonts are in use in a given sample. Many of the glyphs in these typefaces may difficult to tell apart from the same glyph in the other typefaces unless one relies on the position and shape of the slab.

The other prominent association encouraged by these typefaces is the suggestion that they have to do with mainframe computing. In one sense, it is true that OCR and MICR technologies do rely large computing devices, there was never any case in which the stylized typefaces of either OCR or MICR were preferable or even necessary in a screen display situation. These faces were designed for print – specifically, for turning printed text into electronic data – so using them to create a sense that a computer screen interface is historically authentic performs a kind of anamorphic nostalgia in which the historical perception of the original is distorted by descendants of that original form. This association is borne out in the catalog descriptions published with fonts of these typefaces and in keywords assigned to these fonts on websites like MyFonts.com. For example, the website for Monotype Imaging, which distributes “Computer,” makes the following claims about the typeface: “Computer is an all-capitals headline font that immediately implies early mainframe computer technology. Although desktop computers and better screen and printer faces have been available for some time, the type style of the Computer font is still used for futuristic topics” (“Computer Font - Fonts.com”).

Holotypical MICResque

Although MICR-based typefaces discussed above began appearing in videogame arcade cabinets, consoles, and instruction manuals from their earliest inceptions in the 1970s, it was not until the late 70s that such lettering was even possible on videogame screens, and not until the early 80s was it at all common. Adopting the same assumption that type created for mechanical reading signifies a lingua franca for human/computer interfaces (and hybrids), most of the games which used this style of lettering did so as part of some kind of science fiction setting, with some noteworthy exceptions. On the Atari 2600 console, these games include some of the Star Wars games released by Parker Brothers as well as a few other space-oriented shooter games. The table 2-1 illustrates the Arabic numeral glyphs for a number of these, grouping together similar or identical holotypes.

Other games from the early 1980s make use of MICResque design, including *Star Raiders*, most popular in its Atari 400/800 version. As these examples demonstrate, these fonts of MICResque design were not implemented for the purpose of achieving greater clarity. If anything, these designs are less legible for human readers than simpler designs on the same technology. Instead, the fact that programmers chose these faces indicates an attempt to make some kind of associative appeal. Some games from this set adopt a science fiction setting, but that fact alone fails to prove that these MICResque designs have anything to do with science fiction. The subtler point is that these designs invoke an earlier, print-based technology that imposed mechanical constraints on the forms available for letters. Within an environment of similarly restrictive constraint, these designs reference their own sense of being constrained, not merely by conforming to the bitmap grid, but by working in spite of that grid to reference something beyond it. The converse of this relationship between constraint and expression is also true. The strictures of a bitmap grid preclude mimicking other print-based type design (serifs would be difficult to manage at this scale, for example, as would any humanist typeface), so a game programmer wishing to refer to something beyond the screen has a clear target in print-based forms which also employ grids. This, therefore, is the essence of paratypical videogame typography: associations based in mimesis which derive formal properties from the constraints of material technology.

Modernist Precursors

Though their relationship with actual videogames is purely formal, some typefaces and lettering created in the early part of the 20th century bear a surprising resemblance to forms found in videogames of the 1970s. Chief among these designs is the work of the Dutch De Stijl group, an artist group associated with the periodical, *De Stijl*, first published in 1918 (Purvis 25). Theo van Doesburg was the group's founder and sole editor of the journal, and it is his dominant personality which characterizes much of the groups ideals, expressed in a series of manifestos which lay out the group's core ideal, identifying and uniting the universal consciousness of artistic expression (Blotkamp, "Introduction" ix). In practice, this meant "searching for and the most fundamental elements of each separate field of art and then uniting these elements in a well-balanced relationship" (Blotkamp, "Introduction" ix). The group's artistic works tend to emphasize rectilinear forms and primary colors, and the typography employed in the journal itself is strikingly performative (see figure 2-43). Typography itself is an art based on the mechanical reproduction of a limited set of forms, so it seems a fitting medium for explore the aesthetic ideas of De Stijl.

The De Stijl artist most responsible for creating videogame-like letterforms was Vilmos Huszár, who created the famous logo for *De Stijl* seen in figure 2-44. The block letters for the title DE STYL, often erroneously credited to van Doesburg (Ex 92), are formed from uniformly spaced rectangles which create each letter within the space of an identically sized square. Like the bitmap compositions of videogame letters, the unity of each form depends on the successful arrangement of geometrically inflexible components. Huszár's anachronistic references to videogames continue in his *Ornament XXe eeuwse stijl* (Figure 2-45), which bears a resemblance to the representations of binary code I discuss in chapter 3. And his *Compositie II* (Figure 2-46), depicts human figures ice-skating which look uncannily similar to animation sprites such as the ones used for the Atari VCS game *Superman* (1979). Figure 2-47 shows the two side by side, with the figures rearranged for better comparison. Object 2-1 is a simple animation using the same principle as the Atari sprite animation, but taking Huszár's figures as the source image.

A typeface (or, more properly, a lettering style) designed and used frequently by van Doesburg makes another contribution of rectilinear letterforms that have a formal relationship to videogame lettering. This alphabet, shown in figure 2-48, is based on the same square form employed by Huszár as a constraint, but even though van Doesburg's design is significantly more legible, it remains useful primarily for titling and logos.

Piet Zwart, an artist associated with *De Stijl*, used similar lettering throughout the 1920s, with the apparent purpose to emphasize "technology, universality, abstraction, and functionalism" (Purvis 62), ideas which anticipate how this style of lettering is used today. In 1995, type foundry P22 released a revived and expanded version of van Doesburg's alphabet as DeStijl™, and in 1997 The Foundry released their own version as "Architype van Doesburg." Like Huszár's letters, these can be associated with videogames by reference to the forms, but a surprising contextual association also appears in *Wired Magazine*.

Figure 2-49A shows the spine text for *Wired* after its redesign of March 2007.The face is a custom design by Hoefler and Frere-Jones (Editors 26), but its capital W and capital R are close matches for van Doesburg's. Furthermore, the logotype used for each section of the magazine (Figure 2-49B includes the logos used to mark the "Start" and "Play" sections) are similarly rectilinear and echo the forms used by van Doesburg and Huszár, and also by the type designs of Dutch architect H. T. Wijdevald whose lettering for Wendingen is also echoed in the logotype for the digital culture blog *Boing Boing* 2-50.

What are we to make of these associations between *Wired*, which connects these specific typefaces to a context appropriate for videogames? It seems to be the case that *Wired*'s use of these designs does not depend on their referentiality. Rather, the hard edges and rectilinear forms of these samples is consistent with the overall look of the magazine's redesign, and is consistent with previous designs which employed the look of jaggy, pixelized shapes throughout. Perhaps one conclusion we can draw is that if the aesthetic idea Huszár sought was to reduce letterforms to the most fundamental units which could still produce a recognizable character, then maybe the alleged purity of that approach also applies to videogame type design which has the units predetermined by the raster of the bitmaps which store each shape. In other words, if Huszár's design was successful for reasons related to its form, then perhaps those same formal elements, unintentionally re-created in videogame letters accomplish the same aesthetic goals. In one possible example of this, lettering design for the 1982 Atari 2600 game *Blue Print* (CBS Electronics) contains text consistent with a typeface designed by van Doesburg in 1928 (see figure 2-51).

The association between De Stijl and videogames comes full circle with *Pac Mondrian* (punning on Pac-Man and De Stijl artist Piet Mondrian), a project created by the artist collective Prize Budget for Boys ("Pac-Mondrian"). The project consists of a series of playable games programmed in Java which allow players to carry out a game of Pac-Man in a maze based on Piet Mondrian's painting, "Broadway Boogie-Woogie" (Figure 2-52). PBFB states that, "Pac-Mondrian disciplines the syncopated rhythms of Mondrian's spatial arrangements into a regular grid, then frees the gaze to follow the viewer's whimsical perambulations of the painting: a player's thorough study of the painting clears the level" ("Pac-Mondrian").

One final bastion of Dutch graphic design having an interesting affinity with videogame typography is the 20th century designer, Wim Crouwel. In 1967, Crouwel designed an experimental alphabet specifically designed to accommodate CRT compositors and display screens. Contrasting his design with the distortions CRT introduced to a Garamond font, the so-called "New Alphabet" embraces that limitation by constructing each letterform with only right angles (Figure 2-53).

Others of Crouwel's designs feature prominent grids, but it is not necessarily the case that he intended his grids as a response or appeal to technology.

> I am always interested in clarity. It should be clear. It should be readable. It should be straightforward. ... So I started using, gradually, grids for my designs, for my catalogs for museums. I invented a grid, and within the grid I played my game. But always along the lines of the grid so that there is a certain order in it. That is why I use grids. That is why they call me 'gridnik.' For me it's a tool of creating order, and creating order is typography. (Crouwel, comments in Hustwit)

Therefore, for Crouwel, a grid is an expression of the ideal modernist impulse toward simplicity, but the sense in which his designs echo his Dutch forebears while also existing contemporaneously with OCR and MICResque designs provides an important link between the two domains of videogame paratype. The Modernist basis of Crouwel's grid-based compositions suggested that similar, necessarily grid-based designs such OCR, MICR, or bitmap typefaces, may also express Modernist principles of order and clarity. As a paratype of videogame typography, the New Alphabet invokes clarity as a condition of order and implies that typographic expression within videogames is itself "clear" in the sense that Crouwel uses the term. However, the widely varied and contingent textuality which actually exists within this allegedly idealistic grid-like framework demonstrates the extent to which the "game" of typography is more subjective than Crouwel seems to suggest. At least, it is not typically subject to the eye of a master like Crouwel.

Two typefaces by Italian designer Aldo Novarese connect the Modernist impulses of Crouwel within paratypical and holotypical applications. Microgramma, first published in 1952 as a capitals-only typeface, and Eurostile, released in 1962 with as a completion of Microgramma, borrow their basic rounded-square form from technology. In his 1964 book, *Alfa Beta*, Novarese also describes his basis for these typefaces within contemporary (1950s and 60s) architecture.

> The examples that we have grouped under the title: "square shapes" are typical expressions of the trends of this century, and have arisen to match the font to current architectural preferences.

> The squared, compact shape, indeed, is now quite familiar. It is present, even predominant, in everything that surrounds us. And fonts – today, as in the past – blend with the expression of today's world. So samples, even if distinguished, of older writing demonstrate that every style – be it Bodonian, Venetian, Egyptian, or Linear – can modify its shapes (especially with regard to curves) from rounder ones to those that are more squared and sharply angled without excessively modifying the original shape.
>
> The squared shape is the typical expression of our century, as was the round arch from which the Roman stone working style arose; or as the pointed arch which evolved into the Gothic character. (Novarese, qtd. in Blackwell 106) [13]

Novarese's illustration (Figure 2-54), however, shows a train window as one of many places from which the Microgramma square derives. This comparison anticipates Jacob Rabinow's discussion of the aesthetic influence of technological restrictions, but it performs the opposite transition. Whereas Rabinow argues that the forward-slanting design of bus windows arose from a consequence of photographic technology which associated forward-slanting lines with rapid motion, Novarese seems to draw aesthetic inspiration from a technology which may be square-shaped only by necessity.

Blackwell also notes a comparison that seems a likely explanation for Eurostile's continued success, the television screen. Figures 2-55 and 2-56 show the artwork on the box for an early gaming console, the Coleco Telstar Marksman from 1978. The console is a late PONG-clone, with the addition of two game-modes which make use of a light gun. As such, it is a very typical gaming device of the 1970s, both in terms of its technology and design. The logotype for the game combines Helvetica and Avant Garde ITC typefaces, but the description text, set in Eurostile, makes clear the affinity between the font and the adjacent television screen illustrations, and brings full-circle the discussion of paratypical videogame typography.

Eurostile appeared on many other game fliers, boxes and artwork, and it is still prevalent in contexts where it is meant to invoke technological or futuristic style. For example, the Blip box art in figure 2-3 uses Eurostile (ironically) for the text "No TV set required." Eurostile is also the

13 I am grateful to Mark Vasani and Richard Paez for their assistance in translating this passage from Italian.

typeface of choice for the bibliographic software Zotero and is probably most visible as the logotype for insurance company GEICO. If the success of Eurostile does have anything to do with its invocation of the shape of television screens, it is significant to note that this is an increasingly nostalgic inference as the use of more rectangular flat screen and high definition televisions becomes more widespread. Crouwel's New Alphabet was also designed specifically with reference to CRT technology, so its historical paratypical association becomes an increasingly dated reference to technology even though the formal (i.e. rectilinear) identity continues to drive associations with constraints of technology.

Conclusion

This chapter begins with an admonition to take great care with type. The categories and definitions proposed and explored here are indeed tentative. The term paratype offers a useful way into a conversation of influence, intention, and constraint in the field of textuality that surrounds videogames, but its ability to determine or delineate those boundaries is not reliable. Still, the common feature of some logical or empirical relationship with videogames has united practical developments in the technology of machine input with the aesthetic ideals of European modernism, and the result reveals the uniquely situated domain of videogame textuality. While it is not necessarily the case that the aesthetic viability of typefaces designed for videogames prove (owing to their resemblance, say, of van Doesburg's alphabet design) that the neo-Plasticist principles of aesthetics were correct, neither is it the case that Huszár's figure design prophesied the coming of sprite animation. It is also not the case that typefaces like Moore Computer, Westminster, and Data Seventy have any historical connection to videogames. Rather, these relationships and associations demonstrate the powerful and uniquely malleable ability for typographic form to reference its means of reproduction. The next chapter explores videogames and videogame technology as means of and context for typographic reproduction.

Figure 2-1. Magnavox Odyssey logo. (Image from Baer)

Figure 2-2. Star Trek arcade cabinet. (Image from "Star Trek (1972)")

Figure 2-3. "Blip," a table-top electronic game by Tomy. (Image from Morgan)

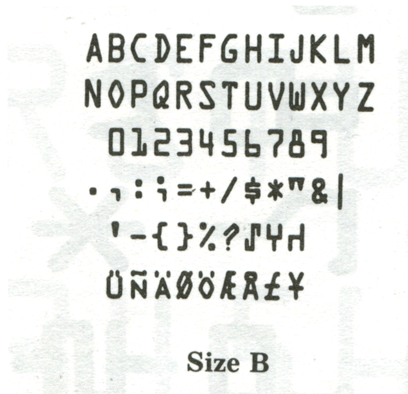


Figure 2-4. Standard OCR characters established by USASI X3.17-1966. The font derived from this standard is commonly referred to as OCR-A. This image is a scan of the character set at size "B." (Scan from "USASI X3.17-1966" 38)

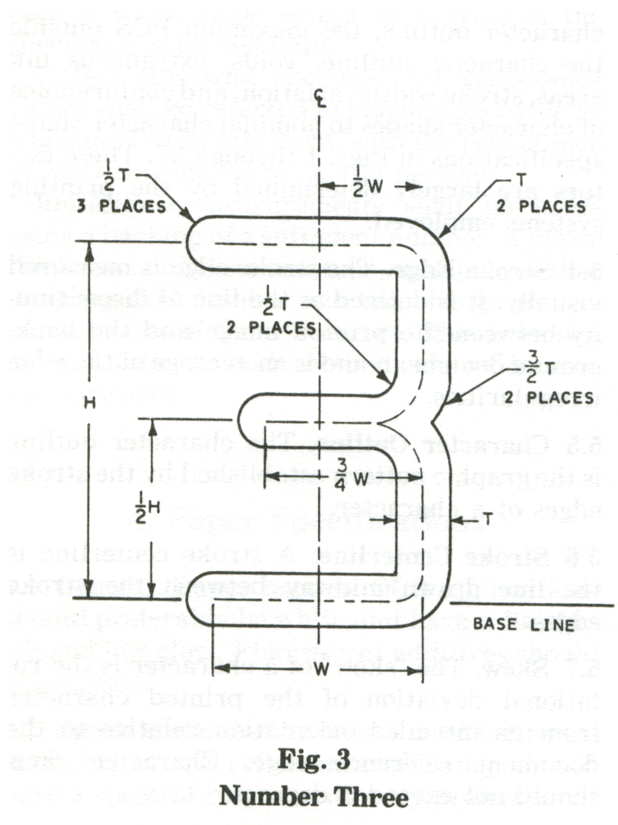


Figure 2-5. The detailed specification for the OCR numeral 3. (Scan from "USASI X3.17-1966" 12)

A	B	C	D	E	F	G	H	I	J	K	L	M	N
O	P	Q	R	S	T	U	V	W	X	Y	Z	a	b
c	d	e	f	g	h	i	j	k	l	m	n	o	p
q	r	s	t	u	v	w	x	y	z	0	1	2	3
4	5	6	7	8	9	!	@	#	$	%	^	&	*

Figure 2-6. A sample character set for a font of OCR-B, also known as ISO-B, first published in 1967 as a more aesthetically pleasing alternative to OCR-A. (Image from "OCR-B 10 BT : Style Details : MyFonts")

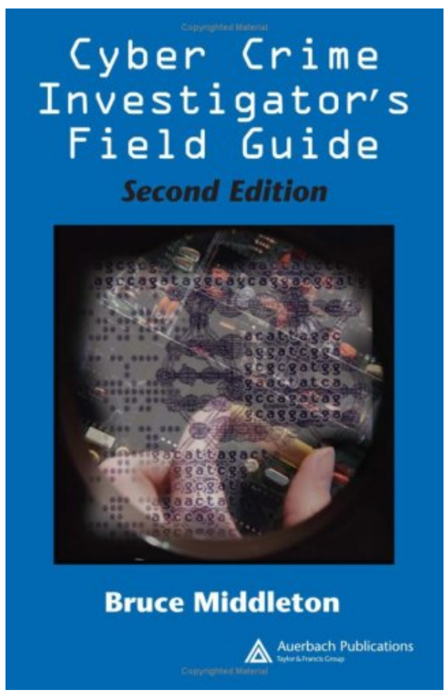


Figure 2-7. Cover of *Cyber Crime Investigator's Field Guide* illustrating OCR-A in use as reference to computer technology.

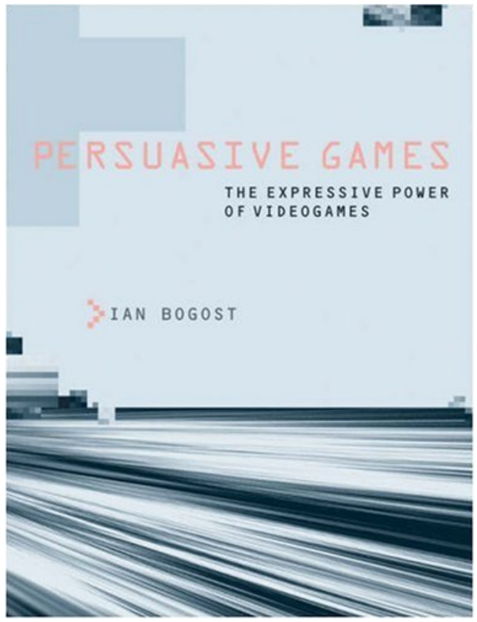


Figure 2-8. Cover of *Persuasive Games* by Ian Bogost.

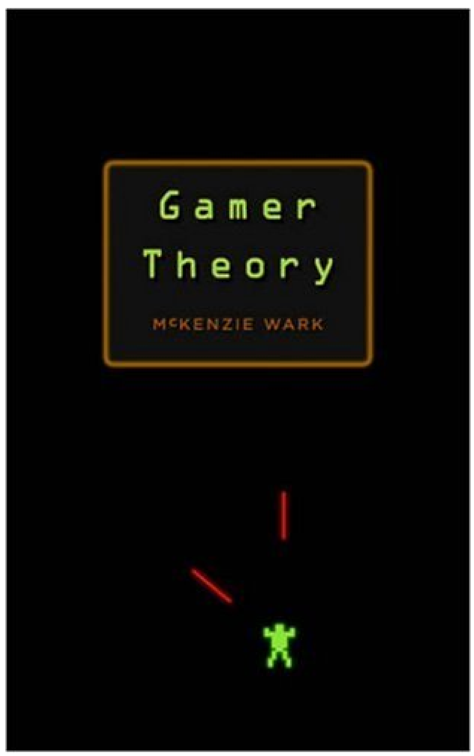


Figure 2-9. Cover of *Gamer Theory* by McKenzie Wark.

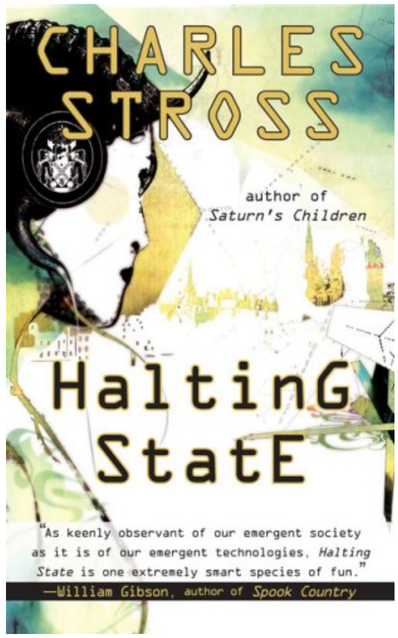


Figure 2-10. Cover of *Halting State* by Charlie Stross.

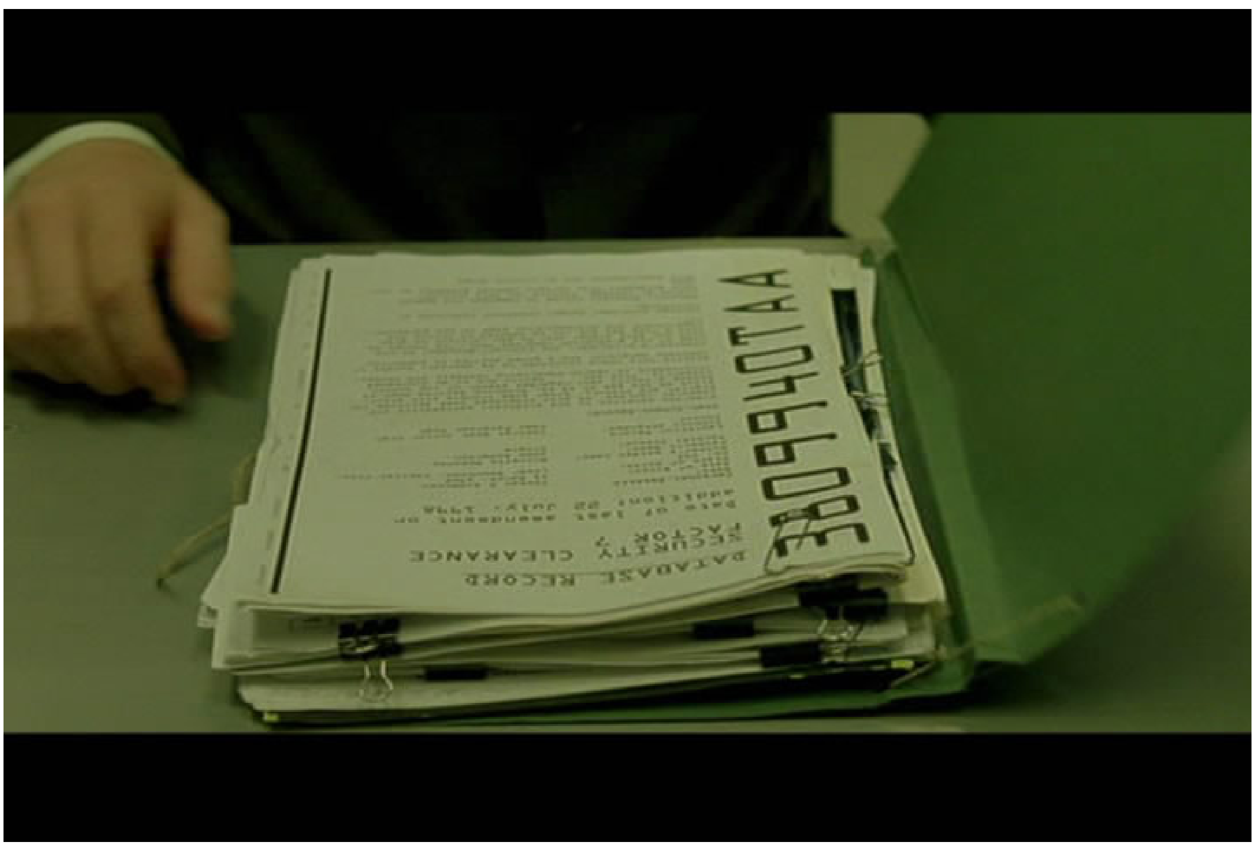


Figure 2-11. In this brief shot from *The Matrix*, Agent Smith's file on "Neo" is written in OCR-A. (L. Wachowski & A. Wachowski)

Figure 2-12. This screenshot from the videogame *Enter The Matrix* depicts the character selection screen, which is the same interface used in the films for the monitors which provide visual access to the Matrix. The text “GHOST”, “NIOBE,” and “PWER FLUX –OK-“ as well as the numerals are all rendered in OCR-A. (Shiny Entertainment)

Figure 2-13. OCR-A within a heads-up display for an android assassin. Image from the pilot episode of *Terminator: The Sarah Connor Chronicles* (Nutter).

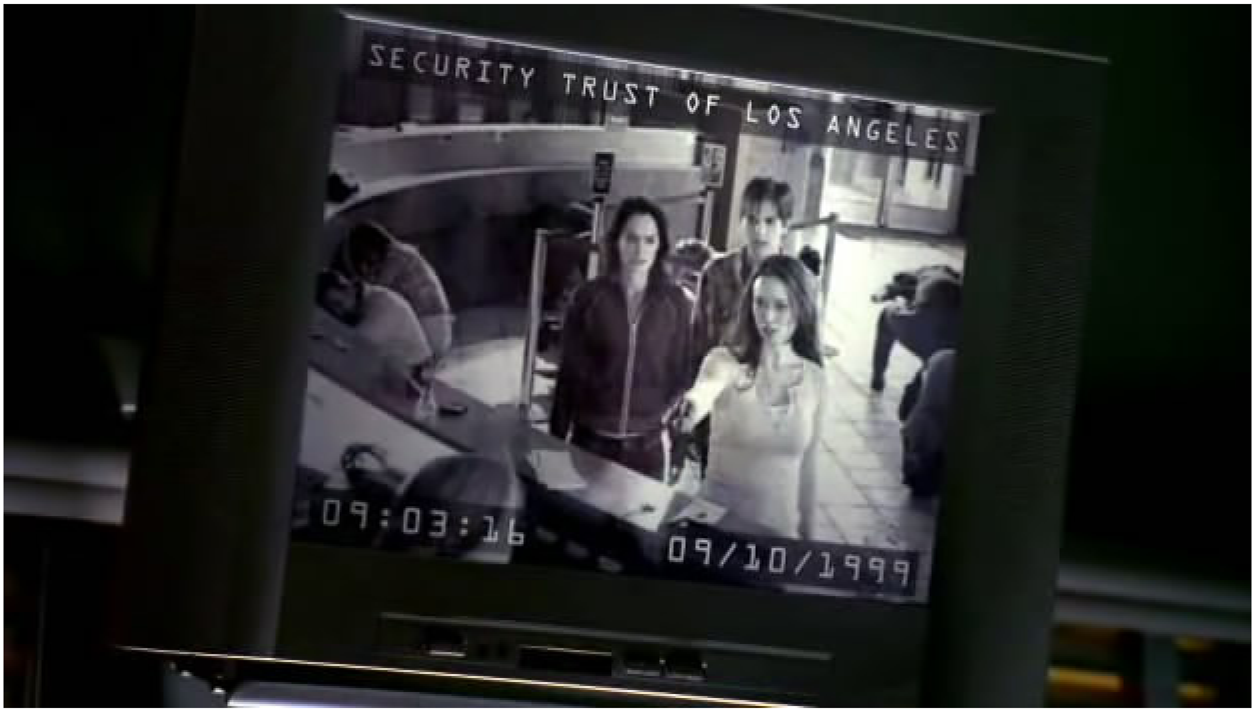


Figure 2-14. OCR-A depicted within a contemporary (late 1990s) surveillance camera. Image from the pilot episode of *Terminator: The Sarah Connor Chronicles* (Nutter).

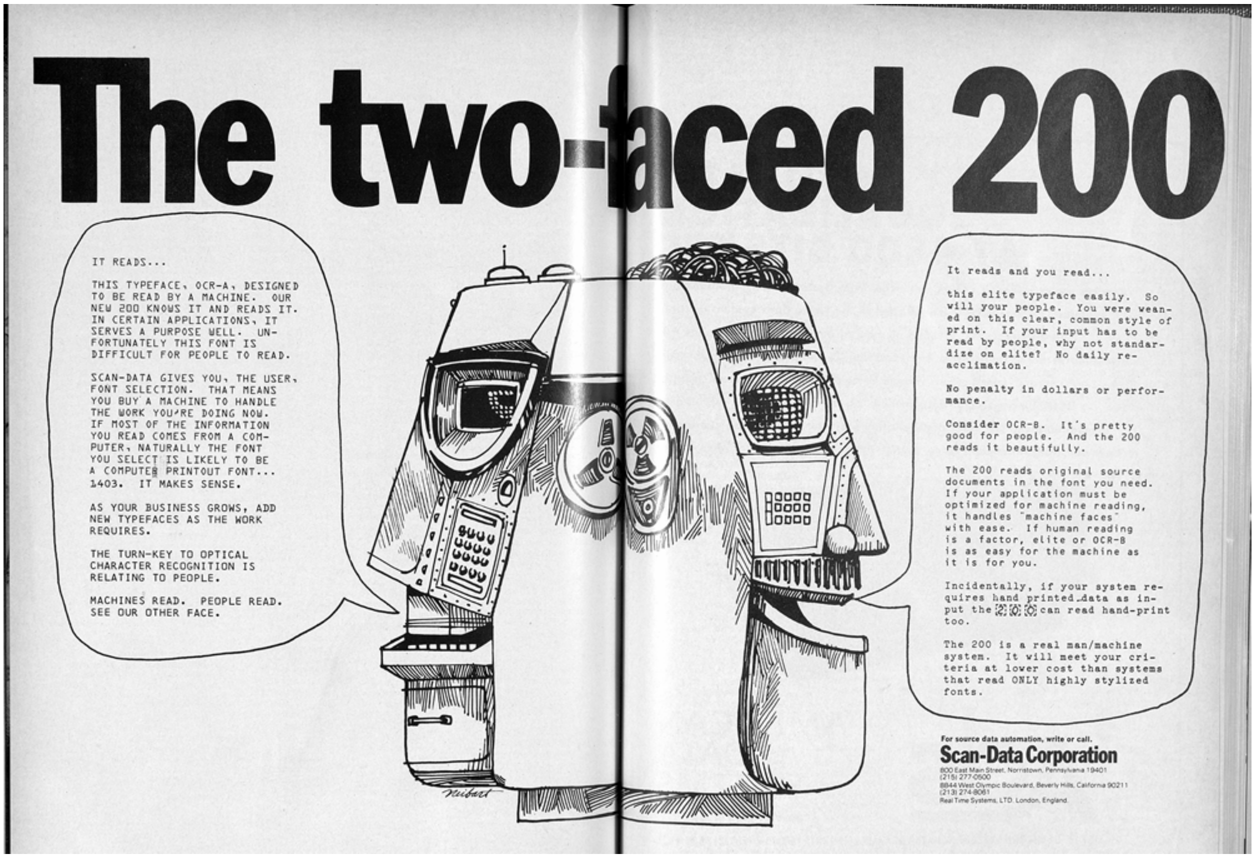


Figure 2-15. Advertisement for Scan-Data 200, an OCR reading device. (Scan-Data Corporation)

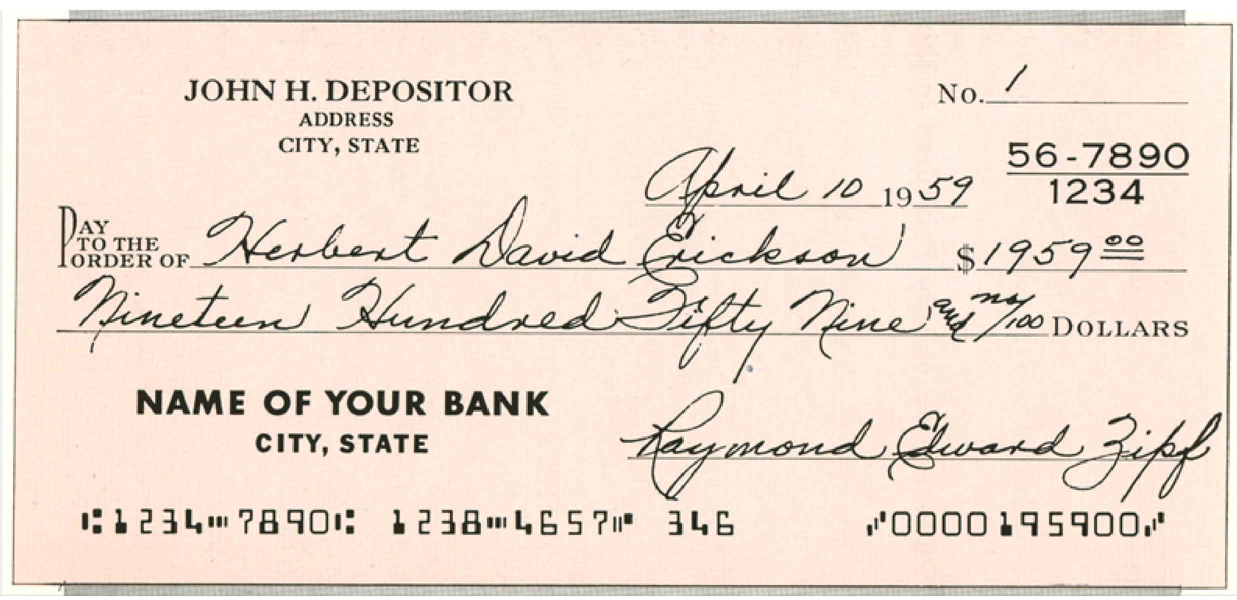


Figure 2-16. Sample E-13B characters used in a “Check of the Future” in ABA Document 147 (Bank Management Commission, Technical Subcommittee 2)

Figure 2-17. CMC-7, an alternative to E-13B still in use in some European banking systems. (Scan from Blackwell 113)

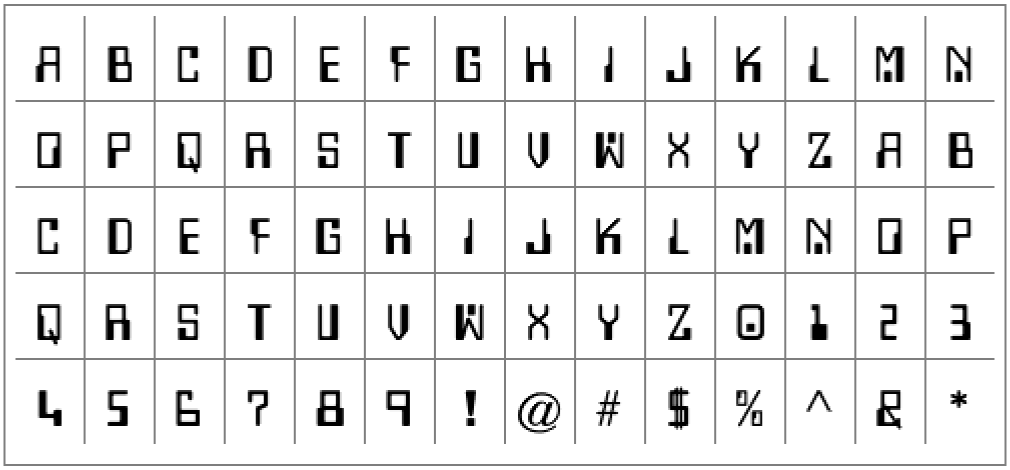


Figure 2-18. “Computer” originally sold as “Moore Computer” by VGC. Its current design is owned by Monotype Imaging.

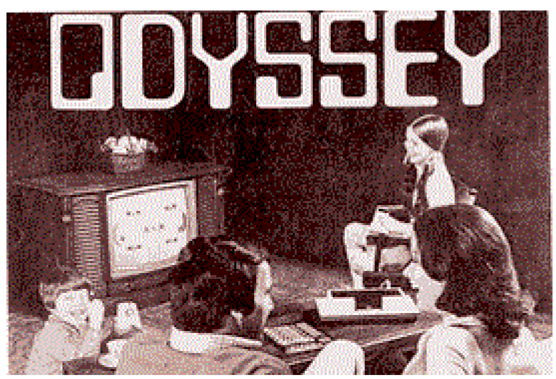


Figure 2-19. The logotype for the Magnavox Odyssey (1972) systems uses Moore Computer.

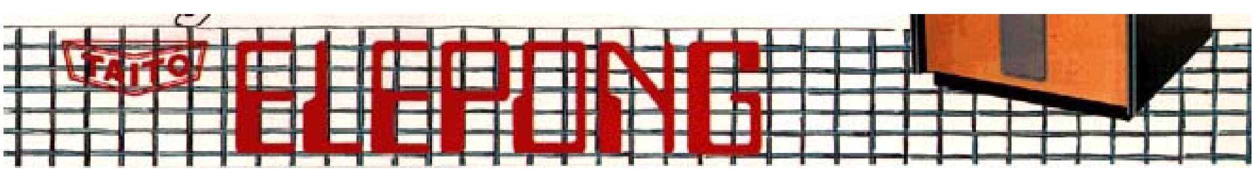


Figure 2-20. “Elepong,” featuring a variant Moore Computer (the O and N use different slabs). Detail from flyer image. (“Video Game: Elepong, Taito”)

Figure 2-21. Coleco Telstar (1976) featuring a logotype in Moore Computer.

Figure 2-22. Logotype for vintagecomputing.com, combining Moore Computer and a typeface with echoes of Theo van Doesburg. ("Vintage Computing and Gaming | Archive » Retro Scan of the Week: "So You Want to Be a Video Games Inventor"")

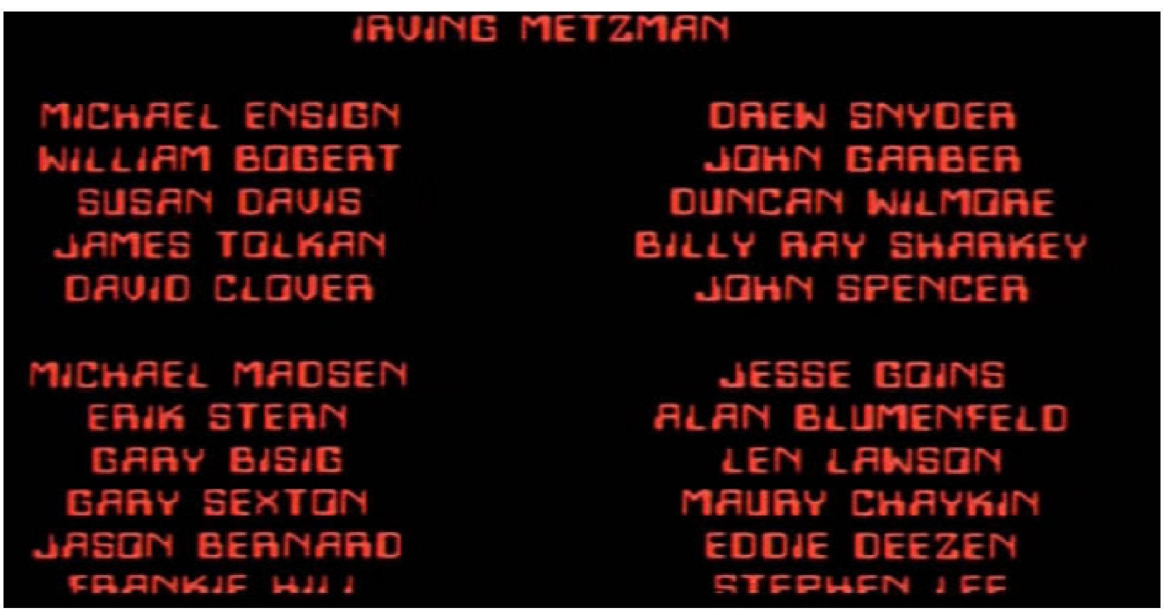


Figure 2-23. End credits for *WarGames*. (Badham)

Figure 2-24. Mechanical Itchy heads-up display. (Archer)

Figure 2-25. Moore Computer's M, N, W and 0 (zero) include floating slabs.

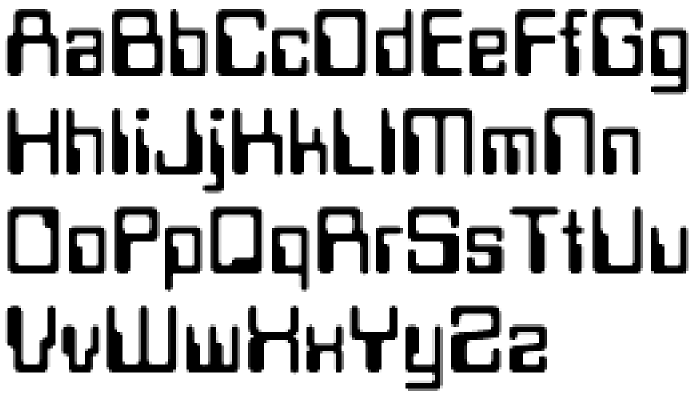


Figure 2-26. Sample characters from Data 70. Designed by Bob Newman and published in 1970 as a Letraset original.

Figure 2-27. A) The Arabic numerals from Data 70, bearing only a general resemblance to B) E-13B.

A B C

Figure 2-28. A) Comparing numeral 1 for E-13B (left) and Data 70 (right); B) comparing numeral 4; C) comparing numeral 0.

Figure 2-29. 0,O, D, and Q from Data 70.

Figure 2-30. “Star Trek” Arcade Cabinet. (“Star Trek (1972)”)

Figure 2-31. "Computer Space Ball" flyer employing Data 70. ("Video Game: Computer Space Ball, Nutting Associates")

Figure 2-32. "BiSci Blood Type Challenge," a learning game using Data 70. (Janzen)

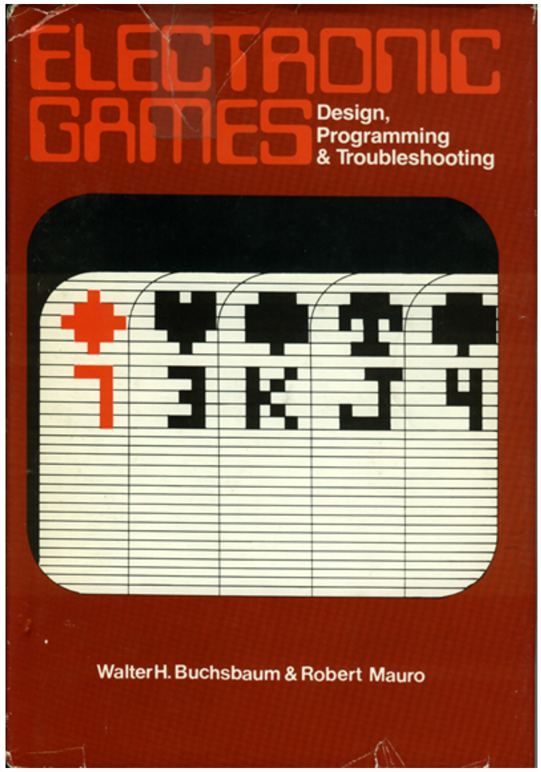


Figure 2-33. Cover, *Electronic Games: Design, Programming, and Troubleshooting* (1978).
(Buchsbaum & Mauro)

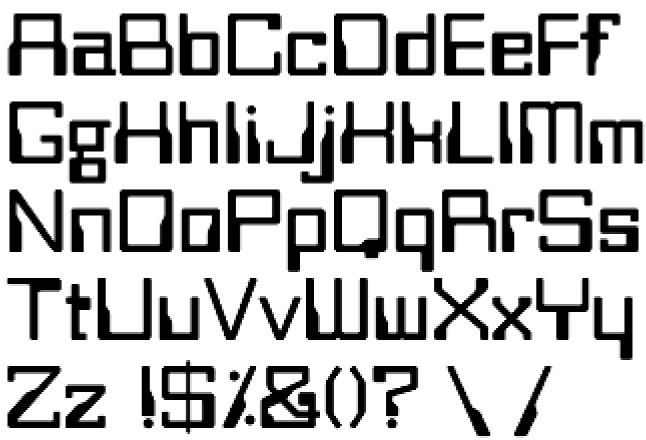
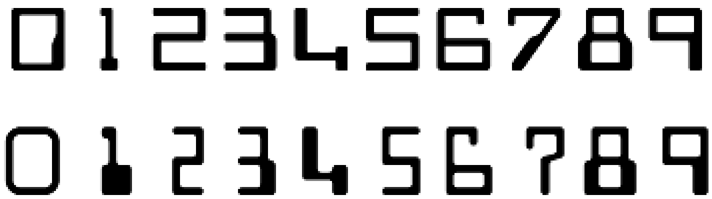


Figure 2-34. Sample of Orbit-B BT.



A

B

Figure 2-35. A) Orbit-B's numerals. B) E-13B.

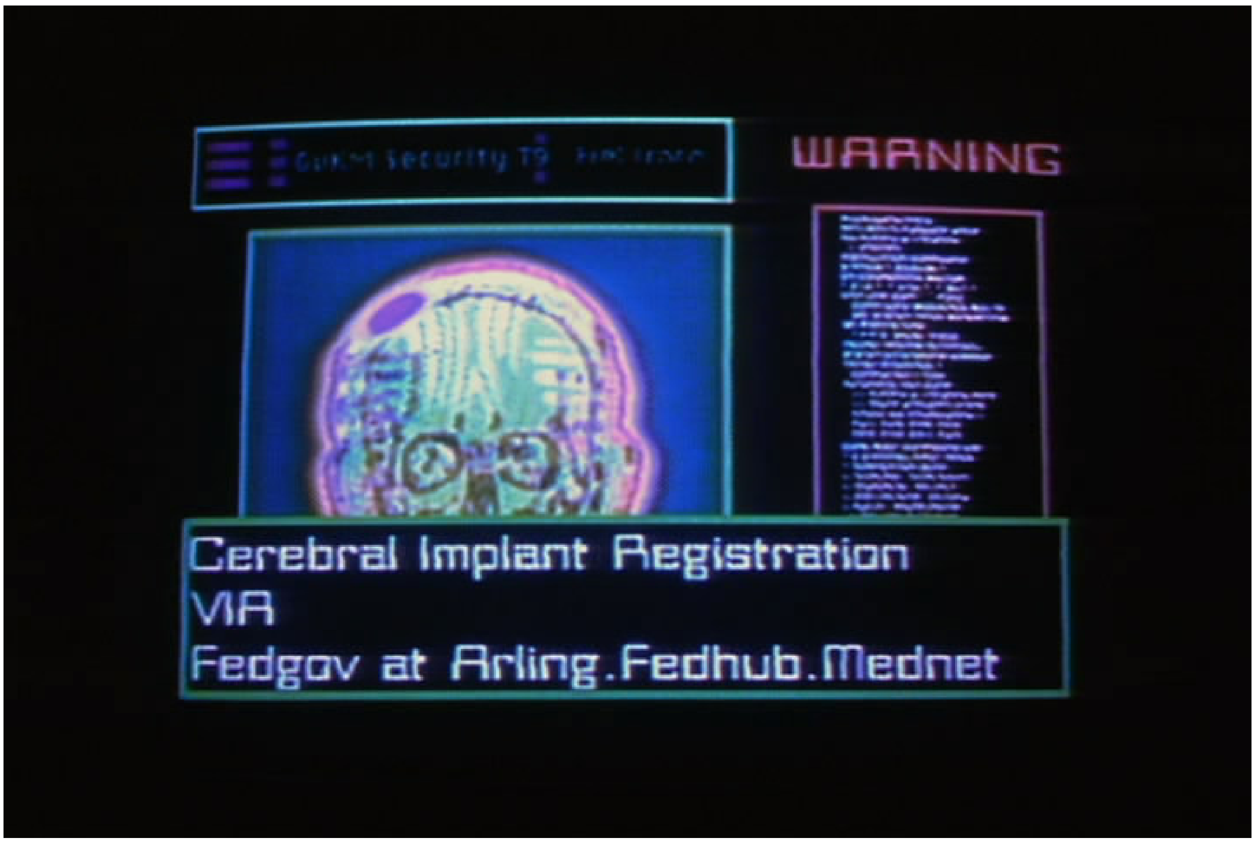


Figure 2-36. Orbit-B in use in a screen interface in the film *Johnny Mnemonic* (Longo).

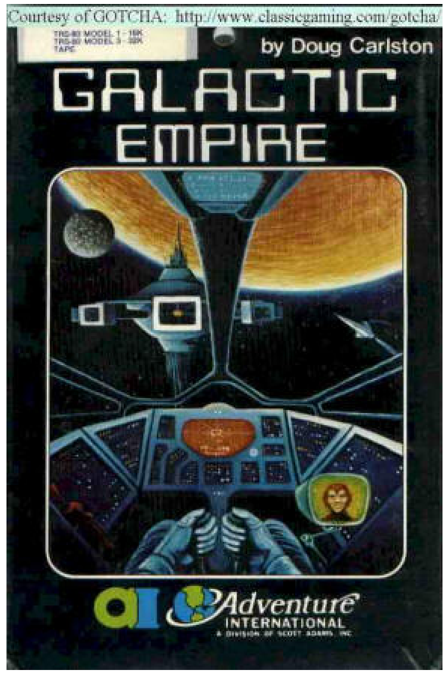


Figure 2-37. Orbit-B on the cover of the 1982 release of Brøderbund's *Galactic Empire* (Falk).

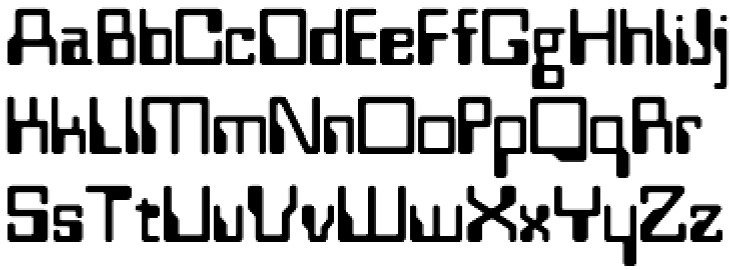
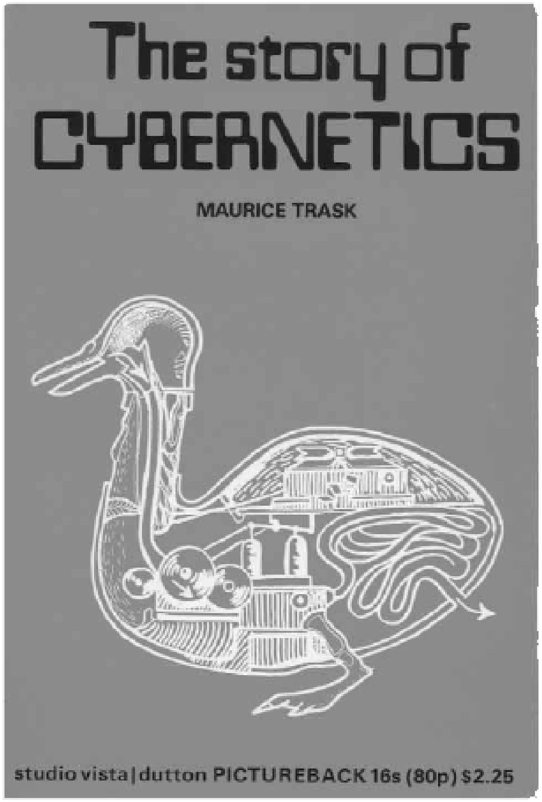
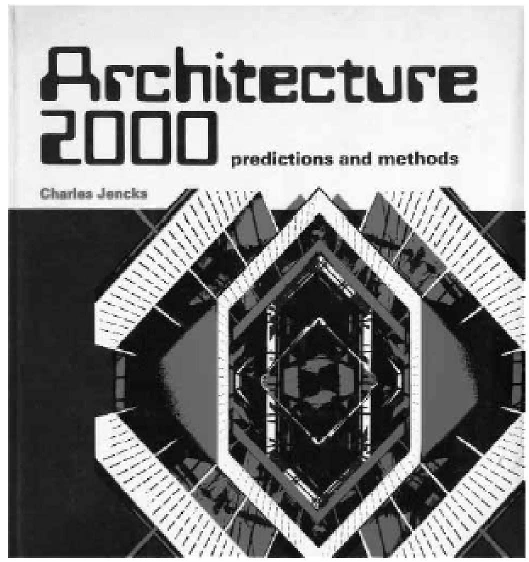


Figure 2-38. Sample of Westminster typeface.

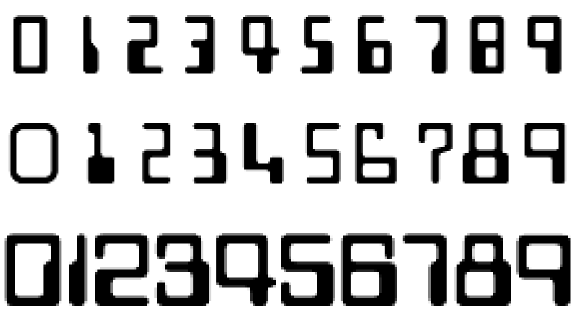


A



B

Figure 2-39. Westminster in use on covers of A) Maurice Trask's *The Story of Cybernetics* (1971) and B) Charles Jencks's *Architecture 2000* (1971). (Owens & Reinfurt 148)



A
B
C

Figure 2-40. Comparing numerals for A) Westminster B), E-13B and C), Data 70.

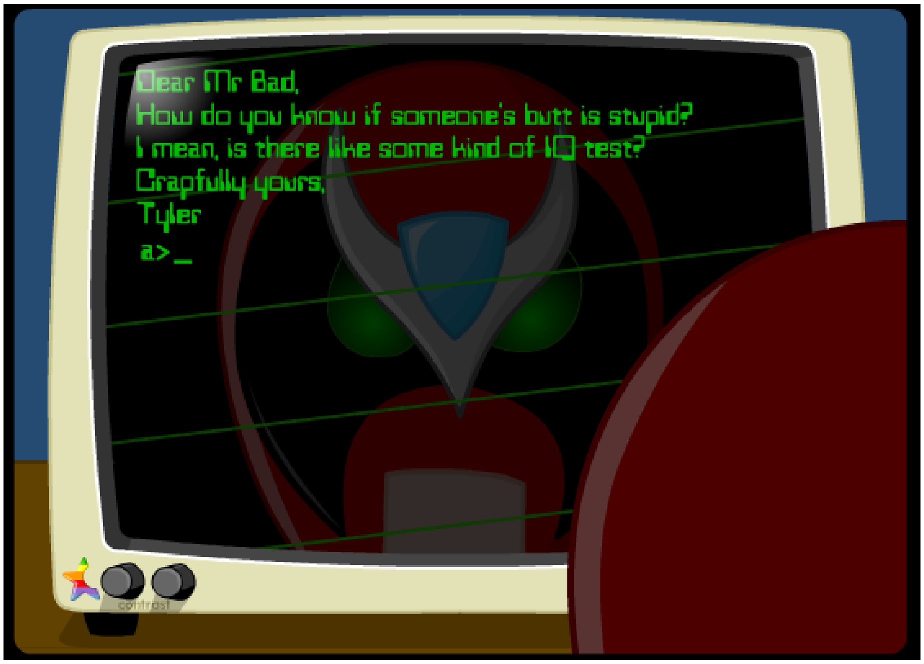


Figure 2-41. Westminster in use as simulated screen font for Strongbad's Tandy 400 (Matt Chapman & Mike Chapman, "Butt IQ")

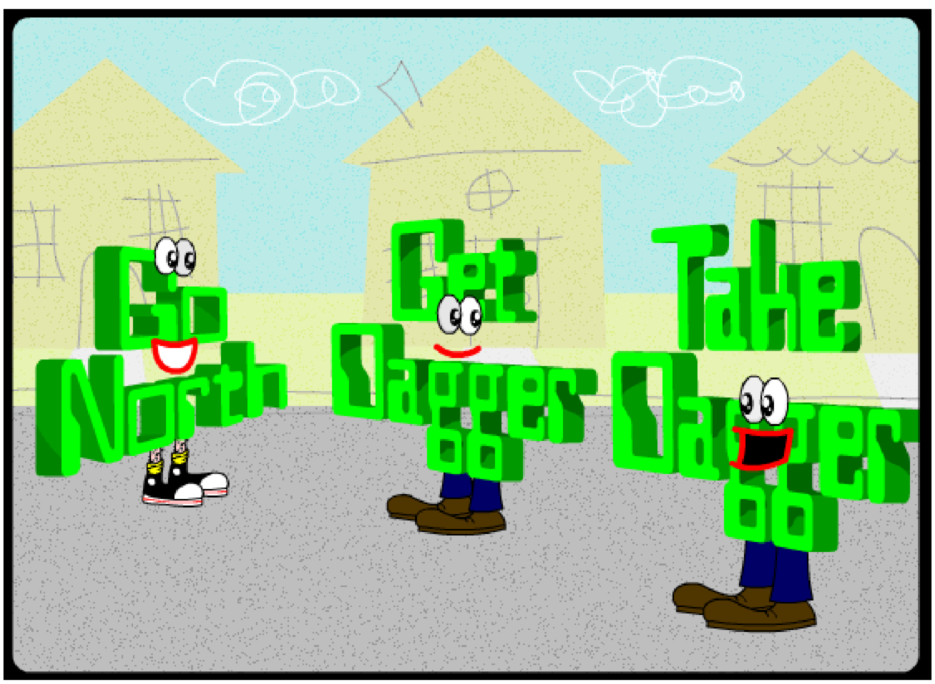
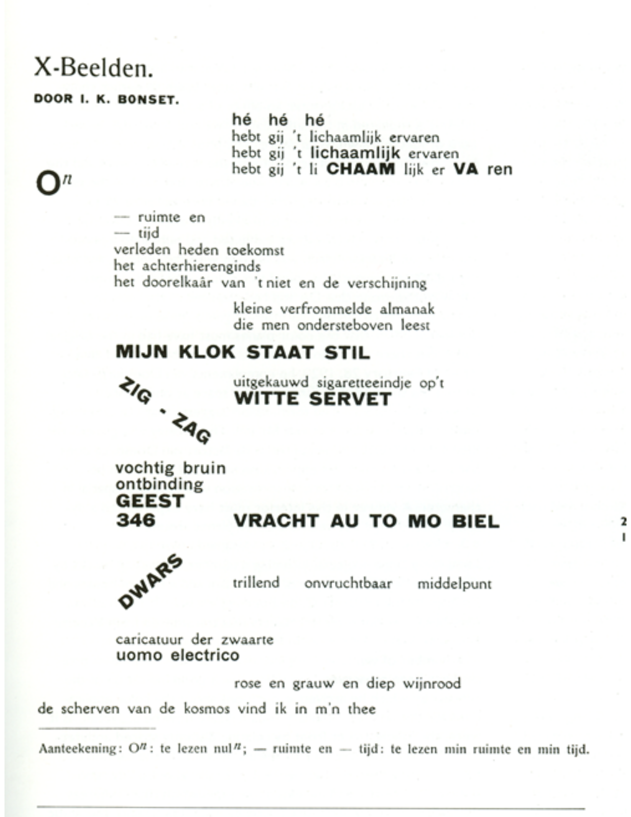


Figure 2-42. "Thy Dungeonman" cartoon from the Strongbad Email, "Web Comics." (Matt Chapman & Mike Chapman, "Web Comics")



X-Beelden.

DOOR I. K. BONSET.

hé hé hé
hebt gij 't lichaamlijk ervaren
hebt gij 't **lichaamlijk** ervaren
hebt gij 't li **CHAAM** lijk er **VA** ren

O^n

— ruimte en
— tijd
verleden heden toekomst
het achterhierenginds
het doorelkaär van 't niet en de verschijning

kleine verfrommelde almanak
die men ondersteboven leest

MIJN KLOK STAAT STIL

ZIG - ZAG

uitgekauwd sigaretteeindje op't
WITTE SERVET

vochtig bruin
ontbinding
GEEST
346 **VRACHT AU TO MO BIEL**

DWARS

trillend onvruchtbaar middelpunt

caricatuur der zwaarte
uomo electrico

rose en grauw en diep wijnrood

de scherven van de kosmos vind ik in m'n thee

Aanteekening: O^n: te lezen nul^n; — ruimte en — tijd: te lezen min ruimte en min tijd.

Figure 2-43. "X-Beelden" ("X-images") by Van Doesburg, 1919 or 1920. (Blotkamp, "Theo van Doesburg" 31)

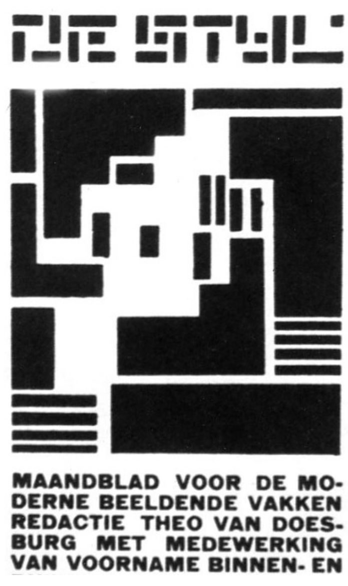


Figure 2-44. De Stijl logo, designed by Theo Van Doesburg and Vilmos Huszár.

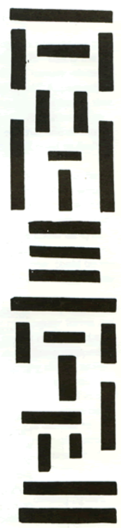


Figure 2-45. Vilmos Huszár, *Ornament XXe eeuwse stijl* (1917).

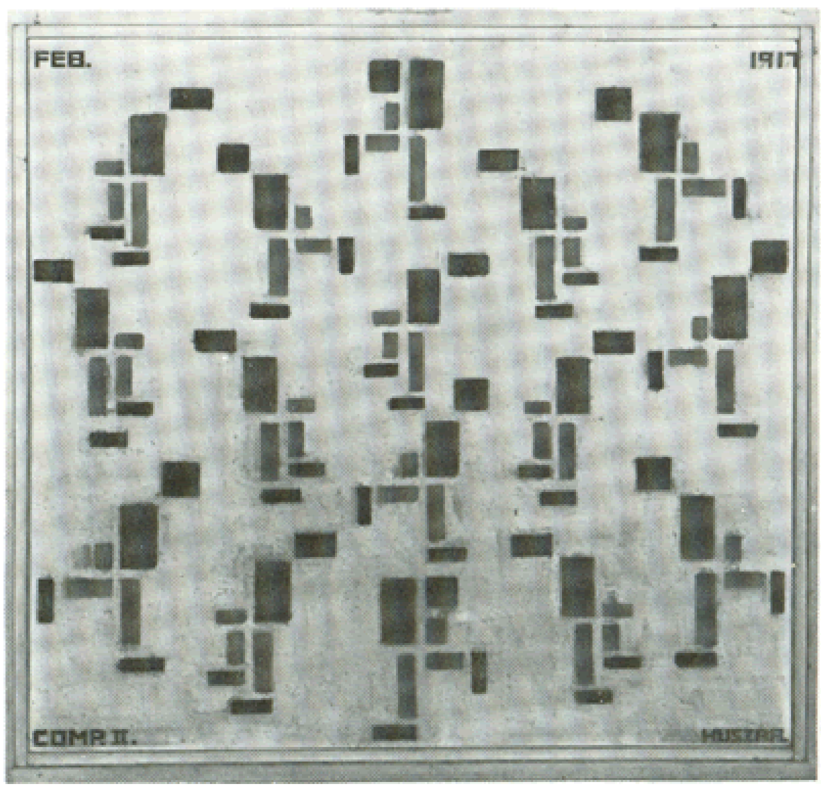


Figure 2-46. Vilmos Huszár, *Composite II* (1917).

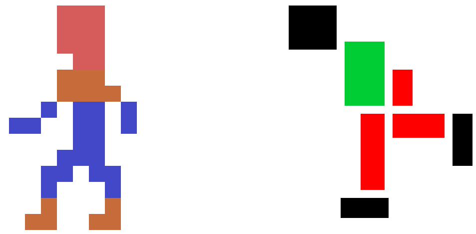


Figure 2-47. Ice skating figure from Huszár's *Composite II* compared with similar sprite figure used in *Superman* (1978) for Atari VCS. (Images enlarged and enhanced to show similarity.)

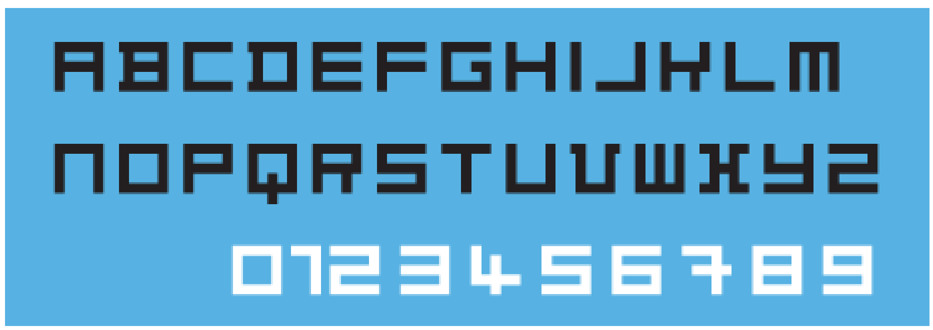


Figure 2-48. Sample of "Architype van Doesburg," a font released by The Foundry after van Doesburg's original 1919 alphabet design. (Wikipedia contributors, "Architype Van Doesburg")

Figure 2-49. *Wired Magazine* spine text and section logos.

Figure 2-50. *Boing Boing* logotype.

A

B

Figure 2-51. A) Text “START” from *Blue Print* [screenshot captured with Stella]. B) Text “START” rendered in P22 De Stijl, a font based on van Doesburg’s 1919 alphabet.

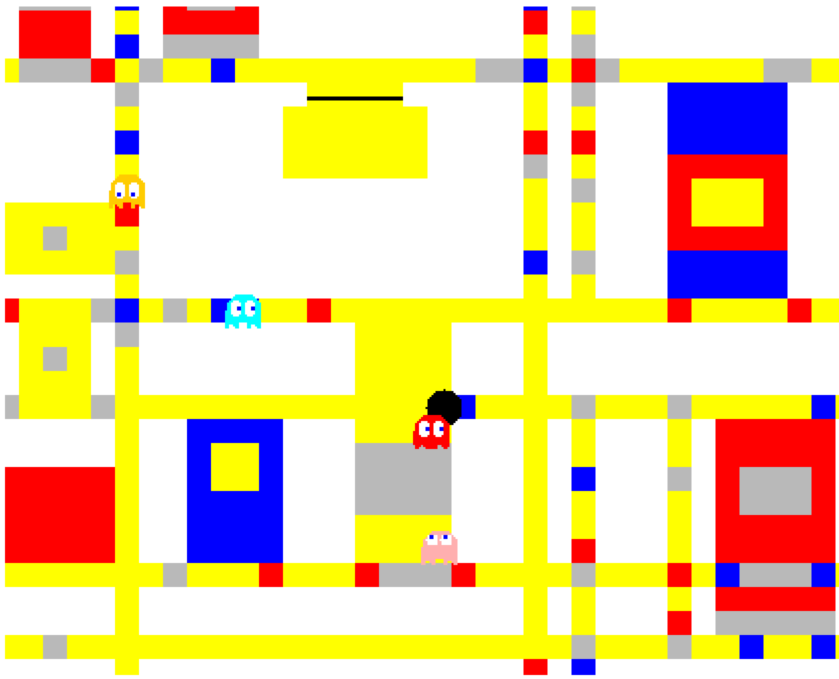
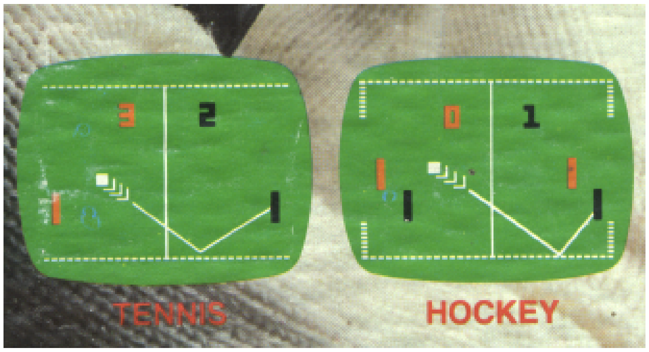


Figure 2-52. *Pac-Mondrian*.

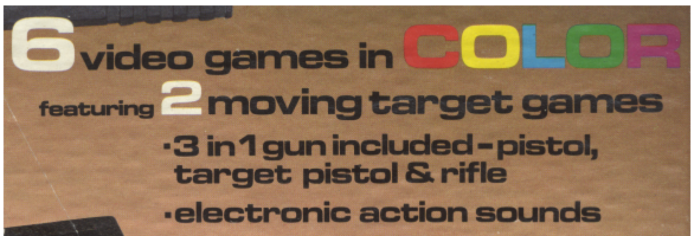
Figure 2-53. Cover of Wim Crouwel’s pamphlet, *New Alphabet: An Introduction for a Programmed Typography*. (Scan from Broos 83)

Figure 2-54. Page describing “Eurostile” from Aldo Novarese’s *Alfa Beta* brochure. This image appears in Blackwell, page 106.

Figure 2-55. Box art for Coleco Telstar Marksman (1978) featuring a Eurostile font alongside illustrations of television screens.



A

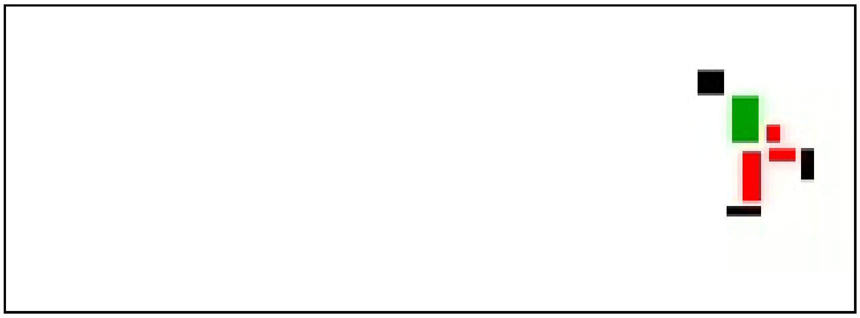


B

Figure 2-56. A) Detail of television screen images on Coleco Telstar Marksman box. B) Detail of text on Marksman box.

Table 2-1. Implementations of MICResque numerals in Atari VCS games.

Style Sample	Implementations
	• *Squeeze Box* (US Games: 1982) • *Death Star Battle* (Parker Bros: 1983)
	• *Marauder* (Tigervision: 1982)
	• *Star Voyager* (Imagic: 1982) • *Star Wars – The Arcade Game* (Parker Bros: 1983)
	• *Yar's Revenge* (Atari: 1981) • *E.T. – The Extra-Terrestrial* (Atari: 1982) • *Saboteur* (Prototype) • *The A-Team* (Prototype)



Object 2-1. A simple animation created by using the skating figures in Huszár's painting as sprites. Click in the rectangle above to start playback; alternatively, click on this text to download the file directly. (.avi file 70KB)

CHAPTER 3
HOLOTYPES AND HARDWARE

hol · o · type *(n)* – any instance of typographic form which is continuous with and representative of a specified textual medium.

The previous chapter approached the paratextual domain of videogame typography from the outside in. Focusing on "paratypical" typefaces, chapter 2 draws connections between the ontological, textual conditions of videogames and the historical, mechanical origins of the typefaces which make similar or contiguous appeals to technological constraint as an aesthetic function of mediality. This chapter reverses this approach to work instead from the inside out. Therefore, if the ultimate conclusion of chapter 2 is that typefaces like Data Seventy tell us something about videogame artifacts because their origins in mechanical reproduction prefigure the electronic textuality of videogames, then the central premise of this chapter is that the typographic structures in videogame artifacts can tell us something about electronic textuality in general through the specific domain of the videogame as a site for and medium of inscription. The purpose of this chapter is to describe holotypical videogame typography, where *videogame typography* refers here to the design and arrangement of alphanumeric artifacts of expression within the paratextual domain of videogames and the modifier *holotypical* identifies a particular subset of videogame typography which defines aesthetic patterns relevant to other subsets. A videogame holotype, therefore, is a specific instance of videogame typography to which other instances of type (paratype) may be compared in order to determine an association with videogame textuality. Accordingly, The central question this chapter seeks to answer is, "What can typographic forms in videogames tell us about the structures shaping videogame expression?"

Before proceeding to analyze several key forms of holotype, the term *holotype* and the significance of the prefix *hol-* require explication in this context. To return briefly to the domain of botanical taxonomy from which the terms holotype and paratype derives, the key function worth noting is the singularity of the holotype as a taxon-generating specimen. Article 9.1 of the International Code of Botanical Nomenclature defines *holotype* in the following way: "A holotype of a name of a species or infraspecific taxon is the one specimen or illustration … used by the author, or designated by the author as the nomenclatural type. As long as a holotype is extant, it fixes the application of the name concerned…" (International Botanical Congress 12). In this context a *type* is a nomenclatural type, an individual member of a species that provides, through its existence, a definitive norm for the larger set of individuals that may be described as part of that taxon. For videogame typography, the relevance is that any instance of type appearing in a videogame is in some way definitive for the larger set (including paratypes) of all instances of videogame typography. The act of naming some typographic artifact as some kind of *videogame* type depends (as a nomenclatural set) on the identity and characteristics of videogame holotypes. This approach is most consistent with the context-based definition offered in the previous chapter, because *any* typographic expression in a videogame may be considered definitive. *Hol-* has numerous other significations as well, mostly in contexts where it establishes concepts of wholeness or completion, as in *holocaust* (literally, "wholly burnt"). This prefix also forms the basis of two similar, etymologically intertwined words which will be significant for a later chapter. For the present discussion, the terms *holograph* and *hologram* are both relevant to clarifying the application of *holotype* in the context of videogame typography and its implications for understanding videogame textuality.

Holograph and *hologram* bear an unusually similar etymology in which both are constructions of *holo-* (derived from the Greek ολος [holos], relating to "whole, or entire") and -*graphos* (γράφειν [graphon], "to write"). Their meanings are quite different, however, as is the relationship each posits between a concept of wholeness and a practice of inscription. *Holograph* (as a noun) typically refers to a manuscript, sometimes specifically one that is written by the author's hand. A *hologram*, however, is an optical phenomenon related to a kind of photographic recording which inscribes a total image that can be rebuilt from any part of it. This is the sense of the term which describes a physical model of the Universe; the so-called holographic paradigm, proposed by David Bohm and Karl Pribram. In popular use, *hologram* generally refers to the three-dimensional variety of this technology. In relation to writing, the sense of wholeness captured by *holograph* is one of authorial intent; the authority of the document is said to rest on its being wholly the work of an author. In relation to photographic inscription and related quantum neurological models, *holograph* invokes a sense of wholeness as a consequence of a property of light and refraction. In terms of a system (like a room or a galaxy), the whole is a property of all of that system's parts. The *holo-* in my *holotype* relates to both of these "wholenesses," because specific typographic forms in videogames emerge from the material substrate of the game text and, in the case of Atari VCS games, may be said to bear the individual traces of the programmers in a much more personal or "authorial" manner than otherwise available on later platforms. These traces reflect the material affordances of the medium as well as the individual "hand" of the programmer because the constraints of the VCS platform require programmers to craft custom fonts for each game. As a holotype, therefore, the letter composed in this manner contains the influence of the programming code and data which it is intermeshed with in the game's ROM data. In this way, the videogame letter becomes an emblem for the entirety of the game and even gestures toward the broader culture of videogame when it is reused and adapted in paratypical contexts.

As a question of interiority and exteriority, respectively, holotypical and paratypical instances of type are not simply distinguished by their being "in" a videogame versus being "around" a videogame. Furthermore, in the sense that both descriptors are qualities of *paratext*, the game text to which the typographic is relatively "para" is not sufficiently clear, as becomes obvious when considering text within videogames. For example, in the game *Mag Max* a generally bland shooter title for the NES console, the main menu screen features more text on screen than at any other time within the game. However, one may reasonably argue that the menu screen is not the game itself, but rather an entry point to the actual game. The question, therefore, is whether the text of *Mag Max* consists of the interactive, rule-based experience of playing the game, or whether *Mag Max* is the mediated experience of the running the software on a specific configuration of hardware. These two alternatives underlie some profound disagreements about how best to study videogames, and their dependence on textual differentiation highlights the significance of typographic analysis for videogame study.

Therefore, rather than defining *holotype* in a way that depends on a specific medial context (for example, "on the screen") or a specific textual context (for example, present during gameplay),[1] I propose to focus *holotype* as that which uniquely reveals the internal stratification of either of either of those alternatives. The point is that alphanumeric forms have close contact with the compromises and constraints faced by game programmers and graphic designers, and letters and numerals retain traces of that contact in a way that representational or iconic game pieces do not. Therefore, *holotype* is always peritextual (Genette's term) in the sense that its presence is contiguous with the text itself, but in addition to providing "entry" into the text a holotype also invites entry into that which makes the existence of the text possible. It is in this sense that the *holo-* prefix is appropriate – the sense in which holotypical designs constitute a discursive expression of total videogame architecture and design.

This chapter explores the concept of holotypical videogame textuality by examining a number of phenomena and technologies which contribute to an understanding of this term. Since the chief content of a holotypical expression is its form, some of this exploration will involve

1 Even this may not qualify as immanently part of the game experience. For example, the "high score" display that was a fixture of 1980s arcade games is a record of a prior instance of play intruding on the current one.

uncovering the origins of formal conventions and constraints common to videogame typography, including the origin of the 7-segment figure and different approaches to storing and retrieving alphanumeric characters, including the resulting influence on character design. This is offered from a historical perspective, seeking to orient the discussion around the first appearances of particular videogame textual artifacts and postures. Another dimension of holotypical textuality is its implications for considering videogames as a site of and means for inscription. The central premise of discussing game textuality presumes a significance for writing, but as a means for inscription, the sense in which games, or any digital artifact, should be considered writing technology is not entirely clear. The practice of leaving easter eggs (hidden content left in games by their programmers), however, is one phenomenon which does present itself as an important mode of writing for videogames. The middle section of this chapter analyzes some important easter eggs, including Warren Robinett's signature hidden within *Adventure*. While this may not be the first videogame easter egg, I argue that it may be the most important – partly because of its reputation as the first easter egg, and partly because it demonstrates so well the unification of the concept holotype with its etymological relative, holograph. Finally, the chapter concludes with a consideration of the screen as a metaphor and agent for textuality. Taking seriously Nick Montfort's critique of screen essentialist approaches to studying New Media,[2] this section argues for a deeper understanding of the literal screen in relation to textuality in order to provide a means for re-evaluating various metaphors indicative of digital utopianism. This concept of an ontology driven by an aesthetic metaphor is discussed further in chapter 4 where I propose "fuzzy" and "jaggy" models for videogame textuality.

2 This term, "screen essentialist," appears in Montfort's 2004 presentation at the MLA Convention, "Continuous Paper," where Montfort is critical of a tendency toward naturalizing the computer screen as a given of electronic culture, when in fact works such as ELIZA and "Colossal Cave Adventure" were first experienced on paper teletype terminals. Matthew G. Kirschenbaum picks up this term in *Mechanisms* as one necessary component in building a formal materialism of digital texts.

Hardware

The first section of this chapter is devoted to a survey of key developments in videogame technology as it relates to the production of text in videogames. Videogame history is closely intertwined with developments in computing, so a full historical analysis of the technology is beyond the scope of this chapter (or this study). Instead, by focusing on specific technologies that had significance regarding the production of alphanumeric characters on videogame screens, this section provides a technical background for the discussion that follows and sets up the concerns which the next section addresses such as the advent of score keeping as part of the game text and the question of the videogame as an inscription technology. In addition to discussing specific platforms and their typographic affordances, I also analyze, where appropriate, the particulars of different methods for composing alphanumeric forms, such as the 7-segment system and the various bitmap grids (e.g. 3 x 5, 5 x 7, 7 x 9, etc.) which are employed by various technologies. These two lines of inquiry address similar concerns, but they are treated separately because some character-generation techniques span across several different platforms, and some platforms allow several to coexist within the same hardware.

It should be noted that the historical perspective I am choosing in the section specifically avoids the broader question of textual production in modern computing. By choosing instead to focus on games – a medium (or genre) of text for which the importance of alphanumeric or verbal text is debatable – I come at the textual question by avoiding the operative framework of normativity. The videogames under particular emphasis here are those which operate primarily on a graphical mode of representation. The alphanumeric signifiers which share a screen with these predominantly visual signifiers introduce an alternate textual vector, but as such, operate within a significantly visual paradigm. As the following discussion illustrates, the inclusion of any text or numeric display is secondary to the production of visuals, and many of the earliest forms of games lacked either. As systems did begin incorporating score display and simple text, it was often implemented with the strictest parsimony, and it was not until their third generation that videogame consoles were capable of generating text from a complete alphabetic character set.

This focus on predominantly visual game systems also necessarily excludes primarily textual games. Text adventure games like *Colossal Cave Adventure*, *Zork*, and the numerous examples of so-called interactive fiction produced since the mid-90s do clearly have a visual dimension. In fact, one popular interactive fiction interpreter, Gargoyle, is advertised for its superior typography: "In this computer age of typographical poverty, where horrible fonts, dazzling colors, and inadequate white space is God, Gargoyle dares to rebel!" (Tor). Furthermore, texts such as Adam Cadre's *Photopia* use typographic characteristics (color, in this case) to communicate shifts in the space of the story-world. It is also a significant feature of many early games that they were originally programmed and played on computer terminals which output their responses to printer rather than a screen. Nick Montfort has noted the difference this makes in terms of gameplay, observing, for instance, that having a printout of one's entire game thus far made it unnecessary for the game to repeat descriptions for areas already explored by the player. The player simply has to read back through the printout to find the previous description (Montfort, "Continuous Paper"). It is impossible, therefore, to ignore the visual, typographic properties of these texts, especially when considering the question of videogame grammatology. However, the decision to focus on games which normalize visual representation predominantly is based in the premise that analyzing typography in these contexts where alphanumeric representation is secondary provides a clearer view into the materiality of videogames as a medium. This is because the attending hardware for producing alphanumeric is similarly ancillary to the game programming. Therefore, the relationships among visual, verbal, and ludic representation can be mapped onto both formal and physical layers, and the extent to which those two forms of materiality come into contact and resist one another is fundamentally characteristic of the videogame as a unique form of electronic textuality.[3]

Platforms

As it is used here, the term *platform* refers to the environment (hardware or software) which underlies a digital work. A platform is the environment which brings the game text into existence and which determines its characteristics to a greater or lesser extent, and the platforms which make videogames possible are not always used solely for ludic purposes. In many cases, the technology necessary to bring videogames into existence was originally created for a different purpose, and it is not until the integrated circuit chips of the mid 1970s (for example, the AY-3-8500 or "Pong on a chip") that hardware dedicated to videogame-specific tasks such as keeping score and generating a play field were cheaply available. The following section discusses these developments in videogame technology where they have relevance for the generation of letters and numbers on videogame screens.

A number of different technologies can make claims for priority as the first videogame, with significant legal and ideological implications for each. Any answer to the question "what was the first videogame or computer game" necessarily depends on definitions of highly loaded and contestable terms like "game," "video," and "computer." Temporarily leaving aside the implications of "video" in the label "videogame" and instead considering the more general term

3 Besides text adventures and interactive fiction, the other predominantly textual genre of computer game which will have to be omitted in this discussion is the so-called Rogue-like or textmode games. These games were designed to take advantage of UNIX terminals which could output only ASCII text, so the game elements are all typographic characters which are being used to represent game objects and characters. The player, for example, is typically represented with an '@,' walls are build with pieces of '|' or '-,' and monsters are typically depicted with letters which mimic their physical appearance and/or which pun on their names. Bats are represented with 'B,' for example, and kestrels are 'K's. See figure 3-1 for a screenshot of *Rogue*.

While these games offer a fascinatingly typographic experience, they are technically graphical games since the representational status of the typographic characters is more hieroglyphic than textual. Mark J.P. Wolf excludes roguelike games from the "text adventure" genre for this reason (Wolf, "Video Game Genres" 274), and Brett Camper notably describes enemy characters in *Beast* (a later ASCII-based game) as "H-shaped," rather than simply H's (Camper 153).

"computer games,"[4] several implementations combining computing input and control with some form of projected display can stake legitimate claims of "firstness." The earliest recorded claimant is a "Cathode-Ray Tube Amusement Device" patented in 1947 by Thomas T. Goldsmith, Jr. and Estle Ray Mann with patent #2455992. The only graphical output of this device seems to be a single missile trajectory: "a cathode ray tube is used upon the face of which the trace of the ray or electron beam can be seen." Further graphics are suggested as "pictures of airplanes" which may be placed on the face of the screen, and even animated explosions: "The game can be made more spectacular, and the interest therein both from the player's and the observer's standpoint can be increased, by making a visible explosion of the cathode-ray beam take place when the target is hit" (Goldsmith, Jr. & Mann). The device lacks any textual display, however, and even the structure for score keeping or rudimentary game rules seems absent from the inventor's description. The typographic question, however, is whether any display apparatus was or could have been used for text. And in the case of this unnamed missile simulator, the answer appears to be no.

OXO

OXO, designed in 1952 by Alexander Douglas is another early experiment which some consider the first computer game (Kirriemuir 22; Rutter & Bryce 7; Carey 60). OXO consists of a simple game of Tic-Tac-Toe (Noughts and Crosses) implemented on an EDSAC mainframe, and it ostensibly employs the alphabetic characters "X" and "O" as game pieces. Semantically, these symbols are not alphabetic in any linguistic or verbal sense, but even as game pieces, the typographic symbology of X and O extends from alphabetic contexts. In other words, whether the glyph, X, is a "cross" or the 24th letter of the English alphabet, it can be re-created by typing a letter X on a keyboard. Thus, the aesthetic qualities exhibited on the output screen of the EDSAC do express a particular technological constraint that relates to its underlying programming and hardware.

4 This is a question I deal with in some detail in Chapter 1.

In the EDSAC simulator shown in figure 3-2, the graphical output appears in the circular screen image at the top left. Although this is clearly graphical output, it is important to note the method by which this output was accomplished. Normal input and output on the EDSAC was accomplished through paper tape, but three CRT monitors indicated the status of data currently in process on the machine. Martin Campbell-Kelly explains as follows:

> One of the most important checking aids on EDSAC was a monitor tube on which the 32 words in one of the 16 main store-delay lines, or "tanks," could be displayed in binary. It was thus possible to watch the progress of a programme during its execution. This practice was known as "peeping." Programmers often arranged to store all the interesting numbers in a programme in the same tank so that peeping would yield as much information as possible. EDSAC was a very slow machine by later standards, obeying around 600 orders per second, so it was quite possible to gain a reasonable impression of what was happening in a programme by observing the monitor tube when the programme ran at full speed. (24 - 25)

In other words, the output displayed on the monitor was a literal visualization of the machine's binary code in the moment it was being processed. The purpose of this display was to allow for debugging programs, so the display generally held no graphical significance at all. During operations such as loading or calculating, the CRT display tank might look something like the image in figure 3-3, which depicts a "peep" at tank 0 while processing a simple "Hello, world" program. In other words, utilizing the peep monitor for graphical output symbolic forms at all was a significant repurposing of its main use, and as an emblem of the machines inner workings, the design graphical characters accomplished in this way are a direct expression of the device's memory storage and display hardware. As perhaps the first instance of videogame holotype, *OXO* provides an excellent demonstration of the core concept of uniting visual textuality with physical materiality that is a crucial function of videogame textuality.

Tennis for Two

William Higginbotham's *Tennis for Two* (1958) is another early creation often credited with being the first computer game. However, like Goldsmith and Man's "Amusement Device," *Tennis for Two* lacked text display and did not employ any representational figures other than a point of light simulating the trajectory of a tennis ball. The game allowed players to engage in a game tennis (depicted in side-view rather than the top-down perspective later employed by *PONG*) rendered on an oscilloscope.

Spacewar!

Spacewar!, a space combat game created by MIT students Steve Russell, J.M. Graetz and Wayne Wiitanen to run on the PDP-1 mainframe computer, was far more influential than either *OXO* or *Tennis for Two*, but it still lacked text. Although it was fairly complete as a game, including a setting, representational figures, movement and combat action, the original versions of the game lacked any on-screen text or even score keeping. The introduction of these typographic elements came as these early videogame-like texts reached the critical milestone of monetization.

Galaxy Game

In many ways, *PONG*, by Atari, can be considered the first commercial arcade game that led directly to the creation of an industry and culture of gaming, but it was not the first. *Computer Space*, a space-based combat game similar to *Spacewar!*, and *Galaxy Game*, a multi-player adaptation of *Spacewar!* on a PDP-11 (Pitts), both predated *PONG* and initiated the formal relationship of mixed graphical and textual display in a videogame environment. The history of how Nolan Bushnell came to create and market *Computer Space* is well documented;[5] *Galaxy Game* is frequently overlooked, perhaps because of its obscurity (only one model was ever constructed), but it does appear to be the first commercial arcade game, preceding *Computer Space* by two months (Pitts). *Galaxy Game* cost 10 cents per game, or 25 cents for three games, and players had to keep track of their fuel, and number of games remaining. Players also had access to a menu of options before the game loaded where they could set variables such as gravity, the number of players, and hyperspace (Panofsky). The game was installed at Stanford University,

5 See, for example, Burnham, *Supercade: A Visual History of the Videogame Age*; Baer, *Videogames: In the Beginning*, Herman, *Phoenix: The Rise and Fall of Home Videogames*; and Kent, *The Ultimate History of Videogames*.

where it had been built by Bill Pitts, Hugh Tuck, and Ted Panofsky (Pitts). Though its scope was limited, *Galaxy Game's* significance as the first commercial game is further amplified by its also being the first videogame to combine on-screen textual display with gameplay. The fact that this innovation is precipitated and made necessary by the breaking up of play segments into monetized chunks is important because it indicates the connection between videogame text and the cultural capital of the medium at its earliest inception. As such, this is a significant instance of holotype. According to Ted Panofsky, who designed the original display circuitry, the text characters were all emulated in the game software, in much the same way that later devices like the Atari VCS would store and retrieve graphical bitmaps for each character. However, like *Spacewar!, Galaxy Game* employed a vector-based display:

> The text was displayed as a vector character set, entirely done in software; there was no hardware text generator. The display hardware was entirely vector based with the software loading end-points and intensity. It then gave a go command and the hardware drew the vector and gave an interrupt when the vector was complete. The software main program created a vector list and an interrupt routine went down the list. The list was repeated at the refresh rate. (Panofsky)

This method is significant and worth noting as an important holotype because the character generation was, in essence, a secondary goal of the game system. The PDP-11 and the CRT monitors that constituted the physical hardware of the game were first and foremost intended for the production of graphic images and the management of their control, just as was the case with *Spacewar!*. By improving on *Spacewar!*, the makers of *Galaxy Game* essentially repurposed a display system originally intended for the production of symbolic graphics. Like other instances of holotype, this relationship of text and image, and the sense in which symbolic graphics are produced through the same material and programmatic means as alphanumeric anticipates the symbolic function of game-like typography when, as paratype, it creates associations with a situation of technical congruence with graphical signification.[6]

6 At the time of this writing, the *Galaxy Game* machine is stored in the Computer History Museum in Mountain View, California. At its most previous location on display at Stanford University, the game was still functional as

Computer Space

Computer Space made a similar contribution to the improvement of text on videogame screens, by including statistical information about the game on screen during gameplay (see figure 3-4). Designed by Nolan Bushnell and Ted Dabney in 1971 for Nutting Associates, Computer Space was arguably the first commercial arcade game (Wolf, "Arcade Games of the 1970s" 36), its preference for that title over Galaxy Game owing to the fact that more than one were made. One can find a few different versions of the story of how *Computer Space* came to be so similar to *Spacewar!*, but the end product was an adaptation of the *Spacewar!* concept in a much cheaper package that could be mass-produced. Replacing the mainframe microcomputer with discrete logic circuitry (no CPU or RAM, in other words), Bushnell and Dabney created a device intended to sit alongside pinball machines and other electronic, pay-to-play amusements. Available in one and two-player models, *Computer Space,* offered gameplay for 25 cents ("History"). and it involved piloting a rocket or spaceship on a star field, firing missiles at UFO Peloponnese, and avoiding the UFOs' missiles. A back lit panel below the game's screen offers the following description: "A simulated space battle that pits computer-guided saucers against a rocketship that you control" (Qtd. in "History"). Though it was not a commercial success, *Computer Space* played an important role in bringing computer games to a wider audience, and its highly constrained textuality foreshadowed developments that would come later.

Like *Galaxy Game*, its immediate but isolated predecessor, *Computer Space* required some means for tracking a player's progress so that gameplay could be broken up into monetized time periods, in this case, 90 seconds per play (Bushnell 4). This is communicated to the player through three rows of numerical display at the right of the playing field. Figure 3-5 shows the descriptive labels affixed to the fiberglass case adjacent to their positions on the screen. The fact that the UFOs,

recently as 1997, but it is not presently offered as a "live" exhibit. As such, I have been unable to acquire a photograph of it in operation and have only been able to infer the actual aesthetic quality of its typography from first-hand accounts in e-mail correspondence with museum staff and Ted Panofsky..

rockets, missiles and star field all can overlap with the numerals (appearing, therefore, to exist *behind* the numerals on a shallow z-axis), this can be seen as a prototypical heads-up display (HUD) scenario , and the placement of explanatory labels on the game casing position these numbers literally and figuratively at one edge or layer of the game's textual space. The circuitry which generates these numerals, however, is contiguous with the backbone circuitry of the entire game, and as such place the computation and display of the score at the core of *Computer Space*'s medial ontology.

Figure 3-6 shows a selection of a technical schematic for the circuit board labeled "SYNC-STAR." As its name suggests, this board controlled the horizontal and vertical synchronization of the output video, as well as generating the star field. It also kept track of variables like the score and remaining time and in the section highlighted in figure 3-6 included a circuit for converting binary-coded decimal values (BCD) to 7-segment numeric figures. This conversion will be explained further in a later section, but it is worth noting here for its flexibility. This circuit is essentially a funnel through which all numeric values are processed and sent as video output to the screen. Here, individual numeric glyphs are constructed from data processed as 4-digit binary data to produce familiar shapes out of 7 segments which lack any signification until combined with one another to form meaningful shapes. In this way, the circuit shown here in figure 3-6 is a significant point of contact between human and machine reading. As such, the construction of these number shapes is fundamentally different from the method which accomplishes other graphics such as the rocket ship and saucers.

As shown in figure 3-7, the method for generating the rocket ship and saucer images required hardwiring the outlines into the circuit board. By casting a grid by overlapping pin-outs of a multiplexer chips, and anchoring specific points of overlap, this circuit was able to create points of light on the video output. This method, therefore, constitutes a kind analog sprite animation by providing four depictions of the rocket ship in different positions of rotation. By rotating these four images 90 or 180 degrees, this display circuit can use these images like frames in an animation depicting smooth rotation in 360 degrees. This is radically different from the means that produces the numbers in figure 3-6 because there is no internal representation of "rocketness" and therefore no interpretation required between human and machine understanding. The shape of the rocket alone is in this way less emblematic of videogame expression because its hardwired origins are, from the machine's point of view, arbitrary. However, the essentially unremarkable design of the 7-segment figures mitigates against their being emblematic or signifying anything beyond the quantity or ordinal value they stand for. Instead, it is the contiguity of these number images with the rocket ship and star field, sharing the same screen space though occupying separate signifying layers,which marks this as an important holotypical landmark in videogame history. This is however, in contrast with *PONG, Computer Space*'s far more famous successor. Although PONG utilized a similar method for generating the 7-segment numerals,[7] the location of the score display on the screen placed it above the simulated tennis court. In typographic terms, therefore, PONG offers a different order of textuality than Computer *Space*.

Magnavox Odyssey

At the same time that *PONG* was kick-starting the arcade game industry, the Magnavox Odyssey was pioneering the home console or "TV Game" market. The original Odyssey played 28 games, which could be selected by inserting circuit jumper cards. The display capabilities of the Odyssey were still rather minimal, so most of the games it offered required the use of included game pieces (cards, chips, etc.), and because the game lacks digital scorekeeping or score display, it offers a unique branch in the history of text in videogames. "Graphics" for the Odyssey were

7 I have not been able to locate an actual circuit schematic or detailed photograph of the *PONG* Arccade cabinet's circuit board, but it is a relatively safe guess that it employed the same BCD to 7-segment conversion as *Computer Space*, most likely relying on a 7448 chip. Al Alcron, who designed and built the PONG circuit board, has described his goal as creating a simpler, less expensive circuit than *Computer Space* (Deuel).

created by using partially transparent overlays which were attached to the player's television screen. For example, figure 3-8A illustrates the screen overlay for the game Haunted House. The game involves "lighting" areas of the haunted house image, as specified by cards the player draws (McCourt). The mostly opaque overlay acts as a barrier between the game world and the player's world, but it is also constitutive of the game world in a way that anticipates the later textual function of robust heads-up display interfaces in more graphically immersive games. As a site for textual inscription, however, the Odyssey's typographic features are limited only by the receptivity of the plastic to the ink printed on its surface. Still, few overlays included text, and most games which involved textual instruction did so through cards or prompted players to record their own score. Still, even though the Odyssey lacked the ability for generating alphanumeric characters, its dependence on peripheral objects marks it as a uniquely tactile gaming systems. In a formal sense, the games played on the Odyssey include printed cards as part of the symbolic structure of the game, so Odyssey games at least occupy a continuum with computer games placing all symbolic pieces on their screen. This is not, however, a formal argument. In relation to the typographic capabilities of its computational hardware, it is worth noting simply that the Odyssey's designers chose a novel approach to overcoming its limitations.

Pong on a chip

Atari released its home version of *PONG* in 1975, and an explosion of clones soon followed, made possible by General Instruments' successfully producing a cheap integrated circuit which could replicate the functionality of *PONG*'s and Odyssey's discrete logic circuitry. GI's AY-3-8500, also released in 1975, was the most successful of several variants on the so-called "pong in a chip" theme, and it was used in dozens – possibly hundreds – of different home game console devices (Winter). What made the 8500 so successful, besides its relatively low price, was that it could be easily combined with other chips to produce apparently different game devices, each focusing on one of its four PONG variant games. It also kept score and displayed two-digit score values (counting up to 15) for each player, so in that regard it is a notable improvement on many earlier non-integrated circuit systems such as the Magnavox Odyssey and the first home *PONG*. Figure 3-9 is an illustration from the data sheet released with the AY-3-8500, demonstrating the graphical output of its various components. Interestingly, because the output is generated as a sequence of timed pulses, the width of graphically unique characters is given as a unit of time. For example, the illustration in 3-9 is illustrating that the components of the numeral forms are 1 μs (one microsecond or one millionth of a second). The vertical dimension of letter segments is restricted to units of width and the repetition of pulses at that width and location along a sequence of scan lines. As shown in figure 3-10, the individual pixel units which compose each of the letterforms are at least 1 μs wide, and 4 scan lines tall (for NTSC) for 6 lines tall (for PAL).[8] This effectively creates the possibility of a 3 x 5 pixel grid, which is sufficient for creating each numeral in the same fashion as with the 7-segment form. This is significant because the numerals on the 8500's grid are generated with a different internal representation than the 7448 offers. Moreover, the figural representation of the score numerals depend for their compositions on units (scan-lines) which are joined in the video signal itself, as opposed to later systems which employ a screen buffer for composing an entire frame in software before passing the signal to the television. In visual terms, the 7-segment figure output of *Computer Space* and *PONG* have a good deal in common with one another, but contrasting the two (as in figure 3-11) reveals in important difference: whereas the 7-segment numeral is constructed out of 7 modular sub-units that can be combined differently, the 3 x 5 numeral is constructed out of 15 identical sub-units whose modularity is a function of their location on the spatial plane of the screen. Further developments

8 Televisions utilizing the NTSC format (i.e. all those in North America) generate a television image by passing a beam across the phosphor screen of the television set 525 times per frame. Each pass is stacked vertically with regard to its neighbors to form the x-dimension of the screen image's raster or grid. In PAL televisions (used in Europe and elsewhere), there are 625 lines per frame. Accordingly, GI's pong-in-a-chip came in two varieties: AY-3-8500 for PAL and AY-3-8500-1 for NTSC (Winter).

in videogame memory made more complex figures possible, but the grid structure or bitmap continued to be the major path along which videogame typography developed. Figure 3-12 is an illustration of a later version of the integrated game circuit, which included alphabetic characters for use in card games. Few of these card-game generating chips were included in gaming systems, but the composition of letterforms out of 5 x 5 pixel grids are similar, though not identical, to the letterforms which appear in the Channel F console.

Channel F

As videogame systems evolved. the next major development was introduced by the little known Channel F system, first released by Fairchild Semiconductor in 1976. Its main innovation was that game programming was stored in ROM memory which was burned onto removable cartridges. Players could then purchase new game cartridges which they could play in their Channel F. In typographic terms, the Channel F was more flexible than its predecessors, but this flexibility introduced its own limitations. Whereas the pong-in-a-chip consoles generated the numerals as a direct representation of an internally stored quantity, leaving no opportunity for an engineer intervention into the typographic representation of the numerals, the Channel F offered a similarly standardized set of numerals (Figure 3-13) which could be referenced in displaying score. The images of these numerals were stored as bitmap data within the system's BIOS memory,[9] so they were in this way similar to the automatically generated numerals employed on the AY-3-8500 chip. However, the flexibility of the method for storying ROM data on a game cartridge the opportunity to create their own graphics, and several Channel F programmers used this opportunity to create their own numeral sets. Doing so, however, required sacrificing memory which could otherwise be used for storing graphics.

9 The BIOS also included images for W, ?, T, M, and X, which were used to create a menu function when a game loaded on the machine. By toggling to the S setting, for example, players can set the maximum score for the game at hand. These bitmaps are documented in Sean Riddle's dissassembly of the BIOS which is available, along with other documentation, on www.veswiki.com (Riddle, "chanf.asm").

The default numerals stored in the Channel F BIOS are built out of 5 x 8 bitmap grids where the characters themselves take up the first 5 bits, leaving 3 bit columns for spacing. Figure 3-13 illustrates how this creates glyphs which effectively consist of 5 x 5 grids. Since the graphical output of the Channel F builds a uniformly rectilinear grid at a resolution of 128 x 64, the characters are form a pleasing, symmetrical square. In formal terms, the Channel F separates display data from a programmatic representation of numeric value in a way that parallels the modularity of the system's design. Similarly, the uniformly 5 x 5 design of the characters echoes the solid shape of the ROM cartridges that contain the game programming. As such, the relationship between the game programming and the Channel F core BIOS depends on the same kind of modularity and ability to be exchanged which characterizes movable type's departure from handwritten script.

Atari VCS

In a further improvement on flexibility, the Atari VCS or Atari 2600 , released a year after Channel F in 1977, left the design of alphanumeric characters entirely up to the programmers and designers of particular games. In other words, unlike the Channel F which stores selected numeral and letter shapes in its central memory, the Atari VCS, which lacks a BIOS, requires programmers to design their own letter and number shapes and store them within a game's graphics.[10] As Ian Bogost notes, the VCS was not a platform built for text, and the purposes it was built for introduce specific, affective constraints.

> In fact, the entire hardware architecture of the Atari [VCS] was crafted to accommodate *Pong*- and *Tank*-like games. The device's memory architecture and hardware register settings provide access to a play field backdrop, two player sprites, two missiles, and one ball. ...

10 This was only after a method had been devised for printing characters on the screen at all. As Warren Robinett recalls, "Doing text on the Atari 2600 was quite hard, even to get a 12-character line, which was what Basic used. It was thought to be impossible at first, but sometimes some clever programming of the 2600 display routine (which was called the "kernel") could surprise you. The 2600 display hardware, although very limited, was quite flexible. A 2600 programmer, Bob Whitehead, came up with a way of doing a chess display, which had 8 independent "chess characters" per line. I think the techniques he used suggested that it might be possible to even get 12 characters. Another 2600 programmer/designer, Dave Crane, succeeded in doing this" (Robinett, "Re: Question about Basic Programming").

> These constraints are not only physical ... but conceptual: the hardware was designed for games like *Pong* and *Combat*, artifacts based on tennis-like attributes. While ROM size 2600 cartridges eventually increased, new game concepts required VCS programmers to manipulate the hardware's affordances to create new play experiences. The VCS offers a striking example of how the structure of a technology platform exerts expressive pressure on the software created to run on it. (Bogost, *Unit Operations* 59)

Alphanumeric characters are, therefore, simply one kind of visual data stored within the ROM image burned into the game cartridge. With every game programmer designing his or her own unique game font, the Atari VCS experienced a tremendous typographic expansion and innovation in terms of uniquely expressive letter design, making it one of the most interesting consoles for typographic analysis. Also, the sense in which type design is not built in to the programming affordances of the device mean that game design on the Atari VCS has a uniquely textual dimension. Designers either created their own font or consciously (and likely tediously) implemented a pre-existing character set, in some cases conforming to company standards. In either case, the fact that the construction of a font takes on an intriguingly autographic dimension.

In historical terms, this expansion of alphanumeric forms parallels the rapid growth of the game industry and the development of major new game genres. For both of these reasons, the Atari VCS is an important console to study, as are the hundreds of games published for it, but this method of bitmap storage for typographic characters, coupled with the large number of games presents a challenge to any qualitative analysis. By using a piece of software developed especially for this purpose, I have identified a number of significant patterns that indicate aesthetic trends in game and graphic design for Atari 2600 games coinciding with advancements in programming techniques.[11]

11 This software, which I call "ROMscrape," is described in detail in Appendix A. ROMscrape consists primarily of a web-based search engine interface that allows users to input bitmap search patterns. This is similar to other software – such as ROMsearcher (Covell, "ROMSearcher"), which identifies probable sequences of ASCII references in binary data, and DiStella (Colbert), which separates assembly language instructrions from bitmap data – but ROMscrape offers a specifically visual interface to the code and (to deal with the potential individuality of programmers' character sets) is useful for finding close matches of visual patterns.

Like the Channel F, the Atari VCS is built around an 8-bit processor – in this case an MOS 6507 – but the Atari lacks a BIOS or other core software for running games. Instead, the Atari VCS is built around a custom Television Interface Adapter (TIA) circuit, originally codenamed "Stella." As a result, the earliest Atari games were subject to the limitations of the CPU and TIA chip, such that the 4 KB of addressable memory in the 6507 was originally thought to be its limit. Later, bank switching technology would expand on this capacity, but the games programmed for the VCS are driven by an aesthetic of simplicity that has its origin in the cost-cutting measures used to keep the console affordable. Accordingly, the earliest games, such as *Combat* (1977)*, Breakout* (1978), and *Golf* (1978) contain only 2KB of data, including all programming and graphics. In these tightly constrained circumstances, there is little room available for complex typography. Those 2KB games which do employ scorekeeping typically display numerals using 3x5 grids, with some minor variation. Table 3-1 compares some of these numeric character sets, illustrating similar visual created through different memory storage patterns.[12] The variations in the numerals 1, 4, and 7 demonstrate designer's individual preferences, and the identical characters which occur across several games demonstrate the constraints of parsimony (utilizing the absolute minimum detail to consistently convey the numeral) or possibly a corporate specification. In any case, even with this coarse granularity, the VCS's propensity toward experimentation and individualization is evident, but each character set created for an Atari game is a textual trace of its programmer, even if that

12 One noteworthy feature of graphics in Atari games, in terms of the relationship between storage and display, is that graphics are generally stored "upside-down" in the program listing to take advantage of a programming technique for efficiently processing the video signal. As Kirk Isreal explains, "One somewhat confusing thing is that (usually) Atari graphics are stored upsidedown in the program listing. This is because usually you have a positive number keeping track of how many lines are left to go as you're drawing the player, and this number is decreasing as you go through the scanlines. Combine that with the fact that the memory offset operation *adds* a number to the base memory location for the graphics, and it usually ends up making more sense to store the things bottom to top (Israel "Happy Face"). In other words, the graphic data is stored as a sequence of bytes, where each byte is represented as a sequence of 1s and 0s. 1s correspond to positive space; 0s to negative space. When the game program draws the screen one scan line at a time, it also proceeds through the sprite graphic one byte at a time. Since the TIA chip is capable of addressing sprites within prescribed limits of 8-pixel lines, it is convenient to decrement the number of pixels remaining, and since the code within the game which references the player sprite simply points to the address (offset) where the sprite data is stored, it makes sense to simply add the remaining line count to the offset location instead of performing a separate calculation to determine the offset.

trace is identical to other games' fonts.

Later, games with more ROM data would utilize more complex figures, as bank switching technology left more memory available to letters and numbers. Whereas a set of numerals at the minimal 3 x 5 dimension occupies only 50 bytes of memory (or 60 bytes if the figures include a blank byte for line spacing), a common 6 x 7 pattern with one byte of line spacing for each character consumes 80 bytes. 6 x 7 numeric patterns exhibit a range of variation similar to the 3 x 5 characters, but the 6 x 7 pattern illustrated in figure 3-14 is unusually consistent across a number of games, particularly those published Parker Brothers. Several Atari games from 1982 and following employ a 7 x 7 pixel grid, illustrated in figure 3-15, and many third party publishers used the 6 x 8 patterns in figure 3-16, which appears in dozens of game cartridges.

These character sets share many features, despite their being composed of differently dimensioned bitmap grids, and their designs are worth noting because they appear to invoke a common referent. In the case of arcade games adapted to the Atari console, the referent is clearly the font used in those games, the so-called "Namco" font which will be discussed in a later section of this chapter. What is significant in terms of the character sets used in Atari games is that the flexibility allows sufficient complexity that distinct patterns and styles begin to emerge. Whereas the constraint of a 7-segment figure dominates its expressive discourse, a 7 x 7 grid allows enough freedom to create distinct styles. As the numeral designs in figures 3-14 - 3-16 show, a distinct style can also represent a company or publisher identity. A non-similar design puts this into perspective.

Figure 3-17 illustrates 6 x 6 pixel bitmaps of the numbers 0 – 9 as the appear in Atari VCS game cartridges published by Mattel through their subsidiary M Network. These designs are nearly identical to the numerals used in the system font built into Mattel's Intellivision console. The only slight difference occurs with the numeral 0. In the Intellivision system font, and most of the M

Network games produced for Atari VCS, the 0 is widened to 7 pixels so that it can include a dot in its center. The specific numeral set in figure 3-17 is extracted from *Burger Time*, and its lack of a dot expresses the flexibility of the VCS as a typographic platform at the same time that the rectangular shape of the letters emblematizes the constraint imposed by a system font. Furthermore, the fact that this *Burger Time* for the VCS is based not on the original arcade version by Data East (which makes use of the "Namco Font"), but on the Intellivision version of *Burger Time* which necessarily uses the Intellivision system font. In this way, the design of alphanumeric characters in *Burger Time* provides a means for uncovering the textual history of the game, a history that additionally traces a route through the ecology of licenses in the early 1980s. Typography, in this case, becomes a tool for unpacking the formal materiality of the game text and through it gaining insight into game culture.

Intellivision

As has already been mentioned, the Intellivision console, which was released in 1979 by Mattel as a competitor to the VCS, was the first game console to include a full character set as part of its hardware. The character set, which may also be correctly referred to as a font since it is a material implementation of a design used in other contexts, includes Arabic numerals, upper and lowercase letters, several punctuation marks and a few non-alphabetic symbols such as arrows.[13] The distinctiveness of this design suggests it was part of an intentional marketing identity strategy, and its visual characteristics at first glance seem to belie the greater processing power of the Intellivision console compared to the VCS. Its rectangular forms and restriction to a 6 x 7 grid recall the much tighter constraint of the Channel F platform. On the other hand, the Intellivision font does include interesting typographic characters such as serifs on the lowercase i and a, and consistent spurs on y, g, d, p and r (see figure 3-18). These serifs and spurs, along with the

13 The complete bitmap listing for this font is included in Appendix B. Also, an Intellivision enthusiast going by the name "jaysun" has created a TrueType font based on this design, and released for free download at <http://www.angelfire.com/sc/pete2049/GSpot/Fonts.htm>.

imbalance reflected in two-story glyphs such as 8 and 2, also echo some of the De Stijl-inspired lettering H.Th. Wijdeveld created for *Wendingen*.[14] As a unique character set which can be immediately recognized and associated with the console, the Intellivision font serves as a specific holotype, suggesting not only videogames, but also a specific videogame publisher, as the M Network games published for VCS consoles demonstrate.

This retention of a particular typographic form across multiple versions of games like *Burger Time* is important because it speaks to the constitution of the *Burger Time* text in the sense that it is a single experience, unified by character, setting, and basic gameplay, manifested differently across platforms. Where those differences are a concession to limitations of a given platform, the fact that the Intellivision's platform-specificity demonstrates the kind of movement that can occur across material chains of signification such that the presence of the Intellivision font in the VCS version of Burger Time complicates the sense in which the VCS game is an adaptation of the coin-operated original, which uses a version of the Namco font. Whereas the VCS *Burger Time* could plausibly stand in for the coin-operated original if its programmer, Ron Surratt, had used the 7 x 7 design common in Atari-published adaptations, the fact that Surrat used the Intellivision font means that the VCS game becomes a signifier for the Intellivision version. The fact that the Intellivision *Burger Time* is far superior graphically to the Atari port (in terms of fidelity to the coin operated original) suggests a trajectory of degradation along the signifying chain that is consistent with Mattel's aggressive advertising campaign contrasting the Intellivision's superior graphics with those of the VCS. Additionally, the relationship expressed among these three Burger Times creates a context in which Matthew Kirschenbaum's concept of formal materiality may be said to apply across multiple instantiations of the same text. The presence of Intellivision font in the VCS *Burger Time* is an artifact of its formal materiality. While

14See, for example, the font "Wendingen," sold by Acmefonts: <http://www.type.co.uk/index.php?p=a_to_z&let=*&id=10716>

Kirschenbaum's definition of the term as "the imposition of multiple relational computational states on a data set or digital object" (*Mechanisms* 12) is referencing system-dependent multiplicity, I will argue in chapter 5 that broadening the definition to treat the social text of Burger Time as a "data object" and thus including separately platformed Burger Times as constituting computational states is in keeping with the spirit of the idea. Furthermore, the specific kind of materiality to which videogame textuality is subject moves across and among platforms in a manner that is uniquely related to technological and legal affordances (that is, in terms of intellectual property licensing). Type design plays an important role in this sequence because the graphical fidelity of the successive implementations of the Burger Time text is progressively worse; each new iteration of Peter Pepper (the game's protagonist) is, therefore, in an analog relationship with its predecessor(s) on other platforms. Significantly, the numbers in both Intellivision and VCS Burger Times are digitally identical. What differences *are* evident between the two (their physical, on-screen dimensions, for example) include hardware specific constraints beyond the platform and include factors specific to each individual instance of playing the game.

GCE Vectrex

The GCE Vectrex console is unique in many ways. First released in 1982, the Vectrex is a self-contained game system built around a CRT monitor. The game programming hardware, built around a MOS 65A02 processor, is contained within the monitor, and a combination joystick and button pad controller fits into a panel on the front of the device. As its name suggests, the Vectrex employs a vector-style display, rather than the common raster display of televisions and computer monitors. Whereas a raster constructs images by piecing together a mosaic of pixels, lighting up only those that are necessary to outline a shape, a vector display draws lines (vectors) between calculated points. The effect is to create crisp graphics that move and scale much more smoothly than their grid-based competitors. Vectrex images are still subject to the pixelization of the monitor's phosphor screen, but this effect is barely perceptible and is mostly offset by the natural diffusion of light across neighboring phosphors. Despite the advantages in terms of crispness, the main limitation of a vector system is its inability to depict surfaces as having textures, so most objects are rendered with the skeleton-like effect of creating vectors that describe only the objects edges. Color is another challenge since the Vectrex could only output monochrome lines. To offset this limitation, and to create the playing field other context-setting graphics, Vectrex games ship with a semi-transparent plastic overlay that attaches to the monitor about .3" above the surface of the screen – in other words, between the player and the game.[15]

Letters and numbers on the Vectrex present two interesting properties relevant to their textuality. First, the characters are generated from ROM data stored as bitmaps, the consummate raster method, that are then painted onto the screen as a stack of horizontal bars which complete the letter shape. Figure 3-19 pairs the bitmaps several characters from the system font[16] with their appearance on a Vectrex screen. The second important property of text on Vectrex screens is the sense in which the overlays provide a contiguous typographic surface, upon which alphanumeric figures are diegetically parallel to in-game text. Even the descriptive phrase "on-screen" is literally true both of text generated by the vector display system and of text printed on the overlay. The juxtaposition this creates is significant because it, like the bitmap to vector conversion method, demonstrates its holotypical quality through the difficulty it poses to emulation.

Figure 3-20 contrasts Vectrex numbering rendered within an emulator with similar numbering on a Vectrex. The emulator image is a photograph of a laptop screen with the Vectrex game Mine Storm running in the emulator software MESS (Multiple Emulator Super System). The laptops is running Windows XP with a resolution at 1024 x 786. MESS excels at emulating a platform's computational environment (its formal material conditions), but it has no means for

15 These overlays constitute a special, physically reified heads-up display interface, a subject I turn to in more detail in chapter 5.

16 The complete character set for the Vectrex system font is included in Appendix B.

emulating features of games that incorporate physical hardware in a haptic interface with the user. Arguably, no emulation ever achieves the full experience of playing the original, since every game console or coin-operated machine relies on some physical apparatus as part of what constitutes its medial existence,[17] but the raster grid of a standard LCD laptop screen presents a specific obstacle to the painted-in figure style. Because the position of vectors and points can be calculated proportionally, the position of a horizontal line within the figure must be rounded to the nearest available pixel line. Because the image loses some brightness at the spaces between screen pixels, the lines appear to be distorted and some are brighter than others. Also, the figures in the Vectrex image are tilted on a slight oblique angle. According to a repair manual published by GCE, this symptom may indicate a worn out chip (specifically, IC301), but like the Vectrex's distinctive hum (a consequence of an improperly shield power supply), this tilt is an endearing expression of the console's unique character and important place in videogame history. In other words, this "defect" – which MESS ignores, thereby effectively emulating an idealized Vectrex – behaves as a signifier or signature of the console's materiality.

The method by which the Vectrex renders typographic characters is also interesting because of how it differs from other vector-based games. This method is based in a system for converting bitmap information to a vector-friendly mode in which a single bit of a bitmap creates a short horizontal bar instead of a rectangular pixel. A continuous row of bars creates an unbroken line, mimicking the sweep of the scan line on raster systems. Vector display has been part of videogaming since the medium's earliest forms; *Spacewar!* and *Galaxy Game* ran on PDP microcomputers which displayed output on a vector display. *Spacewar!* lacks text, and *Galaxy Game*'s text design is unclear, but several coin-operated games released in the late 1970s employed

17 Most obviously, game consoles include unique handheld controllers, and there is currently no means for emulating a handheld controller through any haptic feedback means. Games running in MESS are interfaced with the player's keyboard and mouse, which provides a functioning if ludically impoverished experience of gameplay, but even if one were to construct a USB interface for a vintage controller, other more subtle details always elude perfect emulation.

vector graphics. *Asteroids*, designed by Ed Logg and Lyle in 1979, utilizes a vector display, as do *Tempest* and *Battlezone* (both released in 1980). However, the figures designed for these games were composed of vectors which outline the shapes of individual characters (see figure 3-21). The difference is that outlined letters are more sparse, and occupy less of the screen surface than their bitmap-approximated alternatives on the Vectrex. As such, *Asteroids*-style lettering is arguably less legible, but it is also ontologically more stable because the internal cohesion of a glyph is accomplished through vectors which literally connect with one another. Vectrex-style letters, on the other hand, require that the viewer's optical faculties construct the letters out of less visible material. Vertical contiguity is drastically undercut by the space between the horizontal vectors. Even this potential shortcoming is, however, ameliorated by the diffusion of light across the phosphor screen. This diffusion is subsequently spread further by the colored overlay, reinforcing the overlay's significance relative to the specific textuality of the Vectrex.

The problem with overlays, from a perspective of outlining videogame textuality, is that they are physical artifacts whose relationship to the game itself are something more than paratextual. Whereas it is certainly the case that some overlays, such as those employed on the original *Breakout* coin-operated game, merely provide color to otherwise monochrome delays, discursive overlays such as those used in Vectrex games serve a much more concrete world-building function. As part of a game's heads-up display, the Vectrex overlay presents itself diegetically (in the Aristotelian sense of being narrated). It is also, literally and figuratively, the lens through which the rest of the game space is relayed to the player. Furthermore, it is not simply the case that the overlay is a frame for relaying the space of the game and suturing its diegesis (in the sense of "story-world") to the reader's experience of it. In many cases, the overlay provides information that is perspectively *behind* the action of the action of the game. For example, figure 3-22 juxtaposes images of the game *Spike* viewed both with and without its overlay. As these images demonstrate the overlay supplies the background of a city skyline and combines with the generated vector image to depict the events of the game as taking place in that city Literally, the skyline is in front of the game characters (that is, between the generated image and the player), but *figuratively* it is behind them.

Within this complex interplay of layers, the presence of textual and numeric signifiers provides an additional complication; some are literally in front of the screen image, and others are literally "on" the screen, but what is the figurative, phenomenological status of these characters in terms of the composite screen image? In the *Spike* example, the numbers 1 and 2 at the top right and left corners of the screen exist are "on" the screen because they occupy the same visual field that would otherwise be visible in the absence of the overlay. The numbers themselves and the attending graphics of Spike (the protagonist) and his kidnapped girlfriend Molly are opaque, but they merge visually with the transparent "background" image of the skyline. The effect is that Spike and Molly are in the foreground of the game but are interacting with the setting by standing on something in the background, creating a complete circuit around the game image itself, which uses an oblique vanishing point perspective that points to a separate "distance" off to the right of the enclosure. Other text on the overlay provides simple instructions to the player; "Ladder / Cage," "Kick Left," "Kick Right," and "Jump" assign actions to each of the four buttons on the Vectrex joypad.[18] Like any heads-up display, this textual disposition acknowledges (in this case, grammatically) the agency of the player within the game world as well as some sense of her responsibility to this world. Unlike other HUDs, however, the overlay creates a visually interferential textuality that intercedes within the discourse space of the game console. As such, it destabilizes common metaphors about videogame form, such as the use of the screen (within game

18 The text for button one, "Ladder / Cage," is interesting because it describes the objects upon which actions can be taken, rather than a more standard "Use" button that engages non-movement actions like opening in-game doors. The presence of this text is intended to help the player understand her relationship to Spike, but its grammatic tense leaves its message somewhat vague. One might reasonably expect a ladder or cage to appear upon pressing this button.

studies and new media studies more generally) as a figurative stand-in for the boundaries of a videogame's discourse.[19] As the example of the Vectrex overlay demonstrates, the videogame screen is a complex space, and using the term as a figure for game space carries a lot of unacknowledged baggage, some of which will be examined more closely in a later section.

Famicom and others

The year 1983 is a typical landmark in historical discussions of videogames. This is the year of the great videogame industry crash, when the glut of cheap consoles on the market and the drop in consumer demand led to dozens of game companies shutting down production. The industry would limp along and later see new life with Nintendo's Famicom or Nintendo Entertainment System, but the first years of the 1980s are also a transition in terms of computational power, memory, and game design. As such the means for producing text on videogame screens becomes more standardized and, while still interesting, progressively less expressive of material affordances. Most consoles, from the Intellivision onward, included complete system fonts, and many provided subroutines by which programmers could load their own custom fonts along with game data. This streamlining was also enhanced by implementations of ASCII or similar standards such as ATASCII.[20]

Other systems in the videogame console generations following the crash contain peculiar typographic artifacts and textual dispositions, which are worth mentioning. For example, the ColecoVision, first on the market in 1982, offered greater fidelity to the arcade originals it ported to its system than either the VCS or Intellivision (Forster 51). It did include a system font very similar to many coin-operated games (not the Namco font, however), but an alternate BIOS

19 James Newman's use is typical: "In games like *Tomb Raider* or *Super Mario*, just as in Friedman's *Civilization*, the primary-player may not see themselves as any one particular character on the screen, but rather as the sum of every force and influence that comprises the game" (Newman, "The myth of the ergodic videogame").

20 The Atari 400/800 line of personal computers, for example, included a custom character set and a built-in table for addressing text. Taken together, this font and table of letters (shown in figure 3-23), make up ATASCII. As a font, it has some features in common with the Namco font, which it may have indirectly inspired, as discussed below.

includes a rather different font that is wider and dramatically stylized.[21] The origins of the alternate font are not clear, but because it is integrated at level of the system ROM, games can be emulated using either the original or the modified BIOS, yielding otherwise identical games with very different typography (see figure 3-24).

Also, Atari's 8-bit family of personal computers, which introduced the ATASCII character set, have the ability to make use of its character set handling to create graphics. The system's character set is defined as a table of numbers which reference bitmaps stored in memory, and a subroutine makes it possible for programmers to design their own font to replace the original. This has obvious aesthetic or typographic appeal, but Chris Crawford, in his classic handbook, *De Re Atari*, takes it further to describe the many advantages of this technique, noting that its ability to change character sets while a program is running allows programmers to create simple animations out of redesigned characters. In terms of memory use, this method consumes less RAM than would be the case with straightforward bitmaps. As a result it is conceivable that an entire game could be programmed for an Atari 400/800 (or, possibly, comparable 8-bit computers like the Commodore 64) that generated graphics entirely through the character set. This is similar, in a formal material sense, to the nature of graphics in "text mode" games like *Rogue*, but in the context of videogame textuality, it is also worth noting the relationship this establishes between text and image in the videogame environment. The fact that any given visual object within a game rendered by these techniques could be (in an invisible modality) text, undercuts the argument that image and text are separate paradigms of representation within the videogame. At the very least, this calls attention to the visual nature of typographic characters that is similar to the diegetic lever supplied by the Vectrex's overlays. While the question of representational mode does depend on form and access to specific orders of symbology through the senses, the fact that the digital ontology of images and

21 The origins of this alternate BIOS are unclear, but it is readily available from hobbyist websites. It is possible that the font was introduced in one of the licensed ColecoVision derivatives such as the DINA or Telegames Personal Arcade. Both BIOS fonts are included in Appendix B.

text are indeterminate in these environments is an especially appropriate disposition for the already imagetextual modality of videogameplay.

Videogame console generations nearly always emphasized the system's graphical rendering power or processing speed, so even though most systems resemble contemporaneous personal computers in many respects, text handling is generally an afterthought, if it is mentioned at all. One interesting analysis of text rendering as a comparison point for consoles come in a review of various versions of Super Mario Brothers. Comparing an original NES, the Generation NEX and FC Twin NES simulators, and the Virtual Console included within Nintendo's Wii, Ben Kuchera of *Ars Technica* writes,

> The text test! The NES looks okay, nothing great or terrible going on there. The Generation NEX holds up well, although you can see an issue or two with the question-mark block. More about that in a second. The FC Twin does very poorly with text. It looks terrible in pictures and doesn't get better since the text never moves in the game. Ick. The Virtual Console makes everything pretty though. (Kuchera)

What is interesting here is that, in terms of text legibility, the Virtual Console is found to be a superior rendering of *Super Mario Brothers* than the original console for which it was programmed. From the images he provides, the difference appears to be largely the result of the author's HD TV compressing the image in different ways, but the criticism of the NES here demonstrates that there is more impacting the appreciation of videogame lettering than fidelity to some original, and legibility is an important concern. Still, the dimension of holotypical referentiality through typeface design cannot be ignored, especially in the case of *Super Mario Brothers*. The system font for the NES console shares many distinct features with the "Namco font" then ubiquitous in arcades.[22] This in turn was based on a character set produced by Atari in the 1970s, so the presence of certain letter shapes in Kuchera's example references an idealized typeface which is not imagined but implemented on a broad scale (with slight variation) on dozens if not hundreds of game machines. The comparison in which the NES falls short is not, therefore, a

22 The complete character set for this font is included in Appendix B.

case of a simulation preceding its simulacrum, but a signification system in which the referent of the type design and rendering is the textuality of the the videogame industry.

Holotypical Forms

In the previous sections, I have organized my discussion of videogame textuality around specific platforms and their typographic affordances. In the next section, I turn toward broader, cross-platform patterns in type design for videogames. Because videogame systems tend toward greater flexibility in terms of type design, and therefore move away from the machine-specific constraints which characterize holotypical typographic expression, the letterforms which are adapted across platforms – sometimes retaining evidence of prior constraints – warrant their own analyses. In this section, I consider the ontological distinction between the 7-segment figure and the similar 3 x 5 or 3 x 6 figures which appear on numerous platforms. I then proceed to discuss dedicated character generator ROM chips that appeared in several gaming and computing systems, and I conclude with a discussion of the so-called "Namco Font."

The 7-segment form

The first style of numeric character rendering to appear in a consistent way on videogame screens was the 7-segment figure. These figures could be generated relatively easily with the discrete logic circuitry employed in most video and arcade games before cheaper programmable CPUs such as the MOS 6502 were widely available. For example, *Computer Space* employs this kind of character generation by a special circuit built specifically for this purpose. As shown in figure 3-4, this display reports the player's score (the label "Rocket" is printed on the cabinet adjacent to this display field), the computer opponent's score ("Saucer"), and the amount of time or fuel remaining in the current game ("Time"). The 7-segment form also commonly appears in vector-based games, like *Asteroids*, that utilize the outline method for drawing alphanumeric characters (as opposed to the Vectrex's quasi-bitmaps). Considered as a general form, therefore, the

7-segment form is appropriate in many contexts where constraints mandate parsimony, and in the sense that it is composed of modular segments that can be arranged in different formats to produce different figures, the disposition of the form is aesthetically digital. In other words, there is no necessary relationship between any given segment and a number form it is part of, other than the one-bit property of whether that segment should be “on” or “off” within the requested figure. This digital logic, however, is not necessarily computational or even electronic in origin.

The 7-segment form is today most commonly seen on digital display devices like wrist watches, and it is also common in non-electronic displays such as the marquees used in billboards to update the price of gasoline. In those non-illuminated displays, states of “on” or “off” managed merely through the presence or absence of colored bars. Still, the method for displaying the form do originate with illumination, at least as early as 1908 with a patent for an “Illuminated Announcement and Display Signal.” This device, invented by Frank W. Wood, selectively completed or opened electronic circuits connected to specially placed electric lamps. Since the patent includes the method for arranging these lamps as part of its original claims, Wood’s is likely the original implementation of this method, and the description of the need for the digital nature of this invention:

> In the numerous attempts which have heretofore been made to produce a signal display system of this character, one of the principal difficulties encountered has been the necessity for providing a very large number of electric lamps, and hence a corresponding number of independent circuits, for each letter, figure or character produced. The primary object of this invention is therefore to reduce to a minimum the number of lamps employed within a given field to produce the various characters required, and to this end I form such characters by means of certain elementary blocks of light so constructed and arranged that the characters will be clearly and distinctly outlined. (Wood 3)

In other words, whereas prior devices created figures through unique arrangements of lamps for each number or letter, Wood’s invention used a single arrangement that could emulate others in a manner convincing enough to be legible at distances. Thus, whether the base units of a 7-segment form are lamps, vectors, or raster beams, their innovation is their ability to be arranged into something else, and their unit operational structure is a function of the sub-units formal materiality. For example, in a later invention for illuminated display, the figures (in this case, both letters and numbers) are composed from illuminated tubes. This device, invented in 1937 by Thomas Ross Welch, utilized a much larger set of segments, but it expressed the modular logic of Wood's device to an even greater degree:

> It is an object of the invention to provide a sign comprising a desired number of letterformers, each of which letterformers consists of an arrangement of glow tubes, for example, neon tubes, so placed and connected with a simple control means that selected sections of these glow tubes may be caused to illuminate, thereby presenting to the eye of an observer a desired character or symbol. (Welch 1)

In this way, although the controls of the device are technically analog, its operating principle is, in essence, digital, and it relies on the digital property of alphabetic writing in which letters can be arranged into words independent of individual signification. Welch's sign composes letters and numeric digits out of discrete and modular segments (any combination of 26 "letterformer" segments could be selected) and displays them in a sequential manner controlled from electrical input. As a principle for organizing character-generation, the flexible, modular nature of these "letter-formers" illustrates the same principle of parsimony and modularity appropriate for game devices. It also demonstrates one approach to the problem of designing alphanumeric glyphs from uniform, discrete components that are geometrically generic. As a theoretical font, therefore, Welch's electric sign argues for treating the square and the circle as the fundamental unit of letters and numerals (see figure 3-26).

In early videogames discrete circuitry stores information and generates imagery by manipulating the pathways and waveforms of electrical current through series of logic gates. David L. Heiserman's 1978 textbook, *How to Design and Build Your Own Custom TV Games*, describes the importance of numeric information and the basic method for generating the digit display, which relies on two essential circuits:

> Virtually all fast-action video games call for automatic scoring, and of course it is nice if timed games have some provisions for displaying the elapsed time or time remaining of the play. The circuitry is practically identical in either case, a control circuit that generates binary numbers for scoring or time and a display circuit that generates the appropriate numeric figures on the screen. (Heiserman 381)

Figure 3-27 is Heiserman's illustration of a 7-segment display circuit and figure 3-28 is its accompanying BCD (binary-coded decimal) to 7-segment conversion table. This illustrates the standard usage whereby the lowercase letters a, b, c, d, e, f, and g refer to the seven segments in sequence. The segment a, for example, is at the top of the figure. In practice, the circuit works by assigning 4 digit binary input (measured in this case by electrical current through 4 input pins labeled A, B, C and D) to the proper combination of seven output pins, passing current (or none) as indicated by the logic in the table. This table refers specifically to the table employed by the SN7448 chip, manufactured by Texas Instruments, which would work in combination with other chips on a game circuit board. *Computer Space*, for example, stores each of the numeric values related to gameplay (score, time, etc.) in individual chips, which then pass the correct set of signals – 4 separate current paths to create a 4-digit BCD value – to the display circuit.

The logical states which produce these images do so as a condition of a logical state for the machine as a whole, which is stored and maintained throughout play as the specific path of electrical current through the sequence of chips. Like the lamp-based segmented displays, the 7448 chip converts discrete and non-signifying units into a cohesive whole, such that there is an arbitrary relationship between any individual unit and any specific figure including that unit. With regard to ontology, the 7-segment display generated by the 7448 chip has more in common with Wood's illuminated display device than the vector-based systems, which create segments by connecting points located geometrically and tracked in software, because the 7448 relies on physical wiring and a particular state of energy distributed through that wiring. The logical interpolation which occurs between BCD and 7-segment has a physical substance that upholds its formal properties. Software-based systems, such as bitmaps, support these formal characteristics through potentially several layers of abstraction which convert ROM data into the visual form of numbers. The method by which stored information (a player's score stored in binary data) becomes visual information data (score displayed on screen) is a shift in which the arbitrary relationship of signifier (for example, a 3 displayed on screen) to signified (the quantitative record of 3 prior scoring events) is realized in a sense that is materially different from bitmap storage and retrieval, where the relationship is ironically more analogous in nature.

Character generator ROM chips

As integrated circuit technology condensed many routine computing tasks into cheaper, self-contained units, several chip manufactures produced chips with the dedicated purpose of generating text. Like the integrated circuits which were capable of generating complete PONG-like games, these character-generating ROM (Read-Only Memory) chips would be included in systems solely for that dedicated purpose of drawing text characters on the monitor or television screen. These so-called character generator ROMs generally contain a 128 character set that could be accessed via standard ASCII codes or some variation or subset. For example, in the early 1970s, Signetics produced a chip, designated 2513, which is capable of generating 64 characters – capital letters, numerals, and a few punctuation and mathematical symbols. The characters themselves are generated out of a 5 x 7 pixel grid, so the resulting forms are of a similar density to the ones appearing in Atari VCS games (see figure 3-29). Mitchell Waite, in his *Computer Graphics Primer*, describes the 2513 as the "most coarse" and "cheapest" character-generating ROM, suggesting that glyphs of the 3 x 5 and 5 x 5 density are not worth installing a dedicated chip. Whether grids of this density are considered "coarse" or just highly stylized, more expensive (and expansive) character-generator chips continued a trend toward mimicking more curves and making more gestures toward emulating print typefaces.

The MCM6570 chip, produced by Motorola, provided the full 128 characters of the ASCII standard, providing full upper and lower-case, numerals, punctuation, mathematical symbols, and several Greek characters. Furthermore, one advantage of the 6570 is that it allowed users to program their own fonts into the chip, after creating the font information and coding into IBM punch cards or ASCII Paper Tape Punch (Motorola). To avoid this time-consuming step, Motorola provided a series of chips which came pre-programmed with a font of characters, including standard 128 character ASCII as well as some interesting variations. These characters are generated out of a 7 x 9 pixel matrix that allows for a wider variety of character designs, and the datasheet for the chip includes 9 such variants. Some of the variations change the function of the chip entirely – MCM6573, for example, includes Roman capital letters and a set of Kiragana – but other variations are more subtle (Motorola). In their 1978 book on building "TV games," Walter Buchsbaum and Robert Mauro make use of the MCM6576, which comes programmed with a typical serifed design. Mitchell Waite, on the hand, uses the MCM6571 as his example, which contains a sans-serif font and Greek letters. Figure 3-30 illustrates some of the differences between the two chips.

What these variations within the constraint of a 7 x 9 grid demonstrate is the degree to which a distinct style and expressive pattern can be achieved – even with this rather coarse environment. In terms of the resulting textual ontology for devices that used these chips, the fact that character data exists in read-only memory in physically separate chips implies that the device in question is not fundamentally textual in the same sense that a system with integrated fonts. The ROM data is logically and programmatically the same as character ROM stored within a game or a system's BIOS, but because the resulting character designs themselves are not even system-specific, the device or game text as a whole is less autographically textual. Unlike systems with BIOS-embedded fonts or which required game programmers to come up with their techniques for creating text at all, the computer and gaming systems which made use of 2513 and 6570 chips do not reveal the same interrelatedness of visual and verbal textuality, nor do they encourage the interpretation that the constraints and affordances exhibited in typeface design can be emblematic of a particular device's general aesthetics.

The "Namco" Font

One final example of videogame holotype addresses the diffusion and influence of type design across multiple platforms. The so-called Namco Font, as shown in figure 3-31 on the attract mode screen for *Ms. Pac-Man*, has already been referred to as a possible source for the common 7 x 7 number style of Atari VCS games, and (less so) the system font of the Nintendo NES, but the font now widely associated with arcade gaming saw its widest dissemination in Namco arcade games produced during the 1980s. Figure As a distinct character style, versions of this font often appear in contexts which reference videogame culture, and one popular TrueType font based on this design (Joystix, by Ray Larabie) is distributed on websites like MyFonts.com with keywords such as "arcade," "computer," and "videogame," as well as the names of specific games where the typeface design it emulates appeared – *Frogger*, *Galaga, Galaxian*, and *Pac-Man* (Larabie).

Information about this design has been difficult to locate because it is so prevalent on arcade systems – particularly those produced in Japan in the early 1980s – that its origins and evolution are difficult to trace. One massive resource for studying this font has been assembled by a Japanese hobbiest who signs his or her name only as "qtchicks." He or she has assembled a massive web resource containing images of all fonts included in Namco games, which qtchicks refers to as the Namco Font Museum (qtchicks).[23] What is remarkable is how little the font changes over its many iterations; the major improvements are in offering multiple colors and alternate embossing or

23 The full title is "NAMCO '70~'80's ARCADE VUDEOGAMES [sic] FONT MUSEUM\," and it is available at http://qtchicks.hp.infoseek.co.jp/fonts.html. The text is all in Japanese, but automatic translation software makes it navigable, although some web pages contain an encoding error. Setting the brower's coding manually to "Unicode (UTF-8)", thereby overriding the assigned "Japanese (Shift_JIS)" encoding, will ensure that the text can be interpreted properly. Throughout the Museum, qtchicks refers to the font as "Namco Font," so I have adopted that usage here. Whether it ever had an official designation is presently a matter of speculation.

shadowing effects. This relative uniformity suggests that the font was simply copied from one game to the next. Whether this was simply a relatively easy solution to an otherwise tedious task, or whether it was required as a company policy at Namco (or the other arcade game manufacturers who used it, including Bally / Midway and Taito), this exchangeability exerts an intriguingly allographic propensity on arcade game textuality: within the paratextual domain of videogames, the font is invisible, but in any other contexts, it immediately suggests videogames. With regard to expression, therefore, Namco Font's relationship to the formal materiality of arcade game hardware extends the holotypical referentiality of videogame textuality.

According to qtchicks, the Namco Font was first created by Atari. In 1974, Namco purchased the rights to distribute Atari games in Japan, and in 1978, Namco entered the arcade game market itself with their *Breakout*-like game *Gee-Bee* (Kent 76). *Gee-Bee,* gameplay of which involves removing bricks from walls by bouncing a ball with a paddle, has much in common with *Breakout*, which may be explained by the fact that Namco had released a port of Atari's *Super Breakout* earlier that year. As qtchicks explains, Namco engineers simply borrowed the character set from *Super Breakout* and applied it to *Gee-Bee*, adding several colors and inverted bitmaps for each character. From there, it quickly saw use in games by other Japanese game manufacturers, including Taito and Sega, but its existence before *Super Breakout* is unclear.

The story of the origins of the first *Breakout* game is one of the most well-known pieces of videogame lore,[24] but *Super Breakout* was one of a large number of sequels and derivative titles Atari Games cranked out in the late 1970s. According to James Hague's "Giant List of Classic Game Programmers," *Super Breakout* was programmed by the prolific Ed Logg, who would later produce the classics *Asteroids* and *Centipede* (Hague). Where Logg acquired the design is

24 Steve Jobs, who would later go on to found Apple Computers, worked as an engineer at Atari. Offered a chance to receive a bonus for completing a game within 4 days, with an added incentive for creating it with as few TTL chips as possible, Jobs farmed most of the work to his friend Steve Wozniak, who completed it in the alloted time. Jobs shared half of the bonus with Wozniak (in the amount of $350), but did not tell Wozniak about the extra bonus earned for reducing the machine's cost (on the order of $4000).

uncertain, but the font in *Super Breakout* does have some features in common with some lettering designs in Atari VCS games of 1978. Only a few games include full alphabets, mainly owing to the large memory cost required to craft an entire alphabet, and those which did were likely to be educational in nature. One such cartridge, designed by Warren Robinett, was BASIC Programming, an instructional cart meant to introduce players to computer languages. BASIC is written in human readable language, and one of the cartridge's advertised features was that with the aid of this cartridge, " ... you'll have the VCS game printing your messages..." (Atari 5), a full alphabetic character set was necessary.

Warren Robinett, who would later create the easter egg in *Adventure*, recalls that he first had to address the problem of putting text on the screen. In a personal e-mail to me, Robinett relates the story that fellow programmer Bob Whitehead laid the groundwork by coming up with a way to put 8 "characters" per line on screen to create a chess game. (In this case, characters were chess pieces). Dave Crane adapted this method to create a character-generating kernel that could output 12 characters per line, and Robinett simply copied Crane's code, including the character set (Robinett, "Re: Question about Basic Programming"). [25] There are some differences between Robinett's BASIC character set and Namco Font, which are compared in figure 3-32, which suggests the possibility of a third source. However, some key features are present, such as the subtly humanist axes in two-story figures like 2, 5, and S. Other identifying features are absent, however; BASIC lacks the humanist 0 and 8, for example.

Whatever its true source, Namco Font's popularity owes to more than simply utility or legibility. It exudes a clear style that, though inflected by the rigid 6 x 8 grid that encloses it, still manages to exhibit the grace and balance of a typeface defined for print. In its TrueType

25 The only Atari VCS game from this era credited to Dave Crane that includes a full alphabet is an unreleased prototype for a *Boggle* game. However, the characters in this game do not match Robinett's in BASIC Programming, so it is possible that Crane used a different version of his kernel, or that Robinett used a different source for his characters.

incarnation as Joystix, it retains the bitmapped jagginess because this is crucial to the harmony of the forms. As a videogame holotype, therefore, the font remains distinct from similarly constrained forms which have no historical association with videogames.

Videogame Grammatology

This chapter has so far discussed the nature of type and textuality within videogames, but has so far only implicitly addressed the question of what text in videogames does. This study is based largely in the premise that videogame textuality consists of a unique type of expression, depending on medially situated and discrete layers of representation specific to technological arrangements and autonomous, rule-structured environments. In this case, textuality is the broad sense through which videogames compose themselves as texts, and videogame typography is the specifically alphanumeric dimension of textuality manifested in letters, numbers, and their positions on the screen. What remains assumed and unaddressed, however, is the degree to which videogames may be considered inscriptive media artifacts: how and in what way are videogames subject to the institution of literacy and inflected by the history of writing technology? Addressing these question are three phenomena endemic videogameplay: displaying score on the game screen, "easter eggs" messages hidden within games, and the figural space of the screen itself. The remainder of this chapter briefly addresses each of these.

The Keeping of the Score

In examining the materiality of digital texts, Matthew Kirschenbaum offers a provocative grammatology of the hard drive as an inscriptable technology and site for writing. In addition to considering prevalent metaphors, such as the idea of "writing to a drive" to reference saving data, Kirschenbaum is also interested in the physical properties of drive technology which allow information to be stored and retrieved by appropriately named read/write heads within the hard drive enclosure (Kirschenbaum, *Mechanisms* 70). Ultimately, Kirschenbaum proposes a series of "grammatological primitives" on which the hard drive lies. With regard to the material trace of writing, therefore, a hard drive is random access, a signal processor, differential, volumetric, motion-dependent, planographic and non-volatile (Kirschenbaum, *Mechanisms* 89). Each of these principles addresses formal and forensically material properties of a specific technology, but since videogames depend only on technological conditions in the broad sense (that is, there are many different technologies which can create the conditions for considering a text or interactive experience a videogame), a different approach must instead address the construct of the videogame as a medial disposition general to computational environments, while still incorporating the inflections of those technologies specifically created for playing videogames.

As discussed earlier, the first uses of mixed text and image on screen during gameplay occurred in the form of displaying a player's score. Significantly, games like *Galaxy Game* and *Computer Space* included scorekeeping and display, whereas their formal (and in the case of *Galaxy Game*, programmatic) predecessor *Spacewar!* lacked such a display. Similarly, the first generation version of the Magnavox Odyssey lacked the ability to display score on screen, relying on the players to track the score themselves. This externalization of scoring is in keeping with other games on the console that required the use of playing pieces such as cards or poker chips, because the formal conditions which maintain the state of the game are upheld by the players in a sense which marks gameplay on the Odyssey as something fundamentally different from gameplay on other devices. *Spacewar!* and Tennis on the Odyssey share the common element that an instance of play on the devices was initiated by social conditions specific to the play event. *Spacewar!* was originally conceived as a hack to demonstrate the capability of the PDP-1 computer, and Odyssey Tennis was marketed as a TV Game – an extension of the television around which the family could gather for a game night. Both imply conditions of congeniality which are not assumed by *Galaxy*

Game and *PONG*, both of which were played in public and, significantly, required money to initiate an instance of gameplay. Play is, therefore, placed in a monetized framework.

More importantly, the equity of the now-monetized framework is signified by the intervention of numerals displayed on the screen that provide a textual intervention into the diegesis of the game world, thus maintaining the temporal heterogeneity of an instance of gameplay. In *PONG*, for example, the play field is bound by the edges of the screen (a player scores when her opponent fails to heed the advice printed on the cabinet, "Avoid missing ball for high score") and a scoring event occurs when the ball passes beyond the edge of the screen behind either player. Horizontal lines at the upper and lower edges of the play field describe the boundaries of play, but the score displayed at the upper edge of the screen exists within the world of the game as part of its simulation. Diegetically, it must exist at one remove from the field of play, but it is part of the mediated experience of the text and underscores the practice common of early systems to describe what is now commonly labeled a game a "game simulator." In relation to the player's experience of the game world, therefore, the significance of the displayed score is that it visually and logically separates instances of play from past and future instances of play, inserting a textual displacement of temporality that diachronically isolates a play event that is otherwise synchronically and diegetically identical to every other instance of play. In other words, when the ball fires to begin a volley in Odyssey Tennis, that play field and the rule-conditions which define the player's agency within that magic circle are identical to any previous volley. There are, of course, differences at a phenomenological and microscopic level (degradation in the device's circuitry or increasing burn-out on the screen's phosphors), but in formal terms, the diegetic comportment of the game world is unchanged. In *PONG*, however, each volley is marked with the traces of prior volleys. In figure 3-33 a score of "11 | 7" says, "there have been 18 prior volleys; in 11 of these, player 2 failed to prevent the ball from exiting the screen on her side; in 7 prior volleys, player 1 failed to prevent the ball from exiting the screen on his side." These statements issue from the the game world and address the formal, ludic conditions of play based on a rule set that is enforced by the circuit logic of the game machine and a historical record maintained by the path of electrical current through a sequence of transistors. The signifiers are addressing the numeracy of the game within a syntagmatic ontology of a game of *PONG* by producing typographic interventions at the boundaries of play – both the visible boundaries of the screen and the formal boundaries of the game space.

The score display in *PONG*, *Galaxy Game*, and *Computer Space* are important because they initiate this textual structure, but games continue to exhibit similar textual structures that report on the status of the game. This gesture toward the outer world serves as a discursive act that, by taking the form of alphanumeric text, writes the game's present status into the world. The game itself is internally coherent, so long as the text display reliably records and displays play sequences, so its gesture to the outer world serves the additional function of reinforcing its boundary. Jesper Juul addresses the question of the boundary between game worlds and real worlds by framing it as a question of coherence, arguing, "It is a hallmark of a coherent world game that the bounds of the game space are reasonably motivated by the fictional world" (Juul, *Half-Real* 166). Additionally, although Juul is not specifically focusing his argument on textual dispositions in games, he seems to at least tacitly acknowledge the border-making function text can provided – even if that is a concession to a failure in the game's diegetically presenting its boundaries:

> The relationship between game space and world space becomes more interesting in games with more elaborate fictional worlds, where the end of the game space has to be marked in more subtle ways than by using a white line or a wall. In *Battlefield 1942*, the player approaching the edge of the game space informed by a **textual** message, "Warning! You are leaving combat area. Deserters will be shot." This is known as invisible walls: The fiction gives no clue that the world ends, but for no apparent reason, the game space ends. (Juul, *Half-Real* 165)

Juul's comment is addressing two kinds of boundaries: the fictional boundary of the battlefield, and the material boundary of the battlefield's programmed geography. Though, for Juul, the fact that this boundary is erected textually is incidental, this presentation of the relationship between the player-character and the game world through alphanumeric signifiers performs the same world-building function as the score and *PONG*, though with the addition of a spatial dimension. In both cases, the act of inscription is performed by the game world, and it sustains the persistence of the game to the world around it.

The practice, first appearing in arcade games meant for public places, of recording and displaying high scores further displaces the textual locus of the game into the world of the player. A high score occupies an interesting position with regard to the game text; depending on the game in question and how one defines the textual domain of a videogame, the high score may be considered epitextual (in the case of games which display a high score table while in attract mode) or peritextual (in the case of games which include a high score from a prior game on screen during the current gameplay). As a form of inscription, the high score negotiates presence and absence in a more traditional manner, signifying not only the existence of a prior instance of play (thus extending the syntagmatic dimension of play), but also the fact that some now absent individual played the game and may have recorded her initials in the high score table. In the case of high score, the written record exists beyond the boundaries of a specific instance of play, and in the case of a high score table, it may even be used as an enticement to other players – offering the challenge of placing one's own name in the hall of fame. But if the concept of what constitutes a videogame is expanded to include the machine and programming that bring it into existence (its medial substrate), then high score functions similar to in-game scoring by broadcasting the present state of the game world, in the form of the current best performance, to the world around it. Where players have the option of entering their initials, both game and player participate in acts of inscription that extend the text of a specific game session by creating a record of it.

This practice continues in contemporary games, but it is enacted in different forms and by different means. In one interesting case of the written word standing in for displaced presence, online multiplayer games (in which players participate online by connecting to a central server) collect and record statistics on player performance, which may be displayed on dedicated tracking websites. In first-person shooter games like *Call of Duty*, for example, the server tracks information about each player's performance, including that player's number of frags (kills) per game and more skill-specific details such as which weapons they have been fragged by the most often. Third party software then collates this information into profile pages that can be displayed on a web page or a so-called banner image that the player can insert into his or her signature on messageboards. These banners display basic information about the status of the server, the player's recent performance, and whether the player is presently playing a game. Figure 3-34 illustrates statistics banners for two players, inhumanplumber and molecularr. These images are generated dynamically from the third party service, Game Tracker, so the information on these images reflects the status of the server and both players at the time I saved the images to my computer, April 27, 2008, at 4:39PM ET and 4:42PMET, respectively. Inhumanplumber is one of the top ranked players on the server, and this banner indicates that he or she was playing *Call of Duty* at 4:39PM on a Sunday and that he or she was winning. Molecularr, on the other hand, is a less experienced player whose rank on the server indicates that he is only an above average player and that at 4:42PM on a Sunday, molecularr was not playing *Call of Duty*.[26] In both cases, up to the minute information about the game world is made visible for others (potential players, most likely) who are not presently participating. Like the score displayed at the top of a *PONG* screen, these Game Tracker banners report the moment by moment state of the game, and like the high score

26 Molecularr is, in fact, the screenname of the author.

table displayed on a *Ms. Pac-Man* cabinet, these banners provide evidence of prior states of play with reference to the presence or absence of specific players.

The act of entering one's name (or screen name) in connection to an instance of play creates an autographic record of play and presence that connects the formal materiality of the game with the forensic materiality of a specific session of play. Formal materiality is invoked because the graphic of one's score becomes part of the formal context of the next game, especially if one has achieved the highest score on a game like Space Invaders (Figure 3-35) where the high score occupies part of the screen during gameplay. In terms of forensic materiality, the high score record creates a unique marker of a specific person's presence. This inscription can be considered forensic (after Kirschenbaum's use) because it references the individual uniqueness of a specific media artifact as opposed to digital information's symbolic ability to be copied to other media. In another sense of the term forensic, the high score record provides evidentiary data about the activities of individuals with regard to the game. In two infamous cases of player's who died while playing the videogame *Berzerk*, their last recorded high scores have become part of their stories. In 1981, 19 year-old Jeff Dailey suffered a heart-attack after posting a score of 16,660 (Vasvari & Kirmse), and Peter Burkowski similarly died from a heart attack following an intense session at *Berzerk*. As a contemporary newspaper report puts it, "in fifteen minutes of play, he wrote his initials at least twice in the 'Top Ten' on the *Berzerk* screen" (Kiesling 14). The use of the word "wrote" in this account is worth noting because the autographic significance of Burkowski's having been present to create the record overrides the technical detail that no literal act of writing occurred. Burkowski instead manipulated the joystick and buttons on the screen in order to select his initials, but the act of writing here refers to the broader sense in which Burkowski created a specific state of the game world that was manifested in a special pattern of electricity passing through transistors and integrated circuits and distinguished formally from other games played on the same machine by attaching his name to that unique disposition of game state and electrical current. As a motivation and means for inserting textual information into videogames, score display and high score reporting form a typographic conduit into the logic of a videogame's core textuality and, therefore, constitute a crucial example of holotypical text.

Easter Eggs

The practice known as "easter eggs" is another form of videogame textuality which has an important signatory valence. In its most common use, the term easter egg refers to instances where some information is hidden with a game in such a way that players can only access it using special knowledge or skills. Typically, these are humorous or self-congratulatory in nature, which is consistent with the fact they exist somewhere between the formal world of the game and the medial, software artifact of the game as a set of code or instructions. Conceptually, easter eggs are related to hacks, mods and fossils, but one important difference is that easter eggs reflect some sense of intention on the part of the programmer and as such represents a more or less direct communication from the programmer to the player, employing the formal space of the videogame's fictional construct as a medium. Hacks and mods, on the other hand, typically involve a player directly altering game code, and fossils are typically left over pieces of code which were not meant to be found.[27]

Warren Robinett's easter egg, hidden in the Atari VCS game, *Adventure,* is one example of videogame holotype that captures the sense of a platform-specific constraint as well as the autographic nature of a digital record. Robinett's game was based on the text adventure game *Colossal Cave Adventure*, which included a the "magic word XYZZY" written on the wall of a cave. Like *Colossal Cave*, Robinett's *Adventure* included graffiti on its cave walls – in this case, the phrase "Created by Warren Robinett." Intrepid players can access this room by retrieving an

27 Ruffin Bailey analyzes these interrelated phenomena in his essay, "Hacks, Mods, Easter Eggs, and Fossils," in which he argues that the digital substrate of videogames reveals clues about their design which in turn provides access to authorial intention in a manner and degree unique to the affordances of digital information storage.

invisible dot whose presence in a specific room causes one of its walls to become permeable. This difficult-to-access image, shown in figure 3-36, is often reported (possibly inaccurately) as the first instance of a game designer including hidden content within a video game. The fact that he chose to leave his name, and his reasons for doing, so invoke the sense of an author's manuscript suggested by the term *holograph,* which is discussed earlier in this chapter as one valence of the term holotype.

The story of how Warren Robinnett came to leave his signature on a wall in his groundbreaking Atari VCS game *Adventure* is a legend of early videogame history, and it illustrates the inscriptive nature of the easter egg concept. With this game, Robinnett achieved several important and influential milestones,[28] but programmers working for Atari at this time (1978) did not receive any public recognition for their work, which was one of the factors later leading several disgruntled former Atari programmers to found Activision. Robinnett also took matters into his own hands, altering his game design to serve this purpose of providing recognition. In his words, "I created the secret room in order to hide my signature in the game" (Robinett, "Foreword" xviii). Considering this signature image as an autograph within the formal boundaries of the game diegesis, the applicability of the term *holograph* here refers to the fact that Robinett worked alone to create the design, programming code, and graphics of the game, so his signature as an auto*graph* completes the game's totality and argues for its status as a videogame holograph or single author manuscript. As an example of diegetic inscription, the signature itself becomes a videogame holotype both for its recognizable aesthetics and for the way it operates within the game as the ultimate final goal for the player to achieve.

Significantly, the image of the signature itself is not static, but rather it pulses with the same animation effect (rapidly rotating its color through the full pallet) that also highlights the game's

28 These innovations include navigable space organized in a room-to-room mode and the ability to carry an inventory of objects.

ostensible objective, a chalice guarded by three dragons. Object 3-1 is a short video clip of the animated easter egg; object 3-2 shows the same animation in effect on the chalice. The reason for using this animation is in one sense, to accomplish an aesthetic aim: in Robinett's words, "Once into the secret room, well, hey, why hold back at that point? My name filled the screen like a throbbing, multicolored movie marquee" ("Foreword" xviii). But it also links the signature to the chalice in an important way that distinguishes Robinett's signature from other, possibly earlier, examples of in-game credits.

Notwithstanding the historical and cultural significance of Robinett's easter egg, there is some evidence that his may not have been the first appearance of a digital signature within a console videogame. The Fairfield Channel F was a cartridge-based device first released in 1976, predating the Atari VCS by at least one year. Like the Atari, the Channel F relies on an 8-bit CPU (in this case, a Fairchild F8) and plays games through programmable ROM data loaded by cartridges. The Channel F never achieved the same popularity as the Atari console, but it achieved several firsts, including what may be the first appearance of videogame easter eggs in console games, which include three instances of a programmer leaving his name somewhere in the game. Due in part to the relative obscurity of the console and the somewhat arbitrary means for accessing the hidden content, the Channel F easter eggs have apparently gone unnoticed until relatively recently. The first, discovered by Channel F enthusiast and hacker Sean Riddle, can be found in the game's demonstration cartridge (Riddle, "Channel F info"). This cartridge shipped with the console, and it walked players through a demonstration of its complex controller and setup menus. At the conclusion of the demonstration, if the player simply holds down buttons 1, 3, and 4 part of the text for the final screen of the demo is replaced with the text, "MICHAEL K. GLASS" as shown in figure 3-37. The other two, also uncovered by vintage game hobbiests, are more difficult to access. In the game Video-Whizball, one must play through an entire game on any setting, and then start a new game, specifying game type "43" and setting the maximum score to "67."[29] Doing so reveals the text, REID-SELTH, visible in figure 3-38, the "signature" of programmer Bradley Reid-Selth. Finally, the game *Alien Invasion* (a *Space Invaders* clone) includes a similar credit, also revealing REID-SELTH after the proper setup sequence is executed (see figure 3-25).

The question of whether these Channel F Easter Eggs came into being before Warren Robinnett's is largely semantic (depending on one's definition of the term), and making an argument either way requires relying on some vague chronology. Bradley Reid-Selth has even been quoted that he learned of the easter egg concept by hearing that Atari and Activision programmers were creating them. Whatever the case, the main difference between Robinett's signature and Reid-Selth's and Glass's is that the means for accessing the hidden content in the Channel F games bears an arbitrary relationship to gameplay. The only way a player might encounter any of the three easter eggs mentioned above would be by accident, which explains why they were not documented until nearly 30 years after their publication. Furthermore, there is no diegetic justification for the intrusion of Reid-Selth's name into the play field of Video Whizball. Robinett's signature, on the other hand, exists as a stable object within a narrativized game world. Like any other object in the game, the signature has the property of solidity such that the player's avatar dot bounces off it like a wall. More importantly, the game provides hints at its existence. Ruffin Bailey explains:

> The 2600 will only easily display a maximum of two complex objects called "sprites" (usually called "player graphics" on the 2600) on the screen with each frame. When more needed to be present, Robinett circumvented the 2600's limitation by making *Adventure*'s frames flash quickly enough (with two different objects displayed per frame) to let the gamer understand that more objects were there, though with the side effect that the items would seem to strobe constantly. Robinett put an extra object into the room that contained the secret dot to ensure that the room would flash when the player entered. (R. Bailey)

In this way, the signature exists within the game world as an inscription, but it also exists within the ludic framework of the game's quest narrative. Furthermore, the means by which the easter

29 "43" in hexadecimal notation equals "67" in decimal notation. For more detailed instructions on unlocking these Easter Eggs, see http://members.cox.net/seanriddle/chanf.html.

egg's existence are revealed are intimately connected to the affordances of the platform – exploited here toward a subversive meta-level communication that speaks to the specific corporate culture in which the game was created. In other words, the easter signature of Warren Robinett provides access to the social text of *Adventure* because it extends across the material, ludic, and discursive properties of the game as text. Furthermore, the fact that the image of the signature is clearly inflected by its constraints raises it to holotypical status for the textuality of *Adventure* and the VCS more generally.

Conclusion: Phosphor Burn

Throughout this chapter, the concept of videogame holotype has been dealt with largely as a question of code, taking for granted (or simply ignoring) the presence of the television or monitor screen as a crucial component of any videogame's materiality. For example, most of the character sets and fonts discussed here have been illustrated by printing their bitmap code as opposed to capturing a screenshot with an emulator or photographing a videogame during play. In part, this was done to bypass for now the problematic figure of the screen as a site for videogame discourse, but a true concept of videogame holotype should also include the aesthetic influences of screen technology. The next chapter explores this in some detail, with reference to both paratypical and holotypical videogame typography, but for the present grammatological discussion, the screen demands attention as another site for videogame inscription.

CRT television sets create an image on their screens by passing electron beam across a grid of phosphorescent dots, which luminesce when struck by electrons. Monochrome display devices have one phosphor per screen pixel, and color displays pass beams across red, green, and blue phosphors within each screen pixel. The contact between electron and phosphor is for all practical purposes infinitessimally brief, but it leaves a lasting trace. Each contact degrades the phosphor slightly until it no longer luminesces as it originally did, and the result is a ghost image burned into the screen by prolonged exposure to an unchanging image. Since many coin-operated videogames remain running constantly, screen burn-in is a common problem on arcade machines. Figure 3-40 shows a monitor exhibiting burn in from *Pac-Man*. The clearest elements burned into the screen are the pieces of text, in Namco Font, at the top and bottom of the screen, so it is reasonable to consider this property of phosphor screen display a form of textual inscription. The fact that CRT technology forms the basis of photocompositor (typesetting) technology first available in the 1960s and 1970s further connects the CRT and phosphor screen to printing, and it also underscores the effect of compositing technology on the aesthetics of type.

Although the burned text or images is generally unnoticeable during gameplay, its mark on the screen is indelible. So as a mode of inscription, CRT burn-in serves as a different kind of record. Like the high score display, burn-in utilizes the screen to supply a representation of past instances of play (in this case, all past instances), but the reference it accomplishes is specific to the physical hardware of the videogame as technology. In other words, burn-in does not occur in a necessary relationship with videogames, and it need not be a record of videogameplay at all. However, the prevelance of burn-in on arcade games from the 1970s and 80s indicates the importance of the screen in determining the aesthetics of type designed for display within videogames. Besides burn-in, phosphor based screens also exhibit light bleeding when neighboring phosphors pick up stray electrons, and the passing of, say, a white object across an otherwise black screen can leave an apparent trail as pixels continue to luminesce for a second or two after the beam passes. These effects form an important dimension of videogame expression, so much so that the VCS emulator Stella includes an option for simulating phosphor effects within its display. More importantly for type design, the distortion or fuzzying effect of the CRT screen is a factor which must be accounted for in designing legible type for screen display, and where this has been the case in videogame typography, the character design reflects this influence. The next chapter discusses competing paradigms for representing videogame typography – "fuzzy" and "jaggy" – in order to make the claim that each alternative addresses a different critical ontology for game analysis as an extension metaphors common among videogame and new media scholarship that treat the screen as a figure for media discourse.

Figure 3-1. Screenshot of *Rogue*, the original "textmode" game, which employs ASCII characters for its graphics.

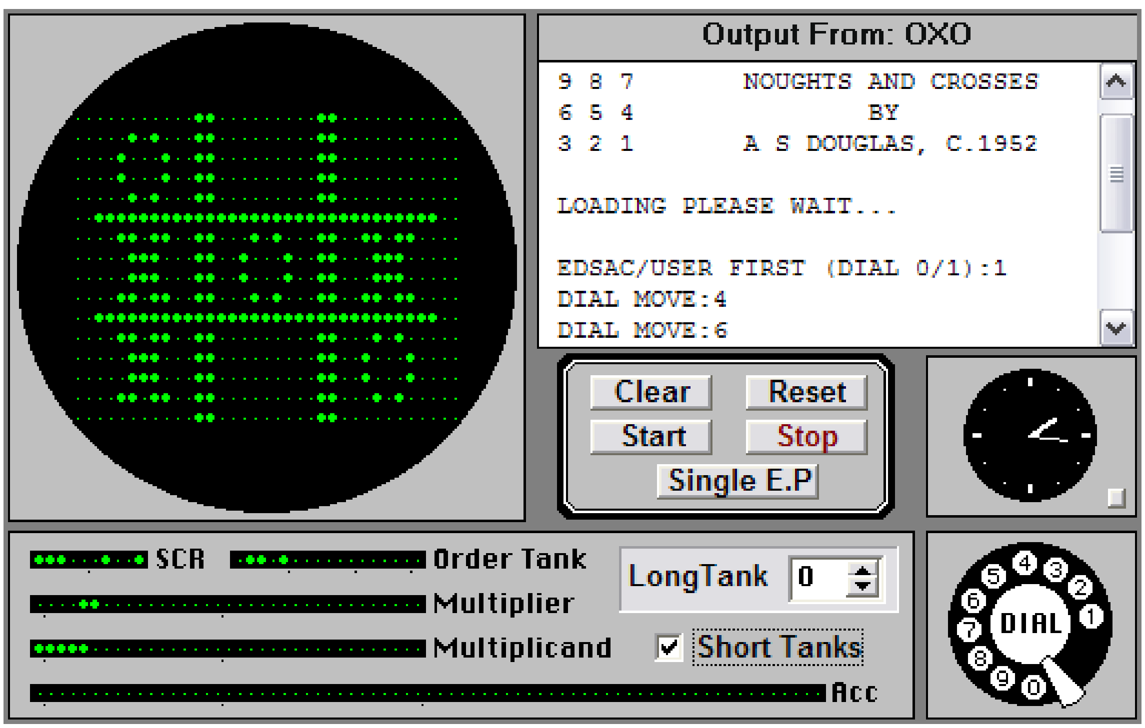


Figure 3-2. Screenshot showing "OXO" running in an EDSAC simulator created by Martin Campbell-Kelly.

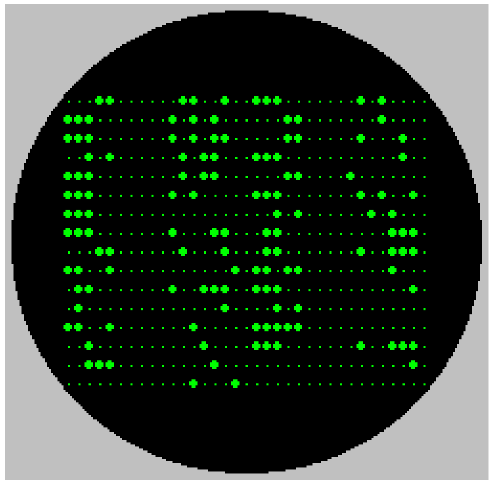


Figure 3-3. Display output of EDSAC processing a "Hello, World" program.

Figure 3-4. Screen image of Computer Space (1972) in action. Photograph by Kevin Armstrong.

Figure 3-5. Detail of information display utilized in *Computer Space*. Photograph by Kevin Armstrong.

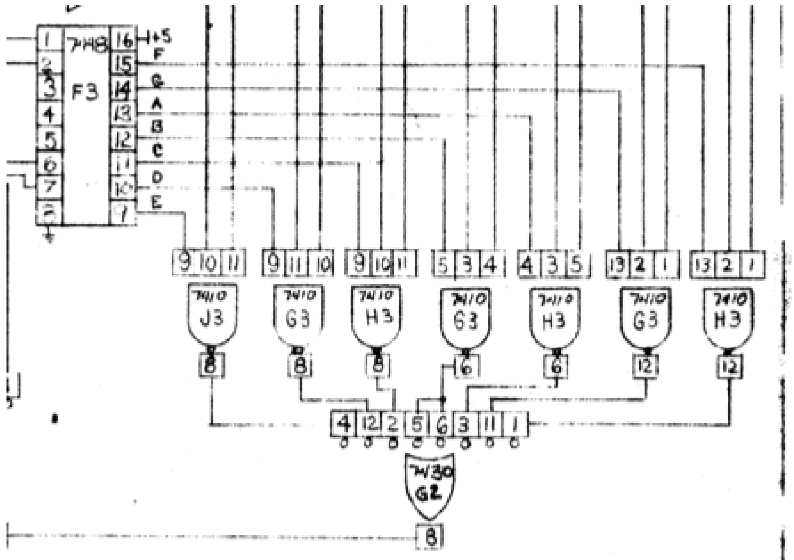
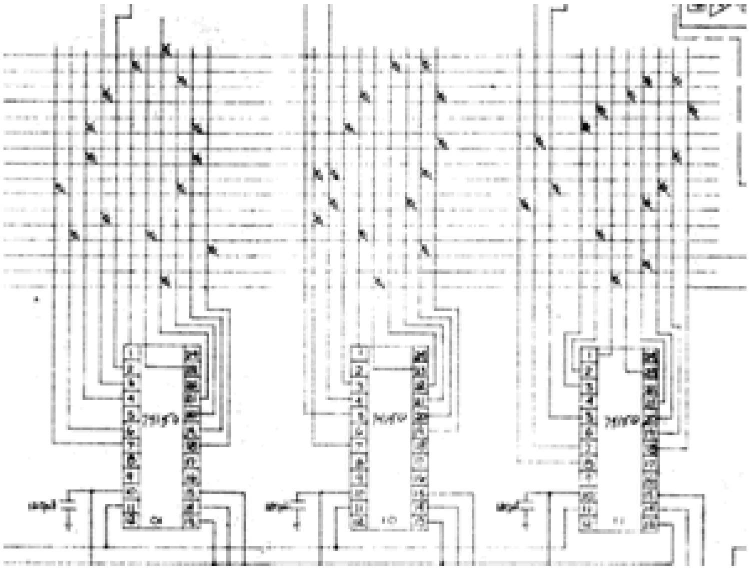
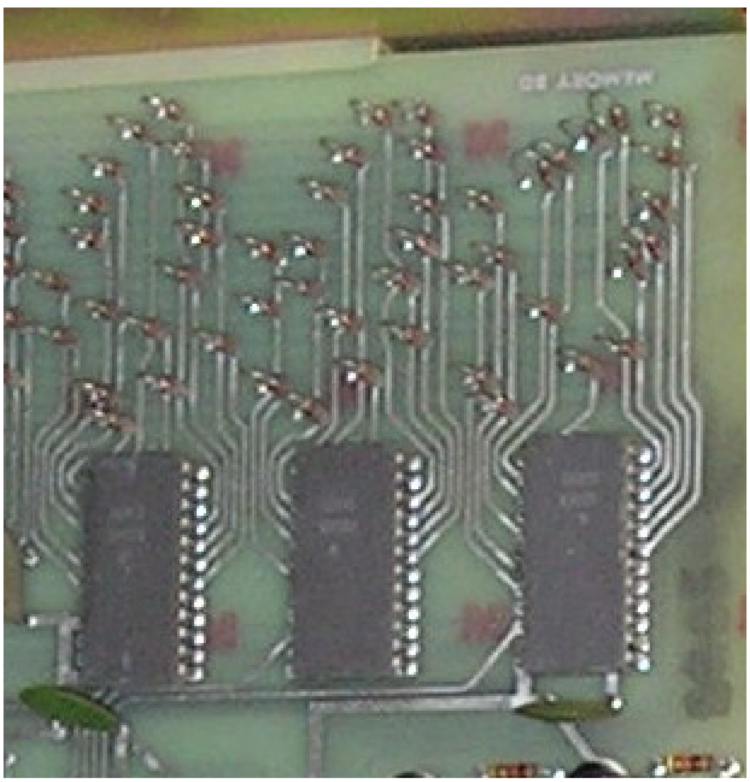


Figure 3-6. Detail of schematics for *Computer Space's* "SYNC-STAR" circuit board, display circuit. This section is the display circuit, centered around a TI 7448 BCD to 7-segment converter.



A



B

C

Figure 3-7. Rocket ship image generation as analog sprites. A) Detail of circuit board diagram for "MEMORY BOARD." B) Photograph of *Computer Space* machine interior with Memory Board visible. Note the Rocket and Saucer outlines on the left. C) The rocket ship image on screen during gameplay. Photographs by Kevin Armstrong.

Figure 3-8. Magnavox Odyssey overlay for *Haunted House*. (McCourt)

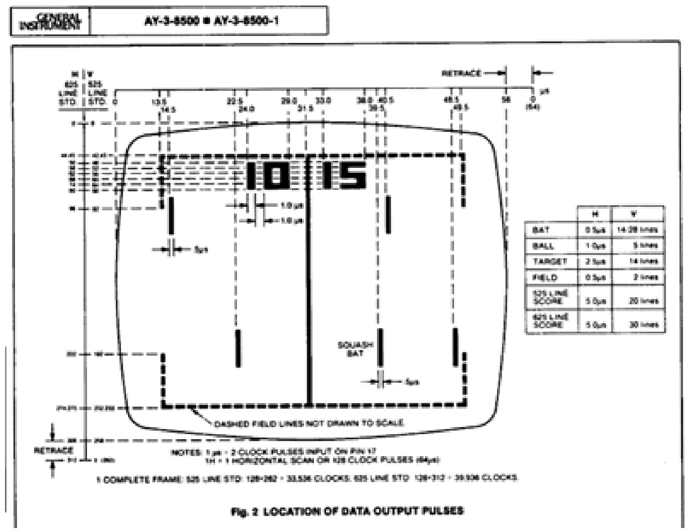


Figure 3-9. Datasheet for General Instruments' AY-3-8500 chip. The so-called "PONG in a chip" that led to so many PONG clones in the mid-1970s.

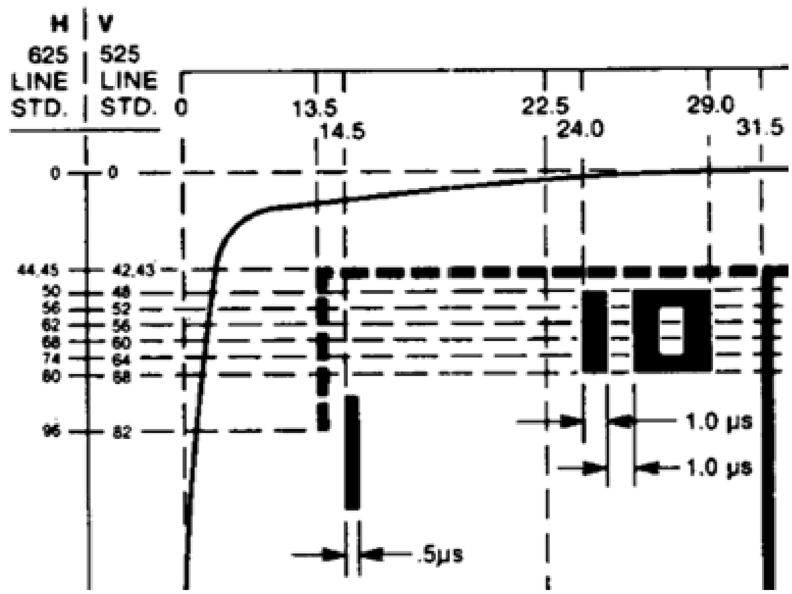


Figure 3-10. Detail of figure 3-9, highlighting the measurements in timing (horizontal) and number of scan lines (vertical) for generating the pixel grid which forms the score.

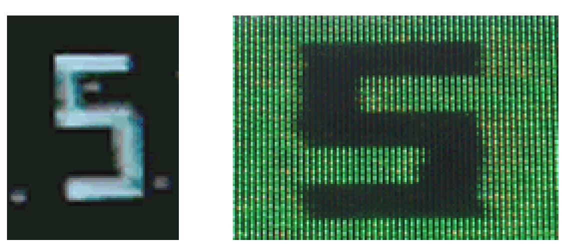


Figure 3-11. Comparing 7-segment numeral 5 from *Computer Space* with 3 x 5 form in a *PONG* clone.

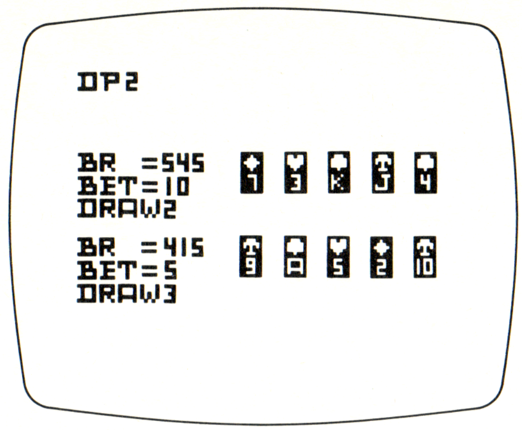


Figure 3-12. Screen illustration of GI AY-3-8810 chip, playing Draw Poker. Image from Buchsbaum 236.

```
XXXXX··· ··X····· XXXXX··· XXXXX··· X···X··· XXXXX··· XXXXX··· XXXXX··· XXXXX··· XXXXX···
X···X··· ··X····· ····X··· ····X··· X···X··· X······· X······· ····X··· X···X··· X···X···
X···X··· ··X····· XXXXX··· XXXXX··· XXXXX··· XXXXX··· XXXXX··· ···X···· XXXXX··· XXXXX···
X···X··· ··X····· X······· ····X··· ····X··· ····X··· X···X··· ···X···· X···X··· ····X···
XXXXX··· ··X····· XXXXX··· XXXXX··· ····X··· XXXXX··· XXXXX··· ···X···· XXXXX··· XXXXX···
```

Figure 3-13. Channel F 5 x 5 numerals, extracted from ROM image of BIOS (SL31254.ROM).

```
··XXXX·· ···XX··· ··XXXX·· ··XXXX·· ····XX·· ·XXXXXX· ··XXXX·· ·XXXXXX· ··XXXX·· ··XXXX··
·XX··XX· ··XXX··· ·X···XX· ·X···XX· ···XXX·· ·XX····· ·XX···X· ·X····X· ·XX··XX· ·XX··XX·
·XX··XX· ···XX··· ·····XX· ·····XX· ··X·XX·· ·XX····· ·XX····· ·····XX· ·XX··XX· ·XX··XX·
·XX··XX· ···XX··· ··XXXX·· ····XX·· ·X··XX·· ·XXXXX·· ·XXXXX·· ····XX·· ··XXXX·· ··XXXXX·
·XX··XX· ···XX··· ·XX····· ·····XX· ·XXXXXX· ·····XX· ·XX··XX· ···XX··· ·XX··XX· ·····XX·
·XX··XX· ···XX··· ·XX····· ·X···XX· ····XX·· ·X···XX· ·XX··XX· ···XX··· ·XX··XX· ·X···XX·
··XXXX·· ··XXXX·· ·XXXXXX· ··XXXX·· ····XX·· ·XXXXX·· ··XXXX·· ···XX··· ··XXXX·· ··XXXX··
```

Figure 3-14. A common 6 x 7 numeric character set employed in several games produced for Atari VCS, including the Parker Brothers games *James Bond 007*, *Popeye*, *Super Cobra*, and *Garfield*, as well as the Activision titles *Ice Hockey*, *Realsports Tennis*, and *Realsports Basketball*. The adult line of games from Mystique / Playaround also employed this 6 x 7 set. I have not included memory locations for these bitmaps because they appear in different places in different games, but this specific set was extracted from a ROM image of an *Ice Hockey* cartridge (Icehockey.bin).

```
········ ········ ········ ········ ········ ········ ········ ········ ········ ········
··XXX··· ···XX··· ·XXXXX·· XXXXXX·· XX··XX·· XXXXXX·· ·XXXXX·· XXXXXXX· ·XXXXX·· ·XXXXX··
·XX·XX·· ··XXX··· XX···XX· ·····XX· XX··XX·· XX······ XX······ ·····XX· XX···XX· XX···XX·
XX···XX· ···XX··· ·····XX· ·····XX· XX··XX·· XX······ XX······ ····XX·· XX···XX· XX···XX·
XX···XX· ···XX··· ··XXXX·· ·XXXXX·· XXXXXXX· XXXXXX·· XXXXXX·· ···XX··· ·XXXXX·· ·XXXXXX·
XX···XX· ···XX··· XXX····· ·····XX· ····XX·· ·····XX· XX···XX· ···XX··· XX···XX· ·····XX·
·XX·XX·· ···XX··· XX······ ·····XX· ····XX·· ·····XX· XX···XX· ··XX···· XX···XX· ·····XX·
··XXX··· ·XXXXXX· XXXXXXX· XXXXXX·· ····XX·· XXXXXX·· ·XXXXX·· ··XX···· ·XXXXX·· ·XXXXX··
```

Figure 3-15. A common 7 x 7 numeric character set used in games manufactured by Atari for their VCS console. Notably, the games using this set were adaptations from arcade originals. This specific set was extracted from a ROM image of *Pole Position* (Polepsn.bin).

```
·XXXX···  ··XX····  ·XXXX···  ·XXXX···  ···XX···  XXXXXX··  ·XXXX···  XXXXXX··  ·XXXX···  ·XXXX···
XX··XX··  ·XXX····  X···XX··  X···XX··  ··XXX···  XX······  XX···X··  X····X··  XX··XX··  XX··XX··
XX··XX··  ··XX····  ····XX··  ····XX··  ·X·XX···  XX······  XX······  ····XX··  XX··XX··  XX··XX··
XX··XX··  ··XX····  ····XX··  ···XX···  X··XX···  XXXXX···  XXXXX···  ···XX···  ·XXXX···  XX··XX··
XX··XX··  ··XX····  ·XXXX···  ···XX···  XXXXXX··  ····XX··  XX··XX··  ··XX····  ·XXXX···  ·XXXXX··
XX··XX··  ··XX····  XX······  ····XX··  ···XX···  ····XX··  XX··XX··  ··XX····  XX··XX··  ····XX··
XX··XX··  ··XX····  XX······  X···XX··  ···XX···  X···XX··  XX··XX··  ··XX····  XX··XX··  X···XX··
·XXXX···  ·XXXX···  XXXXXX··  ·XXXX···  ···XX···  XXXXX···  ·XXXX···  ··XX····  ·XXXX···  ·XXXX···
```

Figure 3-16. A common 6 x 8 numeric character set used in dozens of games produced for Atari VCS. Notably, third party developers such as Activision tended to use this character set while in-house Atari games are more likely to use the 7 x 7 designs in figure 3-12. This specific character set was extracted from a ROM image of *Barn Storming* (Barnstrm.bin).

```
·XXXXXX·  ·XXXX···  ·XXXXXX·  ·XXXXXX·  ·XX··XX·  ·XXXXXX·  ·XXXXXX·  ·XXXXXX·  ·XXXXXX·  ·XXXXXX·
·XX··XX·  ···XX···  ·XX··XX·  ·····XX·  ·XX··XX·  ·XX·····  ·XX·····  ·····XX·  ·XX··XX·  ·XX··XX·
·XX··XX·  ···XX···  ·····XX·  ·XXXXX··  ·XX··XX·  ·XXXXXX·  ·XXXXXX·  ····XX··  ··XXXX··  ·XX··XX·
·XX··XX·  ···XX···  ·XXXXXX·  ·····XX·  ·XXXXXX·  ·····XX·  ·XX··XX·  ···XX···  ·XX··XX·  ·XXXXXX·
·XX··XX·  ···XX···  ·XX·····  ·····XX·  ·····XX·  ·XX··XX·  ·XX··XX·  ··XX····  ·XX··XX·  ·····XX·
·XXXXXX·  ·XXXXXX·  ·XXXXXX·  ·XXXXXX·  ·····XX·  ·XXXXXX·  ·XXXXXX·  ··X·····  ·XXXXXX·  ·XXXXXX·
```

Figure 3-17. The numeric character set used in a series of games produced by Mattel in 1983 for the Atari VCS console. These are identical to the numerals used in the system font built into Mattel's Intellivision console. This specific set was extracted from a ROM image of *Burger Time* (Burgtime.bin).

```
········  ········  ········  ········  ········  ········  ········  ········  ········  ········
XXXXXXX·  ··XXX···  ·XXXXXX·  ·XXXXXX·  ·XX··XX·  ·XXXXXX·  ·XXXXXX·  ·XXXXXX·  ·XXXXXX·  ·XXXXXX·
XX···XX·  ···XX···  ·XX··XX·  ·····XX·  ·XX··XX·  ·XX·····  ·XX·····  ·····XX·  ·XX··XX·  ·XX··XX·
XX·X·XX·  ···XX···  ·····XX·  ··XXXX··  ·XX··XX·  ·XXXXXX·  ·XXXXXX·  ····XX··  ··XXXX··  ·XX··XX·
XX·X·XX·  ···XX···  ·XXXXXX·  ·····XX·  ·XXXXXX·  ·····XX·  ·XX··XX·  ···XX···  ·XX··XX·  ·XXXXXX·
XX···XX·  ···XX···  ·XX·····  ·····XX·  ·····XX·  ·XX··XX·  ·XX··XX·  ··XX····  ·XX··XX·  ·····XX·
XXXXXXX·  ·XXXXXX·  ·XXXXXX·  ·XXXXXX·  ·····XX·  ·XXXXXX·  ·XXXXXX·  ··XX····  ·XXXXXX·  ·XXXXXX·
········  ········  ········  ········  ········  ········  ········  ········  ········  ········
```

Figure 3-18. Select characters from Intellivision system font.

```
·XXX····     XXXXX···     XXXXX···     XXXX····     X···X···
X···X···     X·X·X···     X···X···     X···X···     XX·XX···
·X······     ··X·····     X···X···     X···X···     X·X·X···
··X·····     ··X·····     X···X···     XXXX····     X·X·X···
···X····     ··X·····     X···X···     X·X·····     X···X···
X···X···     ··X·····     X···X···     X··X····     X···X···
·XXX····     ··X·····     XXXXX···     X···X···     X···X···
```

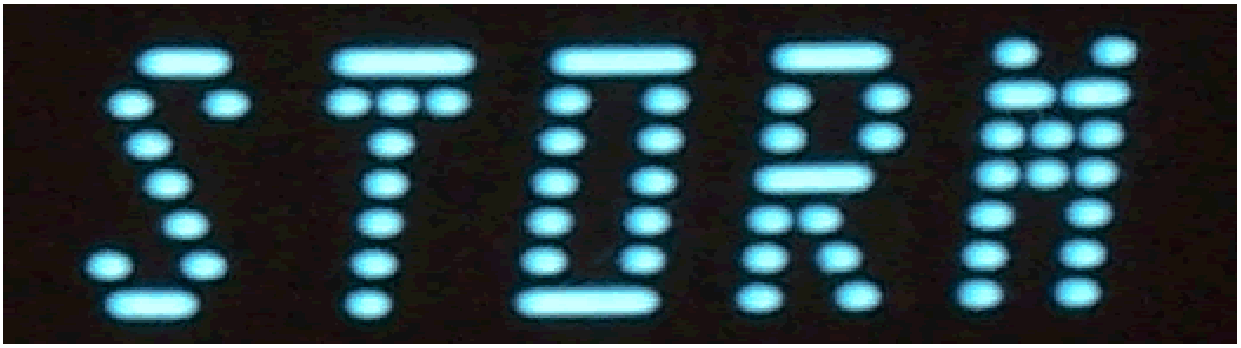
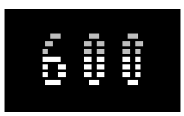


Figure 3-19. Select characters generated by Vectrex, pairing each bitmap with its on-screen rendering. Photograph from *Mine Storm*.

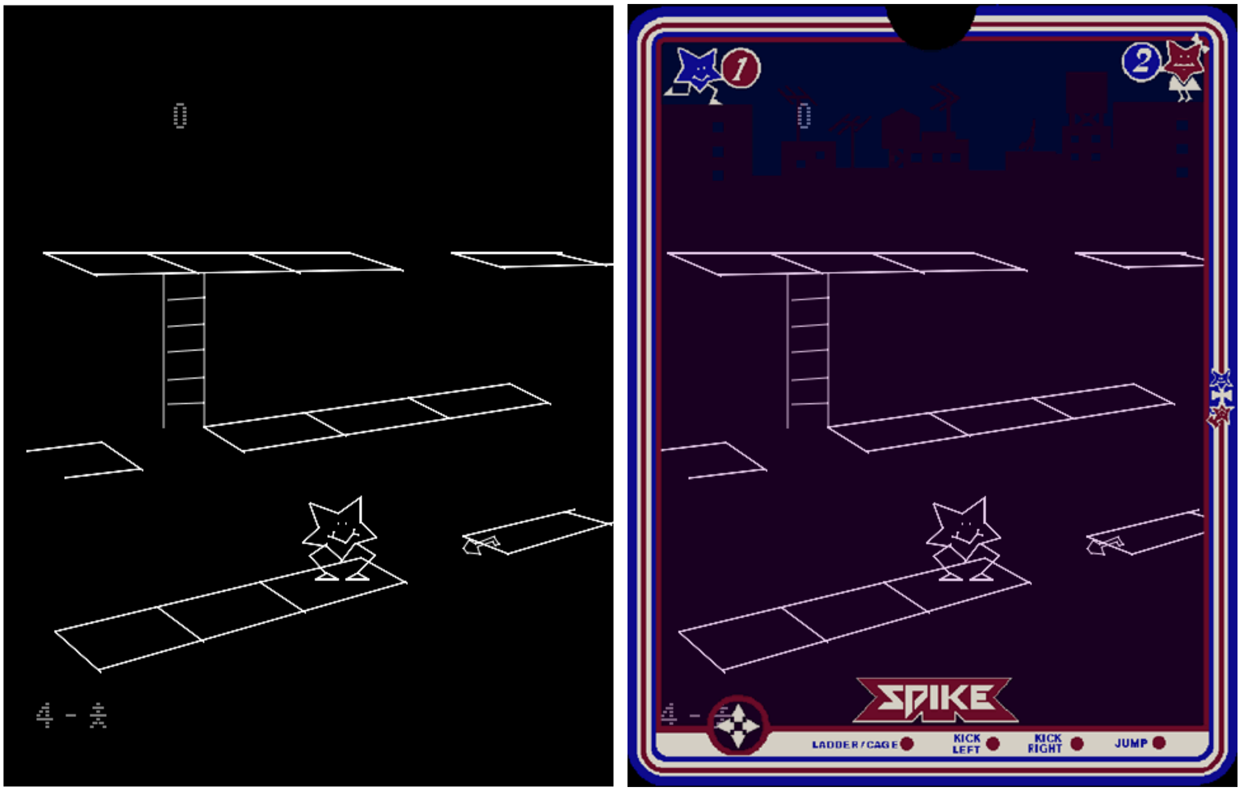


A

B

Figure 3-20. A demonstration of one difficulty in emulating Vectrex gameplay. A) Numerals displayed in a Vectrex emulator (screenshot captured from MESS emulator, running Vectrex in DirectDraw graphics mode at 1024 x 728 resolution). B) Similar numerals displayed on Vectrex screen (photograph).

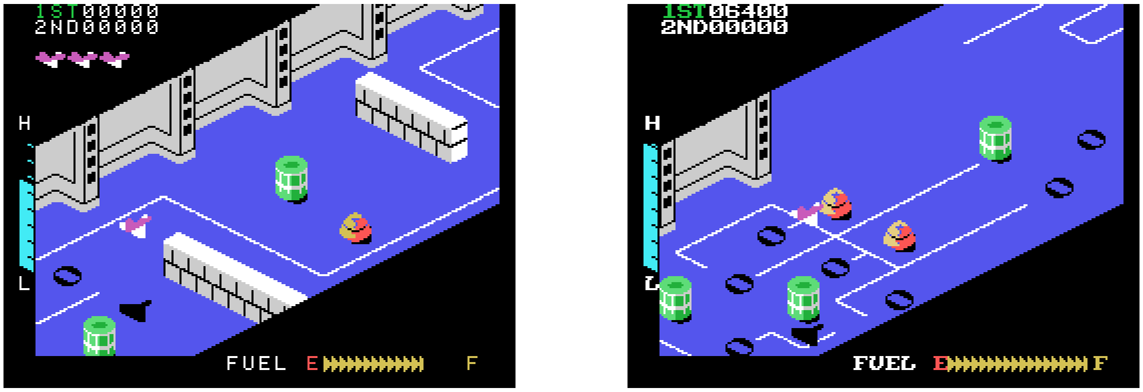
Figure 3-21. Screenshot from the vector-based coin-operated game *Battlezone*. Note that *Battlezone,* like most other vector-based games composes alphanumeric glyphs by outlining their shape.



A B

Figure 3-22. Screenshots of Vectrex game *Spike* A) with overlay, B) with no overlay. Each image was captured from a Vectrex emulator running in MESS. However, since the emulator software applies the overlay as a transparent PNG that is placed in a layer above the running emulation (a combination formally quite analogous to the physical layer created by the transparent overlay), I created the image in B by using a graphics editing program to combine the screenshot in A with the transparent overlay image used by MESS.

Figure 3-23. ATASCII character set. (Image from "ATASCII").



A B

Figure 3-24. Screenshots of the same game (*Zaxxon*) emulated with A) the original ColecoVision BIOS and B) the alternate system font BIOS.

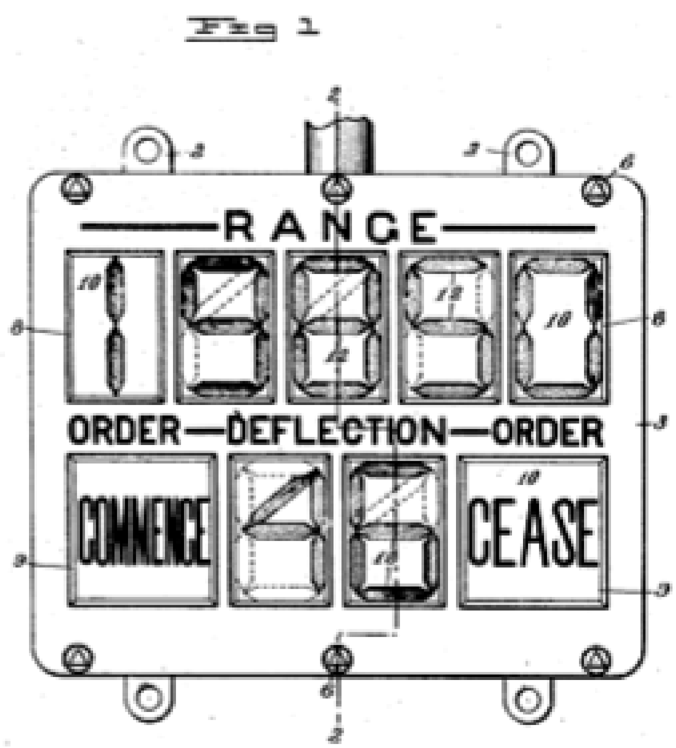
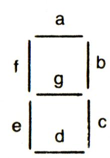


Figure 3-25. The 7-segment figure generator patented by Frank W. Wood in 1908 (Wood).

Figure 3-26. Thomas Ross Welch's 1937 design for a text-displaying Electric Sign. Note the importance of uniform geometric shapes in making up the individual characters.



A

DEC	BCD IN D C B A	SEGMENT OUT a b c d e f g
0	0 0 0 0	1 1 1 1 1 1 0
1	0 0 0 1	0 1 1 0 0 0 0
2	0 0 1 0	1 1 0 1 1 0 1
3	0 0 1 1	1 1 1 1 0 0 1
4	0 1 0 0	0 1 1 0 0 1 1
5	0 1 0 1	1 0 1 1 0 1 1
6	0 1 1 0	0 0 1 1 1 1 1
7	0 1 1 1	1 1 1 0 0 0 0
8	1 0 0 0	1 1 1 1 1 1 1
9	1 0 0 1	1 1 1 0 0 1 1
G=1	X X X X	0 0 0 0 0 0 0

B

Figure 3-27. Illustration of the 7-segment truth table. (Scanned from Heiserman 236)

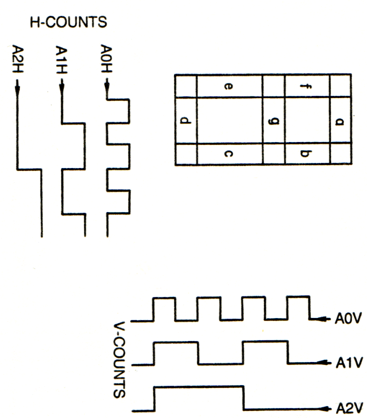


Figure 3-28. Illustration of display circuit for 7-segment generator.

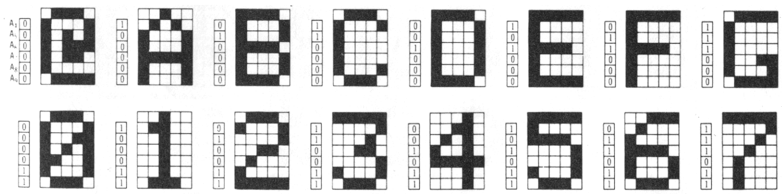


Figure 3-29. Selected characters from 2513 character-generator ROM (Signetics).

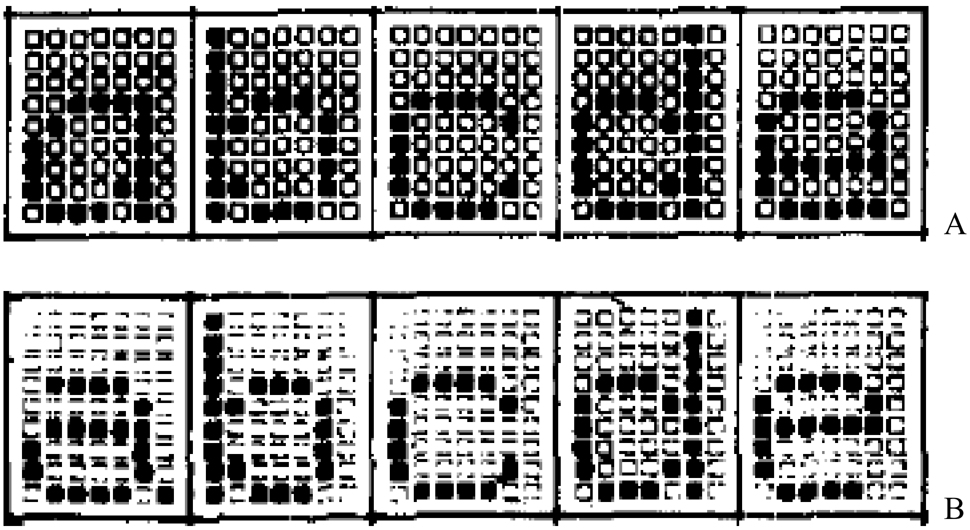


Figure 3-30. Comparing serif and sans-serif characters on A) MCM6576 and B) MCM6571 character-generator ROM chips.

Figure 3-31. The “opening credits” for Namco’s *Pac-Man*, featuring “Namco Font” for all text.

A

B

Figure 3-32. Comparing sample characters from a) BASIC Programming and B) Namco Font.

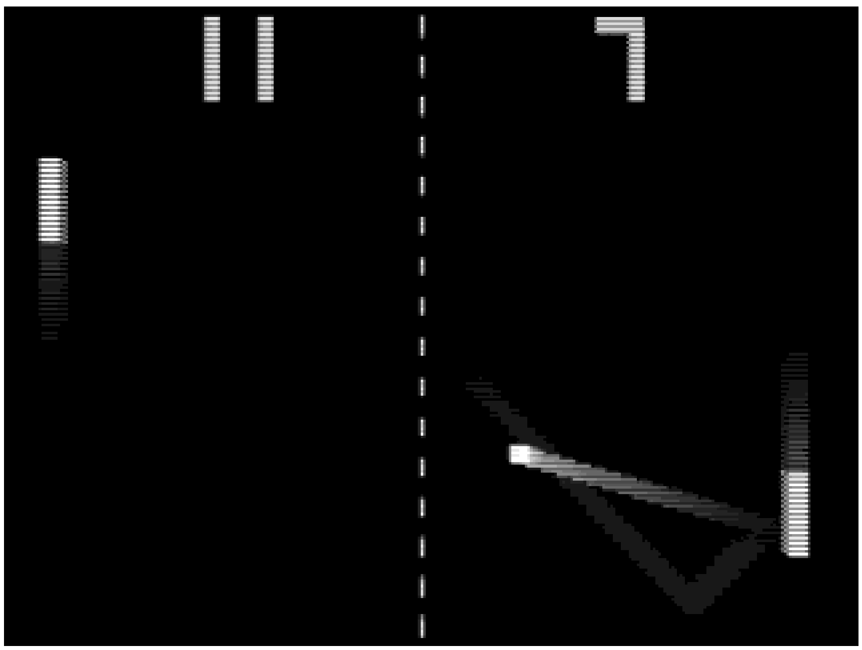
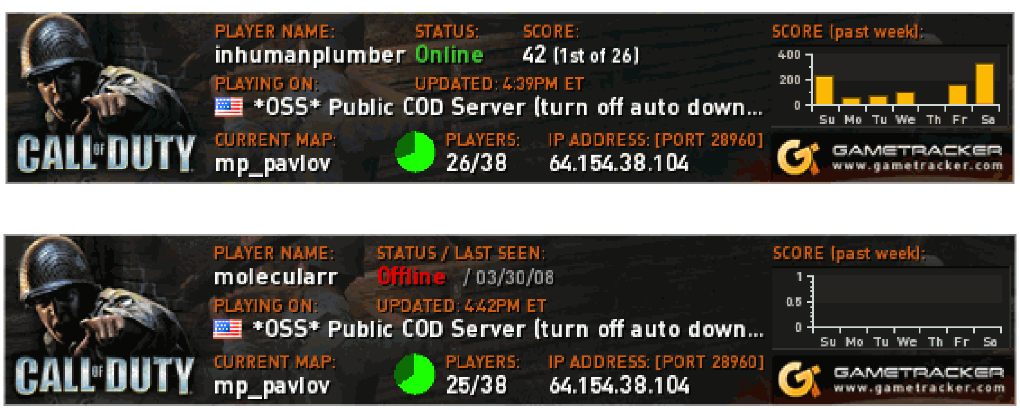


Figure 3-33. *PONG* screenshot. Score is 11 to 7. (Screenshot from "8-Bit-Nirvana: Atari Pong")



A

B

Figure 3-34. Game statistics generated by Gametracker, monitoring an active server at IP address 64.154.38.104. A) One of the top players on this particular server, inhumanplumber, who was online at the time this image was captured. B) A less committed player, molecularr, whose current status indicates that he is offline.

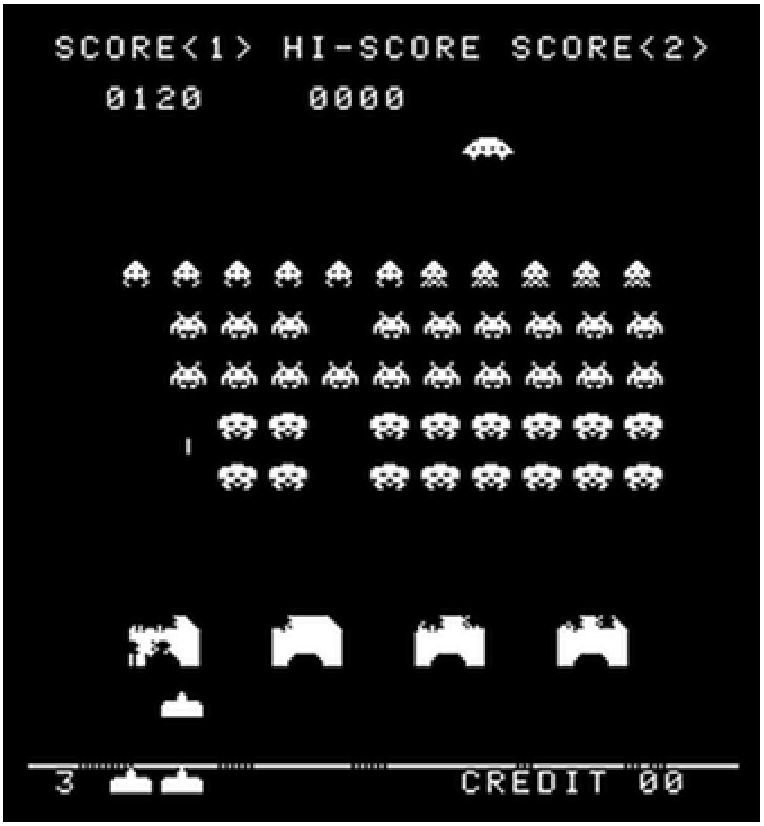


Figure 3-35. Screenshot of *Space Invaders*, showing position of high score (in this case, 0000).

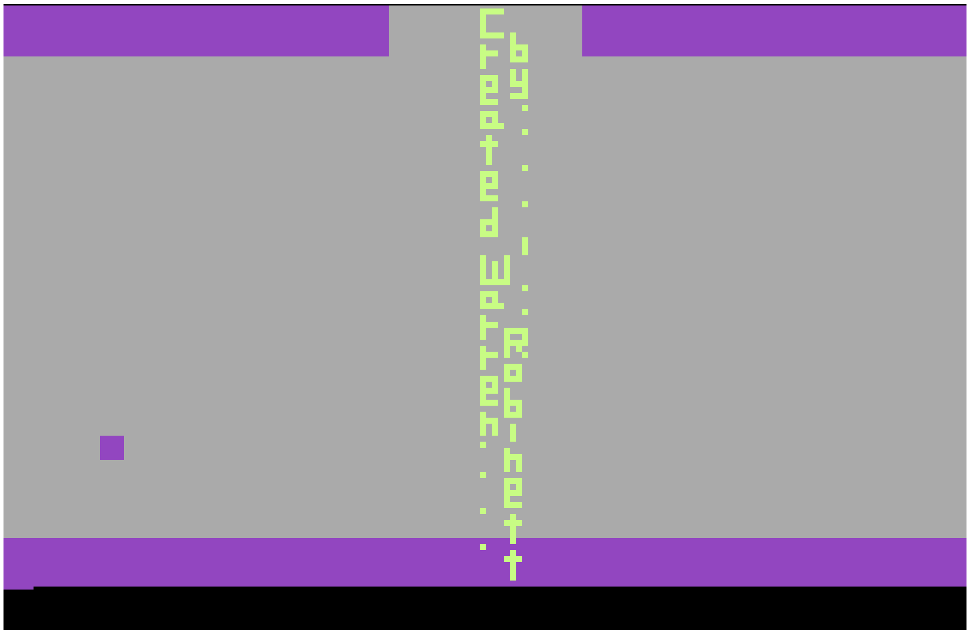
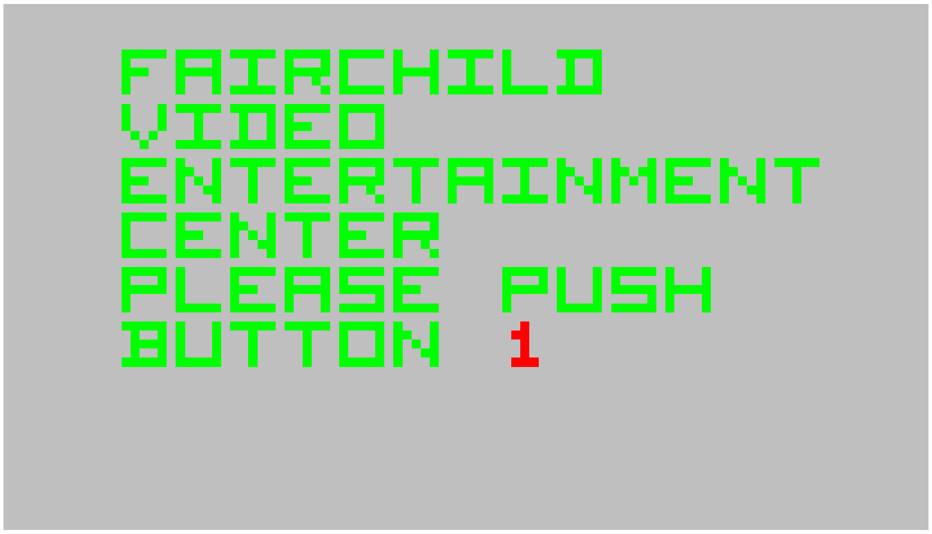
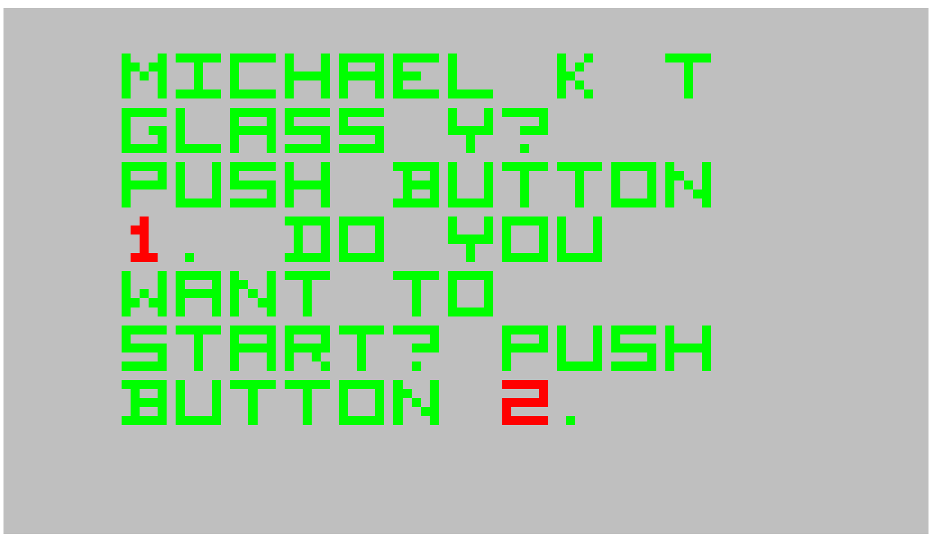


Figure 3-36. Warren Robinett's "autograph" in *Adventure* for the Atari VCS console. Screenshot captured from the VCS emulator Stella.

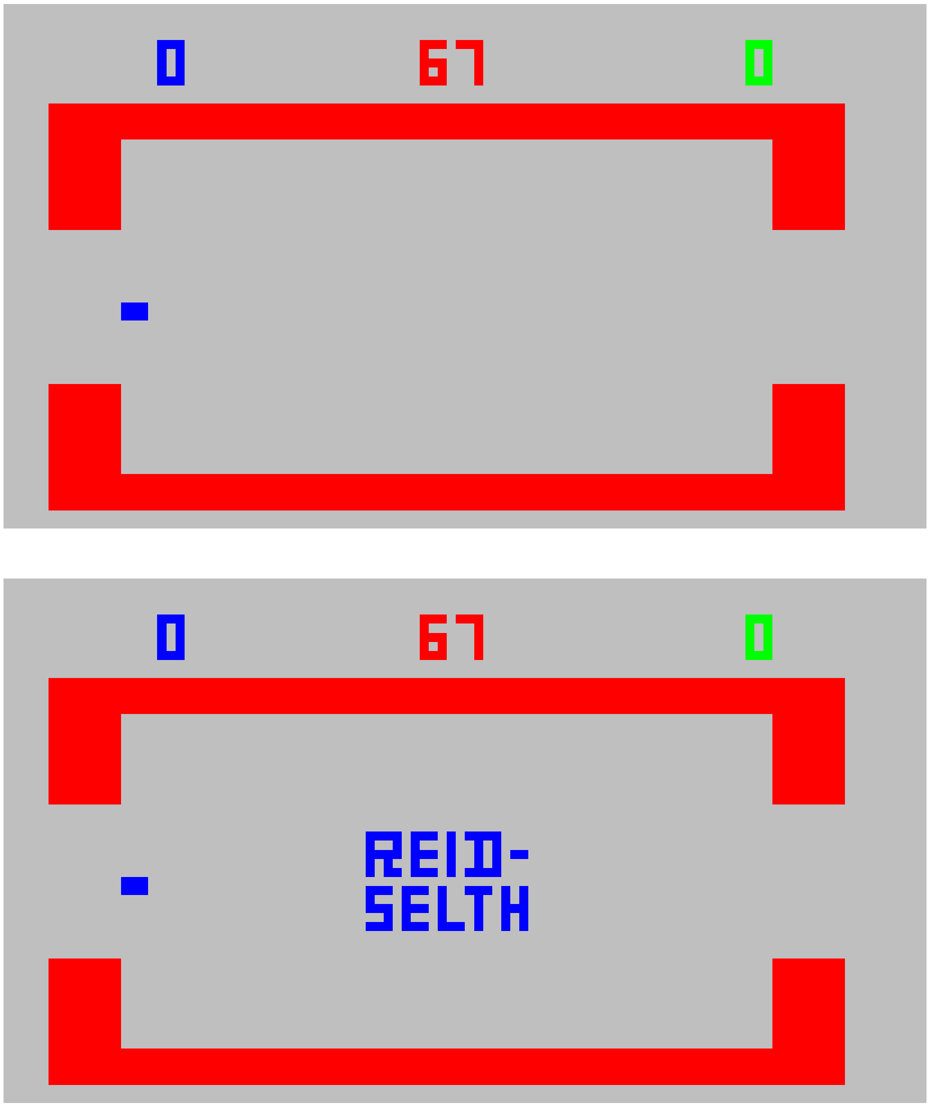


A



B

Figure 3-37. The first Channel F Easter Egg? A) Concluding text in demonstration cartridge. B) Revealing the programmer's name, MICHAEL K. GLASS. Both screenshots captured from Channel F emulator running in MESS.



A

B

Figure 3-38. Easter Egg in Channel F game *Video-Whizball*. A) Normal game playing field. B) Playing field with revealed text. Both screenshots captured from Channel F emulator running in MESS.

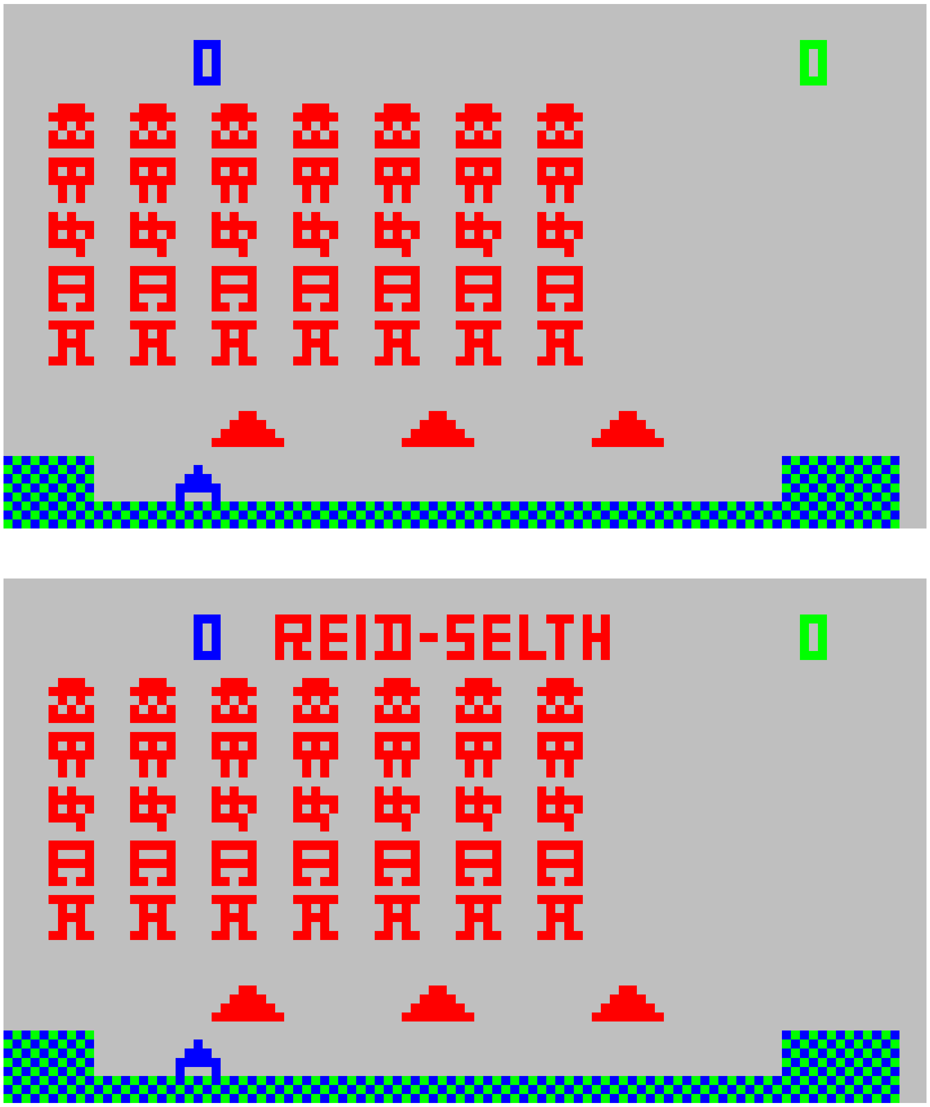


Figure 3-39. Easter Egg in Channel F game *Alien Invasion*. A) Standard gameplay field. B) Playing field with revealed text.

Figure 3-40. An example of extreme screen burn-in, suffered at the hands of a *Pac-Man* game.

Table 3-1. Sample numeric character sets employing 3 x 5 pixel grids for composing numerals. Note the variation, especially with 1 and 4. Memory addresses of these bitmaps have been omitted to conserve space. Most of the examples below double the numerals in order to cover units and tens places simultaneously. Similarly, some also transpose the character in the tens position across a bilateral axis. This is in order to anticipate and take advantage of the TIA chip's built-in ability to mirror sprites across a scan line.

Game	Bitmap patterns									
Breakout	·XXX·XXX ·X·X·X·X ·X·X·X·X ·X·X·X·X ·XXX·XXX	··X···X· ··X···X· ··X···X· ··X···X· ··X···X·	·XXX·XXX ···X···X ·XXX·XXX ·X···X·· ·XXX·XXX	·XXX·XXX ···X···X ··XX··XX ···X···X ·XXX·XXX	·X·X·X·X ·X·X·X·X ·XXX·XXX ···X···X ···X···X	·XXX·XXX ·X···X·· ·XXX·XXX ···X···X ·XXX·XXX	·X···X·· ·X···X·· ·XXX·XXX ·X·X·X·X ·XXX·XXX	·XXX·XXX ···X···X ···X···X ···X···X ···X···X	·XXX·XXX ·X·X·X·X ·XXX·XXX ·X·X·X·X ·XXX·XXX	·XXX·XXX ·X·X·X·X ·XXX·XXX ···X···X ···X···X
Combat	····XXX· ····X·X· ····X·X· ····X·X· ····XXX·	··X···X· ··X···X· ··X···X· ··X···X· ··X···X·	XXX·XXX· ··X···X· XXX·XXX· X···X··· XXX·XXX·	XXX·XXX· ··X···X· ·XX··XX· ··X···X· XXX·XXX·	X·X·X·X· X·X·X·X· XXX·XXX· ··X···X· ··X···X·	XXX·XXX· X···X··· XXX·XXX· ··X···X· XXX·XXX·	XXX·XXX· X···X··· XXX·XXX· X·X·X·X· XXX·XXX·	XXX·XXX· ··X···X· ··X···X· ··X···X· ··X···X·	XXX·XXX· X·X·X·X· XXX·XXX· X·X·X·X· XXX·XXX·	XXX·XXX· X·X·X·X· XXX·XXX· ··X···X· XXX·XXX·
Indy 500	····XXX· ····X·X· ····X·X· ····X·X· ····XXX·	··X···X· ··X···X· ··X···X· ··X···X· ··X···X·	XXX·XXX· ··X···X· XXX·XXX· X···X··· XXX·XXX·	XXX·XXX· ··X···X· ·XX··XX· ··X···X· XXX·XXX·	X·X·X·X· X·X·X·X· XXX·XXX· ··X···X· ··X···X·	XXX·XXX· X···X··· XXX·XXX· ··X···X· XXX·XXX·	XXX·XXX· X···X··· XXX·XXX· X·X·X·X· XXX·XXX·	XXX·XXX· ··X···X· ··X···X· ··X···X· ··X···X·	XXX·XXX· X·X·X·X· XXX·XXX· X·X·X·X· XXX·XXX·	XXX·XXX· X·X·X·X· XXX·XXX· ··X···X· XXX·XXX·
Outer Space	·····XXX ·····X·X ·····X·X ·····X·X ·····XXX	···X···X ···X···X ···X···X ···X···X ···X···X	·XXX·XXX ···X···X ·XXX·XXX ·X···X·· ·XXX·XXX	·XXX·XXX ···X···X ··XX··XX ···X···X ·XXX·XXX	·X·X·X·X ·X·X·X·X ·XXX·XXX ···X···X ···X···X	·XXX·XXX ·X···X·· ·XXX·XXX ···X···X ·XXX·XXX	·XXX·XXX ·X···X·· ·XXX·XXX ·X·X·X·X ·XXX·XXX	·XXX·XXX ···X···X ···X···X ···X···X ···X···X	·XXX·XXX ·X·X·X·X ·XXX·XXX ·X·X·X·X ·XXX·XXX	·XXX·XXX ·X·X·X·X ·XXX·XXX ···X···X ·XXX·XXX
Space Invaders	XXX··XXX X·X··X·X X·X··X·X X·X··X·X XXX··XXX	·X····X· ·XX··XX· ·X····X· ·X····X· XXX··XXX	XXX··XXX X······X XXX··XXX ··X··X·· XXX··XXX	XXX··XXX X······X XXX··XXX X······X XXX··XXX	X·X··X·X X·X··X·X XXX··XXX X······X X······X	XXX··XXX ··X··X·· XXX··XXX X······X XXX··XXX	··X··X·· ··X··X·· XXX··XXX X·X··X·X XXX··XXX	XXX··XXX X······X X······X X······X X······X	XXX··XXX X·X··X·X XXX··XXX X·X··X·X XXX··XXX	XXX··XXX X·X··X·X XXX··XXX X······X X······X
Dodge 'Em	XXX··XXX X·X··X·X X·X··X·X X·X··X·X XXX··XXX ········	·X····X· XX····XX ·X····X· ·X····X· XXX··XXX ········	XXX··XXX ··X··X·· XXX··XXX X······X XXX··XXX ········	XXX··XXX ··X··X·· ·XX··XX· ··X··X·· XXX··XXX ········	X·X··X·X X·X··X·X XXX··XXX ··X··X·· ··X··X·· ········	XXX··XXX X······X XXX··XXX ··X··X·· XXX··XXX ········	XXX··XXX X······X XXX··XXX X·X··X·X XXX··XXX ········	XXX··XXX ··X··X·· ·X····X· X······X X······X ········	XXX··XXX X·X··X·X XXX··XXX X·X··X·X XXX··XXX ········	XXX··XXX X·X··X·X XXX··XXX ··X··X·· XXX··XXX ········

Object 3-1. Video capture of Warren Robinett's Easter Egg in *Adventure* (emulated in Stella). Robinett's "marquee" graffiti is highlighted by making use of the same animation technique that identifies the chalice as the player's objective. Click the image to start playback; alternatively, click on this text to download the file directly. (.avi file 6MB)

Object 3-2. Video capture of the shimmering chalice, the player's ostensible objective in *Adventure* (emulated in Stella). In this clip, the player returns the chalice to the gold castle, which rewards the player by shimmering along with the chalice. Click the image above to start playback; alternatively, click on this text to download the file directly. (.avi file 4.9MB)

CHAPTER 4
FUZZY AND JAGGY: AESTHETIC AND ONTOLOGICAL DISPOSITIONS OF VIDEOGAME TYPE

Text and typography in and around videogames – whether played on arcade cabinets, consoles, or emulators – exhibit aesthetic properties that emphasize their programmatic origins (as bitmaps, for example) as well as the influence or interference of the display hardware itself. In this chapter, I hope to demonstrate that by understanding the different appearances of type in videogames and, to an important extent, the hardware that generates these effects, we can arrive at a better comprehension of videogame expression, textuality[1] and material culture. In other words, if we consider videogames to be cultural artifacts that can communicate complex, subtle or persuasive ideas, then even the smallest elements of the game, including the size, dimension, and individual clarity of pixels can contribute to that cumulative articulation. And since expressive typography is a practice whereby designers modify letter shapes, sometimes subtly, in order to communicate emotions, associations, and ideas, then the shapes of letters in videogames must also be examined for their expressivity, both within videogames and in the materials that surround and support videogames.

Of particular concern in this chapter is the sense in which specific material conditions (such as rendering software and display hardware) influence the appearance of alphanumeric shapes to express ideas or a set of assumptions about the game artifact. As evidence of this practice in videogames and gaming culture more generally, I offer that when graphic designers have borrowed or imitated typographic forms from videogames and used them in other media such as print or higher-resolution web graphics, they typically choose to mimic one of two categories of effects that I am going to be comparing – *jaggy* or *fuzzy*. These two qualities, realized through distinct

1 In this chapter, I use the term *textuality* in reference to the idea of the videogame *as* text, not simply the manifestation of text within the game. The term *videogame text*, on the other hand, will usually refer to actual, alphanumeric characters in or related to videogames. In order to avoid confusion, I will also use the term *videogame object* to refer to the game-as-text, and it is hoped that the differences will be clear from the context.

visual properties, can in turn mobilize textual associations across different genres and time periods. By recognizing the technological, material origins of these differences, we can begin to see competing patterns of expectations about videogame text that emerge from different points of view regarding the game object and outline a critical approach to videogame textuality that takes this into account.

It is my argument that the so-called *jaggy* appearance of most videogame-based text foregrounds programmatic constraint as a type of revisionary nostalgic aesthetic, whereas the so-called *fuzzy* property anticipates a different layer of material constraint, the display surface or screen, foregrounding instead a version of the text in which the traces of the programmatic constraints are less evident. I will be showing that videogame text in its original forms often demonstrates latent fuzziness, but the main distinction I am drawing here is not simply historical. Neither in the case of fuzzy nor of jaggy type should the textual form be considered an ideal one in the sense that it could be called more historically accurate than the other, so it is important to note that both fuzzy and jaggy videogame type are inherently ephemeral and present only within or in reference to an instance of playing a game – present or absent to a degree which depends on variable constraints of form including the available screen resolution, amount of addressable memory in the processor and the efficiency of the machine language. This is the primary reason why the reification of these patterns in graphic design offers a useful body of evidence for analysis, even though that reification necessarily introduces distortions of its own which may themselves be fuzzy or jaggy. Still, it is the relative differences among these representations that I am interested in here because the analysis that each extreme end of the fuzzy/jaggy continuum generates is most useful when compared to its opposite. Similarly, the question of intentionality with regard to type design and rendering is a topic that I hope to avoid except as a way of anchoring observations of difference. In other words, the purpose intended for a typeface may have little to do with its received interpretation or its use by designers, much less what the design signifies with regard to a particular audience, but intent is at least one of the discourses embedded in type. In fact, conflicting or confused intention can be a useful tool for unpacking the textuality of type design, as in the case of fonts exhibiting the influence of E13B that I discuss in chapter 2.

In terms of how it contributes to videogame studies, the question of difference as it pertains to typography opens a broader approach to the production of meaning or semiosis in games and the relationship of such meaning-making to a game's rhetorical expression. In the context of semiotics, fuzziness is an appropriate descriptor for a property of lettering because it takes on additional valence as a logical term describing a category of ontology in which entities can possess degrees of membership to a set.[2] Therefore, as a response to prior approaches to videogame analysis (much of which has a structuralist basis in semiotics), and in the context of differences among typographic signifiers, fuzziness is an ontological category (both in the sense of a type of ontology and as a type of category in a larger ontology) that points a way toward a fuzzy critical approach resonating with contemporary post-structural theories of cultural production.

To advance this argument, I am specifically focusing on examples of type that originate in or refer to the videogame industry's nascent period of the late 1970s and early 1980s. This era is useful for a number of reasons. First, the limitations on games effected by the hardware and software then available for designing and playing them result in more distinct and constrained forms. Though I will be arguing that similar constraints remain present even in modern games, the distortions and the designers' attempts to mitigate distortion are much more apparent when, for

2 XML (eXtensible Markup Language) is an example of an information structure (ontology) that can employ fuzzy logic as a principle of its organization. Like HTML, XML communicates relationships among entities in a set by nesting elements within other elements. The containing element is, therefore, said to be the "parent" of each "child" element that it encloses between its opening and closing tags. Elements can also be modified by defining attributes within the opening tag of that element that modify or further define that element without modifying its children or parent. In a fuzzy XML framework, child elements would be modified with attributes specifying the degree to which that child belongs to its parent set. To extend the family metaphor, the effect could be that an element is a member of multiple families or a "half sibling" of other elements at its same hierarchical position. I am mentioning this application of fuzziness because it is often used for managing the retrieval of semantic textual information and because "ontology" (as in a philosophical model of being) is important for my discussion of videogame expression. For more on fuzzy ontologies, see Sanchez, Elie, *Fuzzy Logic and the Semantic Web*.

example, the pixels are easily visible as individual units, rendered at a width of several millimeters each.

Second, the so-called "retro gaming" aesthetic is prevalent in gaming culture and elsewhere in graphic design associated with the seventies and eighties. In these contemporary works, the adjustments programmers made to accommodate the constraints in original gaming platforms survive today as vestigial aesthetic properties that invoke the memory of the historical console.

A third reason why the late 70s and early 80s is an appropriate period for typographic analysis is the fact that many game consoles from this era (the Atari VCS, for example) lacked a native system for rendering text, which meant that designers had to compose each required letter as a bitmap and program a call to that bitmap or sprite in order to display the letter or numeral. This cumbersome approach may explain technically why the earliest games for the Atari rarely contain text any more complicated than a title screen and numbers for reporting score or levels. For the purpose of critical analysis, however, this method of storing and retrieving textual information also means that the text is fundamentally (programmatically) part of the game's graphical environment and, therefore, part of its diegesis. In this way, fuzziness and jagginess are relevant to the entire content of the game, but their presence as features of paratextual typographic materials is an important way to approach analyzing videogame text.

On the subject of jaggy effects in graphic design, Matthew G. Kirschenbaum, whose use of the term jaggy I have adopted here, notes that "jaggies remain emblematic of the radical new ontologies of the medium [of computing]" (Kirschenbaum, "Word as Image in an Age of Digital Reproduction" 142), and further connects that emblematic status to an assumption related to material technology. Discussing Zuzana Licko's approach to type design for early Macintoshes, Kirschenbaum writes, "what began as a material limitation in hardware and display technologies was quickly accepted, adopted, and adapted as an integral aspect of the medium's aesthetic identity, an identity that has remained iconic and intact to this day, long since the technological base has shifted beyond the crude conditions Licko describes" ("Word as Image in an Age of Digital Reproduction" 142). Licko acknowledges this elsewhere, noting that the rise in popularity of fonts like those in her "Lo-Res" font package (Figure 4-1) owes a direct debt to videogame type:

> For these designers, and their audiences, who grew up playing videogames and now surf the Internet, low resolution type is no longer an alien, difficult-to-read, crude computer phenomenon. ... Through our everyday encounters with computers, the idiosyncrasies of bitmaps are disappearing – visible pixels are becoming accepted as the natural mark of the computer, like brush strokes on an oil painting. (Licko)

Though Kirschenbaum and Licko refer mainly to jaggy type in relation to its figuring of the computer medium, I argue that the same emblematic status exists for videogame type itself, and that as an emblematic property, the jaggier a type sample (that is, the lower its implied resolution), the more likely it is that the jaggy type is meant to specifically suggest videogaming rather than computing more generally. Moreover, when taken together, the dual qualities of fuzziness and jagginess as they are manifested in games and in artifacts of gaming culture reveal the richness of the aesthetic identity Kirschenbaum discusses and its close involvement with the material dimension of videogame interaction. Understanding the implications of these properties is an important component of a critical approach to videogame materiality and mediality.[3]

The Differences

In practice, fuzzy and jaggy properties can both be more or less present in any given sample. Also, as I will elaborate, both properties can be extended and modified in different ways, conveying various inflections on the associations they express, and it is easiest to see these differences between the two in extreme or exaggerated cases, even more so in print or high-resolution graphic design where fuzziness or jagginess must be an intentional part of the design. To

3 As I am using it here, the term *materiality* refers to the sense in which a game is made up of some material, "physical" substance that influences its expression and reception. *Mediality* is a related term that refers to the content of the game as it expresses and is influenced by its medium; in this case, the medium of the videogame is a conceptual category defined by what makes videogaming unique. In other words, *materiality* addresses a game as a specific, historical object, and *mediality* addresses a game as a mode of expression. Both concepts are aspects of videogame textuality.

facilitate this discussion of text associated with videogames, I adopt the term *videogame paratype* to refer to this category of typography where a contextual association with videogames operates through the referentiality of the type design in question.[4] Paratype is, therefore, important for this discussion not only because it demonstrates the cultural significance of fuzzy and jaggy styles, but also because it can be seen as a material reification of visual properties that are otherwise ephemeral and technically unique to each instance of gameplay. In other words, each native instance of video game type will be influenced by factors like the age and focus of the television monitor or the level of RF interference.[5] When a designer chooses to imitate videogame type in some other context, she chooses a particular degree of fuzziness or jagginess that imitates the ephemeral properties of a hypothetical game experience and modifies it according to the aesthetic goals of the new context. Of course, that new context may introduce its own set of constraints or distortions,[6] so the process of transferring relevant visual features from one medium (videogames) to another (print, for example), also demonstrates the recursive status of signification for game text. In other words, videogame paratypography generally works by attempting to mimic the *production* of videogame text by creating print-based approximations of screen-based effects like scan lines and halation. However, what actually occurs in the creation of paratype is an analogical demonstration of the *reception* of videogame text in which the properties of the receiving medium (the amount or density of ink) perform a function analogous to the subjective receiving apparatus of videogame text – both phenomenologically (where the image printed on paper stands in for the

4 The term *paratype*, used in this context, takes on some different meanings and associations which are developed in Chapter 2. Tthe term is already in use within taxonomic nomenclature to designate individual elements of a set that are of the same order as, but distinct from that set's holotype (the originary member or first specimen of the set).

5 RF (Radio Frequency) interference, also known as EMI (ElectroMagnetic Interference), occurs when an electronic receiver device like a radio or television intercepts unintended signals from an external source. The result is that the primary signal is decayed or distorted by the presence of signal noise (Wikipedia contributors, "Electromagnetic interference").

6 Some of the examples below are illustrations of screen images from advertisements for videogames published in comic books. The relatively cheap paper and rapid color printing processes in the time period under consideration (early 1980s) introduced distortions which mimic, exaggerate, or lessen effects of jagginess or fuzziness inherited from the original game image.

flattened image on the player's retina) as well as culturally (where the factors influencing the selection and arrangement of videogame paratype stands alongside the discourse influencing the design and display of the original videogame typeface). This recursion reveals the immanently discursive value of videogame type as a basis for destabilizing and disrupting formal analyses of videogames. If one considers videogame typography to be an emblem of game ontology, it is a category with an uncertain lineage, comprising artifacts that flicker between first and second-order signification and between text and paratext. In other words, videogame type is always already fuzzy. Before exploring the implications of this fuzziness as an interpretive model, I will first examine the visual manifestations and details of both jaggy and fuzzy videogame type.

Jaggy Type

I am using the term *jaggy* to refer to text that features jagged, pixelized edges like the first several examples which follow. While both jagginess and fuzziness are relative terms which in most samples are not exclusive of one another, each can manifest in different ways that will signify different material associations. The examples in figures 4-2, 4-3 and 4-4 demonstrate the basic features of jaggy type: stair-stepped angles and curves instead of smooth arcs, a sharp distinction between foreground and background colors, (on-screen) pixels aligned flush with the actual screen's pixels, and rectilinear units which build each letter shape out of uniform bits or pixels and imply a containing grid. Each of these examples also demonstrates a stylized pixelization in the sense that their designers almost certainly intended to create a jaggy effect.

Figure 4-2, the logotype for the journal, *Game Studies,* exaggerates its jaggy property by placing a visible separation between the simulated pixels. The uniform letter-spacing is consistent with the width of the characters' strokes, so the total image implies a strict grid framework in which individual units have been darkened while others have been left transparent, mimicking how an LCD monitor displays images. By contrast, the *Destructoid* logotype (Figure 4-3) retains and further exaggerates the stair-stepped jagginess, but spaces the letters closer than their implied pixel widths should allow. The result is more typographically effective than the *Game Studies* logotype, but *Destructoid* has sacrificed the building-block or grid constraint in order to achieve this balance. In that sense, it features less jagginess than *Game Studies*.

Figure 4-4, an image from the web comic *Penny Arcade*'s advertisement for their 2007 convention, demonstrates a jaggy effect on a different scale. In this case, the designer has chosen a font specifically intended for use as a bitmap that retains its hard edges rather than incorporating anti-aliasing. That is, the stair-stepped upper edges of the capital *A* character are not a consequence of the operating system's native font rendering that normally approximates a curve with the smallest incrementation possible. Instead it is a pre-defined saw tooth which emphasizes jagginess in order to reference videogaming. Normally, the default font-rendering schemes on Windows and Macintosh computers apply anti-aliasing to textual glyphs by slightly blurring the pixel edges in order to lessen the saw-tooth effect. The *PAX* banner overrides the anti-aliasing as an intentional effect that mimics that kind of lower-resolution effect within the higher-resolution display on which the image was captured. Though its "jags" are less pronounced than Game *Studies*' or *Destructoid*'s, the *PAX* banner still operates within the jaggy aesthetic.

Besides their appearance, what all of these examples have in common is an intentional resemblance to classic videogame text – specifically a stylized version of videogame text that adopts a jaggy aesthetic as its basic style. Figure 4-5 shows a screenshot from the Atari VCS version of *Dig Dug* (1982) that demonstrates jagginess unconsciously or in its natural state. Unlike this screenshot, the examples from the three paratypes above demonstrate jagginess self-consciously as an explicit reference to the naturally jaggy type supposedly native to lower-resolution displays. In other words, the shapes of the letters in videogame paratype can be considered typographically expressive in the sense that their appearance invokes a videogame antecedent that reinforces the context of the image itself (a game studies journal, a game blog, and a gamer convention, respectively), and they derive this association by simulating the low resolution technology that constrained the original forms of videogame typefaces. The screenshot from *Dig Dug* is un-self-conscious in its jagginess because its appearance actually is a consequence of the technology used to display it – in this case, the open source Atari VCS emulator, Stella, created by Bradford W. Mott.[7] This is significant because the emulation naturalizes jaggy text as a default, bypassing the sometimes fuzzying effects of historical display hardware (aging color televisions, for example). As later examples will show, the fuzzing or blurring that does occur in some display environments actually enhances the appearances of shapes such that, for example, stair-stepped arcs appear to have greater fidelity to a curved line when the stair-stepped pixels bleed into one another.

Though it is not my argument that either the fuzzy or jaggy effect is more faithful to its original form, it is important to note the resemblance between figures 4-2, 4-3 and 4-4 and the screenshot captured from Stella. Like most emulators, Stella attempts to recreate a software environment or platform but does not claim to recapture any actual situation of playing the games in their original forms. Features such as the tactile response of Atari's joysticks or the heavy mechanical clicks of toggling the console's switches are simply beyond the pale of software emulation. In one sense, it is tempting to view the Stella image as more pure or faithful to the original than an image derived from plugging an actual Atari device into a modern television because the screenshot captures Stella's output directly from the computer's video card and thereby avoids any interference or distortion from the monitor. Therefore, it seems that Stella actually improves on the original gaming experience because it optimizes the output for the crisper displays of modern computer monitors, rather than the comparatively more primitive displays of 1970s and

7 Because Stella is open source, many individual contributors have added to its features and ported it to various other platforms. Noted contributors are listed as members of the Stella Team. See <http://stella.sourceforge.net/theteam.html> for more information and a current list of major contributors.

1980s TV sets. By contrast, MAME (Multi-Arcade Machine Emulator), another open source emulator, provides a means for simulating the blurring effects of display devices common to older arcade machines.[8] It is not necessarily the case that these simulated effects are actually more accurate, but the existence of this feature acknowledges the difference between fuzzy and jaggy graphics in actual gameplay and the similarity between the un-filtered image in figure 4-1 and the logotypes in figures 4-2, 4-3 and 4-4 betray a design choice based on an assumption about the natural state of videogame type.

Whether or not any of the jaggy designs above are actually based on someone's experience with an emulator, what I am arguing is that this jagginess is simply one way of portraying the constraint of display hardware and that even this version bears the marks of influence from its digital origin. The alternative – text that is apparently or intentionally fuzzy either as a result of its native display condition or as an intentional design choice – also betrays its electronic origins, but it does so by arresting or calling attention to the production of the game image at a different stage of its formation, reception, or transmission. Accordingly, text that is jaggy stakes out a position closer to the bitmap sprite[9] that is stored in the assembly code for the game, whereas text that is fuzzy positions itself after several layers of interference and degradation.

To summarize the properties of the jaggy style, Table 4-1 illustrates each of the basic features identified in the examples given above. These examples are meant as somewhat exaggerated cases, but even among these, it is clear that jagginess is an important feature of videogame type, both as paratype and in native examples. Its use is widespread in logotype and graphic design meant for use in a videogame context, and its exaggerated pixelization is a standard marker for technological nostalgia. Fuzzy type accomplishes these same associations, even though it is somewhat less

8 The options in a standard Windows distribution of MAME include "scan lines" and differently sized apertures, all of which are accomplished by overlaying semi-transparent PNG images over the video output of the emulator. The expressed purpose of these overlay effects is to improve the authenticity of the experience, and users can create their own images in order to finetune the artifactual fidelity of their experience.

9 In the context of videogames, a "sprite" is a small unit of graphics code which is called whenever a given object or state of an object is needed.

common, so its connotative difference from jaggy type is one way to expose discursive functions of videogame typography.

Fuzzy Type

Fuzziness is a somewhat less well-defined visual trait, so its examples are accordingly more difficult to characterize. Nevertheless, their existence as a distinct category and type of expression demonstrates that there are important expressive differences between fuzzy and jaggy type which bear upon our understanding of videogame aesthetics and expression. Most importantly, fuzzy type distinguishes itself by blurring the boundaries between the shape of the letter and the background. Whereas jaggy type (in screen-based media) is designed in such a way that its edges align closely with the display device's pixels, fuzzy type in the same media blends more or less gradually (as allowed by resolution) with its background. The essential "blockiness" of the form might still be visible, but the shapes of the blocks are less uniform and their boundaries less discrete. In its different forms, fuzziness can demonstrate different valences or essential qualities that express different ideas about the media it references as well as its actual medium. Like jagginess, the different examples and uses of fuzziness define a broad category of type, so the examples below illustrate some prominent and even exaggerated examples.

Though the edges of this first image are relatively crisp, figure 4-6 demonstrates one kind of fuzziness that exaggerates the specific effects of anti-aliasing on the text. Anti-aliasing is a transformation of images or text in screen-based media that introduces an algorithmic blurring to simulate that shape at a higher resolution. The blurring that is foregrounded in Brucker-Cohen's banner image actually creates an impression of a more continuous line when the text is reduced to a 12pt size. Figure 4-7 shows the difference the blur makes by reducing the text in the image to what may have been its original size. The image on the left preserves the blurring before scaling the image, while the one on the right trims out the exaggerated blur before scaling. The effect is subtle, but the main difference is that the weight of the lines in 4-7A appears greater because the single pixel wide lines "borrow" weight from the lightly colored surrounding pixels. Exaggerating and enlarging this effect, as Brucker-Cohen has done, emphasizes the tension between software and hardware constraints. Anti-aliasing technology exists in order to overcome distortions which would otherwise appear as a result of the screen's pixel grid, so grotesquely exaggerated anti-aliasing points out the proprietary discourse involved in minimizing the presence and influence of the screen.

Figure 4-8 is a different kind of fuzzy effect in a videogame paratype that specifically invokes a videogame textual sensibility. The full context (Figure 4-9) depicts the point of view of the interior of an X-wing fighter and suggests putting the player in the pilot's seat, defending the Rebel Alliance by defeating the Death Star. The text's referentiality here is automatically twofold. On the one hand, the text adopts the information display (HUD) visible in the cockpit, but more importantly, it mimics a low resolution videogame display, such as the one produced by the Atari VCS. In fact, the textuality expressed in the HUD this text references already owes more to videogame representations of heads-up display than to any referent in the real world. Like jaggy type, the individual units that make up each letter shape in figure 4-8 are each visibly distinct from their neighbors, but the makeup of each dot is subject to the slight distortions and indeterminacies of the printing process, resulting in a slightly rounded shape. Looking even closer at each dot (Figure 4-10) reveals the sub-units, halftone dots, that form the color of the lettered dots. In this case, fuzziness manifests in the shape of the units which form the letter shapes, as well as the optical illusion which creates a slight halation at the junctures of the lines.

Blip magazine was a short-lived publication appearing in the heyday of the arcade and home console boom (1982) and disappearing in the subsequent industry crash that occurred in 1983 and following.[10] Published by Marvel Comics, *Blip* apparently did little to stand out from the crowd of other gaming magazines, but the design of the logo (Figure 4-11) reveals an interesting allegiance to the appearance of videogame typography. The lines are relatively crisp, but the fuzziness is apparent in the outlines of the characters and the horizontal gaps which simulate scan lines on CRT monitor, the same kind of lines that can be simulated in MAME when running it from a command line with the suffix "-effect scan line." In this particular design, the scan line effect is further exaggerated by the way the visible portions of the letter "bulge" outward between the lines. This logotype may be considered fuzzy because, like the *Death Star Battle* paratype, the individual units are not uniformly rectilinear.

Figures 4-12 and 4-13 show detailed views of an ad that appeared in *Batman #365* for the never-released title *James Bond as Seen in Octopussy*. The type in the screen image is apparently designed as a faithful representation of this title's gameplay. Though we can not be sure of this, the design and use of the text in this example appears consistent with other games from this era and for this platform, so it is relatively safe to assume that this image is faithfully representative. At any rate, it is positioned rhetorically as an accurate account of play. However, the typographic detail apparent in figure 4-13 works against the claim of authenticity, since the visual features of the numerals 1045 appear to have smooth angles and curves. If this is a faithful representation, it is faithful to how the text would have appeared on a slightly unfocused CRT monitor where the halation and phosphorescence of the monitor might have accomplished a smoothing effect similar to the effects of anti-aliasing. Furthermore, the fine detail of this example has exceeded the capabilities of the printing mechanism so that the printed characters themselves appear blurry on the page. This is fuzziness of two orders that takes on significance as a discursive property because the second instance of fuzziness (distortions resulting from the printing process) obscure and

10 The industry blog *GameSetWatch* offers an interesting note on *Blip*'s place in history: "it launched February 1983 and published its final issue in August, thus becoming the first magazine 'victim' of the Atari crash" (Gifford).

prevent access to the intentionality of the original instance (a designer's attempt to faithfully mimic the text on a CRT screen).

Half-Life 2 features a gracefully minimal HUD which calls the player's attention to particular elements in an unusual and subtle way. In figure 4-14, the player has just activated a wall-mounted power unit to recharge his avatar's HEV (Hazardous EnVironment) suit. In the logic of the game's diegesis, the suit is what projects the HUD to the player-character, so the suit indicates that it is receiving energy by increasing the brightness of the HUD's power indicator. That it does so by revealing halation and scan lines raises an interesting question about the display technology. If the player-character is indeed wearing a helmet,[11] then the game-world explanation for the appearance of fuzzy effects must be that the HUD must be projected onto the surface helmet's face plate using a kind of raster display. In this case, the sudden manifestation of fuzziness reinforces the connection between the player and the player-character by drawing attention to the second screen within the fictional helmet. Thus, fuzziness performs a diegetic function in terms of how the diegesis of *Half-Life 2* is constructed.

Finally, as a counterpoint to the *Dig Dug* logotype in figure 4-5, figure 4-15 is an image from the same Atari port of *Dig Dug* as played on an actual TV with an Atari VCS. The phosphorescence of the screen and other factors like the unavoidable RF on the connection all contribute to the fuzziness of this image. This is a TV from the late-1990s, so its focus is likely crisper than many TVs may have been when the game was originally played in the early 1980s. Still, one can easily recognize some effects similar to that of anti-aliasing; the curves on the upper edges of letters are smoother and more consistent with smoothly curved lines. Unlike the previous examples of fuzzy type, this photograph of *Dig Dug*'s logotype contains the features identified above but in a way that is not self-conscious. The apparent distortions are all natural to the current

11 There is some debate about this. Gordon Freeman is usually depicted without a helmet, but since he is rarely (if ever) visible on screen in *Half-Life 2*, and the informatic display appears only when Gordon is wearing the suit, it seems logical to conclude that he must be wearing a helmet during gameplay.

display situation of my TV's 13-inch screen. Taken together, all of these examples indicate some common features for fuzzy videogame typography: non-rectilinear sub-units, smooth or feathered edges, screen based distortion (i.e., raster lines and anti-aliasing), greater contiguity among sub-units, and an overall formal indeterminacy.

Plotting the Differences

The previous two sections offered a broad survey of typographic qualities associated with videogame type. Since type that is predominantly jaggy or fuzzy may contain a combination of several different features, in this section I formulate an arrangement of typographic properties that distinguish fuzziness from jagginess. Though what follows may resemble a formal ontology, it is not my intent to form an exhaustive account of videogame type or to essentialize either fuzzy or jaggy type with reference to discrete and necessary qualities. All of the oppositions discussed below should be considered provisional observations for the sake of discussion, and their resulting emplotment is meant to reveal the conceptual implications of fuzzy and jaggy text for understanding the materiality of video games.

The first typographic feature related to concepts of fuzzy and jaggy is the sense in which text can be either sharp or smooth in the way it is presented. Figure 4-16 illustrates these alternates as a continuum. In screen-based media, sharpness is manifested by aligning the pixel edges of the letter figure flush with the edges of the screen. In other words, there is a clear boundary between two fields of color that embraces the grid of the actual display device. In print media, sharpness appears as fields of solid, uniformly shaded color with clear and precise boundaries. Softness, on the other hand, manifests in screen media as a gradation of color value from one field into the next. This either requires a resolution high enough that individual pixels are not obvious (i.e., so that altering the color pixel-by-pixel appears smooth) or it can appear as a result of bleeding or halation on some monitors. In print media, softness is manifested as any of several blending, shading, or feathering techniques or by imprecise printing (e.g., when an excess of ink causes the figure to bleed into its background). Sharpness may be generally associated with jaggy type, and smooth with fuzzy, but this is not necessarily the case. Figure 6 above is predominantly fuzzy, but its features are definitely sharp.

Videogame type can also be categorized by the degree to which its forms are granular or fluid. Granularity here refers to the separation of the pixels, dots, or other sub-units that compose the letters. By contrast, contiguity refers to the connection of the pixels, dots, or other sub-units, particularly as they bleed into or overlap one another. Figure 4-17 opposes these two attributes. Type that is more granular emphasizes the influence of the grid on the letterform, and type that is more contiguous emphasizes the ability of the letterform to overcome the constraints of the grid. Granularity is more likely to be associated with jaggy type, and fluidity is more likely to be associated with fuzzy type. But this is even less consistently true than in the case of the sharp/smooth distinction. Figure 4-8 is predominantly fuzzy, for example, but it exhibits a good deal of granularity because of the way the pixel units are separated. Furthermore, the printing dots visible within the pixels manifest fuzziness by working against the sharpness of the forms. In another sense, the fact that granularity is distributed through two distinct surfaces (the printed page itself and the fictional screen which displays the characters) also contributes to its fuzziness. Figure 4-4 exhibits a fair amount of fluidity because contiguous pixels merge with no visible boundary between them, but the jagged edge of the letterforms and the prevailing sharpness of the form associate it more closely with jagginess.

To illustrate how these two oppositions (sharp/smooth and granular/fluid) contribute to defining fuzziness and jagginess, figure 4-18 shows each opposition pair as an axis on a Cartesian plane. In this case, the x-axis corresponds to the sharp/smooth continuum, and the y-axis corresponds to granular/fluid continuum. In each of these, the values express a range of possible quality assignment (for example, more or less fluidity), with the region around the 0,0 coordinates remaining neutral. The diagonal lines correspond to the jaggy vector (extending into the top-left quadrant) and the fuzzy vector (extending into the lower-right quadrant). These vectors essentially divide the territory into two halves, indicated in the graph by the bisecting dashed line. Following the vectors' arrows, samples of type which more clearly feature sharpness and granularity are more likely to be associated with jagginess, and samples which feature softness and fluidity are more likely to be fuzzy. In this way, samples of videogame type can be plotted according to their visual characteristics and associated with either category. The graph may work in reverse as well. That is, it may be the case that isolated type samples which lack any contextual association with videogames, may gain the association of that context if their relative emplotment on the chart places them at the extremity of either side of the continuum.

Other qualities help define fuzzy and jaggy type, but these two oppositions, sharp/smooth and granular/fluid, help classify the majority of the samples. Other qualities like rectilinear bowls and counters, an implied resolution with visual raster grids, and various versions of decay can also contribute, but all of these can be wrapped up in one final means of comparison, medial layering. In nearly all of the examples of either fuzzy or jaggy type, the difference is important because it reveals some history of medial transformation. In most, there is some sense in which the form mimics shapes, texture, or other details that are based in another medium. Even the Lo-Res font collection, which Licko designed specifically as an expression of screen-based aesthetics, owes formal characteristics to print-based letterforms.[12] The other examples all contain some kind of medial discourse that implies a narrative of their adaptation. In most cases, the designs imply that they have been transferred from a lower resolution display to a higher one or (in the case of the

12 Licko has stated that she designed fonts like Oakland specifically for the screen because many contemporary fonts (in the 1980s) attempted to transfer popular typefaces into the unwieldy grid of the computer. The success of Licko's creations, as well as the ubiquitous Chicago (designed by Susan Kare for the Apple Macintosh), depends on their using the grid effectively as a nascent creative environment with very different constraints on legibility.

Death Star Battle and *007* ads) to print. The typical result in these cases is that the new medium in some way exaggerates the constraints of the original medium, and it does so by using the tools of fuzziness and jagginess. This discussion of referentiality can be summarized as a vector which accounts for the mediality of a videogame type sample in terms of its discursivity. Whether in reference to specific game or to a particular display technology, mediality is depicted as a kind of layering in which the videogame text (in-game, logotype, or paratype) refers to some display device other than the one on which the user is currently accessing it. Because this is accomplished by invoking some kind of constraint, these instances of type also refer and call attention to the constraints of their own display medium.

In this way, any instance of videogame type is composed of three medial layers, at least one of which is a videogame. Figure 4-19 illustrates these layers with an advertisement for *The Empire Strikes Back* (Parker Brothers, 1982; Atari 2600). The first layer is an image of the game itself, a sample from the actual, historical game, which is placed above the advertisement in the diagram. This relationship illustrates the idea that the game image's influence on the print image is interferential. That is, the printed image is inflected by the implied presence of the game image by manifesting or stylizing its properties. In other words, the printed image adopts the aesthetic qualities of the game image *using* the tools available to the printed image and subsuming it under the visual rhetorical patterns of the print ad. The final layer is the projected, imaginary image of the game screen as the printed image depicts it. In the diagram, this appears behind the print image because it is ultimately inaccessible and has no necessary relationship to the game itself. This final layer is necessary because of the indeterminacy implied in the relationship between the first two. Given a printed sample, we have no way of knowing whether the original referent was itself fuzzy or jaggy, so while the first layer is the actual antecedent, this projected layer is the imagined antecedent. In this way, the *James Bond* example cited earlier becomes more interesting since no known production copies exist. In other words, there is no way of assessing the fidelity of the printed image to the original, and the imaginary layer is, therefore, the most faithfully real artifact of the nonexistent game.

Of course, there are some significant problems with this model, the most obvious being that each layer in the triad above is subject to cultural and technological influences in its own right that impact the fidelity of each in different ways. Specifically, the domains of fuzziness and jagginess manifest across and among the three layers. That is, fuzziness or jagginess can be apparent within each layer, but each can also be the means by which one layer relates to the other. Ultimately, the advantage of this model lies in the fact that when it comes to analyzing the visual appearance of typographic forms, one need not invoke a representative quality of the text or numeric object, except in the sense that it refers to its corollary in other medial contexts. In other words, there is no such thing as a real 0, so the relationship illustrated in figure 4-19 demonstrates how similarity and difference among examples of videogame type do not deal only or primarily with the fidelity of the representation but, rather, express a discourse of mediality. That discourse can be enhanced and better clarified by combining it with the graph in figure 4-18. In this combination, the mediality vector becomes a z-axis along which the fuzzy/jaggy graph is extended at each of the three moments.

This final graph, figure 4-20, attempts to combine the previous two graphs in a broad theory of videogame materiality. The result, however, reveals a number of problems with this emerging model. To begin with, the text is anchored paratextually. Even if the printed image is largely faithful to its screen antecedent, the fact that its discursive power resides in reference to a non-local, ephemeral manifestation of game makes it a dubious anchor for any kind of system because it is subject to its own constraints, which are completely independent of any game. The game image itself, in reference to its materiality, projects fuzziness or jagginess as a consequence of subjective and alterable material condition, so its respective relationship to the print image (in terms of fuzziness and jagginess) depends on at least two indeterminable interferences – the infelicities of print technology, and the artist's personal experience of actual gameplay. This game image must also be further qualified by the media used to record and transmit the image, and similarly, the means by which I have captured and recorded the image here have an impact on its ability to contribute to a persuasive argument about difference. Most importantly, the idea that different versions of the game image exist in relation to each other runs the risk of essentializing each state as a discrete or unitary existence or implying that each is a variously inflected instantiation of a primary or ideal image. This latter implication would be especially tempting if the images in question referred to real-world objects, but since these game images including constantly changing elements like the player's score suggest at the very least that any approach to their textuality must account for their constantly-shifting referent if it is to encompass the full range of videogame expression . Moreover, comparing the printed image to an actual screenshot hints at technological determinism by fixing each of their visual rhetoric in terms dictated by an arbitrary other. In short, the approach outlined in the last several figures is itself inherently and excessively jaggy because it employs sharply granular constituents to develop its model, much like many of the articles in the journal *Game Studies* which rely on similarly structuralist strategies. It is ironic, therefore, that among the examples of jaggy type, the *Game Studies* logotype (Figure 4-2) is the most obviously jaggy. As an alternative, what is needed is a fuzzy approach that begins with the technological and cultural affordances of the medium and embraces the complexities and subtleties of its materiality.

Type on the Television Screen

Videogame typography is a relatively recent entry in the history of type, but its fuzzy and jaggy qualities and the relationship of those properties to medial discourse have an important predecessor in type design for television, especially as they relate to continuity and fluidity. Furthermore, an understanding of the differences between type for television and type for videogames helps clarify the relationships between the code and the interface levels of Montfort's model. The history of typography is closely entwined with the technology used to print and display type, and the CRT raster screen is one case where typography has had to adapt to the felicities of the new display medium. Hermann Zapf notes in a 1968 article that "technical developments have changed our alphabet since the invention of typecasting…today there may be the special problems of reading machines; tomorrow, perhaps, electrostatic printing methods or developments in laser-beam techniques"(Zapf 351). Or, to put the question in terms of an even wider range of evolution, "just as the monk's quill produced the alternating thin-and-thick strokes of his characters and the hot metal type of the first line-casting machines determined the present standards for spacing, the typography of tomorrow is being shaped by today's tools of graphic communication" (Spangler 38 - 39). The *Journal of Typographic Research* (now *Visible Language*) is a valuable resource for studying this phenomenon of type evolution, and several articles in its first volumes (beginning in 1967) report on research dealing with the typographic considerations of raster and CRT technology[13] as it determines typography.[14] In one such study Rudi Bass, then a graphic designer at CBS, discusses the research that went into developing their CBS News 36 typeface. The challenge was to optimize type for an unpredictable set of factors that could degrade its legibility:

> The cathode ray tube (CRT) face introduces new problems into the aesthetic-technological relations that determine legibility. … Final quality control rests in the hands of the viewer. Widely varying reception conditions, and more often than not, an aging and badly-tuned

13 In addition to its common use in television monitors, cathode ray tubes (CRT) are the core composing medium for phototypesetting machines like the Linotron 505 (produced by Monotype in 1967) which are also notable for storing and retrieving font information digitally (Wallis). The Linotron 505 was capable of resolving characters at 650 vertical lines-per-inch or the higher quality 1300 lines-per-inch (Frutiger, "Letterforms in Phototypography" 333).

14 The fact that these research documents appear alongside analysis of book design and contemporary trends in concrete poetry indicates the scope and depth of this excellent publication. For other articles in this journal on CRT printing and display technology, see Frutiger, "Letterforms in Phototypography," Crouwel, "Type Design for the Computer Age," Mergler and Vargo, "One Approach to Computer Assisted Letter Design," and Shurtleff, "Relative Legibility of Leroy and Lincoln/MITRE Fonts on Television."

> home screen affect the quality of reproduction. To overcome what seems to be bad reception, many viewers exaggerate brightness and contrast which only adds to the deterioration of the letterforms. (Bass 257 - 358)

Successful design, they discovered, involved balancing the stroke width of characters (so as to avoid their disappearing under too-harsh contrast) with the amount of negative space within and between characters, as illustrated in figure 4-21. One specific problem was the tendency with certain fonts for counters and interior angles to become filled in with excess light, completely fusing some counters and causing adjacent characters to bleed into one another.

Their solution was to punch out lacunae at the inner corners of letterforms which had the effect of trapping excess light. The resulting letterforms (are more legible and more open because of a different order of distortion imposed upon the letterform prior to its being transferred to a screen and broadcast. This technique is similar to the solution Matthew Carter found in 1978 for preparing Bell Centennial (Figure 4-22). This commissioned typeface was to be used in phone books, so it had to sustain legibility at very small point sizes and on cheap, thin paper (Sherman). Like CBS News 36, Carter's design anticipates that the letterforms (in this case, composed of ink) will tend to overflow and fill in narrow counters, so the solution is to extend negative space to absorb that excess.

Both Bell Centennial and CBS News 36 are adapted for situations with obvious and significant differences from videogames, but the sense in which they have been adapted for specific use means that when they are abstracted from that use, they still bear the marks of their adaptation. This is how typography can be discursive. What each of these typefaces demonstrates, moreover, is a logic behind composing letterforms that takes into account at least two levels of translation which will occur: transmission and display. *Transmission* here refers to the means by which visual information is encoded or encapsulated for transportation to a remote audience. The act of writing is always about addressing an absent audience, so designing typefaces to account for the hazards of transmission is a basic act of any instance of writing. In the case of television fonts, transmission involves an original signal being encoded, broadcast, and translated by appropriately tuned receivers. In terms of the phone book, transmission is the process of photocompositing and printing the book. The other level, *display*, refers to the television reception (in the case of CBS News 36) and the printed page (in the case of Bell Centennial). In other words, while transmission manifests its discourse in generality, display manifests through singularity – specific instances including aleatory effects which, though technically a subset of the transmissive discourse, gain significance by being non-normative with regard to the transmitted set.

Returning to *Berzerk*, the discourse embedded in the Countdown-like numeric characters is analogous to the discourse of transmission and display proceeding from CBS News 36 and Bell Centennial. In other words, if Countdown is a design that foregrounds the effects of halation on unadapted letterforms, then the use of that form in such a way that the display medium reintroduces the characteristic distortion expresses the idea that the screen is an important part of the aesthetic composition of the game image. Brignall's commentary with Countdown is originally ironic in the sense that it mimics (in print) a variety of accidental screen-based distortion, implicitly mocking the failures of the screen as a typographic medium.

The simplified, jaggy version of Countdown that appears in the *Berzerk* emulation is ironic in a different sense because it reverses the logic of the original, fuzzy version that was already a demonstration of that which Countdown was meant to comment on. In terms of the levels in Montfort's model, typographic form is a valuable carrier of the discursive relationships among each level because the primary signification of typographic shapes proceeds without reference to an intrinsic ideal or objective correlative: there is no such thing as a natural 3 in the real world. The versions of the Countdown/*Berzerk* 3 which convey style, connotation, and commentary reflect, therefore, on the context where they appear and the means by which they are displayed.

Additionally, none of these 3s in the various incarnations of *Berzerk* raise a question of fidelity to an original 3 or prime example because each can be expressed in terms of the others. The fact that these interrelated expressions can in turn relate to the material constraints of raster or vector display as well as the discrete logic of digital encoding affirms the close relationship among the five levels in the platform studies model and the importance of unpacking the discursive nature of their configuration. In this analysis, typographic expression is one answer to Montfort's suggestion that the difficulty in demarcating the interface level (for example, whether or not to include paratextual material such as cabinet art) might be addressed by a "similar multi-level model of material analysis" (Montfort, "Combat in Context" fn 4). I have already introduced the concepts of transmission and display as two possible levels of materiality. The final section of this chapter explores a multi-level, fuzzy-ontological model including these levels.

Conclusion: Toward a Fuzzy Critical Approach

The implied challenge in Aarseth's remark is echoed with reference to typography by John Cayley a few pages later: "After all, do constraints that are imposed on the manipulation of pixels in order that they produce the outlines of letters tell us anything about those letters or the words which they, in turn, compose?" (Cayley 208). Cayley's answer to this rhetorical question seems to be "no,"[15] and he is insisting instead that the letter has always been digital in the sense that its powers for abstraction and description are unmatched by any other human technology. Aarseth's logic is similar, and the way he separates his three "aspects" could be phrased as a rhetorical question similar to Cayley's: "Does the material/semiotic aspect of a videogame tell us anything about its rules or how it is played?" In order to answer both questions in the affirmative, one must consider the visual status of fuzzy and jaggy videogame type. Kirschenbaum states that jaggy type

15 See the *First Person* discussion thread on this essay, including responses by Nick Montfort and Johanna Drucker, at http://www.electronicbookreview.com/thread/firstperson/programmatology. In this discussion, Cayley affirms the importance of the pixel and its material discourse but remains suspicious of its predominance as a figure for digitality in both digital theory and practice.

is "emblematic" of computing, so if both jaggy and fuzzy type can be emblematic of videogaming, then the presence and delineation of pixels within text-as-image is an emblem of videogaming that is both icon and analogy.

Aarseth's aspects insist on separating two distinct layers of discourse within the videogame object: rules and signifiers. The first aspect works through the second to produce the third, gameplay. This model relies on a clean break with other media because Aarseth is insisting on the novelty of gaming and its difference from received modes of cultural production such as literature and film. Under this model, in other words, gaming is jaggy. Like jaggy type, Aarseth's and other similar formal models proceed by identifying and accounting for each constituent element and placing it in a determined relationship with other elements. This method is "sharp" (following the definitions outlined in the first section) because it isolates games from other media, and it is "granular" because it separates the components of gameplay into discrete functions. It is fitting, therefore, that among the examples of jaggy type, the *Game Studies* logotype (Figure 4-2) is the clearest and most exaggerated demonstration of jagginess. The individual pixels of the letterforms are identical in size and shape and do not overlap or otherwise come in contact with each other; this is an appropriate emblem for a journal (founded by Aarseth) where many of the articles adopt structural approaches to dissecting games in order to pin down meaning within specific ludological contexts.

Still, the actual makeup of videogame text presents a problem for models such as Aarseth's or Jesper Juul's more complex rendering of "rules" and "fictions" in his book, *Half-Real*. In expressive typography, for example, letterforms hover between word and image, occasionally but never fully resolving to either mode of signification at the exclusion of the other. In the same way, text in videogames (particularly numeric characters) is a visual function of the *rules* of the game which can only be expressed in terms of the "materiality/semiotics" of the game but which in turn signify the relationship between the game's rules and its fictions. In other words, videogame text (for example, in a heads-up display) is both part of some fictional world that the rules generate and part of the rules themselves. The factors which influence and constrain the forms of the numerals pervade the game logic, so the numeric characters are self-reflexive emblems of videogame textuality. In this way (to answer Cayley), the constraints signified by the pixels *do* tell us something about the game and its meaning.

In contrast to jaggy, structurally-inclined models, Bogost's discussion of unit operations seems an attractive, fuzzy alternative. Because the approach can operate at potentially any depth or scope (that is, units can be cultural, psychological, computational, material, etc.), unit analysis demonstrates a degree of "softness" that distinguishes it from structuralist models. The analytical grain can be as fine or coarse as the material dictates, so the delineation of the specific constituents which compose an expressive unit is necessarily imprecise and adaptable. In other words, unit operations can be considered fuzzy for their uncertain boundaries and fluid interchange of constituents. Furthermore, like fuzzy typography, unit operations are no less emblematic or representative of the games they refer to. In fact, because fuzzy type invokes the specific, material artifact of the screen, and because unit analysis can include personal context,[16] the two are well-suited for one another. To put it generally, the ludological school of thought (of which the allegedly narratological school is a logical contradistinction) is more concerned with the idea of games and the formal ideals which link all instances of gameplay. By contrast, Bogost's unit analysis and its intellectual compatriot, platform studies, approach videogames as texts that include the often messy and imprecise infelicities of material culture that help define videogame expression.

Materiality is fuzzy. This is the meaning that fuzzy type ultimately expresses. In relating the ontology of game objects to fuzzy ontology, one might be tempted to state that the core

16 See, for example, Bogost's sample unit analysis of the Tom Hanks film, *The Terminal*. Bogost's reading of it in terms of unit operations includes the irony of his viewing the film in-flight.

characteristic of digital materiality is uncertainty leveraged against the finite determinism of true and false, 1 and 0, that underlies each digital artifact and ensures its faithful reproduction across multiple storage formats. The problem with the statement in the previous sentence, besides the fact that much of it is not exactly true, is that the lesson of materiality is that there is no core, no intent or act of encoding or decoding that is not always already subject to the discourses of mediality – that is the cultural, economic, and technical contexts which give shape and form to media. Furthermore, the adjective "uncertain" covers another falsehood because it implies that the phenomenon of material uncertainty deviates from a normative certainty of some kind, when in fact the lesson of materiality is that its precession is imprecise and external to the phenomena of material culture. In other words, in the examples above, it is not necessarily the case that the examples from the emulator Stella are more faithful to the original game's intent, or that my photographs from the actual Atari VCS are more faithful to how the game was played in its original form. Nor is it even the case that the fuzzy images from the photographed screen are closer to the designer's intent which may have anticipated the distorting effects of the television. What matters is that these differences leverage the discourse of materiality across different media and that typography is a useful vector for expressing and transmitting that discourse. In order to traverse that vector and examine the cross-platform textuality of a single game, the next chapter is devoted to an analysis of the textuality of the game *Berzerk* in different versions.

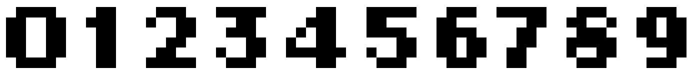


Figure 4-1. Selection from “Lo-Res 9 Wide Bold,” by Zuzana Licko. Part Emigre’s Lo-Res font package. Image and font design © 1995 – 2007, Emigre Inc.

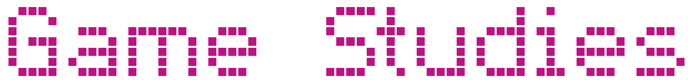


Figure 4-2. Logotype[17] for *Game Studies: the International Journal of Computer Game Research*, <www.gamestudies.org>.

Figure 4-3. Banner logotype (trimmed) from the gaming news and weblog site, *Destructoid*, <www.destructoid.com> Image © 2007 Destructoid LLC.

Figure 4-4. Cropped banner image from the website for PAX 07, the 2007 convention hosted by the webcomic *Penny Arcade,* http://www.pennyarcadeexpo.com. Image © 2007 Penny Arcade.

Figure 4-5. Enlarged screenshot captured from *Dig Dug* (1982) played on the Stella emulator.

17 A *logotype* here refers to a specific use of lettering, possibly through a custom typeface or hand-drawn characters, to convey a specific brand identity or trademark.

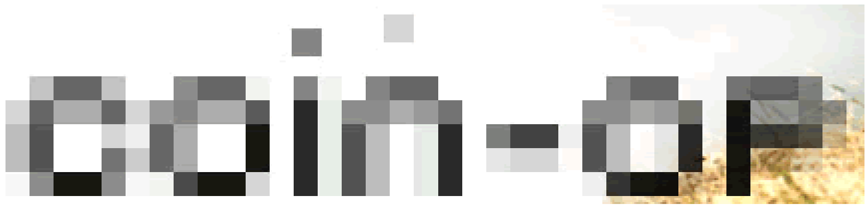


Figure 4-6. Banner image (original size) from Coin-Operated.com, the blog and personal website of digital artist Jonah Brucker-Cohen. Image © 2007 Jonah Brucker-Cohen.

Figure 4-7. Image from figure 6 shown at what may have been its original size. A) With anti-aliasing blurring intact. B) With anti-aliasing removed.

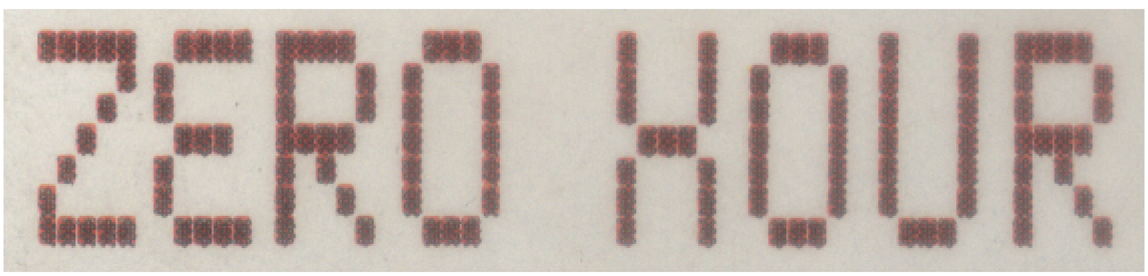


Figure 4-8. Text detail (slightly enlarged) from advertisement for the Atari 2600 game, *Star Wars: Death Star Battle*.

Figure 4-9. Full ad for *Star Wars: Death Star Battle*. Image © 1983 Parker Bros.

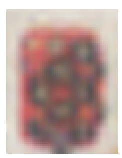


Figure 4-10. A single subunit (dot) from the text in figure 4-8 reveals that it is made up of smaller dots.

Figure 4-11. Title and logo for *Blip* magazine. Image from Gifford. Logo © 1982 Marvel Comics, Inc.

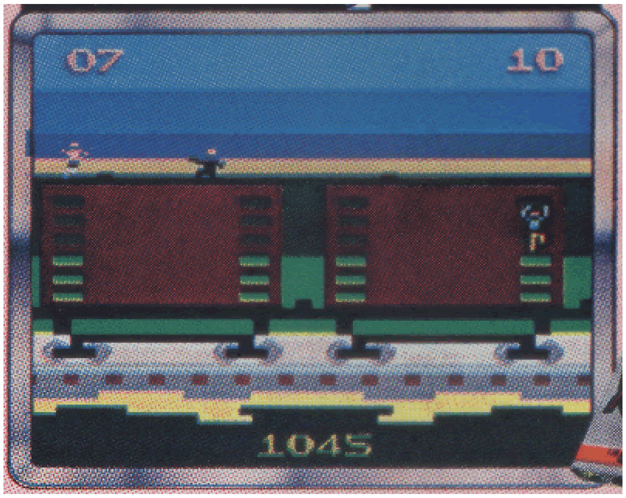


Figure 4-12. Image detail (enlarged) from advertisement for the Parker Brothers game for the Atari 2600, *James Bond as Seen in Octopussy*. These advertisements appeared prominently in several DC comics' titles in late 1983, but the game was never actually released to the public.

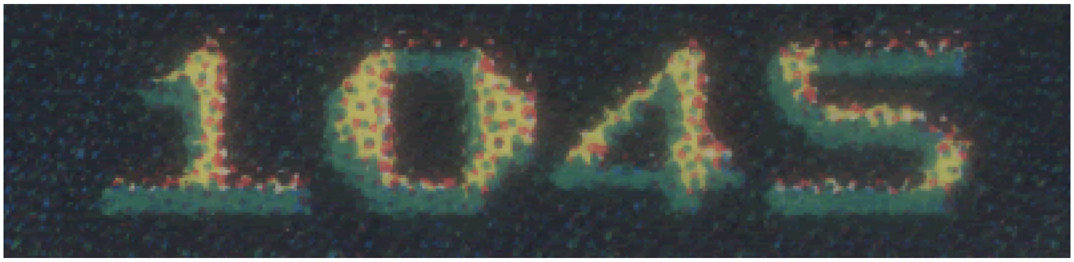


Figure 4-13. Detail (enlarged) from same advertisement in figure 4-12. Advertisements appearing in comics originally referenced screen images through a stylized, often cartoony drawing of gameplay. Gradually, these became more realistic and more likely to be actual photographs of gameplay. Since this game was never released and no known copies exist, it is impossible to say if this is a photographic or cartoonic representation.

Figure 4-14. Screenshot from *Half-Life 2*.

Figure 4-15. Photograph of actual TV displaying *Dig Dug* for Atari VCS.

Figure 4-16. Continuum between visual features, sharp and soft.

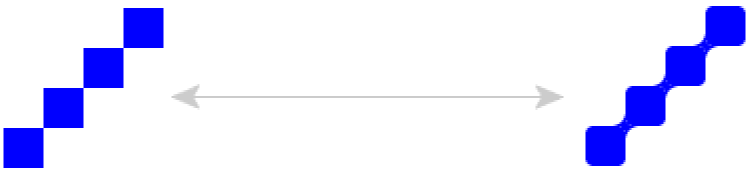


Figure 4-17. Granularity and contiguity, two other features of videogame type.

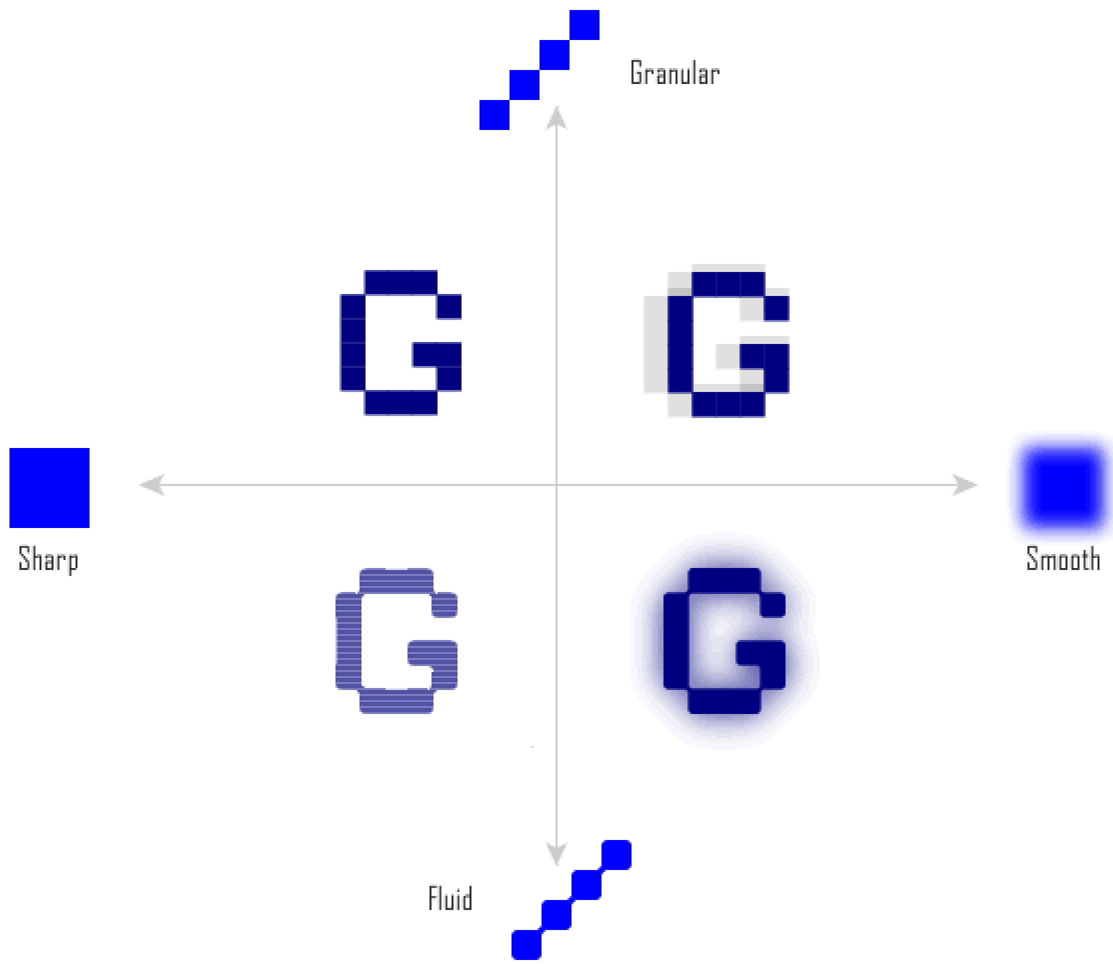


Figure 4-18. Graph of the visual properties sharpness, softness, granularity, and fluidity illustrating how they calculate a given text's fuzziness or jagginess.

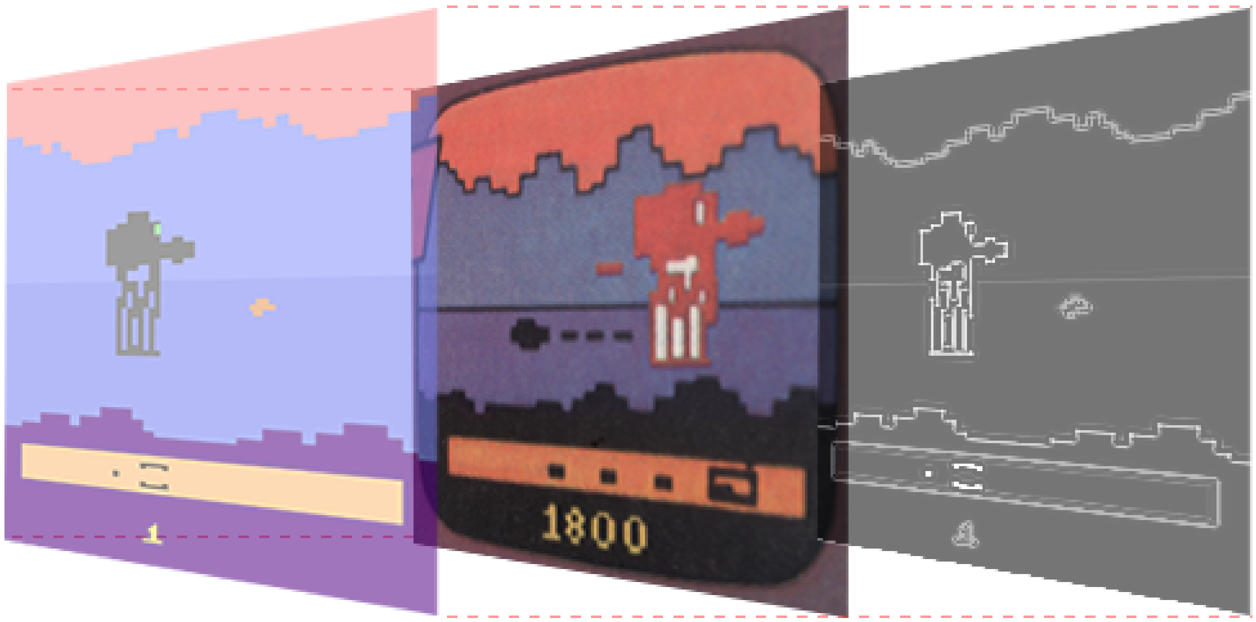


Figure 4-19. An illustration of the primary layers of mediality embedded in the printed image of the game, *Star Wars: The Empire Strikes Back*.

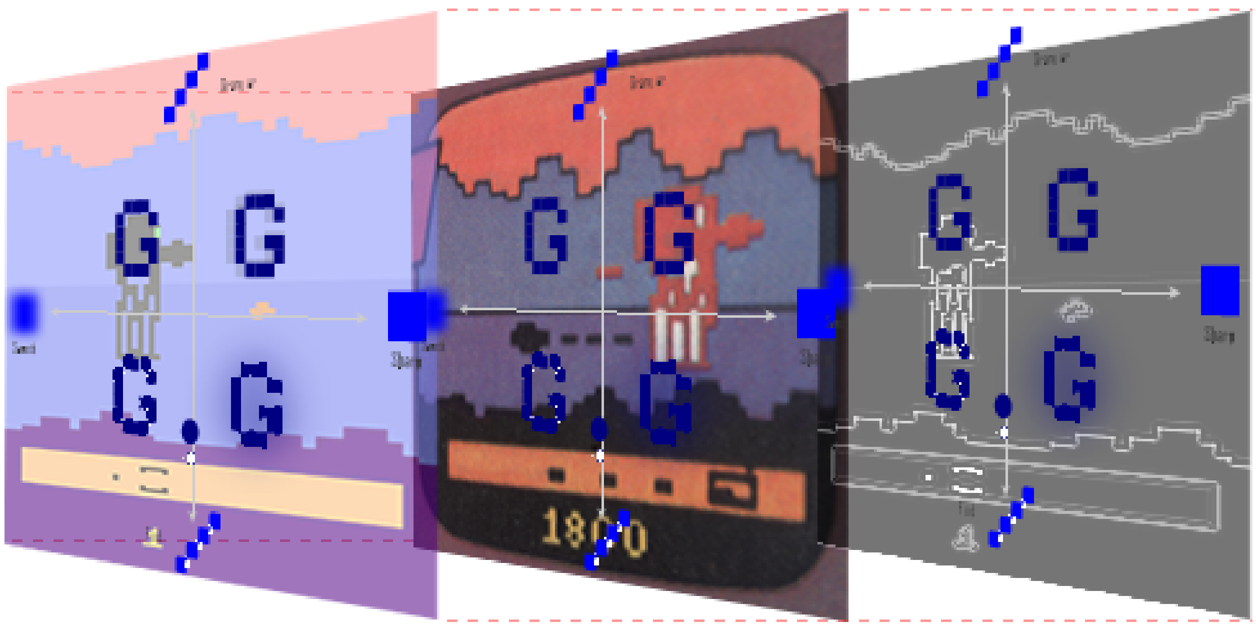
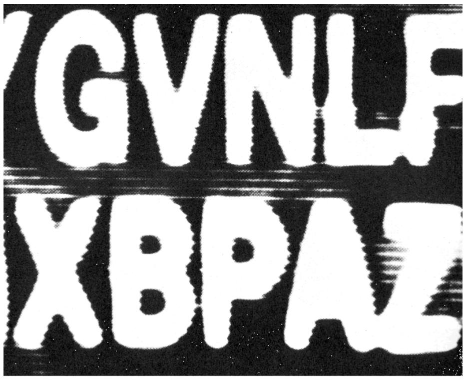
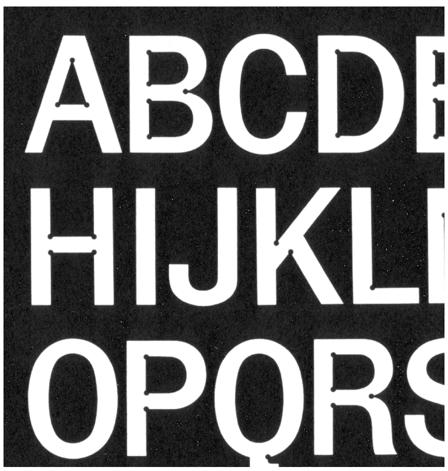


Figure 4-20. Combining the graphs from figures 4-18 and 4-19 projects the medial discursivity of video game type in terms of the sample's visual properties.



A



B

C

Figure 4-21. A) Bass's example of degradation, halation, and flux distorting News Gothic Bold. B) Selection of CBS News 36 (negative), C) CBS News 36 displaying same text as 4-21A with improved form and legibility. Images from Bass, p. 366 – 369.

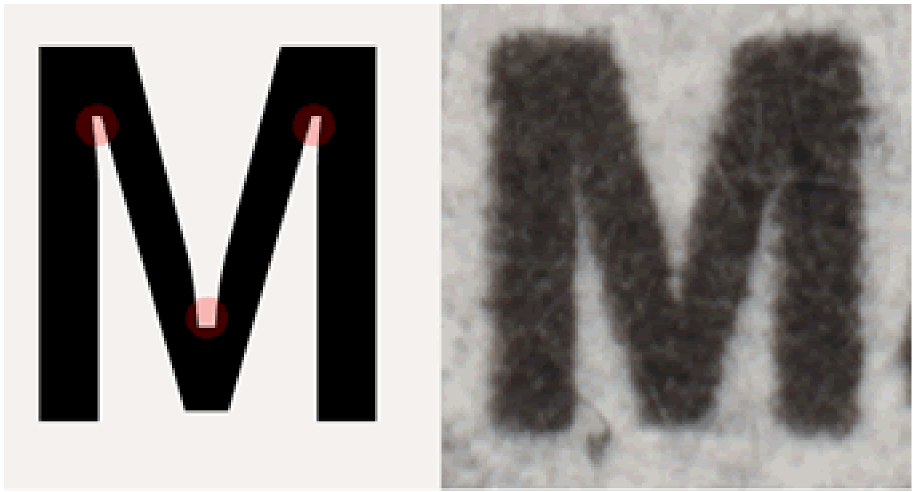
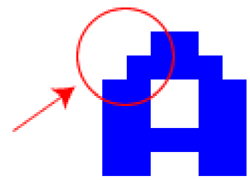
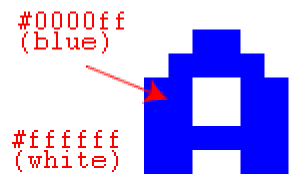
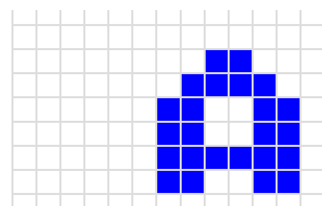
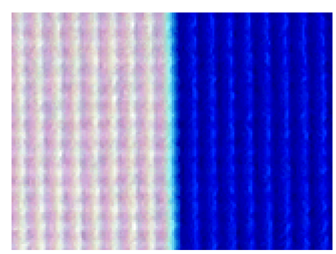


Figure 4-22. Matthew Carter's design for Bell Centennial, a typeface commissioned by AT&T for use in telephone directories. (Image from Sherman)

Table 4-1. Summary and examples of characteristics of jaggy type.

Characteristic	Example*
Stair-stepped angles and curves	
Sharp distinction between foreground and background colors	
Uniform, rectilinear subunits: actual or simulated pixels	
Simulated pixel edges flush with actual monitor pixels	 (Photograph of same image on LCD monitor; contrast enhanced to show detail)

* The images in this example use the font, ATASCII, based on Atari's implementation of ASCII text for their line of home computers.

CHAPTER 5
A TYPOGRAPHIC ARCHEOLOGY OF *BERZERK*

Studying videogames requires an archaeological approach – digging down through layers of culture, context, and platform to learn more about the artifacts. In order to fully understand these objects, we must begin by understanding the technological affordances which their expression relies on – the intertwining materiality of the technical strata which work together in producing a single game image, experience or text. In this final chapter, I offer a stratigraphic reading of multiple versions of the videogame Berzerk (Stern Electronics, 1980). Specifically, I argue that an analysis of the typography of each version reveals unique expressive properties of that respective platform. For example, the typeface emulated in one instance of Berzerk (for the Atari 2600 console) models the form of a popular typeface used in the 1960s and 70s, Countdown, which is often associated (parodically) with computer interfaces, but the extent that it does so depends on the interference of screen artifacts at the level of image generation. Significantly, the degree to which Berzerk's text successfully matches this face depends on the artifacts of a particular, phenomenal game "layer" that often goes ignored, the phosphor screen of common CRT televisions.

For some writers such as Lars Konzack, for whom the narrative and linguistic signifiers of videogames are "ornamental," the possibility of including the material artifacts of the display screen must seem even more superfluous. Ever the ludologist, Konzack states,

> All the semantic meanings of the game are secondary to the gameplay's primary lodologic structure. The signs conveying meaning are indeed superficial, but still help in putting the game into perspective. Two games may have exactly the same gameplay, but by having different ornamental signs and narratives (such as pictures, sounds and/or text) they convey different meanings of what is happening within the game. (95)

But as King and Krzywinska response, "the term 'ornamental' implies a more dismissive attitude towards contextual material than seems to be suggested by the conclusion that different meanings are being conveyed" (74). In other words, if a so-called "ornament" has a profound enough influence on the game that it fundamentally changes the meaning-production of the game text, then it deserves a more privileged place within the conceptual structure of the game. King and Krzywinska later make the following claim about the relationship between gameplay and context, which highlights the similarity of their concept *context* with Genette's *paratext*:

> Gameplay has its own intrinsic appeals. It might be said that these can be heightened by the location of gameplay within recognizable contexts, but this presumes that more than a very rudimentary gameplay can ever exist outside some kind of legible contextual framework. If gameplay is often to the fore, it might be argued that this is only possible as a result of the contexts – broad and more specific, crude or subtle – within which it makes any sense. (King & Krzywinska 75)

Comparing this summation with Genette's "paratext is what enables a text to become a book and to be offered as such to its readers" (1) and noting particularly the mention of legibility and sense-making, the contiguity of this approach to game studies with textual analysis is clear. I argue that the scope of this meaning-making, paratextual field should include both the peritextual "accidentals" of videogame content (including typeface design and numeric arrangement) and the epitextual artifacts surrounding the game (from the arcade cabinet art to instruction manuals), but the lesson in recognizing the concepts, holotype and paratype, as well as fuzzy and jaggy, should that there is a territory between epitext and peritext worth exploration. Genette offers the equation, "paratext = peritext + epitext" (5), and I am arguing that the plus sign marks the spot at which to begin excavating the discourse plane of typographic textuality in videogames.

The material surface of the screen is an important figure in the analysis of videogames, but as Montfort and others have noted, the figure of the screen as a transparent emblem of New Media ontology is conflates its actual position in the history of computational media.[1] In fact, the *opacity* of the screen provides a significant vector of the material culture around videogames, and as such, support using an archaeological metaphor for approaching the game text in a way that includes factors such as the screen. First, the interferences of the screen, radio frequency noise, and any

1 Montfrot criticizes the "screen essentialist" view in his essay, "Continuous Paper."

other intrusions are typically referred to as *artifacts* because they impede image transmission or image quality. In computing, an artifact is "[a] phenomenon or feature that is not originally present or intended in a system and is a by-product of some aspect of processing. For example, the intended smooth gradation of color generally appears on a computer monitor as an artifact consisting of very narrow bands of even color, with each band slightly different from the adjacent bands" (*High Definition* 15). One common example of this sort of artifact manifests clearly in JPEG images (a so-called "lossy" format) that have been greatly compressed, as illustrated in figure 5-1. In this sense, an artifact is a mark on the image which indicates the presence of the technology supporting the image. The distinctive appearance of JPEG artifacts distinguishes these images from, for example, PNG (Portable Network Graphics) images which are in turn labeled "lossless." Therefore, an inference based upon marks on the object allows one to conjecture, like an archaeologist, a historical setting for that object, the sequence of transformations situating it in a technologically specific context.

The second sense in which the term *artifact* is important to the archaeological metaphor is that it emphasizes the historical situation of the game itself. In criticizing prevailing views of textual criticism, particularly those dependent on the ideology of authorship as the sole locus of critical interpretation, Jerome McGann, among others, has proposed an alternative program for textual studies, one which recovers the diachronic and social aspects of literature through "the operation of a complex structure of analysis which considers the history of the text in relation to the related histories of its production, reproduction, and reception" (McGann, *Radiant Textuality* 123). Similarly, Johanna Drucker has written extensively on the importance of the material and visual domains of meaning in literature, and specifically with regard to typography, Drucker argues that it is uniquely suited for studying critical practices. "Because of its interdisciplinary character, the treatment of typography within critical interpretation can be used to trace the transformations in the premises on which both literary and visual arts criticism conceive of their object" (Drucker, *The Visible Word* 1). This chapter seeks to conceive the game object by way of an analysis of its typography and in so doing, to outline a program of study which uncovers the diachronic, culturally material aspects of videogames by way of the circumstances of their production and reception.

One way of indexing those circumstances, as shown in the previous chapter, is through the alternate regimes of representation which I refer to as "jaggy" or "fuzzy." These are two broad descriptors for different means of rendering type in and around videogames; type in videogames may be rendered in a manner which is jaggy or fuzzy, depending on the technology, and type which exists in reference to videogames (for example, on arcade cabinets) may also employ fuzziness or jagginess as a reference to the display rendering which creates either condition. One common instance of jaggy type occurs in most videogame emulators – software which reproduces a game platform – on modern personal computers. Because the monitor's display will generally be far crisper than those available to game players in the early 1980s, the experience of playing the game today is slightly altered. Compare, for example, the two versions of the title screen for Atari's adaptation of *Dig Dug* (Figure 5-2). In the first image, the stair-stepped edges of the letters' diagonal lines create a jaggy line which is only as smooth as the resolution allows. In the second image, the jagginess is minimized and the line appears smoother and more continuous because it benefits from the slight distortion it experiences when deployed on a CRT monitor screen.

Though this type of modulation might still be classed an "accidental" in the terminology of traditional bibliography, these effects are far from unintentional. For example, *Compute!'s Second Book of Atari Graphics* includes a chapter that explains how to take advantage of screen artifacts for improving visual effects.[2] The point is not that either fuzzy or jaggy type is a better or more

2 This text is advising how to program for the Atari 400/800 line of home computers, but the principle is the same for any console system. In addition to coloring effects, the chapter also includes a program for using moiré effects for aesthetic purposes (Pewther 193).

faithful rendition of an ideal design of a character or the designer's intended appearance. Such a position would place the bitmap-encoded version of a character in a position of primacy. Rather, both fuzzy and jaggy afford different mediations of constraint that allow them to reference a specific set of technology (i.e., either an emulator or a physical console) even when a different artifactual layer such as print intervenes. In this way, arcade flyers, t-shirts and website designs can accomplish a specific association which operates conceptually at the layer of the screen. This referential ability of fuzziness and jagginess suggests that understanding the effects of the screen should be part of an approach to videogame study, but in fact, in much of the existing literature on new media, the screen is taken for granted or essentialized as a metaphor.

Levels and Layers

Among the many attempts to formulate a critical concept of videogame structure (ontological or otherwise), one recurring theme is an organizational heuristic based in levels or layers. Nick Montfort's article, "Combat in Context," proposes a five-level model for videogame analysis that he repeats when describing the book series he is co-editing with Ian Bogost (Bogost & Montfort). Interestingly, Bogost and Montfort chose to use a different graphic for portraying the levels. The difference is interesting, even though it may have originated in purely aesthetic considerations, because it places a different emphasis on relationships among the levels. In both the article and the discussion on the Platform Studies website, Montfort arranges game content into the following five levels: 1) platform; 2) game code; 3) game form; 4) game interface; 5) reception & operation (Montfort, "Combat in Context"). The two illustrations appear side-by-side in figure 5-3. The first (5-3A) is from Montfort's *Game Studies* article where the table retains the ordinal markers from their initial listing. The second (5-3B) is from the website introducing the Platform Studies book series. Note that in the example from the website, Montfort and Bogost have removed the ordinals and instead illustrate degrees of separation as a gradation of color such that the *platform* level is the darkest or most dense – a fitting characterization for a foundational substratum. This symbology has significance because 5-3B implies a degree of co-involvement and dependency that is missing from the discrete, padded cells and strict ordinals of 5-3A. In other words, figure 5-3A presents the five levels as being *jaggy*, whereas figure 5-3A illustrates the relationship in a way that is more *fuzzy*.

Besides responding specifically to Lars Konzack's seven-layer model for game analysis,[3] Montfort also implicitly addresses other approaches which stratify videogame content for the purposes of study, including Juul, Aarseth, and Eskelinen. Essentialist stratigraphies are not confined to ludologists, however. Julian Kücklich deals with one such approach in his article, "Perspectives of Computer Game Philology," where he caricatures the much-maligned, so-called "narratological" approach[4] to computer game study that "blindly equat[es] a computer game's *technical* levels of its code and its interface with the *narratological* levels of story and discourse" (Kücklich, "Perspectives of Computer Game Philology"). Kücklich sees this equation as a problem and provides a different (also dually-layered), constructivist model where games are treated as non-trivial machines with embedded trivial machines (adopting Espen Aarseth's terminology) that consist in processes of adaptation. What is important here about Kücklich's straightforward criticism of a narratological model and his response to it is that both employ levels as a metaphor for a conceptual relationship among differently functioning sub-phenomena. In responding to the initial, assumed hierarchy, a different hierarchy is exchanged for the original. I am not arguing that organizing the figurative structure of a videogame by invoking layers or levels is incorrect. Instead,

3 Konzack proposes a method that identifies seven layers: "hardware, program code, functionality, gameplay, meaning, referentiality and socio-culture" (90) . He also acknowledges two "levels" or perspectives on these layers: the "virtual space" of the game (that is, in the game world), and "the playground" (that is, the game board, pieces, players, environment, etc.).

4 I have no intent to rehash the so-called ludology and narratology debate, but it is worth noting how each "side" makes similar but complementary claims which depend on some form of reductionism that may or may not be arbitrary. I offer a broader survey of approaches to videogame textual analysis in Chapter 1.

the prevalence of this rhetorical gesture extends from the same textual logic behind Genette's paratext and McKenzie's social text.

It is also noteworthy that Kücklich arrives at his proposal by closing off the code as external to the process of gameplay and that he does so by invoking the screen as a metaphor: "Usually, the only thing the player knows about the world of the game is what is *displayed on the screen*. However, the player is able to learn about the implicit rules of the game simply by interacting with it for a sufficient amount of time" (Kücklich, "Perspectives of Computer Game Philology," my emphasis). As he is using it here, Kücklich means something synonymous with *actually* or *literally* when he writes "on the screen;" his point is that players only come in contact with the game world as they actually perceive it with their senses. But the figure of speech he chooses here is interesting because it suggests the dual nature by which the videogame screen becomes a physical but transparent threshold between the game world and the real world at the same time that it is a conduit connecting those two worlds. It also indicates how the material structure of the game text influences intellectual discourse about games, even when that discourse is not directly relevant to material structures and their influences. Whereas Montfort criticizes a screen essentialist position, what I am developing here is in essence a screen materialist approach.

Another common stratification of game content worth noting is condensed into the following summation, declared by Aarseth in his "Genre Trouble" essay: "Any game consists of three aspects: (1) rules, (2) a material/semiotic system (a gameworld), and (3) gameplay (the events resulting from application of the rules to the gameworld)" (Aarseth, "Genre Trouble" 48). This is the core taxonomy most clearly articulated by Jesper Juul[5] and most aggressively defended by Markku Eskelinen.[6] The advantage to this ludo-centric approach is that it allows videogames to be analyzed among a longer history of games and play that is as ancient as culture itself. However, if

5 In this passage, Aarseth refers specifically to Juul's article "Games Telling stories?" but his *Half-Real: Videogames between Real Rules and Fictional Worlds* addresses the topic directly and extensively.
6 Aarseth directs readers to Eskelinen's essay, "Towards Computer Game Studies" (36) in the same collection.

in practice this formal separation privileges rules at the exclusion of semantic context (as Aarseth elaborates in this essay), Stuart Moulthroup seems correct in calling the insularity of this allegedly self-contained system, “alarmingly narrow” (48). The fact that Aarseth introduces the second aspect by conflating or combining materiality and semiotics and offering further clarification that he is referring to the “gameworld,” suggests that there is something more complicated going on here that merits more than a single “aspect.” Regardless of how one treats the phenomenon of gameplay, it is clear that some meaning and referentiality must be inferred from the various (and possibly discordant) rhetorics of simulation and representation. Furthermore, although Aarseth does not specifically use the term levels, it is clear from the primacy he places on rules that he intends some degree of stratigraphy (however ordered) in relating the 3 aspects of gameplay he identifies.

Though he is not necessarily a direct response to the ludology conversation, Alexander Galloway’s approach introduced in *Gaming: Essays on Algorithmic Culture* demonstrates one way of understanding how the material/semiotic dimension of the gameworld can influence a game’s ability to express meaning, which is an important way in which Montfort’s levels can interact with one another. As Galloway observes, “the shape and size of Mario in the NES version of *Super Mario Bros.* is determined not simply by artistic intention or narrative logic but by the design specifications of the 8-bit 6502 microchip driving the game software” (32). Furthermore, this sense in which game characters are embodiments of “math made visible” (32) is continuous with control structures that determine formal actions within the game since these are dictated by informatic principles that drive software. Although he is writing about representational graphic sprites, this point of Galloway’s nicely addresses my point in chapter 3 about videogame holotype.

In a different approach, Ian Bogost’s unit analysis creates a method from the object-oriented logic of software, identifying both software (rules) and the gameworld (fiction) as systems constructed on the basis of separate but interdependent units (Bogost, *Unit Operations*). Both Bogost's and Galloway's approaches are consistent with Montfort's five-level model, and taken together, these methods and assertions amount to what Montfort and Bogost have labeled "platform studies." Steven Jones has drawn an explicit connection between platform studies and materialist textual studies, noting,

> At the heart of both traditions, book history and textual studies, is an emphasis on what might be called text technologies, the material methods by which texts of various kinds get made and distributed and received. In this sense, platform studies already is a form of textual studies, focused on objects that are not (primarily) verbal texts. By analogy, we can speak of the "platform" of eighteenth-century print culture as a social framework... (Jones 129)

As the following sections demonstrate, typography can be an important feature of the videogame text for allowing us to examine how a game's technology merges with its social framework. This, in turn, leads us to intertextual associations created through the use of particular numeric character designs. Taken together, the multiple versions of *Berzerk* on different game platforms constitute the *Berzerk* social text or texts, and we can access the technologies of its paratextual provenance through a close examination of its typography.

Berzerk(s)

The first and original incarnation of *Berzerk* was programmed in 1980 by Alan McNeil for Stern Electronics ("Berzerk Videogame by Stern"). The game consists of a "humanoid" character fighting off talking robots and a bouncing smiley face named Evil Otto who chases the humanoid protagonist through a dark maze. In terms of the content of its loosely implied story, *Berzerk* could be considered an aesthetic forerunner to *DOOM* (id Software, 1993), and like *DOOM*, *Berzerk* also generated its share of controversy. *Berzerk* is the first game known to contribute to a player's death (Kiesling 14), and it became a target of early videogame critics who decried its presentation of humanoid-on-robot violence. Thomas Radecki, then chairman of the National Coalition on Television Violence, wrote in 1983, "the object is to kill as many other stick figures as possible, before they kill you … This type of role-playing practice is certain to have long-term harmful effects on the player; it teaches violent reactions" (Qtd. in Sullivan 70). Significantly, Radecki's criticism here does not identify the act of killing as affectively realistic, which is the main objection of those who criticize ultra-violent contemporary videogames such as *Manhunt* and *Postal*. Rather, he objects to a symbolic act which depends not on the resemblance of the Automazeons to humans, but rather on the symbolic relationship between their iconic, stick figure presentations.

Excluding the arcade sequel *Frenzy* (1982), *Berzerk* saw four licensed translations to other platforms, including versions for the Atari 2600, Atari 5200, GCE Vectrex, and a board game by Milton-Bradley.[7] It is interesting to note the varying paratextual renditions of the humanoid, Automazeons, and Evil Otto that accompany the various translations, but it is more important to note how the shapes used to render text reveal a discourse of materiality which, following Montfort's terminology, connect and complicate the platform level with the interface level. There are important differences among each *Berzerk*, and because those differences reflect on their respective platforms, the platform-influenced visual differences in typeface design become figures for differences in gameplay.

This co-involvement seems to follow Montfort's thinking when he recreated the 5-level illustration for the Platform Studies website (Figure 5-3A). Whereas listing platform as the "first" level grants it a place of determined primacy, removing the ordinal and simply placing it at the bottom signifies its foundational status without implying as strong a conceptual dependency on the part of the other levels. The fact that multiple versions of *Berzerk* that are mostly similar can co-exist at all means that, in a purely practical sense, the programmers working on the adaptations at least began with a common form/function level. The differences which do appear in different

7 Milton-Bradley produced a number of board games based on popular video game titles in the early 1980s. Though I will be mainly focusing on the videogame versions of *Berzerk*, the board game version is interesting for depicting Evil Otto with a body.

versions of *Berzerk* illustrate what happens as programmers attempt to retain the same textual and typographic material while changing platforms. The result reveals that a relationship between levels where contingencies and propensities of each platform have enforced compromises (or improvements) in the game form and interface manifest at the level of reception and operation. The technological influence on the reception of the text also extends to the material culture of *Berzerk*, both paratypically and holotypically.

Figure 5-4 shows screenshots from the four licensed versions of *Berzerk*. At first glance, the games appear to be similar, and in fact, at the level of form, one might well argue that these are in most respects identical. Some slight variations do, however, change the gameplay significantly. In the Atari 2600 version, for example, the robots are not programmed to fire their lasers in all eight directions, so the player (who can fire in eight directions) can gain a tactical advantage by approaching Automazeons diagonally. Also, in the Vectrex version, the player's sprite is comprised of a continuous set of vector lines so that there is no gap between the player-character's head and shoulders. This gap in the other versions can allow a laser to pass through without harming the player. Figure 5-5 captures this technique in action. The so-called "bulletproof bowtie" technique is only useful as a last resort, but it works because of choices McNeil made when he designed the stick figure's body. The reason for this design is apparent in the artwork for the original flyer advertising *Berzerk* (Figure 5-6). Despite the fact that the sprite contains a gap between the head and body of the figure, the fuzzying effects of the raster display artifacts blur the edges of the stick figure's form so that his head appears to be connected to the rest of his body. The gap one sees in the emulated, jaggy versions (Figure 5-7) results from the more crisp display technology used by the emulator software and the user's monitor itself. In this way, the figure's form, which depends on artifacts at both the platform and the display levels, also affects gameplay and one's experience or interpretation of the game text.

The Arabic numeral 3 provides a more subtle variation that reveals how pervasive the interactions and influences are among the levels of the game object. Figure 5-8 shows the numeral 3 as it is rendered on the various platforms, each by way of an emulator. The 3s from the Atari 2600 version and the Vectrex version are the most obviously different because the Vectrex display attempts to adapt a raster composition similar to the arcade and 5200 versions for its vector display system and the 2600 employs an entirely different shape for the numeral. The form in figure 5-8B exhibits characteristics fundamental to jagginess, the most prominent of which are the visible pixels which compose the numeral shape itself and constrain its form to that which can be drawn at that output resolution. Figure 5-9 illustrates this compositional logic by highlighting and isolating one such pixel.

The highlighted pixel in figure 5-9A demonstrates a crucial point in terms of Montfort's levels. Montfort's main improvement on Konzack is that each of his levels is considered to be influenced by its cultural context. In other words, the game object is surrounded by context at all layers rather than finding context only in its reception – that is, projecting each expression outward from a categorically ideal, lower-level code that works its way through progressively disruptive levels until finally coming into contact with the real world. Instead, for Montfort, context, and therefore criticism, can intervene at any level and each can be considered in its own terms. As a logical extension to this idea, each level is in turn part of the context of every other level and can be employed as a critical tool to unlock its co-involved fellow levels. The image of the numeral 3 in figure 5-9 highlights a single pixel, which is important because its jaggy appearance and visually discrete sub-units (pixels) mimic the actual digital composition of the game's code. In this way, the fourth level (interface) communicates something about the first level (code), but the degree to which it does so depends on a level which lies external to Montfort's ontology, the material conditions of the display technology.

In the following section, I examine two versions of Berzerk within a framework of platform studies (or simply textual studies), with particular attention to the typography of each. The two *Berzerk*'s I examine are the adaptation for the Atari VCS, ported by Dan Hitchens in 1982 (" Berzerk (Atari)"), and for the GCE Vectrex, ported by Chris King (Woodcock). These are the most appropriate for this analysis because they are the most different typographically, not only in terms of the appearance of their lettering and numbering, but also in the means by which numeric characters are retrieved in the game code. Also, crucially, the Atari VCS and Vectrex are the only two games of the four where I have immediate access to an actual game console. The coin-operated original *Berzerk* and the port for the *Atari 5200* can be played on emulators (MAME and MESS, respectively), but the differences between an emulation and a materially present game is an important point in this analysis.

Berzerk **on Atari VCS**

Berzerk for Atari VCS was programmed in 1982, after the coin-operated game had proven to be largely successful. *Berzerk* was one of many arcade hits that Atari attempted to recreate as they increasingly sought to compete with the videogame market (Kent), but in terms of its fidelity to the original, *Berzerk*'s stick figure graphics appeared to be far more comfortable on the VCS platform than did the similarly adapted *Pac-Man.*[8] Gameplay was altered slightly for the VCS version (mainly, the robots cannot fire diagonally), but the main visual difference was in the typeface used for reporting the player's score. Figure 5-10 illustrates the relationship between *Berzerk*'s binary data and the numeral 3 as generated by the emulator Stella. As discussed in chapter 2, the Atari VCS lacked a native text rendering kernel, so game designers had to allow space in each game's ROM for storing bitmap images of the needed characters. The game program includes instructions

8 The Atari VCS version of *Pac-Man* is frequently cited as one of the worst videogames of all time. For example, William Cassidy of *GameSpy.com* ranks *Pac-Man* for the VCS at #4 on his list of the "10 Most Shameful Games." See <http://www.gamespy.com/articles/492/492996p1.html> for the complete list, which includes other coin-op ports.

that call up a specified segment of code which contains a graphic representation of the Arabic numeral “3.” As an image, that section of code has no inherent three-ness that is interfaced programmatically other than the context of its being called when the digit, three, is required for representing the player’s score or other numeric values.

Of course, it should be noted that presenting these images of the numeral three in the context of this present document involves an additional interference or transformation. In figure 5-10A, I have converted the game’s code into a visual form by processing it with a program that represents each positive value (binary 1) as a capital X and each negative value (binary 0) as a period. The numbers in the left column indicate the memory addresses where the example bytes reside within the game’s ROM file. To create the image in 5-10B (and the similar preceding images), I used Stella’s built-in snapshot function to capture the emulator’s output to a bitmap file. I then enlarged the image in a graphics program making sure that the software did not distort or otherwise attempt to anti-alias the image. The jagginess is maintained, and juxtaposing these two images shows that, in this example, both code and screenshot are jaggy. Representing the code visually does introduce another level of interpretation and interference, but the on/off logic it illustrates is true to the digital character of the storage media. Like the edges of the image in 5-10B, the bit positions in the ROM must be either positive or negative to communicate the correct image data to the screen. The image Stella generates can be considered a rather accurate rendering of the original image, but its fidelity in this example depends on the way Stella draws the image to the screen as well as the way I captured and manipulated the image. Significantly, this process intercepts the image before it is transmitted to the screen buffer, so it effectively ignores any influence that the screen might have on the appearance and referential quality of the image.

The impact of the monitor is, however, quite apparent in figure 5-11 where the image in 5-11B is a different capture of the same numeral 3 represented in figure 5-10B. In this case, I made the initial capture of the image by photographing the monitor at a very close distance and then enlarging that digital photograph. The difference between the screenshot 3 and the photograph 3 reveals that, in the case of 5-11B, the congruity of the generated image to the divisions of the display surface of the screen results in a 3 that is more jaggy, which is significant here because the jaggy 3 creates an illusion of greater fidelity to an imaginary, un-mediated 3 image. Each generated pixel is divided neatly into four display pixels, and the edges of the generated pixels are flush with the edges of the screen's cells. However, as the photograph reveals, the rectilinear structure of the pixels is not actually continuous. Their subdivision reveals their edges not to be true lines at all but, rather, points of light arranged in a matrix which the eye combines into a continuous line when viewing it at a sufficient distance. The difference this makes is that whereas the photographed, enlarged image (5-11B) emphasizes the effects of the screen layer, the screenshot image (5-10B) attempts to ignore it.

Jaggy type is not necessarily reclaiming the aesthetic of the screen grid, as may often implied in overtly nostalgic context, but it is instead accessing the logical matrix of the game's machine code. This is not to say that the jaggy numeral three in figure 5-10B is more authentic to the game experience or a better, more ideal 3, but that it expresses something about the nature of the game image, namely that its digital origin imposes graphical limitations and that the aesthetic effect of those limitations is not necessarily a bad thing. If the imagery of game emulators expresses a rhetoric of nostalgia, it is nostalgia that creates an imaginary ideal game image where the screen as a medium itself is invisible. As an alternative, if figure 5-11 makes an allowance for the regime of the screen as a signifier of mediality, then figure 5-12 puts it right in the foreground.

This final in-game example illustrates a more dramatic influence of materiality. The image is a photograph of the numeral 3 generated by an Atari VCS console and displayed on a normal CRT television. The raster scan lines, phosphorescence, and the resulting halation on the figure are all natural conditions of the display medium that affect the shape of the 3 in important ways. Note the irregularity of the (logical) pixels and their tendency to swell at inner corners and narrow at the endpoints of posts. The corners are all rounded, and the bright, positive space of the shape blends relatively gradually with the black negative space of the field. Like the *Dig Dug* logotype in figure 5-2B, this image is inherently and unself-consciously fuzzy. It clearly emphasizes the influence of the screen on the rendered text image. Moreover, besides expressing a specific quality through the game's interface level that affects its reception level, this image also suggests something about the platform and original source code: specifically, it is possible that this shape was chosen for the VCS port because at a height of 7-pixels, it was more appropriate for the condensed visual and memory space than the 9-pixel high figures in the arcade version. Another instance of this version of the 3 (that is, it is the same in terms of its functional representation within the structure of the Atari VCS game) proves illuminating in this regard.

The 3 illustrated in figure 5-12 performs the same numerical signification within the printed, simulated game as the 3s in both game images in figures 5-9 – 5-12.[9] This image appears in the instruction manual for Atari's adaptation of *Berzerk* for the VCS. It is clearly an illustration rather than a photograph because the lines of the laser beam are unbroken diagonals as opposed to the stair-stepped missiles appearing in the game. The angle these lines illustrate is also far steeper than what actually appears in the game. More importantly, the shape of the numeral 3, while retaining its basic form, has changed somewhat dramatically with regard to the outline of its inner post, as illustrated in figure 5-13B. The line that was previously a short post comprising a single pixel and a single code bit has now become a tapered point. This printed, fuzzy image of the numeral 3 is

9 Achieving a score that contains a 3 in the first position was more difficult than I expected. In order to capture the images from Stella and the Atari, I had to reload the game until I started with an opening field of 3 robots so that, by killing them all, I received a bonus of 30 points. (The game randomly distributes between 3 and 11 robots as it generates each maze). The game displays that bonus until the player exits that room, so I had time to position my camera and take the photograph. This approach was necessary since the Atari VCS lacks a pause button, and since the maximum number of robots per screen is 11, the only possible bonus score (10 points per robot if all are killed) including a 3 would be 30.

quite a departure from the jaggy 3 in figure 5-9, but if anything, it has a stronger resemblance to the fuzzy 3 in figure 5-12. From this association, it is possible to surmise that the unacknowledged graphic designer who prepared this illustration of the screen image was using a CRT display image as a reference. Another possibility is that the artist deemed the narrow post (I.e. single pixel) too small to be legible at the printed resolution of the manual. In either case, fuzziness has here been adopted in order to increase legibility and strengthen an association between a print image and a screen image. This association invites considering the fuzzy screen image (interface level) within the context of the print image (reception and operation level), but the print image could also be a considered part of the interface level since it invokes the interface by adopting the aesthetic regime of the screen at the same time that it negotiates its own constraints. This connection between the print image and the screen image also indicates one possible inspiration for the typeface used in the Atari VCS *Berzerk*: the shapes of the numerals in *Berzerk* appear to be adapted from Countdown, a typeface Colin Brignall designed in 1965 for Letraset.

Table 5-1 shows the numeral forms in Countdown and how they appear to have been adapted to the bitmap grid constraint. The first column contains a sample from Countdown, the second shows a representation of the code for that numeral, and the third column contains a photograph of that numeral on a CRT television screen. Taken together, these images illustrate the visual relationship of the source bitmap and the graphical realization of that shape on a television screen. Some differences are apparent (the variable x-height in Countdown has been made uniform in *Berzerk* and the *Berzerk* forms are proportionally wider), but some key features remain intact such as the asymmetrical weight of lines in the 0s and the alternation of thick and thin lines in the 8s. While it is probably not enough of a similarity to call *Berzerk*'s typography a font of Countdown, the typeface's influence on *Berzerk*'s numeracy is clear.

With regard to the numeral 3, introducing a tapered point seems to strengthen this association with Countdown since one of the original typeface's defining features is the way Brignall used rounded internal and external corners to mimic the effects of halation and light buildup. Whether or not Brignall actually intended to simulate screen display technology of the mid 1960s, the letterforms he created have proven well-adapted for that environment and, therefore, well-suited in other contexts to referencing screen-based media as its adaptation in *Berzerk* and frequent uses in arcade cabinet graphics and posters indicate. Accordingly, we can infer that when Dan Hitchens ported *Berzerk* to the Atari 2600, he chose to imitate this typeface either because of its well-established association with other game and science fiction texts or because its structural characteristics which established that association increased the letters' and numbers' legibility even (or especially) with the presence of light buildup and halation. The bitmaps of the numeral shapes lack this fuzzy effect, so one way to describe the adaptation of Countdown is that *Berzerk*'s version of the typeface is not fully composed until it has undergone the fuzzying influence of the CRT screen.

With that progressive imbrication that produces the resemblance to Countdown, the final composition places the *Berzerk* typeface within a broad textual field. In this way, *Berzerk* can be said to exhibit intertextuality with a comic book from the early 1970s called *Countdown*. This comic features a logotype appropriately set in Countdown, and because it is set in outer space, the association it creates with *Berzerk* suggests a science fiction context for *Berzerk* as well.[10] Furthermore, several other games released for Atari VCS incorporate identical character sets, indicating that these games, including *Defender* (1981), *RealSports Volleyball* (1982), and the prototypes *Bugs Bunny* and *Holey Moley*. According to the database at AtariAge.com, *Defender* and *RealSports Volleyball* were both programmed by Bob Polaro and graphically designed by Alan Murphy , suggesting that these very different games shared a common code base. A different

10 I am grateful to Brian Slawson for drawing my attention to the *Countdown* comic.

implantation of a character set similar to Countdown appears in *Demon Attack* (1982), *Marine Wars* (1983), and *Pooyan* (1982). This connection among these games raises the question of platform and its relationship to videogame textuality. Specifically, the VCS versions of *Defender* and *Berzerk* seem to have more in common than any pairing of the several *Berzerk*s, so how to what degree does the means by which this commonality is leveraged constitute the textuality of any of these games? If the underlying code and hardware of a videogame is at the core of its textuality, then what critical value can be extracted by observing that *Bugs Bunny* and *Berzerk* are textually similar? The problem with this question is that it posits the figurative "core" of textuality in a position that, like the ludo-centric frameworks mentioned earlier, privileging the platform excludes semantic content from the production of meaning. The real lesson of platform studies lies in recognizing the influence of the platform without losing sight of the gameplay as the dominant textual force. Platform studies does, however, create a vector along which other games can be broad into the textual situation, allowing us to infer something about the production history of *Berzerk*.

***Berzerk* on GCE Vectrex**

The Vectrex version of *Berzerk* is also quite different from the arcade version for reasons related to the console hardware, and because the Vectrex is a significantly different platform, it is useful to compare its version to the Atari 2600's. First, the Vectrex uses a more advanced BIOS, which contains ASCII characters that can be accessed within games by employing a numeric code.[11] Unlike the Atari VCS where alphanumeric characters have to be stored in bitmap form within each individual game's ROM, Vectrex games can call on the device's internal library of letter and number shapes. Second, the forms of these characters are interesting because they employ a raster composition method within a vector display environment. letterforms are,

11 Specifically, a built-in routine with the label $F37A handles text strings. Programming documentation and tutorials for the Vectrex are available at <http://www.playvectrex.com/designit_f.htm>.

therefore, composed of apparently unbroken horizontal lines which fill in the internal space of the letter shape. This is different from other vector display systems like those used in the arcade games *Asteroids*, *Battlezone*, and *Tempest*, which draw letter shapes using vector lines that delineate the strokes of the letters themselves (see figure 5-14). In other words, these systems display characters by connecting lines to points placed at intersections or endpoints of letters or numerals. Ontologically, these are similar to the 7-segment character generation discussed in chapter 2.

The fact that the display is only capable of drawing monochrome images in white lines also has an impact on the appearance of shapes on the Vectrex. Color is achieved by using transparent overlays specifically created for each game, which rest in a tray about ¾” from the screen surface. In the case of *Berzerk*, the overlay simply gives the game images a blue tint, but it also serves to soften the appearance of the otherwise harshly bright vector lines. This softening also helps alleviate distortion that occurs in images toward the edges of the screen. A consequence of the electron gun’s method for drawing vectors on the screen causes shapes near the outer edges (like much of the text) to quiver or shake slightly, so applying a colored layer of interference mitigates the distraction this may otherwise cause.

Figure 5-15 shows the effect the overlay has on the display of numerals in *Berzerk*. Other than the color, the main difference is that the overlay introduces a softening or blurring effect on the edges of the lines, leading to greater perceptible continuity within the letter shapes. In other words, the un-modified Vectrex image can be considered jaggy, whereas the overlay causes it to be slightly more fuzzy. It is important to note here that the overlay mitigates the distortion of flickering and vibration by introducing a different order of distortion. In this way, the Vectrex image that is literally dually-layered provides a convenient analogy for the differences between emulated and actual game images: an actual television contains an additional layer (its screen) which the emulated screenshot does not.

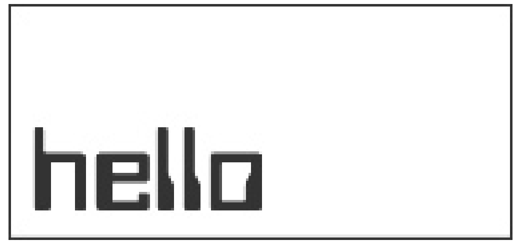
In terms of the game platform and its influence on design and referentiality, the reason a 3 looks the way it does in *Berzerk* for Vectrex is simply that that is how 3s always look in Vectrex games. This rather clearly demonstrates the influence a specific platform must have in determining game content and expression, but it also means that the opportunity for uniquely expressing a relationship among hardware, software, and a typeface does not exist on the Vectrex in the way that it does on the Atari 2600. This shows how technological constraint can encourage creativity and experimentation, which is one possible reason why programming games for the Atari 2600 remains a popular hobby among enthusiasts.

Conclusion

Successful type design for videogames depends on anticipating the propensities and infelicities of each potential layer the image must pass through, so understanding these layers is an important component of a critical approach to videogame textuality. The appearance of alphanumeric characters in videogames and their representations in other contexts which reference videogames can be an important relay for the multi-layered, multiply-contextualized levels of textuality in videogames. This affinity emerges as a result of the unique relationship videogames share with typographic expression: the sense of dependency on the capabilities of technology for constraining or freeing the forms which designers have available to them. A typographic approach to videogame textuality unpacks the expressive content of videogames through the figure of typography in order to better understand the workings of constraint on the videogame form. In both, the effects of constraint are often taken for granted, which is generally the goal of the designer or programmer. However, even contemporary videogames designed for high-definition display output negotiate some forms of constraint, so in order to understand how the affordances of game platforms influence their aesthetics, it is useful to look closely at relics from gaming's past where constraints manifest more clearly.

This present study is at beginning of what should be a fruitful avenue for New Media scholarship, both in regard to videogames and other digital artifacts. As part of a general trend toward more robustly historicized approaches to digital media, of which the Platform Studies collection and the Software Studies Initiative at UCSD are an eminent part, a materialist approach to studying digital texts has much in common with similar trends in textual studies. Typography is a site of meaning production in around videogames that expresses the unique ontology of videogame textuality.

A



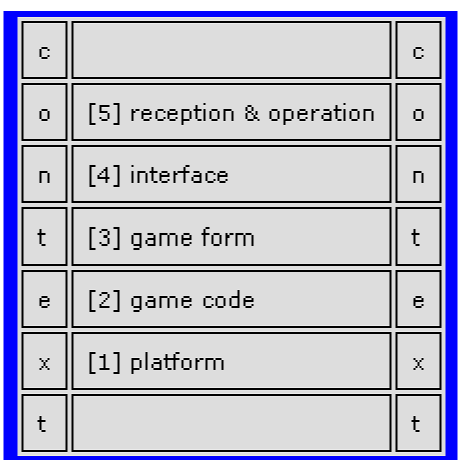
B

Figure 5-1. A) A JPEG image with little compression (95% quality). B) The same image with high compression (20% quality).

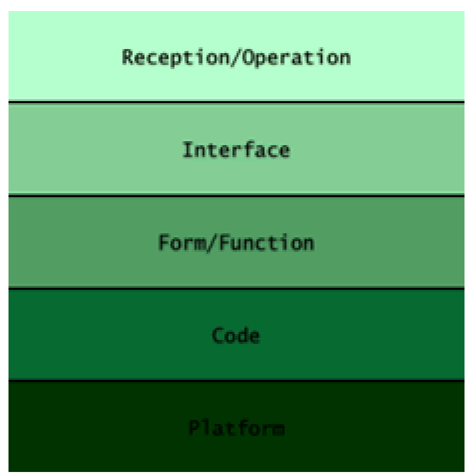
A

B

Figure 5-2. A) Screenshot of *Dig Dug* for Atari 2600 as played with the emulator Stella running on Windows XP. B) A photograph of *Dig Dug*'s title screen as played on an actual CRT television hooked up to an actual Atari 2600.



A



B

Figure 5-3. A) Montfort's table of game levels for computer game analysis; from his article "Combat in Context," <http://gamestudies.org/0601/articles/montfort>. B) An alternate version (size adjusted) of Montfort's table of five levels (with their ordinals removed) from the website introducing Montfort and Bogost's Platform Studies book series from MIT Press, <http://www.platformstudies.com>.

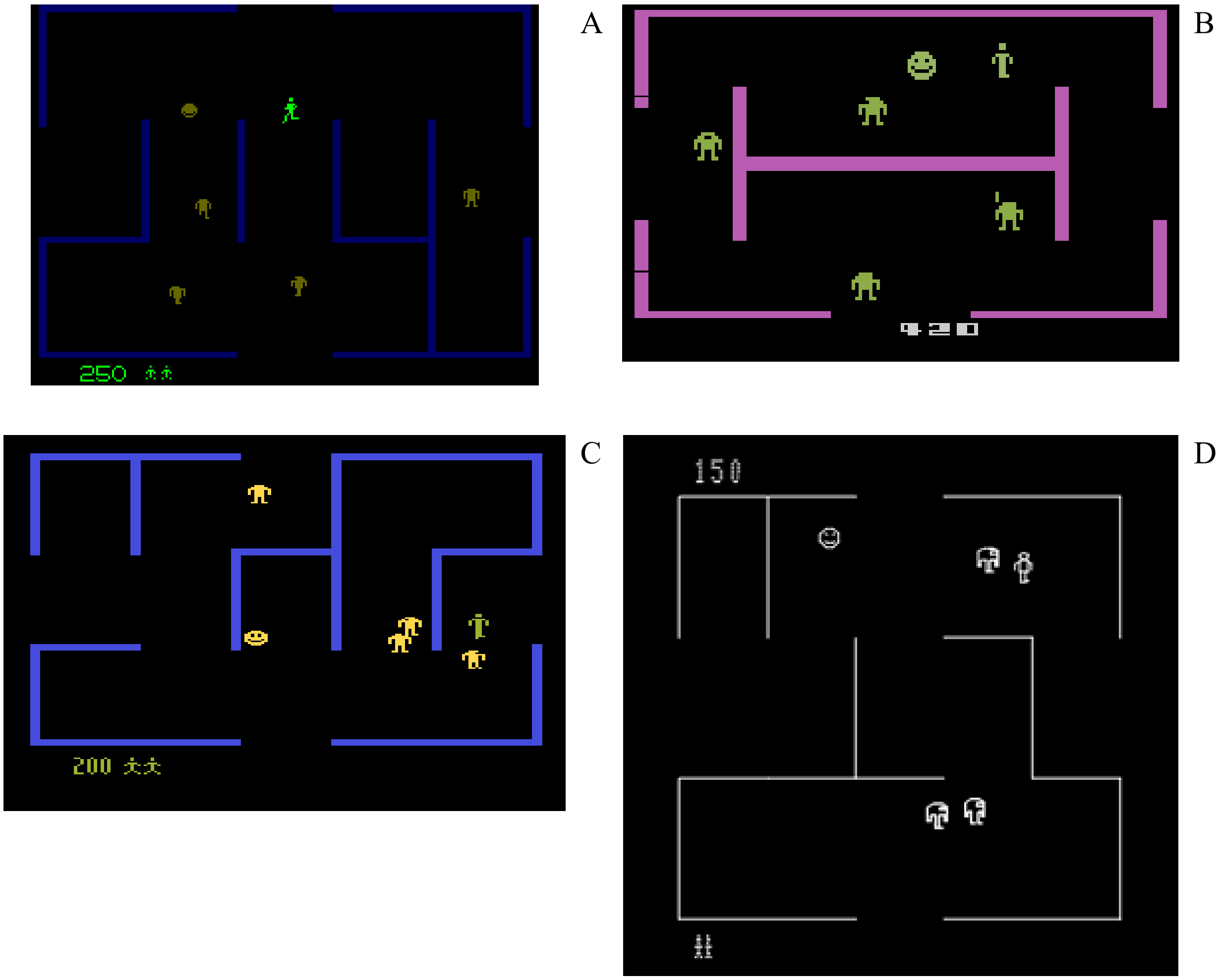


Figure 5-4. An array of the four licensed videogame versions of *Berzerk*. A) *Berzerk* the original arcade game [screenshot from MAME, scaled 25%]; B) *Berzerk* for Atari 2600 [screenshot from Stella, scaled 50%]; C) *Berzerk* for Atari 5200 [screenshot from MESS]. D) *Berzerk* for Vectrex [screenshot from MESS].

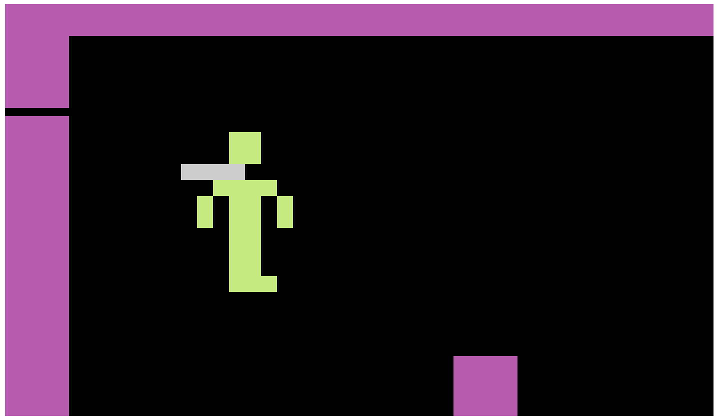
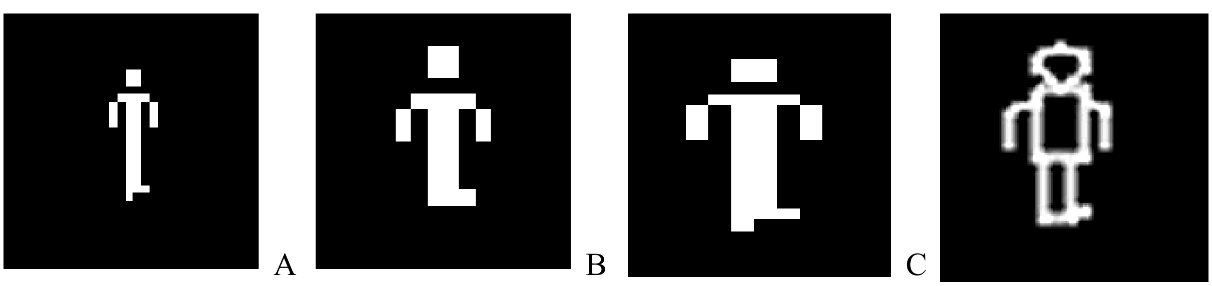


Figure 5-5. "Bulletproof bow tie" technique, demonstrated on *Berzerk* VCS.

Figure 5-6. Detail from flier advertising original *Berzerk* arcade machine. This artwork, appearing on the cabinet and promotional flier for *Berzerk*, is a photograph of the game monitor. It retains the monitor's phosphorescence and blurring instead of the un-mediated bitmap.



A B C D

Figure 5-7. The different humanoid stick figures from the four videogame versions of *Berzerk*. The gap between the figure's head and shoulders appears in all but the Vectrex version. A) Original arcade game [screenshot from MAME]; B) *Berzerk* for Atari VCS [screenshot from Stella]; C) *Berzerk* for Atari 5200 [screenshot from MESS]; D) *Berzerk* for Vectrex [screenshot from MESS].

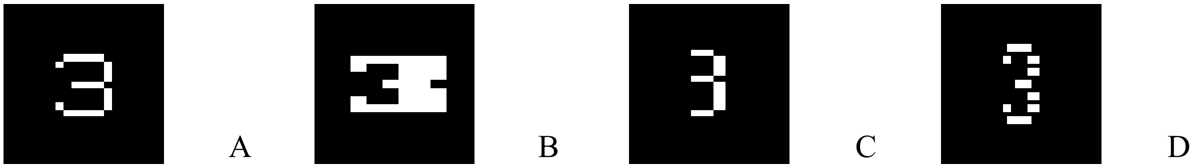


Figure 5-8. An array of different versions of the numeral 3 as depicted in different *Berzerk*s. A) *Berzerk* the original arcade game [screenshot from MAME]; B) *Berzerk* for Atari VCS [screenshot from Stella]; C) *Berzerk* for Atari 5200 [screenshot from MESS]; D) *Berzerk* for Vectrex [screenshot from MESS].

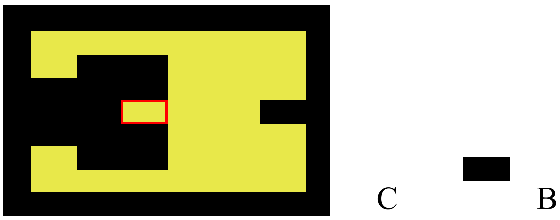
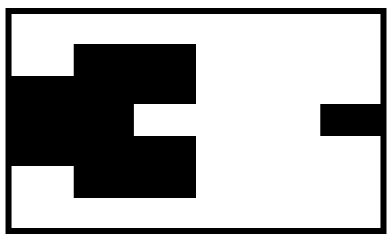


Figure 5-9. A) The numeral 3 from *Berzerk* for Atari VCS [screenshot from Stella]. The red outline highlights a single pixel. B) A solitary pixel from this numeral 3.

```
$3115     .XXXXXX.
$3116     .X..XXX.
$3117     ....XXX.
$3118     ...XXX..
$3119     ....XXX.
$3120     .X..XXX.
$3121     .XXXXXX.
```

A

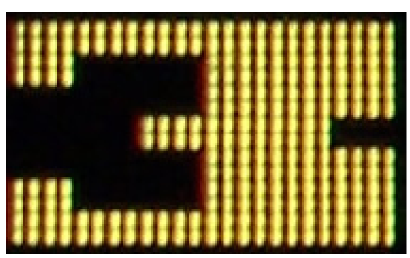


B

Figure 5-10. A) A representation a section of the machine code (ROM) which executes *Berzerk* for Atari.[12] B) The numeral 3 which is generated by the machine code in A).

```
$3115     .XXXXXX.
$3116     .X..XXX.
$3117     ....XXX.
$3118     ...XXX..
$3119     ....XXX.
$3120     .X..XXX.
$3121     .XXXXXX.
```

A



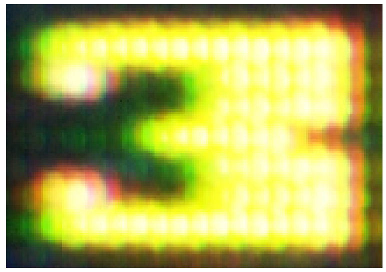
B

Figure 5-11. A) Same code section as figure 9A. B) Photograph of numeral 3 generated by Stella on LCD monitor.

12 The bitmap for the 3 sprite is actually stored upside-down, so the code above and below the 3 are the inverted 2 and 4, respectively.

```
$3115     .XXXXXX.
$3116     .X..XXX.
$3117     ....XXX.
$3118     ...XXX..
$3119     ....XXX.
$3120     .X..XXX.
$3121     .XXXXXX.
```

A

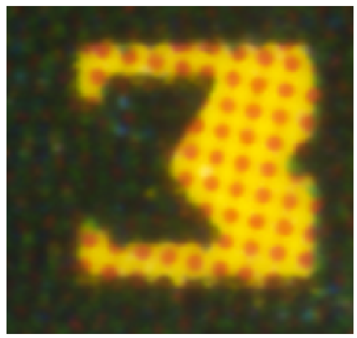


B

Figure 5-12. A) Same code section as previous two figures. B) Photograph (enlarged) of the same 3 generated by an actual Atari 2600 and displayed on a standard CRT television.

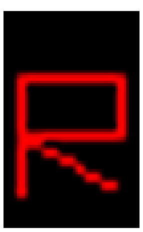
```
$3115     .XXXXXX.
$3116     .X..XXX.
$3117     ....XXX.
$3118     ...XXX..
$3119     ....XXX.
$3120     .X..XXX.
$3121     .XXXXXX.
```

A

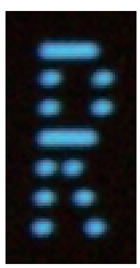


B

Figure 5-13. A print version of the same numeral 3 in *Berzerk* for Atari. A) Illustration of screen image (enlarged) from *Berzerk* for Atari instruction Manual. B) Greatly enlarged numeral 3 from 12A.



A



B

Figure 5-14. A) The letter R as rendered and displayed on a Vectrex (photographed without overlay). B) The letter R as displayed on an emulated *Battlezone* machine.

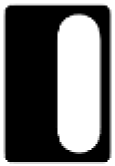
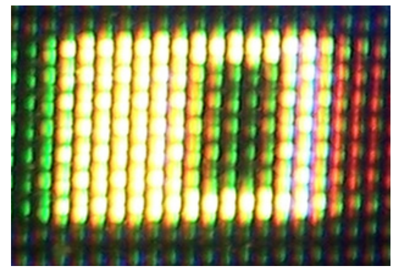
A

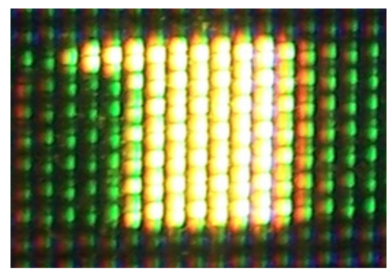
B

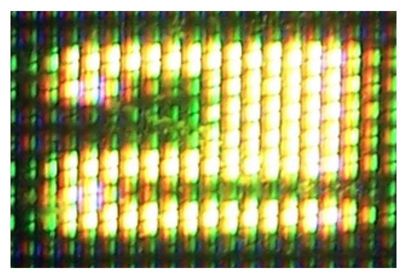
Figure 5-15. A) *Berzerk* score display photographed without overlay. B) Same score photographed with overlay.

Table 5-1. Countdown numerals, *Berzerk* bitmap code, and *Berzerk* screen images.

Countdown numeral characters*	*Berzerk* machine code**	*Berzerk* screen images***

	`3101 .XXXXXX.`
	`3100 .XXX..X.`
	`3099 .XXX..X.`
	`3098 .XXX..X.`
	`3097 .XXX..X.`
	`3096 .XXX..X.`
	`3095 .XXXXXX.`
		

	`3108 ..XXXX..`
	`3107 ...XXX..`
	`3106 ...XXX..`
	`3105 ...XXX..`
	`3104 ...XXX..`
	`3103 ...XXX..`
	`3102 ...XXX..`
		

	`3115 .XXXXXX.`
	`3114 .X..XXX.`
	`3113 ....XXX.`
	`3112 ....XXX.`
	`3111 .XXXXXX.`
	`3110 .X......`
	`3109 .XXXXXX.`
		

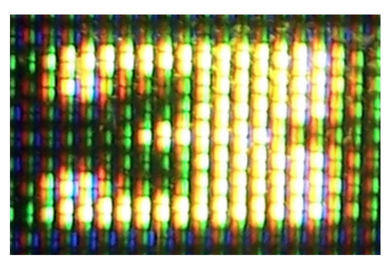
	`3115 .XXXXXX.`
	`3116 .X..XXX.`
	`3117 ....XXX.`
	`3118 ...XXX..`
	`3119 ....XXX.`
	`3120 .X..XXX.`
	`3121 .XXXXXX.`
		

Table 5-1 continued.

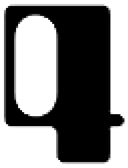
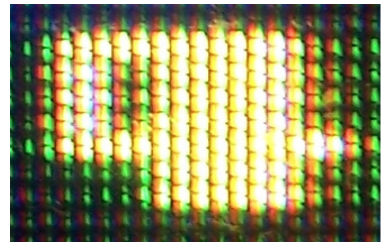
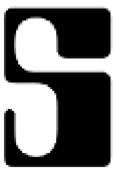
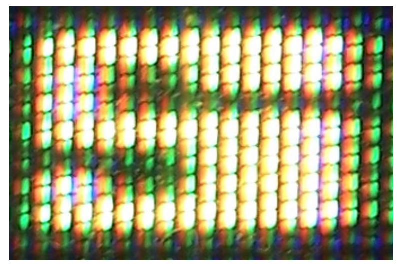
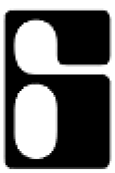
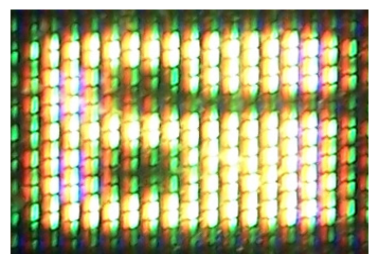
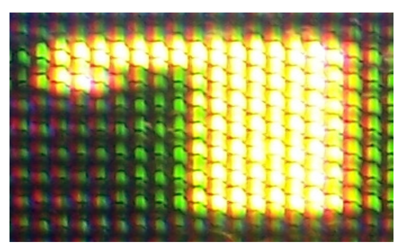
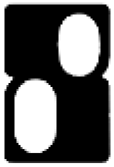
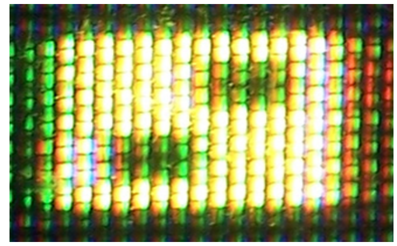
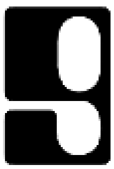
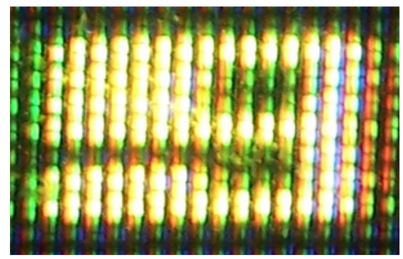
	3129	.XXXXX..	
	3128	.X.XXX..	
	3127	.X.XXX..	
	3126	.X.XXX..	
	3125	.XXXXXX.	
	3124	...XXX..	
	3123	...XXX..	
	3136	.XXXXXX.	
	3135	.X..XXX.	
	3134	.X......	
	3133	.XXXXXX.	
	3132	....XXX.	
	3131	.X..XXX.	
	3130	.XXXXXX.	
	3143	.XXXXXX.	
	3142	.X..XXX.	
	3141	.X......	
	3140	.XXXXXX.	
	3139	.X..XXX.	
	3138	.X..XXX.	
	3137	.XXXXXX.	
	3150	.XXXXXX.	
	3149	.X..XXX.	
	3148	....XXX.	
	3147	....XXX.	
	3146	....XXX.	
	3145	....XXX.	
	3144	....XXX.	

Table 5-1 continued.

	3157 .XXXXXX. 3156 .XXX..X. 3155 .XXX..X. 3154 .XXXXXX. 3153 .X..XXX. 3152 .X..XXX. 3151 .XXXXXX.	
	3164 .XXXXXX. 3163 .XXX..X. 3162 .XXX..X. 3161 .XXXXXX. 3160 ......X. 3159 .XXX..X. 3158 .XXXXXX.	

* Number images sampled from Countdown SH Regular, owned by Scangraphic Digital Type Collection. Preview images available at <http://www.myfonts.com/fonts/efscangraphic/countdown-sh/>.
** Code is printed in inverse of its actual ROM order to line up the images for comparison.
*** Photographs taken with a Kodak EasyShare CX7430 (4.0 Mega Pixels) at a distance of approximately 9 inches from screen surface.

APPENDIX A
ROMSCRAPE: A SOFTWARE METHOD FOR ANALYZING VIDEOGAME DATA

This appendix provides information about the design an implementation of a software application I developed to advance the research conducted throughout this study. Collectively bearing the name ROMscrape, this software involves several scripts written in the programming language Perl and culminates in a web-deployed search engine interface that allows users to search binary data for specific patterns of interest. The searching algorithm is non-discriminate, but for the research objectives for this study (particularly, the analysis of different typeface designs employed in Atari VCS games, as discussed in chapters 2 and 3), the data in question are the character designs for numbers and letters. These are stored as bitmap data in ROM memory, and because this character data is not programmatically accessible in a semantically consistent manner, an algorithm was employed that produced a series of visually associational indexes.[1] These indexes group together chunks of binary code with reference to their similarity as images and allow for search queries to return chunks exhibiting high degrees of similarity.

The following sections discuss the design and development of this alpha version software. It is presented here in anticipation of future wider releases as well as to demonstrate, in a more general sense, the applicability of data mining software for humanities research in digital artifacts. In this way, it is hoped that ROMscrape can contribute to the emerging field of Digital Humanities, where computer applications are developed to support humanistic research agendas such as literary analysis. What follows is presented as a narrative of how I identified and addressed a problem in my research, and while I do go into some technical detail, it is not my goal to present this appendix as a scientifically rigorous document.

1 I use the adverb *visually* somewhat reluctantly here because the index is never, strictly speaking, made visible to the user. A more accurate term would be an adverbial transformation of the word *image*, but I am not aware if such a word exists in English.

The Need

In terms of this project, ROMscrape fills an immediate need. As I approached one of the main research questions guiding this study – how have letter designs in console games evolved over time – I found it increasingly helpful to analyze typographic artifacts in videogames both in their "mediated" presentation on actual videogame screens, as well as in their encoded state as binary data. Sometimes a contrast between the two reveals evidence of discursive mediality on the part of type design, as discussed in chapter 4 and in Chapter 5 with regard to *Berzerk*. To make claims of this nature more general than single instances and, therefore, to enable a valid claim as to the uniqueness or normativity of the *Berzerk* example, I needed to quickly evaluate and compare hundreds of games for their typographic properties.

Focusing on the data set of Atari VCS ROMs available from various Atari fan-sites (mainly AtariAge.com), I arrived at a basic set of 528 ROM images,[2] ranging from 2KB to 16KB in size. I had a means for visually accessing binary data,[3] but with approximately 8MB worth of data, the prospect of browsing it byte by byte for interesting patterns presented an enormous obstacle. If estimated generously, with approximately 40 bytes per second (scrolling visually through a column of binary output), I predicted that it would take approximately 56 hours of constant scanning simply to look through all of the code. Adding in time for annotating and collating interesting similarities would extend the time even further. While this amount of time is not technically unmanageable, a better solution would be a software application that performed the tedious scanning for me – discarding meaningless data and retaining interesting patterns. The challenge to such a solution lies in educating software so that it can separate the signal of typography from the noise of everything else, including other graphics and programming code.

2 A ROM image is a digital copy of the data stored in ROM memory on removable Atari game cartridges. This kind of copy is referred to as an *image* because it is a bit-by-bit copy of the complete data stored on the cartridge. It is not the same kind of copying that occurs when files are transferred from one file system (say, a USB drive) to another.

3 This is the Perl script used to extract the character sets included in Appendix B. The full source code of this script is also included in that Appendix.

Similar Applications: ROMsearcher and DiStella

The inspiration and motivation for finding a software solution in ROMscrape derived in part from two pre-existing computer programs. These were were designed to perform a similar function, although not with the goal of humanistic research. The first, ROMSearcher, was designed by Chris Covell in order to find hidden or encoded text strings in videogames (Covell, "ROMSearcher README"). ROMsearcher works by scanning a chunk of bytes for a number sequence matching the ASCII interpolation of the text string in question. It also can search for patterns which use alternatives to ASCII or even the customized implementations of character sets like many Atari VCS games. Most computer systems display text based on the internal representation of numeric values that correspond to specific letters. The string 'APPLE,' would be represented by the following number sequence, (usually reported in hexadecimal notation): x41, x50, x50, x4c, x45, (decimal 65, 80, 80, 76, 69). An ASCII character set is, therefore, an internal table of values which equates x41 with A, x50 with P, x4c with L and x45 with E.

In a stream of binary data such as a videogame ROM image, the bytes which compose this text string would be represented as eight-digit binary encodings of the same value. In this manner, the string APPLE would be represented using the following binary sequence:

```
01000001
01010000
01010000
01001100
01000101
```

ROMSearcher works by looking through binary data for this specific pattern or, failing that, a pattern with the same sequence of numeric distance between bytes. In this way, an alternate character implementation (say, one in which x36 = A, etc.) can be discovered. If such a shift is identified, the game's remaining textual content can be converted and exported.

ROMSearcher is similar to ROMScrape[4] because it searches for textual patterns, but ROMSearcher addresses videogame content at a semantic level. In other words, in order to locate a particular text string, ROMSearcher requires that there be some internal representation of each letter in the string corresponding to numeric data. It thus relies on the same textual logic of the game or file system. The limitation this encounters is that many games include text which is stored as a single bitmap graphic for a word or phrase. Matthew Kirschenbaum discusses this problem of text and text-as-image in the course of a forensic analysis of a disk containing the computer game *Mystery House*. He notes that one important textual feature of the game is a series of mysterious notes scattered around the house. They present an obstacle for textual analysis, however, because "[t]heir 'text' is actually comprised by vector coordinates, which are stored elsewhere on the disk. One consequence of this is that we can't hack the game and discover the text of the notes by using a hex editor" (*Mechanisms* 130). An ideal software solution for extracting text would be one that recognizes the distinctive pattern of a textual character, say, a capital A, regardless of whether it is treated programmatically as a part of image data or textually within a character set.

DiStella Disassembler is another tool which provides a function similar to that aspired to in ROMscrape. As its programmer, Bob Colbert, explains, "Distella allows you to disassemble any 2k or 4k Atari 2600 ROM image into compilable source code!" (Colbert). This functionality is useful to the large community of homebrew Atari developers still writing games for the console because often the best way to learn programming in a particular language (in this case, the Assembly Language for the VCS's MOS 6502 processor) is to analyze and modify functioning examples. Programmers can then make changes to the disassembled code, reassemble it, and play the resulting game in an emulator to see the effect. One significant side-effect of disassembly is that it must separate programming code from game data – information like graphics and sound. "It uses

4 The similarity of the names is coincidental. I had arrived at the designation for ROMScrape prior to learning about ROMSearcher.

an ‘intelligent’ algorithm to distinguish between data and code, and even uses standard labels for the 2600’s registers” (Colbert). Figure A-1 shows a section of the output file from a disassembly of the VCS cartridge *Barnstorming*. ‘

Distella’s ability to quickly separate information from instructions is the basis of its application toward aesthetic ends. Ben Fry, an MIT student, has used Distella in combination with other programming to produce “distellamaps” – visualizations of Distella’s output of code and data – as part of his Visualizing Code project. Figure A-2 is a distellamap of Adventure, in which the famous Easter Egg text, “Created by Warren Robinett,” is clearly visible among the orange-shaded blocks of data.[5] These visualizations are clearly enjoyable on an aesthetic level, and are even available for purchase in poster sizes.

Ben Fry has also created other code visualization projects which seek to make visible the underlying processes of computational artifacts. One such project, Sprite Deconstructulator (see figure A-3), presents and highlights the bitmap contents of *Super Mario Brothers*, illustrating in real time the game’s system for accessing bitmap sprites and thus exposing the logic of the cartridge’s structure to visual inspection. Like Distella and distellamap, Deconstructulator’s function hovers between analysis and aesthetics, so while it is operating in the same artifactual space as ROMscrape, its use does not directly serve the interest of humanistic study. What is needed, and what ROMscrape offers, is a means to return data on queries which will allow researchers to arrive at better understandings of how videogames work across different game genres and (eventually) other platforms in order to better understand how videogames participate in and are a product of human culture and society

5 In these visualizations, data is rendered such that an orange block corresponds to a 1 value, and a 0 is represented by blank space. Instructions are printed using their opcodes, and linked together with curving lines whenever a GOTO call or reference is performed.

Deployment in Drupal

Whereas tools like Distella and ROMsearcher run as command line scripts or with a minimal GUI and are aimed at use for specific tasks, the goal of ROMscrape was to provide a widely accessible and distributed interface on the web. This situation creates a situation of collective inquiry and scholarship – a context in which the database of source material could be expanded by users and, more importantly, searches can be saved, shared, and compared. To accomplish these goals, I chose to develop ROMscrape for deployment within the open-source Content Management System Drupal, first created by Dries Bruyteart and now maintained by a developer community of thousands. Unlike similar web software such as WordPress or Movable Type, Drupal provides a flexible framework and a database abstraction layer which simplifies database queries.

This wide scope and immediate avenue for dissemination did, however, introduce its own problems as it requires search queries and any other calculations to be performed within the relatively limited computational environment of a shared web hosting account. Assessing and overcoming this limitation provided a crucial breakthrough by necessitating the development of a shortcut algorithm. The basic operation of ROMscrape is to take a query pattern and find similar patterns within a set of binary data, but given a dataset of roughly 528 games, occupying approximately 3.5MB of memory, the total number of possible chunks which could match any given query might number as high as 3.2 million.[6] Within the relatively modest resources available for this work (my own computer and a shared web server), the magnitude of this comparison proved far too processor-intensive.

The problem is that the basic comparison is not just between the query chunk and each of the 3.2 million chunks in the data set. Each comparison must be further broken down into a byte-to-

6 Assuming that a query is a pattern consisting of a sequence of at least 4 bytes, each ROM image or file may be divided into as many chunks as there are bytes, minus 4. For example, viable four-byte chunks appear at offset 0 – 3, 1 – 4, 2 – 5, etc. If the file is only 4008 bytes long, however, the highest complete four-byte chunk would appear at offset 4004 – 4008. In other words, there are almost as many viable chunks in a given set as there are individual bytes.

byte comparison, which is itself composed of a bit-to-bit comparison. In essence, therefore, the number of comparisons being performed to execute a simple comparison query of a four-byte chunk (the arbitrarily determined minimum size providing recognizable character data) would actually be on the order of 102.4 million comparisons. This bit-level comparison is necessary because the relevant data involved in any given bit is not its numerical significance with regard to the decimal equivalent of the binary value stored in the byte, but rather its visual significance within an image. Thus, the position of any given bit value with regard to the image it composes must be compared to the equivalent value at the same position in a comparison chunk. Figure A-4 illustrates this by contrasting visual similarity with numeric similarity. Whereas the numeric value of a byte could be expressed in its decimal equivalent and thus compared to other decimals, this single comparison provides no information relevant to visual similarity. This decimal comparison has the advantage of being a single calculation, as opposed to the 8 calculations required to compare each bit with each separate bit, but the decimal value has no necessary relationship to its visual appearance within a bitmap.

The prospect of performing 102.4 million comparisons for each query made obvious the need to find a better way: the tasks of research and analysis require flexibility and responsiveness in a digital tool, and the long-form of comparison was going to take hours for each search. The answer was to find a shortcut – some way to organize the dataset so that obviously or dramatically different search queries would be passed over and more likely matches would receive closer inspection.

A Two-Stage Index

I considered a number of different approaches to this problem, eventually arriving at a method of building an index in two stages. With the goal of reducing the total number of comparisons for a minimally processor intensive deployment on a web server, I determined to limit the number of comparisons within any individual search to 3000. In other words, a given query image would never be compared to any more than 3000 images in the data set within a given pass. In order to accommodate the total data set, this required a two-dimensional array of values and two passes through this index to find the best possible match or matches. Essentially, this structure amounts to a table with up to 3000 columns and 3000 rows. The first pass compares the query to each row header, selecting the best match; the second focuses only on that row, selecting the best items from within that set of no more than 3000 columns. The trick to making something like this successful is making sure that each row header correctly attracts the query to its best possible chance of finding a good match. In other words, the bestness of the row header must communicate the same bestness as its constituents on that row. The first stage in forming this index, therefore, was to determine the most effective possible values for row headers. Several options were considered, but the eventual solution was to create what I call a "fuzzy thumbnail" of a given set of visually similar chunks.

A fuzzy thumbnail is created as a composite of two or more actual chunks or thumbnails. Where the two images differ at a particular bit, the fuzzy thumbnail creates a compromise value between the two. For combinations involving more than two images, this compromise is also weighted to indicate the proportion of values comprising that bit. In this way, the fuzzy chunks provide a reliable best match index for a series of progressively associated chunks, which are formed through the process described below as the first index pass. This concept is illustrated in figure A-5.

Using a Perl script, the first stage of index formation scans each ROM file in the set and extracts every possible four-byte chunk.[7] These chunks are compiled into a table that records the

7 Although the goal of ROMscrape is to allow searching for chunks of any length up to 12 bytes, statistical comparisons demonstrated that the variety of four-byte chunks created a more useful and manageable data set for collation than two-, three-, five-, or six-byte chunks. Variable length is currently accomplished by combining partial chunks within the actual search script built into Drupal. This functionality, however, does not return results with sufficient consistency, so I am reviewing this function for future versions.

number of times a particular chunk appears. The script proceeds by finding a chunk and comparing it to the existing chunks in the table. If an exact match is found, then the total number or "weight" for that chunk is incremented. Otherwise, that chunk is identified as unique and appended to the end of the table with a weight value of 1. This pass is fairly easy to form computationally, since the comparison of chunk to chunk simply determines "match" or "no match," rather than the degrees of difference matching required for the comparative and qualitative analysis in users' search queries, and this first step usually completes with less than one minute of computation on a moderately equipped personal computer.

The output of this step is a table of each existing chunk within the dataset, totaling about 1.5 million. This is a large number, but is significantly smaller than the potential total number of unique four-byte sequences that are possible mathematically, 4,294,967,296.[8] This significant reduction indicates the difference between meaningless data (every possible byte sequence as determined by mathematical possibility) and meaningful data (only those byte sequences which actually appear in the Atari games analyzed). Still, the goal for the search index is to reduce this set of 1.5 million data pairs to a "smarter" set of 3000 pairs, so the initial pass has to be reduced further through a process of folding.

This step is much more time-consuming, and because the newly-folded index requires a progressive structure, I could not distribute the task to multiple computers. Instead, I committed my personal desktop computer to the task, which ended up taking several days to complete. The length of this task, I also discovered, made it necessary to build in the possibility of the script pausing and resuming its processing. This step of index folding works by sorting the table mentioned above and using the 3000 "heaviest" patterns as starting points for indexes. Next, each

8 This number is calculated by considering that each byte consists of 8 bits which may each be 1 or 0. Given a minimum value of 00000000 (decimal 0) and maximum value of 11111111 (decimal 255), the total number of possible values for a given byte of data is 256. Each of those 256 possibilities may be combined with each of the same 256 possibilities in the next byte. So the total value of possible byte combinations for a two-byte sequence is 256^2 or 65,536. Therefore, the total possibly byte sequences for a four-byte chunk is 256^4 or 4,294,967,296.

remaining pattern is matched against the 3000 heavy patterns, locating its best possible match. The two chunks are combined into a fuzzy thumbnail, and the new thumbnail replaces the heavier thumbnail on the stack of 3000.

Once this set of 3000 is determined, the final step in forming the index is to assign each actual chunk within the dataset to its proper index. This required simply processing through each ROM file and matching each chunk to the appropriate index. Since this is not recursive or progressive, it was possible to distribute the processing of this task to several different computers. Eventually involving 6 computers running simultaneously,[9] it was still another two days before the fully indexed data was complete.

At this point, the remaining challenges were relatively straightforward, mainly these consisted of constructing a Drupal module to handle the search, which required a JavaScript front-end for inputting pattern queries and an efficient means for organizing the data within the index. I also incorporated a secondary function for ROMscrape, which retrieves and lists the binary data of each cartridge in the set. This was then cross-referenced with the search results so that a matched chunk can be displayed in the context of the full cartridge it is extracted from.

Sample Query

At this point, ROMscrape was functional enough to begin using it for my research. Object A-1 is a short video of one of the search queries I performed for chapter 5. Having identified a particular shape of interest, the numeral 3 as it appears in *Berzerk*, I wanted to see what other games used that same shape or similar ones. The video demonstrates entering the 3 pattern as a search query, and returning a result which shows six exact matches, including *Berzerk,* the original game I found it in. Clicking on the "View this chunk in context" link takes me to a page illustrating the complete contents of one of these games, *Defender*. The context confirms that this is indeed the same character data that appears in *Berzerk*. Next, clicking on the context link for a different

9 I am grateful to my father, Paul Whalen, for donating processing time on his computer for this purpose.

chunk, one from the game *Space Jockey*, confirms that although this is a similar 3, the context of this character set is not identical, it is, nevertheless, similar to Countdown, the original typeface which may have inspired *Berzerk*'s numbering. Finally, the chunk identified in *Space Jockey* can be modified to include the full 3-character, and used as a new search query. This new query returns the same games already identified as similar, but ranks them differently.

As may be evident in the sample query, there are a number of outlying bugs and infelicities in the deployment of the software. In addition to improvements in the front-end, the initial process of forming the index needs improvement, as well as the search algorithm which collates results for odd-length queries. Essentially, there are a several stages at which an arbitrary rounding or a similar compromise may be excluding data sets. Some basic auditing of the final stage of index building indicates that the final distribution does not match the distribution projected by the cumulative weight of the fuzzy thumbnails. They are similar enough to return reliable results, but the difference indicates that some improvement is possible. The time-consuming task of experimentation with this process will hopefully be eliminated with access to more powerful computing resources in the future. Eventually, ROMscrape will be released to the public on a Drupal website and contributed as an open-source module for others to extend and modify. Long-term features will include the ability for users to save and compare search results, and add new ROM images to the database.

It is my hope that this software proves to be a valuable addition to the field of Digital Humanities, making it possible to mitigate the problems introduced by the heterogeneous nature of text and image in digital environments.

```
START:
       SEI
       CLD
       LDX    #$00
LF004: LDA    #$00
LF006: STA    VSYNC,X
       TXS
       INX
       BNE    LF006
       JSR    LFD47
       LDA    $82
       BNE    LF01B
       LDX    #$01
       STX    $82
       DEX
       JMP    LF64F
LF01B: LDX    #$07
LF01D: LDA    LF0AD,X
       EOR    $85
       AND    $86
       STA    $87,X
       CPX    #$01
       BCS    LF02C
       STA    COLUBK,X
LF02C: DEX
       BPL    LF01D
       LDA    #$21
       LDX    #$00
       JSR    LFDBD
       LDA    #$28
       INX
       STX    CTRLPF
       JSR    LFDBD
       LDA    $DD
```

```
LFDEC: .byte $78 ; | XXXX   | $FE00
       .byte $CC ; |XX  XX  | $FE01
       .byte $CC ; |XX  XX  | $FE02
       .byte $CC ; |XX  XX  | $FE03
       .byte $CC ; |XX  XX  | $FE04
       .byte $CC ; |XX  XX  | $FE05
       .byte $CC ; |XX  XX  | $FE06
       .byte $78 ; | XXXX   | $FE07
       .byte $78 ; | XXXX   | $FE08
       .byte $30 ; |  XX    | $FE09
       .byte $30 ; |  XX    | $FE0A
       .byte $30 ; |  XX    | $FE0B
       .byte $30 ; |  XX    | $FE0C
       .byte $30 ; |  XX    | $FE0D
       .byte $70 ; | XXX    | $FE0E
       .byte $30 ; |  XX    | $FE0F
       .byte $FC ; |XXXXXX  | $FE10
       .byte $C0 ; |XX      | $FE11
       .byte $C0 ; |XX      | $FE12
       .byte $78 ; | XXXX   | $FE13
       .byte $0C ; |    XX  | $FE14
       .byte $0C ; |    XX  | $FE15
       .byte $8C ; |X   XX  | $FE16
       .byte $78 ; | XXXX   | $FE17
       .byte $78 ; | XXXX   | $FE18
       .byte $8C ; |X   XX  | $FE19
       .byte $0C ; |    XX  | $FE1A
       .byte $18 ; |   XX   | $FE1B
       .byte $18 ; |   XX   | $FE1C
       .byte $0C ; |    XX  | $FE1D
       .byte $8C ; |X   XX  | $FE1E
       .byte $78 ; | XXXX   | $FE1F
```

Figure A-1. Selection of output from Distella Dissassembler showing the separation of code and data. This particular example is the output of the game *Barn Storm*.

Figure A-2. Distellamap of Adventure.bin, created by Ben Fry using a modification of Distella Disassembler.

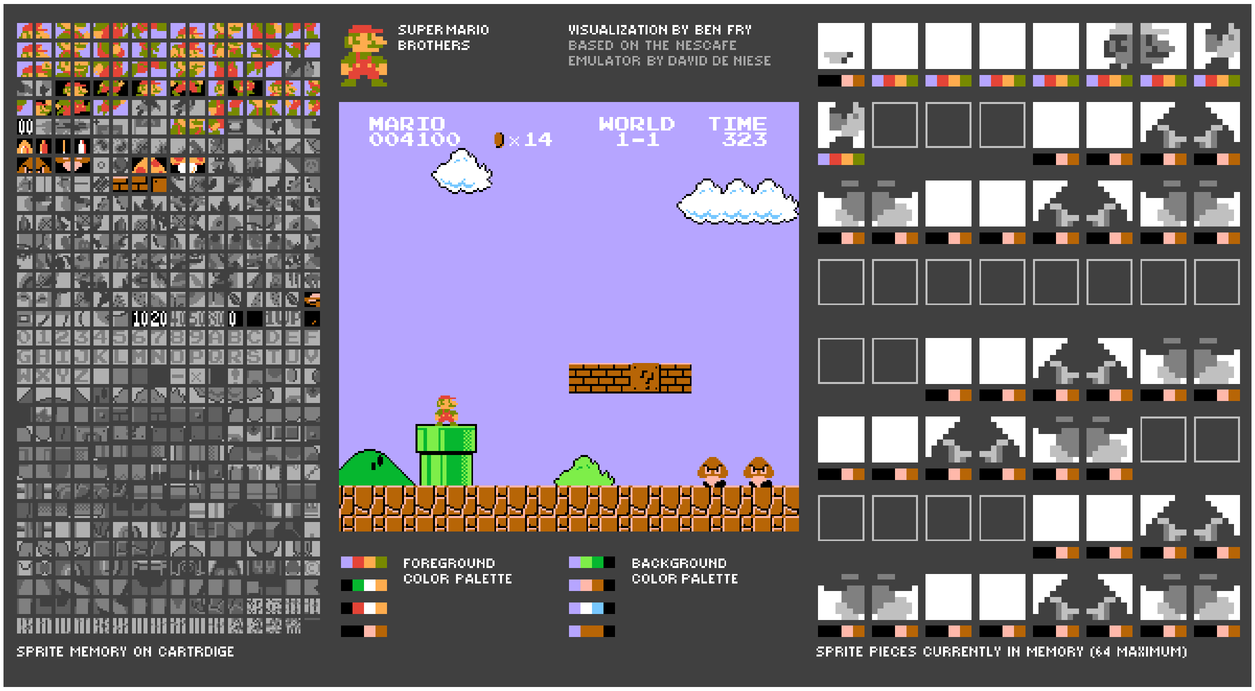


Figure A-3. Screenshot of Ben Fry's Sprite Deconstructulator.

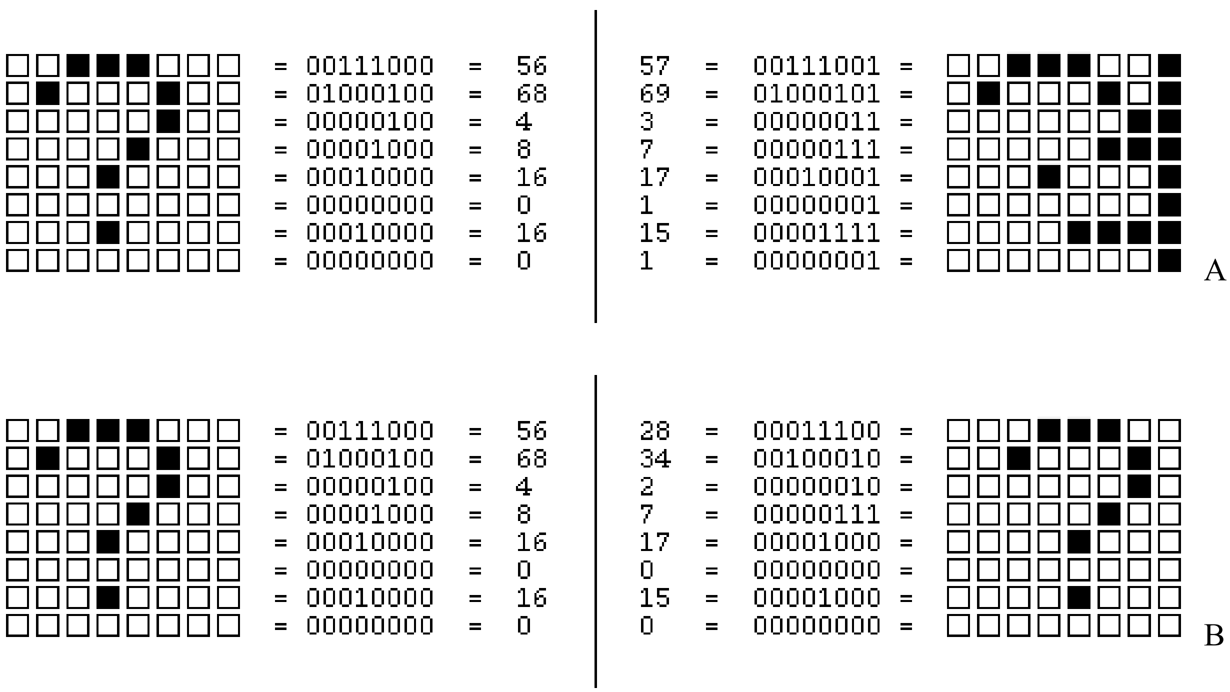


Figure A-4. Contrasting numeric similarity with visual similarity. A) Two chunks with high numeric similarity but low visual similarity. B) Two chunks with low numeric similarity but high visual similarity.

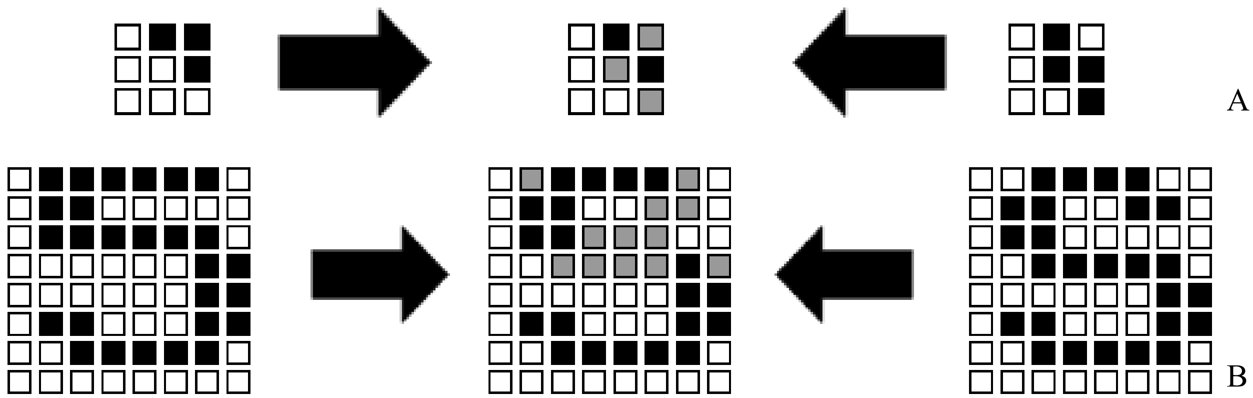
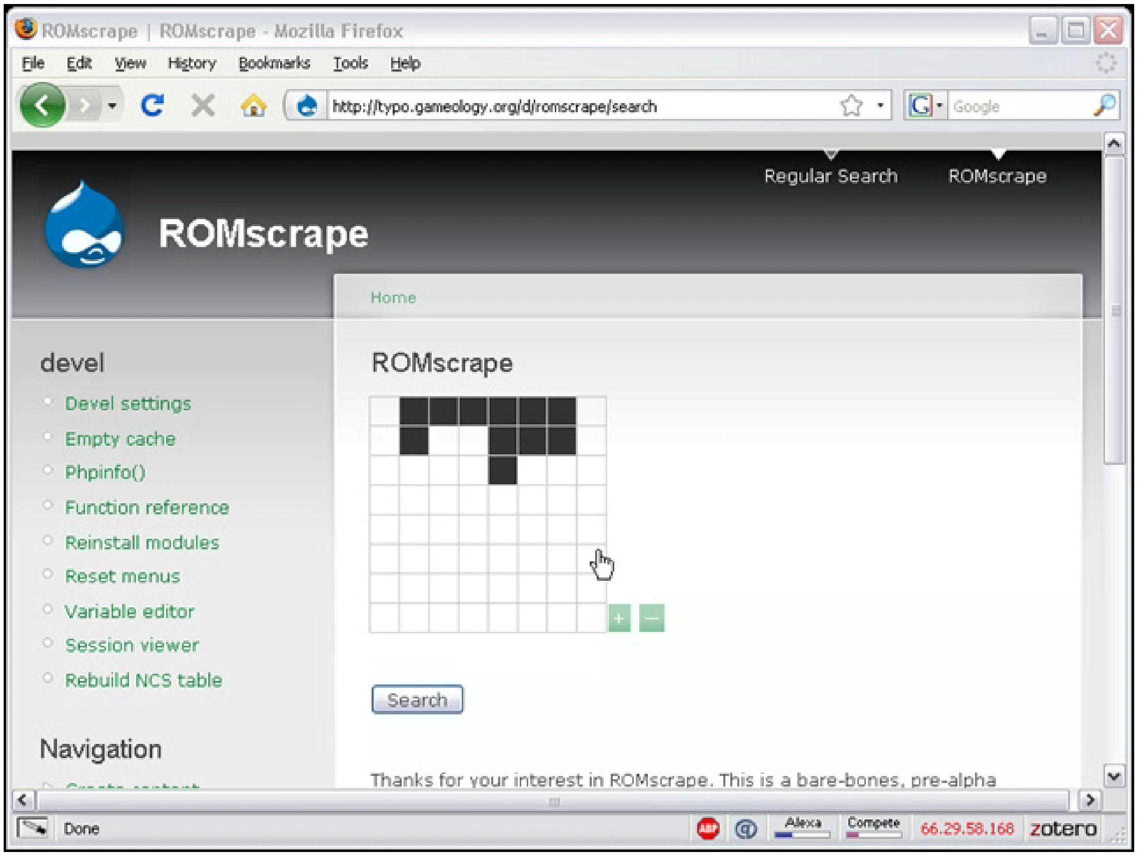


Figure A-5. Forming fuzzy thumbnails. Squares that are not the same value calculate an average (shown in gray) between the two. A) A simple 3 x 3 grid illustrating the point. B) An actual pair of bitmaps which might be combined within ROMscrape's index.



Object A-1. Video capture of ROMscrape performing a search and comparing the results. Click on the image above to start playback; alternatively, click on this text to download the file directly. (.avi file 8.4MB)

APPENDIX B
CHARACTER SETS, TYPEFACES, AND TYPE SPECIMENS

This appendix gathers in one location the various typefaces and character sets referenced throughout the study. Each table below represents the character extracted from the specified console BIOS. In most cases, these BIOS ROM images are readily available on the Internet, where they are distributed in order to run various console emulating software. Comparing these selected character sets reveals the subtle differences in design choices available to system designers, and because the tables are organized chronologically, these choices bear witness to the evolution of certain lettering styles over time.

I employed a simple Perl script to extract the data and format it into appropriately sized tables.

Table B-1. Character set stored in Channel F BIOS ROM.

$0367	XXXXX···	$036c	··X·····	$0371	XXXXX···	$0376	XXXXX···	$037b	X···X···	$0380	XXXXX···	$0385	XXXXX···	$038a	XXXXX···	$038f	XXXXX···
$0368	X···X···	$036d	··X·····	$0372	····X···	$0377	····X···	$037c	X···X···	$0381	X·······	$0386	X·······	$038b	····X···	$0390	X···X···
$0369	X···X···	$036e	··X·····	$0373	XXXXX···	$0378	XXXXX···	$037d	XXXXX···	$0382	XXXXX···	$0387	XXXXX···	$038c	···X····	$0391	XXXXX···
$036a	X···X···	$036f	··X·····	$0374	X·······	$0379	····X···	$037e	····X···	$0383	····X···	$0388	X···X···	$038d	···X····	$0392	X···X···
$036b	XXXXX···	$0370	··X·····	$0375	XXXXX···	$037a	XXXXX···	$037f	····X···	$0384	XXXXX···	$0389	XXXXX···	$038e	···X····	$0393	XXXXX···
$0394	XXXXX···	$0399	XXXXX···	$039e	XXXXX···	$03a3	XXXXX···	$03a8	········	$03ad	XXXXX···	$03b2	X···X···	$03b7	XXXXXXXX	$03bc	········
$0395	X···X···	$039a	X·······	$039f	····X···	$03a4	··X·····	$03a9	········	$03ae	X·X·X···	$03b3	·X·X····	$03b8	XXXXXXXX	$03bd	··X·····
$0396	XXXXX···	$039b	X··XX···	$03a0	··XXX···	$03a5	··X·····	$03aa	········	$03af	X·X·X···	$03b4	··X·····	$03b9	XXXXXXXX	$03be	········
$0397	····X···	$039c	X···X···	$03a1	········	$03a6	··X·····	$03ab	········	$03b0	X·X·X···	$03b5	·X·X····	$03ba	XXXXXXXX	$03bf	··X·····
$0398	XXXXX···	$039d	XXXXX···	$03a2	··X·····	$03a7	··X·····	$03ac	········	$03b1	X·X·X···	$03b6	X···X···	$03bb	XXXXXXXX	$03c0	········
$03c1	········	$03c6	·X·X····	$03cb	X·X·····	$03d0	XX······	$03d5	····X···	$03da	··X·····	$03df	···X····	$03e4	····X···	$03e9	········
$03c2	········	$03c7	·X·X····	$03cc	X·X·····	$03d1	XX······	$03d6	····X···	$03db	··X·····	$03e0	···X····	$03e5	···X····	$03ea	···XX···
$03c3	XXXXX···	$03c8	·X·X····	$03cd	X·X·····	$03d2	········	$03d7	····X···	$03dc	··X·····	$03e1	··X·····	$03e6	··X·····	$03eb	··X·····
$03c4	········	$03c9	·X·X····	$03ce	X·X·····	$03d3	········	$03d8	····X···	$03dd	··X·····	$03e2	·X······	$03e7	·X······	$03ec	XX······
$03c5	········	$03ca	·X·X····	$03cf	X·X·····	$03d4	········	$03d9	····X···	$03de	··X·····	$03e3	·X······	$03e8	X·······	$03ed	········
$03ee	········	$03f3	X·······	$03f8	·X······												
$03ef	XX······	$03f4	·X······	$03f9	·X······												
$03f0	··X·····	$03f5	··X·····	$03fa	··X·····												
$03f1	···XX···	$03f6	···X····	$03fb	···X····												
$03f2	········	$03f7	····X···	$03fc	···X····												

Table B-2. Character set stored in GCE Vectrex BIOS ROM. Each figure is interlaced so that the bytes which compose the image are stored at a distance of 80 from one another.

$19f4	········	$19f5	··X·····	$19f6	·X·X····	$19f7	·X·X····	$19f8	··X·····	$19f9	XX··X···	$19fa	··X·····	$19fb	···X····	$19fc	···X····
$1a44	········	$1a45	·XXX····	$1a46	·X·X····	$1a47	·X·X····	$1a48	·XXXX···	$1a49	XX··X···	$1a4a	·X·X····	$1a4b	··X·····	$1a4c	··X·····
$1a94	········	$1a95	·XXX····	$1a96	·X·X····	$1a97	XXXXX···	$1a98	X·X·····	$1a99	···X····	$1a9a	·X·X····	$1a9b	·X······	$1a9c	·X······
$1ae4	········	$1ae5	··X·····	$1ae6	········	$1ae7	·X·X····	$1ae8	·XXX····	$1ae9	··X·····	$1aea	·XX·····	$1aeb	········	$1aec	·X······
$1b34	········	$1b35	········	$1b36	········	$1b37	XXXXX···	$1b38	·XXX····	$1b39	·X······	$1b3a	X·X·X···	$1b3b	········	$1b3c	·X······
$1b84	········	$1b85	········	$1b86	········	$1b87	·X·X····	$1b88	··X·X···	$1b89	X··XX···	$1b8a	X··X····	$1b8b	········	$1b8c	··X·····
$1bd4	········	$1bd5	··X·····	$1bd6	········	$1bd7	·X·X····	$1bd8	XXXXX···	$1bd9	X··XX···	$1bda	·XX·X···	$1bdb	········	$1bdc	···X····
$19fd	·X······	$19fe	··X·····	$19ff	········	$1a00	········	$1a01	········	$1a02	········	$1a03	····X···	$1a04	··XX····	$1a05	··X·····
$1a4d	··X·····	$1a4e	X·X·X···	$1a4f	··X·····	$1a50	········	$1a51	········	$1a52	········	$1a53	····X···	$1a54	·X··X···	$1a55	·XX·····
$1a9d	···X····	$1a9e	·XXX····	$1a9f	··X·····	$1aa0	········	$1aa1	········	$1aa2	········	$1aa3	···X····	$1aa4	·X··X···	$1aa5	··X·····
$1aed	···X····	$1aee	X·X·X···	$1aef	XXXXX···	$1af0	········	$1af1	·XXX····	$1af2	········	$1af3	··X·····	$1af4	·X··X···	$1af5	··X·····
$1b3d	···X····	$1b3e	X·X·X···	$1b3f	··X·····	$1b40	·X······	$1b41	········	$1b42	········	$1b43	·X······	$1b44	·X··X···	$1b45	··X·····
$1b8d	··X·····	$1b8e	········	$1b8f	··X·····	$1b90	·X······	$1b91	········	$1b92	········	$1b93	X·······	$1b94	·X··X···	$1b95	··X·····
$1bdd	·X······	$1bde	········	$1bdf	········	$1be0	X·······	$1be1	········	$1be2	X·······	$1be3	X·······	$1be4	··XX····	$1be5	·XXX····
$1a06	·XXX····	$1a07	·XXX····	$1a08	···X····	$1a09	XXXXX···	$1a0a	··XX····	$1a0b	XXXXX···	$1a0c	·XXX····	$1a0d	·XXX····	$1a0e	········
$1a56	X···X···	$1a57	X···X···	$1a58	··XX····	$1a59	X·······	$1a5a	·X······	$1a5b	····X···	$1a5c	X···X···	$1a5d	X···X···	$1a5e	·XX·····
$1aa6	····X···	$1aa7	····X···	$1aa8	·X·X····	$1aa9	XXXX····	$1aaa	X·······	$1aab	···X····	$1aac	X···X···	$1aad	X···X···	$1aae	·XX·····
$1af6	·XXX····	$1af7	··XX····	$1af8	X··X····	$1af9	····X···	$1afa	XXXX····	$1afb	··X·····	$1afc	·XXX····	$1afd	·XXXX···	$1afe	········
$1b46	X·······	$1b47	····X···	$1b48	XXXXX···	$1b49	····X···	$1b4a	X···X···	$1b4b	·X······	$1b4c	X···X···	$1b4d	····X···	$1b4e	·XX·····
$1b96	X·······	$1b97	X···X···	$1b98	···X····	$1b99	X···X···	$1b9a	X···X···	$1b9b	X·······	$1b9c	X···X···	$1b9d	···X····	$1b9e	·XX·····
$1be6	XXXXX···	$1be7	·XXX····	$1be8	···X····	$1be9	·XXX····	$1bea	·XXX····	$1beb	X·······	$1bec	·XXX····	$1bed	·XX·····	$1bee	········
$1a0f	·XX·····	$1a10	········	$1a11	········	$1a12	········	$1a13	·XXX····	$1a14	·XXX····	$1a15	··X·····	$1a16	XXXX····	$1a17	·XXX····
$1a5f	·XX·····	$1a60	···X····	$1a61	········	$1a62	·X······	$1a63	X···X···	$1a64	X···X···	$1a65	·X·X····	$1a66	·X··X···	$1a67	X···X···
$1aaf	········	$1ab0	··X·····	$1ab1	·XXXX···	$1ab2	··X·····	$1ab3	····X···	$1ab4	X·X·X···	$1ab5	X···X···	$1ab6	·X··X···	$1ab7	X·······
$1aff	·XX·····	$1b00	·X······	$1b01	········	$1b02	···X····	$1b03	···X····	$1b04	X·XXX···	$1b05	X···X···	$1b06	·XXX····	$1b07	X·······
$1b4f	·XX·····	$1b50	··X·····	$1b51	·XXXX···	$1b52	··X·····	$1b53	··X·····	$1b54	X·XX····	$1b55	XXXXX···	$1b56	·X··X···	$1b57	X·······
$1b9f	··X·····	$1ba0	···X····	$1ba1	········	$1ba2	·X······	$1ba3	········	$1ba4	X·······	$1ba5	X···X···	$1ba6	·X··X···	$1ba7	X···X···
$1bef	·X······	$1bf0	········	$1bf1	········	$1bf2	········	$1bf3	··X·····	$1bf4	·XXXX···	$1bf5	X···X···	$1bf6	XXXX····	$1bf7	·XXX····
$1a18	XXXX····	$1a19	XXXXX···	$1a1a	XXXXX···	$1a1b	·XXXX···	$1a1c	X···X···	$1a1d	·XXX····	$1a1e	····X···	$1a1f	X···X···	$1a20	X·······
$1a68	·X··X···	$1a69	X·······	$1a6a	X·······	$1a6b	X·······	$1a6c	X···X···	$1a6d	··X·····	$1a6e	····X···	$1a6f	X··X····	$1a70	X·······
$1ab8	·X··X···	$1ab9	X·······	$1aba	X·······	$1abb	X·······	$1abc	X···X···	$1abd	··X·····	$1abe	····X···	$1abf	X·X·····	$1ac0	X·······
$1b08	·X··X···	$1b09	XXX·····	$1b0a	XXX·····	$1b0b	X··XX···	$1b0c	XXXXX···	$1b0d	··X·····	$1b0e	····X···	$1b0f	XX······	$1b10	X·······
$1b58	·X··X···	$1b59	X·······	$1b5a	X·······	$1b5b	X···X···	$1b5c	X···X···	$1b5d	··X·····	$1b5e	····X···	$1b5f	X·X·····	$1b60	X·······
$1ba8	·X··X···	$1ba9	X·······	$1baa	X·······	$1bab	X···X···	$1bac	X···X···	$1bad	··X·····	$1bae	X···X···	$1baf	X··X····	$1bb0	X···X···
$1bf8	XXXX····	$1bf9	XXXXX···	$1bfa	X·······	$1bfb	·XXXX···	$1bfc	X···X···	$1bfd	·XXX····	$1bfe	·XXX····	$1bff	X···X···	$1c00	XXXXX···
$1a21	X···X···	$1a22	X···X···	$1a23	XXXXX···	$1a24	XXXX····	$1a25	·XXX····	$1a26	XXXX····	$1a27	·XXX····	$1a28	XXXXX···	$1a29	X···X···
$1a71	XX·XX···	$1a72	XX··X···	$1a73	X···X···	$1a74	X···X···	$1a75	X···X···	$1a76	X···X···	$1a77	X···X···	$1a78	X·X·X···	$1a79	X···X···
$1ac1	X·X·X···	$1ac2	X·X·X···	$1ac3	X···X···	$1ac4	X···X···	$1ac5	X···X···	$1ac6	X···X···	$1ac7	·X······	$1ac8	··X·····	$1ac9	X···X···
$1b11	X·X·X···	$1b12	X··XX···	$1b13	X···X···	$1b14	XXXX····	$1b15	X···X···	$1b16	XXXX····	$1b17	··X·····	$1b18	··X·····	$1b19	X···X···
$1b61	X···X···	$1b62	X···X···	$1b63	X···X···	$1b64	X·······	$1b65	X·X·X···	$1b66	X·X·····	$1b67	···X····	$1b68	··X·····	$1b69	X···X···
$1bb1	X···X···	$1bb2	X···X···	$1bb3	X···X···	$1bb4	X·······	$1bb5	X··X····	$1bb6	X··X····	$1bb7	X···X···	$1bb8	··X·····	$1bb9	X···X···
$1c01	X···X···	$1c02	X···X···	$1c03	XXXXX···	$1c04	X·······	$1c05	·XX·X···	$1c06	X···X···	$1c07	·XXX····	$1c08	··X·····	$1c09	·XXX····

Table B-2 continued.

$1a2a	X···X···	$1a2b	X···X···	$1a2c	X···X···	$1a2d	X···X···	$1a2e	XXXXX···	$1a2f	·XXX····	$1a30	X·······	$1a31	·XXX····	$1a32	··X·····
$1a7a	X···X···	$1a7b	X···X···	$1a7c	X···X···	$1a7d	X···X···	$1a7e	····X···	$1a7f	·X······	$1a80	X·······	$1a81	····X···	$1a82	·X·X····
$1aca	X···X···	$1acb	X···X···	$1acc	·X·X····	$1acd	·X·X····	$1ace	···X····	$1acf	·X······	$1ad0	·X······	$1ad1	····X···	$1ad2	X···X···
$1b1a	·X·X····	$1b1b	X·X·X···	$1b1c	··X·····	$1b1d	··X·····	$1b1e	··X·····	$1b1f	·X······	$1b20	··X·····	$1b21	····X···	$1b22	········
$1b6a	·X·X····	$1b6b	X·X·X···	$1b6c	·X·X····	$1b6d	··X·····	$1b6e	·X······	$1b6f	·X······	$1b70	···X····	$1b71	····X···	$1b72	········
$1bba	··X·····	$1bbb	X·X·X···	$1bbc	X···X···	$1bbd	··X·····	$1bbe	X·······	$1bbf	·X······	$1bc0	····X···	$1bc1	····X···	$1bc2	········
$1c0a	··X·····	$1c0b	·X·X····	$1c0c	X···X···	$1c0d	··X·····	$1c0e	XXXXX···	$1c0f	·XXX····	$1c10	····X···	$1c11	·XXX····	$1c12	········

$1a33	········	$1a34	········	$1a35	··X·····	$1a36	····X···	$1a37	··X·····	$1a38	········	$1a39	········	$1a3a	········	$1a3b	··XXX···
$1a83	········	$1a84	········	$1a85	·XXX····	$1a86	····XX··	$1a87	··X·····	$1a88	·XXX····	$1a89	·XXX····	$1a8a	········	$1a8b	·X···X··
$1ad3	········	$1ad4	·XXX····	$1ad5	X·X·X···	$1ad6	····X·X·	$1ad7	··X·····	$1ad8	X···X···	$1ad9	XXXXX···	$1ada	·XX·····	$1adb	X·XXX·X·
$1b23	········	$1b24	XXXXXXX·	$1b25	··X·····	$1b26	····X···	$1b27	··X·····	$1b28	X···X···	$1b29	XXXXX···	$1b2a	XXXX····	$1b2b	X·X···X·
$1b73	········	$1b74	XXXXXXX·	$1b75	··X·····	$1b76	·XXXX···	$1b77	X·X·X···	$1b78	X···X···	$1b79	XXXXX···	$1b7a	XXXX····	$1b7b	X·XXX·X·
$1bc3	········	$1bc4	·X··X···	$1bc5	··X·····	$1bc6	XXXX····	$1bc7	·XXX····	$1bc8	·XXX····	$1bc9	·XXX····	$1bca	·XX·····	$1bcb	·X···X··
$1c13	XXXXX···	$1c14	········	$1c15	··X·····	$1c16	·XX·····	$1c17	··X·····	$1c18	········	$1c19	········	$1c1a	········	$1c1b	··XXX···

$1a3c	···X····	$1a3d	··X·····	$1a3e	·X···X··	$1a3f	·X···X··	$1a40	········	$1a41	XXXXXXX·	$1a42	XXXXXXXX	$1a43	XXXXXXX·
$1a8c	···X····	$1a8d	·XXX····	$1a8e	········	$1a8f	········	$1a90	·XX·XX··	$1a91	X·····X·	$1a92	XXXXXXXX	$1a93	XXXXXXX·
$1adc	··XXX···	$1add	··X·····	$1ade	········	$1adf	········	$1ae0	X··X··X·	$1ae1	X·····X·	$1ae2	XXXXXXXX	$1ae3	XXXXXXX·
$1b2c	··XXX···	$1b2d	XXXXX···	$1b2e	X·····X·	$1b2f	··XXX···	$1b30	X··X··X·	$1b31	X·····X·	$1b32	XXXXXXXX	$1b33	XXXXXXX·
$1b7c	·XXXXX··	$1b7d	··X·····	$1b7e	·X···X··	$1b7f	·X···X··	$1b80	·XX·XX··	$1b81	X·····X·	$1b82	XXXXXXXX	$1b83	XXXXXXX·
$1bcc	·XX·XX··	$1bcd	·X·X····	$1bce	··XXX···	$1bcf	X·····X·	$1bd0	········	$1bd1	X·····X·	$1bd2	XXXXXXXX	$1bd3	XXXXXXX·
$1c1c	X·····X·	$1c1d	X···X···	$1c1e	········	$1c1f	········	$1c20	········	$1c21	XXXXXXX·	$1c22	XXXXXXXX	$1c23	XXXXXXX·

Table B-3. Complete character set extracted from Intellivision graphics ROM (grom.bin). Originally, this data was stored in a ROM chip within the Intellivision console.

$0000	········	$0008	··XX····	$0010	·XX··XX·	$0018	········	$0020	···X····	$0028	········	$0030	···X····	$0038	···X····	$0040	····XXX·
$0001	········	$0009	··XX····	$0011	·XX··XX·	$0019	···X·X··	$0021	XXXXXXX·	$0029	·XX···X·	$0031	·XXXXX··	$0039	··X·····	$0041	····X···
$0002	········	$000a	··XX····	$0012	········	$001a	··XXXXX·	$0022	XX·X····	$002a	·XX··X··	$0032	·XX·····	$003a	·X······	$0042	····X···
$0003	········	$000b	··XX····	$0013	········	$001b	···X·X··	$0023	XXXXXXX·	$002b	····X···	$0033	··XXX···	$003b	········	$0043	····X···
$0004	········	$000c	··XX····	$0014	········	$001c	··XXXXX·	$0024	···X·XX·	$002c	···X····	$0034	·XX·····	$003c	········	$0044	····X···
$0005	········	$000d	········	$0015	········	$001d	···X·X··	$0025	XX·X·XX·	$002d	··X··XX·	$0035	·XXXXX··	$003d	········	$0045	····X···
$0006	········	$000e	··XX····	$0016	········	$001e	········	$0026	XXXXXXX·	$002e	·X···XX·	$0036	···X····	$003e	········	$0046	····X···
$0007	········	$000f	········	$0017	········	$001f	········	$0027	···X····	$002f	········	$0037	········	$003f	········	$0047	····XXX·
$0048	·XXX····	$0050	···X····	$0058	········	$0060	········	$0068	········	$0070	········	$0078	·······X	$0080	········	$0088	········
$0049	···X····	$0051	··XXX···	$0059	···XX···	$0061	········	$0069	········	$0071	········	$0079	······X·	$0081	XXXXXXX·	$0089	··XXX···
$004a	···X····	$0052	·XX·XX··	$005a	···XX···	$0062	········	$006a	········	$0072	········	$007a	·····X··	$0082	XX···XX·	$008a	···XX···
$004b	···X····	$0053	··XXX···	$005b	·XXXXXX·	$0063	········	$006b	·XXXXXX·	$0073	········	$007b	····X···	$0083	XX·X·XX·	$008b	···XX···
$004c	···X····	$0054	···X····	$005c	···XX···	$0064	········	$006c	········	$0074	········	$007c	···X····	$0084	XX·X·XX·	$008c	···XX···
$004d	···X····	$0055	········	$005d	···XX···	$0065	···XX···	$006d	········	$0075	···XX···	$007d	··X·····	$0085	XX···XX·	$008d	···XX···
$004e	···X····	$0056	········	$005e	········	$0066	···XX···	$006e	········	$0076	···XX···	$007e	·X······	$0086	XXXXXXX·	$008e	·XXXXXX·
$004f	·XXX····	$0057	········	$005f	········	$0067	····X···	$006f	········	$0077	········	$007f	X·······	$0087	········	$008f	········
$0090	········	$0098	········	$00a0	········	$00a8	········	$00b0	········	$00b8	········	$00c0	········	$00c8	········	$00d0	········
$0091	·XXXXXX·	$0099	·XXXXXX·	$00a1	·XX··XX·	$00a9	·XXXXXX·	$00b1	·XXXXXX·	$00b9	·XXXXXX·	$00c1	·XXXXXX·	$00c9	·XXXXXX·	$00d1	········
$0092	·XX··XX·	$009a	·····XX·	$00a2	·XX··XX·	$00aa	·XX·····	$00b2	·XX·····	$00ba	·····XX·	$00c2	·XX··XX·	$00ca	·XX··XX·	$00d2	···XX···
$0093	·····XX·	$009b	··XXXX··	$00a3	·XX··XX·	$00ab	·XXXXXX·	$00b3	·XXXXXX·	$00bb	····XX··	$00c3	··XXXX··	$00cb	·XX··XX·	$00d3	···XX···
$0094	·XXXXXX·	$009c	·····XX·	$00a4	·XXXXXX·	$00ac	·····XX·	$00b4	·XX··XX·	$00bc	···XX···	$00c4	·XX··XX·	$00cc	·XXXXXX·	$00d4	········
$0095	·XX·····	$009d	·····XX·	$00a5	·····XX·	$00ad	·XX··XX·	$00b5	·XX··XX·	$00bd	··XX····	$00c5	·XX··XX·	$00cd	·····XX·	$00d5	···XX···
$0096	·XXXXXX·	$009e	·XXXXXX·	$00a6	·····XX·	$00ae	·XXXXXX·	$00b6	·XXXXXX·	$00be	··XX····	$00c6	·XXXXXX·	$00ce	·XXXXXX·	$00d6	···XX···
$0097	········	$009f	········	$00a7	········	$00af	········	$00b7	········	$00bf	········	$00c7	········	$00cf	········	$00d7	········
$00d8	········	$00e0	········	$00e8	········	$00f0	········	$00f8	·XXXXXX·	$0100	XXXXXXX·	$0108	·XXXXXX·	$0110	·XXXXXX·	$0118	·XXXXXX·
$00d9	········	$00e1	·····XX·	$00e9	········	$00f1	·XX·····	$00f9	·XX··XX·	$0101	X·····X·	$0109	·XX··XX·	$0111	·XX··XX·	$0119	·XX··XX·
$00da	···XX···	$00e2	···XX···	$00ea	·XXXXXX·	$00f2	···XX···	$00fa	·····XX·	$0102	X·XXX·X·	$010a	·XX··XX·	$0112	·XX··XX·	$011a	·XX·····
$00db	···XX···	$00e3	·XX·····	$00eb	········	$00f3	·····XX·	$00fb	···XXXX·	$0103	X·X·X·X·	$010b	·XX··XX·	$0113	·XXXXX··	$011b	·XX·····
$00dc	········	$00e4	···XX···	$00ec	·XXXXXX·	$00f4	···XX···	$00fc	···XX···	$0104	X·XXXXX·	$010c	·XXXXXX·	$0114	·XX··XX·	$011c	·XX·····
$00dd	···XX···	$00e5	·····XX·	$00ed	········	$00f5	·XX·····	$00fd	········	$0105	X·······	$010d	·XX··XX·	$0115	·XX··XX·	$011d	·XX··XX·
$00de	···XX···	$00e6	········	$00ee	········	$00f6	········	$00fe	···XX···	$0106	XXXXXXX·	$010e	·XX··XX·	$0116	·XXXXXX·	$011e	·XXXXXX·
$00df	····X···	$00e7	········	$00ef	········	$00f7	········	$00ff	········	$0107	········	$010f	········	$0117	········	$011f	········
$0120	·XXXXX··	$0128	·XXXXXX·	$0130	·XXXXXX·	$0138	·XXXXXX·	$0140	·XX··XX·	$0148	·XXXXXX·	$0150	·····XX·	$0158	·XX··XX·	$0160	·XX·····
$0121	·XX··XX·	$0129	·XX·····	$0131	·XX·····	$0139	·XX··XX·	$0141	·XX··XX·	$0149	···XX···	$0151	·····XX·	$0159	·XX··XX·	$0161	·XX·····
$0122	·XX··XX·	$012a	·XX·····	$0132	·XX·····	$013a	·XX·····	$0142	·XX··XX·	$014a	···XX···	$0152	·····XX·	$015a	·XX·XX··	$0162	·XX·····
$0123	·XX··XX·	$012b	·XXXXX··	$0133	·XXXXX··	$013b	·XX·XXX·	$0143	·XXXXXX·	$014b	···XX···	$0153	·····XX·	$015b	·XXXX···	$0163	·XX·····
$0124	·XX··XX·	$012c	·XX·····	$0134	·XX·····	$013c	·XX··XX·	$0144	·XX··XX·	$014c	···XX···	$0154	·XX··XX·	$015c	·XX··XX·	$0164	·XX·····
$0125	·XX··XX·	$012d	·XX·····	$0135	·XX·····	$013d	·XX··XX·	$0145	·XX··XX·	$014d	···XX···	$0155	·XX··XX·	$015d	·XX··XX·	$0165	·XX·····
$0126	·XXXXX··	$012e	·XXXXXX·	$0136	·XX·····	$013e	·XXXXXX·	$0146	·XX··XX·	$014e	·XXXXXX·	$0156	·XXXXXX·	$015e	·XX··XX·	$0166	·XXXXXX·
$0127	········	$012f	········	$0137	········	$013f	········	$0147	········	$014f	········	$0157	········	$015f	········	$0167	········

Table B-3 continued.

$0168	X·····X·	$0170	·X···XX·	$0178	·XXXXXX·	$0180	·XXXXXX·	$0188	·XXXXXX·	$0190	·XXXXXX·	$0198	·XXXXXX·	$01a0	·XXXXXX·	$01a8	·XX··XX·
$0169	XX···XX·	$0171	·XX··XX·	$0179	·XX··XX·	$0181	·XX··XX·	$0189	·XX··XX·	$0191	·XX··XX·	$0199	·XX··XX·	$01a1	···XX···	$01a9	·XX··XX·
$016a	XXX·XXX·	$0172	·XXX·XX·	$017a	·XX··XX·	$0182	·XX··XX·	$018a	·XX··XX·	$0192	·XX··XX·	$019a	·XX·····	$01a2	···XX···	$01aa	·XX··XX·
$016b	XXXXXXX·	$0173	·XXXXXX·	$017b	·XX··XX·	$0183	·XX··XX·	$018b	·XX··XX·	$0193	·XX··XX·	$019b	·XXXXXX·	$01a3	···XX···	$01ab	·XX··XX·
$016c	XX·X·XX·	$0174	·XX·XXX·	$017c	·XX··XX·	$0184	·XXXXXX·	$018c	·XX··XX·	$0194	·XXXXX··	$019c	·····XX·	$01a4	···XX···	$01ac	·XX··XX·
$016d	XX·X·XX·	$0175	·XX··XX·	$017d	·XX··XX·	$0185	·XX·····	$018d	·XX·XXX·	$0195	·XX··XX·	$019d	·XX··XX·	$01a5	···XX···	$01ad	·XX··XX·
$016e	XX···XX·	$0176	·XX···X·	$017e	·XXXXXX·	$0186	·XX·····	$018e	·XXXXXX·	$0196	·XX··XX·	$019e	·XXXXXX·	$01a6	···XX···	$01ae	·XXXXXX·
$016f	········	$0177	········	$017f	········	$0187	········	$018f	······XX	$0197	········	$019f	········	$01a7	········	$01af	········

$01b0	·XX··XX·	$01b8	XX···XX·	$01c0	·XX··XX·	$01c8	·XX··XX·	$01d0	·XXXXXX·	$01d8	···XXXX·	$01e0	X·······	$01e8	·XXXX···	$01f0	···X····
$01b1	·XX··XX·	$01b9	XX···XX·	$01c1	·XX··XX·	$01c9	·XX··XX·	$01d1	·····XX·	$01d9	···XX···	$01e1	·X······	$01e9	···XX···	$01f1	··XXX···
$01b2	·XX··XX·	$01ba	XX·X·XX·	$01c2	··XXXX··	$01ca	·XX··XX·	$01d2	····XX··	$01da	···XX···	$01e2	··X·····	$01ea	···XX···	$01f2	·X·X·X··
$01b3	··X··X··	$01bb	XX·X·XX·	$01c3	···XX···	$01cb	·XX··XX·	$01d3	···XX···	$01db	···XX···	$01e3	···X····	$01eb	···XX···	$01f3	X··X··X·
$01b4	··XXXX··	$01bc	XXXXXXX·	$01c4	··XXXX··	$01cc	·XXXXXX·	$01d4	··XX····	$01dc	···XX···	$01e4	····X···	$01ec	···XX···	$01f4	···X····
$01b5	···XX···	$01bd	·XX·XX··	$01c5	·XX··XX·	$01cd	···XX···	$01d5	·XX·····	$01dd	···XX···	$01e5	·····X··	$01ed	···XX···	$01f5	···X····
$01b6	···XX···	$01be	·XX·XX··	$01c6	·XX··XX·	$01ce	···XX···	$01d6	·XXXXXX·	$01de	···XX···	$01e6	······X·	$01ee	···XX···	$01f6	···X····
$01b7	········	$01bf	········	$01c7	········	$01cf	········	$01d7	········	$01df	···XXXX·	$01e7	·······X	$01ef	·XXXX···	$01f7	········

$01f8	····X···	$0200	····X···	$0208	········	$0210	·XXX····	$0218	········	$0220	····XXX·	$0228	········	$0230	········	$0238	········
$01f9	···X····	$0201	·····X··	$0209	········	$0211	··XX····	$0219	········	$0221	····XX··	$0229	········	$0231	··XXXXX·	$0239	········
$01fa	··X·····	$0202	······X·	$020a	·XXXXX··	$0212	··XXXXX·	$021a	·XXXXXX·	$0222	·XXXXX··	$022a	·XXXXXX·	$0232	··XX····	$023a	·XXXXXX·
$01fb	·XXXXXX·	$0203	········	$020b	····XX··	$0213	··XX·XX·	$021b	·XX··XX·	$0223	·XX·XX··	$022b	·XX··XX·	$0233	·XXXXX··	$023b	·XX·XX··
$01fc	··X·····	$0204	········	$020c	·XXXXX··	$0214	··XX·XX·	$021c	·XX·····	$0224	·XX·XX··	$022c	·XXXXXX·	$0234	··XX····	$023c	·XX·XX··
$01fd	···X····	$0205	········	$020d	·XX·XX··	$0215	··XX·XX·	$021d	·XX·····	$0225	·XX·XX··	$022d	·XX·····	$0235	··XX····	$023d	·XXXXX··
$01fe	····X···	$0206	········	$020e	·XXXXXX·	$0216	··XXXXX·	$021e	·XXXXXX·	$0226	·XXXXX··	$022e	·XXXXXX·	$0236	··XX····	$023e	····XX··
$01ff	········	$0207	········	$020f	········	$0217	········	$021f	········	$0227	········	$022f	········	$0237	········	$023f	·XXXXX··

$0240	·XX·····	$0248	···XX···	$0250	·····XX·	$0258	·XX·····	$0260	··XXX···	$0268	········	$0270	········	$0278	········	$0280	········
$0241	·XX·····	$0249	········	$0251	········	$0259	·XX·····	$0261	···XX···	$0269	········	$0271	········	$0279	········	$0281	········
$0242	·XXXXX··	$024a	··XXX···	$0252	·····XX·	$025a	·XX··XX·	$0262	···XX···	$026a	XXXXXXX·	$0272	·XXXXXX·	$027a	·XXXXXX·	$0282	·XXXXXX·
$0243	·XX·XX··	$024b	···XX···	$0253	·····XX·	$025b	·XX·XX··	$0263	···XX···	$026b	XX·X·XX·	$0273	··XX·XX·	$027b	·XX··XX·	$0283	··XX·XX·
$0244	·XX·XX··	$024c	···XX···	$0254	·····XX·	$025c	·XXXX···	$0264	···XX···	$026c	XX·X·XX·	$0274	··XX·XX·	$027c	·XX··XX·	$0284	··XX·XX·
$0245	·XX·XX··	$024d	···XX···	$0255	··XX·XX·	$025d	·XX··XX·	$0265	···XX···	$026d	XX·X·XX·	$0275	··XX·XX·	$027d	·XX··XX·	$0285	··XXXXX·
$0246	·XX·XXX·	$024e	·XXXXXX·	$0256	··XX·XX·	$025e	·XX··XX·	$0266	·XXXXXX·	$026e	XX·X·XX·	$0276	··XX·XX·	$027e	·XXXXXX·	$0286	··XX····
$0247	········	$024f	········	$0257	··XXXXX·	$025f	········	$0267	········	$026f	········	$0277	········	$027f	········	$0287	··XX····

$0288	········	$0290	········	$0298	········	$02a0	········	$02a8	········	$02b0	········	$02b8	········	$02c0	········	$02c8	········
$0289	········	$0291	········	$0299	········	$02a1	··XX····	$02a9	········	$02b1	········	$02b9	········	$02c1	········	$02c9	········
$028a	·XXXXX··	$0292	·XXXXXX·	$029a	·XXXXXX·	$02a2	·XXXXXX·	$02aa	·XX·XX··	$02b2	·XX··XX·	$02ba	XX·X·XX·	$02c2	·XX··XX·	$02ca	·XXX·XX·
$028b	·XX·XX··	$0293	··XX·XX·	$029b	·XX·····	$02a3	··XX····	$02ab	·XX·XX··	$02b3	·XX··XX·	$02bb	XX·X·XX·	$02c3	··XXXX··	$02cb	··XX·XX·
$028c	·XX·XX··	$0294	··XX····	$029c	·XXXXXX·	$02a4	··XX····	$02ac	·XX·XX··	$02b4	·XX··XX·	$02bc	XX·X·XX·	$02c4	···XX···	$02cc	··XX·XX·
$028d	·XXXXX··	$0295	··XX····	$029d	·····XX·	$02a5	··XX····	$02ad	·XX·XX··	$02b5	··XXXX··	$02bd	XXXXXXX·	$02c5	··XXXX··	$02cd	··XXXXX·
$028e	····XX··	$0296	··XX····	$029e	·XXXXXX·	$02a6	··XXXXX·	$02ae	·XXXXXX·	$02b6	···XX···	$02be	·XX·XX··	$02c6	·XX··XX·	$02ce	·····XX·
$028f	····XXX·	$0297	········	$029f	········	$02a7	········	$02af	········	$02b7	········	$02bf	········	$02c7	········	$02cf	··XXXXX·

$02d0	········	$02d8	····XXX·	$02e0	···XX···	$02e8	·XXX····	$02f0	········
$02d1	········	$02d9	····X···	$02e1	···XX···	$02e9	···X····	$02f1	········
$02d2	·XXXXXX·	$02da	····X···	$02e2	···XX···	$02ea	···X····	$02f2	········
$02d3	·····XX·	$02db	··XX····	$02e3	···XX···	$02eb	····XX··	$02f3	·XXXX···
$02d4	···XX···	$02dc	····X···	$02e4	···XX···	$02ec	···X····	$02f4	···XXXX·
$02d5	·XX·····	$02dd	····X···	$02e5	···XX···	$02ed	···X····	$02f5	········
$02d6	·XXXXXX·	$02de	····XXX·	$02e6	···XX···	$02ee	·XXX····	$02f6	········

Table B-4. Character set extracted from ColecoVision original BIOS.

$5515	·XXXXXX·	$5523	···XXXXX	$5531	·X···X··	$5539	········	$5547	··X·····	$5555	·X·X····	$5563	·X·X····	$5571	··X·····	$5579	XX······
$5516	X······X	$5524	·····X··	$5532	·XX·XX··	$5540	········	$5548	··X·····	$5556	·X·X····	$5564	·X·X····	$5572	·XXXX···	$5580	XX··X···
$5517	X·XXXX·X	$5525	·····X··	$5533	·X·X·X··	$5541	········	$5549	··X·····	$5557	·X·X····	$5565	XXXXX···	$5573	X·X·····	$5581	···X····
$5518	X·X····X	$5526	·····X··	$5534	·X·X·X··	$5542	········	$5550	··X·····	$5558	········	$5566	·X·X····	$5574	·XXX····	$5582	··X·····
$5519	X·X····X	$5527	········	$5535	········	$5543	········	$5551	··X·····	$5559	········	$5567	XXXXX···	$5575	··X·X···	$5583	·X······
$5520	X·XXXX·X	$5528	········	$5536	········	$5544	········	$5552	········	$5560	········	$5568	·X·X····	$5576	XXXX····	$5584	X··XX···
$5521	X······X	$5529	········	$5537	········	$5545	········	$5553	··X·····	$5561	········	$5569	·X·X····	$5577	··X·····	$5585	···XX···
$5522	·XXXXXX·	$5530	········	$5538	········	$5546	········	$5554	········	$5562	········	$5570	········	$5578	········	$5586	········
$5587	·X······	$5595	··X·····	$5603	··X·····	$5611	··X·····	$5619	··X·····	$5627	········	$5635	········	$5643	········	$5651	········
$5588	X·X·····	$5596	··X·····	$5604	·X······	$5612	···X····	$5620	X·X·X···	$5628	··X·····	$5636	········	$5644	········	$5652	········
$5589	X·X·····	$5597	··X·····	$5605	X·······	$5613	····X···	$5621	·XXX····	$5629	··X·····	$5637	········	$5645	········	$5653	········
$5590	·X······	$5598	········	$5606	X·······	$5614	····X···	$5622	··X·····	$5630	XXXXX···	$5638	········	$5646	XXXXX···	$5654	········
$5591	X·X·X···	$5599	········	$5607	X·······	$5615	····X···	$5623	·XXX····	$5631	··X·····	$5639	··X·····	$5647	········	$5655	········
$5592	X··X····	$5600	········	$5608	·X······	$5616	···X····	$5624	X·X·X···	$5632	··X·····	$5640	··X·····	$5648	········	$5656	········
$5593	·XX·X···	$5601	········	$5609	··X·····	$5617	··X·····	$5625	··X·····	$5633	········	$5641	·X······	$5649	········	$5657	··X·····
$5594	········	$5602	········	$5610	········	$5618	········	$5626	········	$5634	········	$5642	········	$5650	········	$5658	········
$5659	········	$5667	·XXX····	$5675	··X·····	$5683	·XXX····	$5691	XXXXX···	$5699	···X····	$5707	XXXXX···	$5715	··XXX···	$5723	XXXXX···
$5660	····X···	$5668	X···X···	$5676	·XX·····	$5684	X···X···	$5692	····X···	$5700	··XX····	$5708	X·······	$5716	·X······	$5724	····X···
$5661	···X····	$5669	X··XX···	$5677	··X·····	$5685	····X···	$5693	···X····	$5701	·X·X····	$5709	XXXX····	$5717	X·······	$5725	···X····
$5662	··X·····	$5670	X·X·X···	$5678	··X·····	$5686	··XX····	$5694	··XX····	$5702	X··X····	$5710	····X···	$5718	XXXX····	$5726	··X·····
$5663	·X······	$5671	XX··X···	$5679	··X·····	$5687	·X······	$5695	····X···	$5703	XXXXX···	$5711	····X···	$5719	X···X···	$5727	·X······
$5664	X·······	$5672	X···X···	$5680	··X·····	$5688	X·······	$5696	X···X···	$5704	···X····	$5712	X···X···	$5720	X···X···	$5728	·X······
$5665	········	$5673	·XXX····	$5681	·XXX····	$5689	XXXXX···	$5697	·XXX····	$5705	···X····	$5713	·XXX····	$5721	·XXX····	$5729	·X······
$5666	········	$5674	········	$5682	········	$5690	········	$5698	········	$5706	········	$5714	········	$5722	········	$5730	········
$5731	·XXX····	$5739	·XXX····	$5747	········	$5755	········	$5763	···X····	$5771	········	$5779	·X······	$5787	·XXX····	$5795	·XXX····
$5732	X···X···	$5740	X···X···	$5748	········	$5756	········	$5764	··X·····	$5772	········	$5780	··X·····	$5788	X···X···	$5796	X···X···
$5733	X···X···	$5741	X···X···	$5749	··X·····	$5757	··X·····	$5765	·X······	$5773	XXXXX···	$5781	···X····	$5789	···X····	$5797	X·X·X···
$5734	·XXX····	$5742	·XXXX···	$5750	········	$5758	········	$5766	X·······	$5774	········	$5782	····X···	$5790	··X·····	$5798	X·XXX···
$5735	X···X···	$5743	····X···	$5751	··X·····	$5759	··X·····	$5767	·X······	$5775	XXXXX···	$5783	···X····	$5791	··X·····	$5799	X·XX····
$5736	X···X···	$5744	···X····	$5752	········	$5760	··X·····	$5768	··X·····	$5776	········	$5784	··X·····	$5792	········	$5800	X·······
$5737	·XXX····	$5745	XXX·····	$5753	········	$5761	·X······	$5769	···X····	$5777	········	$5785	·X······	$5793	··X·····	$5801	·XXXX···
$5738	········	$5746	········	$5754	········	$5762	········	$5770	········	$5778	········	$5786	········	$5794	········	$5802	········
$5803	··X·····	$5811	XXXX····	$5819	·XXX····	$5827	XXXX····	$5835	XXXXX···	$5843	XXXXX···	$5851	·XXXX···	$5859	X···X···	$5867	·XXX····
$5804	·X·X····	$5812	X···X···	$5820	X···X···	$5828	X···X···	$5836	X·······	$5844	X·······	$5852	X·······	$5860	X···X···	$5868	··X·····
$5805	X···X···	$5813	X···X···	$5821	X·······	$5829	X···X···	$5837	X·······	$5845	X·······	$5853	X·······	$5861	X···X···	$5869	··X·····
$5806	X···X···	$5814	XXXX····	$5822	X·······	$5830	X···X···	$5838	XXXX····	$5846	XXXX····	$5854	X·······	$5862	XXXXX···	$5870	··X·····
$5807	XXXXX···	$5815	X···X···	$5823	X·······	$5831	X···X···	$5839	X·······	$5847	X·······	$5855	X··XX···	$5863	X···X···	$5871	··X·····
$5808	X···X···	$5816	X···X···	$5824	X···X···	$5832	X···X···	$5840	X·······	$5848	X·······	$5856	X···X···	$5864	X···X···	$5872	··X·····
$5809	X···X···	$5817	XXXX····	$5825	·XXX····	$5833	XXXX····	$5841	XXXXX···	$5849	X·······	$5857	·XXXX···	$5865	X···X···	$5873	·XXX····
$5810	········	$5818	········	$5826	········	$5834	········	$5842	········	$5850	········	$5858	········	$5866	········	$5874	········

Table B-4 continued.

$5875	····X···	$5883	X···X···	$5891	X·······	$5899	X···X···	$5907	X···X···	$5915	·XXX····	$5923	XXXX····	$5931	·XXX····	$5939	XXXX····
$5876	····X···	$5884	X··X····	$5892	X·······	$5900	XX·XX···	$5908	X···X···	$5916	X···X···	$5924	X···X···	$5932	X···X···	$5940	X···X···
$5877	····X···	$5885	X·X·····	$5893	X·······	$5901	X·X·X···	$5909	XX··X···	$5917	X···X···	$5925	X···X···	$5933	X···X···	$5941	X···X···
$5878	····X···	$5886	XX······	$5894	X·······	$5902	X·X·X···	$5910	X·X·X···	$5918	X···X···	$5926	XXXX····	$5934	X···X···	$5942	XXXX····
$5879	····X···	$5887	X·X·····	$5895	X·······	$5903	X···X···	$5911	X··XX···	$5919	X···X···	$5927	X·······	$5935	X·X·X···	$5943	X·X·····
$5880	X···X···	$5888	X··X····	$5896	X·······	$5904	X···X···	$5912	X···X···	$5920	X···X···	$5928	X·······	$5936	X··X····	$5944	X··X····
$5881	·XXX····	$5889	X···X···	$5897	XXXXX···	$5905	X···X···	$5913	X···X···	$5921	·XXX····	$5929	X·······	$5937	·XX·X···	$5945	X···X···
$5882	········	$5890	········	$5898	········	$5906	········	$5914	········	$5922	········	$5930	········	$5938	········	$5946	········

$5947	·XXX····	$5955	XXXXX···	$5963	X···X···	$5971	X···X···	$5979	X···X···	$5987	X···X···	$5995	X···X···	$6003	XXXXX···	$6011	XXXXX···
$5948	X···X···	$5956	··X·····	$5964	X···X···	$5972	X···X···	$5980	X···X···	$5988	X···X···	$5996	X···X···	$6004	····X···	$6012	XX······
$5949	X·······	$5957	··X·····	$5965	X···X···	$5973	X···X···	$5981	X···X···	$5989	·X·X····	$5997	·X·X····	$6005	···X····	$6013	XX······
$5950	·XXX····	$5958	··X·····	$5966	X···X···	$5974	X···X···	$5982	X·X·X···	$5990	··X·····	$5998	··X·····	$6006	··X·····	$6014	XX······
$5951	····X···	$5959	··X·····	$5967	X···X···	$5975	X···X···	$5983	X·X·X···	$5991	·X·X····	$5999	··X·····	$6007	·X······	$6015	XX······
$5952	X···X···	$5960	··X·····	$5968	X···X···	$5976	·X·X····	$5984	XX·XX···	$5992	X···X···	$6000	··X·····	$6008	X·······	$6016	XX······
$5953	·XXX····	$5961	··X·····	$5969	·XXX····	$5977	··X·····	$5985	X···X···	$5993	X···X···	$6001	··X·····	$6009	XXXXX···	$6017	XXXXX···
$5954	········	$5962	········	$5970	········	$5978	········	$5986	········	$5994	········	$6002	········	$6010	········	$6018	········

$6019	········	$6027	XXXXX···	$6035	········	$6043	········	$6051	·X······	$6059	········	$6067	········	$6075	········	$6083	········
$6020	X·······	$6028	···XX···	$6036	········	$6044	········	$6052	··X·····	$6060	········	$6068	········	$6076	········	$6084	········
$6021	·X······	$6029	···XX···	$6037	··X·····	$6045	········	$6053	···X····	$6061	·XXX····	$6069	XXXX····	$6077	·XXXX···	$6085	XXXX····
$6022	··X·····	$6030	···XX···	$6038	·X·X····	$6046	········	$6054	········	$6062	X···X···	$6070	·X··X···	$6078	X·······	$6086	·X··X···
$6023	···X····	$6031	···XX···	$6039	X···X···	$6047	········	$6055	········	$6063	XXXXX···	$6071	·XXX····	$6079	X·······	$6087	·X··X···
$6024	····X···	$6032	···XX···	$6040	········	$6048	········	$6056	········	$6064	X···X···	$6072	·X··X···	$6080	X·······	$6088	·X··X···
$6025	········	$6033	XXXXX···	$6041	········	$6049	········	$6057	········	$6065	X···X···	$6073	XXXX····	$6081	·XXXX···	$6089	XXXX····
$6026	········	$6034	········	$6042	········	$6050	XXXXX···	$6058	········	$6066	········	$6074	········	$6082	········	$6090	········

$6091	········	$6099	········	$6107	········	$6115	········	$6123	········	$6131	········	$6139	········	$6147	········	$6155	········
$6092	········	$6100	········	$6108	········	$6116	········	$6124	········	$6132	········	$6140	········	$6148	········	$6156	········
$6093	XXXX····	$6101	XXXX····	$6109	·XXXX···	$6117	X···X···	$6125	XXXXX···	$6133	·XXX····	$6141	X··X····	$6149	X·······	$6157	X···X···
$6094	X·······	$6102	X·······	$6110	X·······	$6118	X···X···	$6126	··X·····	$6134	··X·····	$6142	X·X·····	$6150	X·······	$6158	XX·XX···
$6095	XXX·····	$6103	XXX·····	$6111	X·XXX···	$6119	XXXXX···	$6127	··X·····	$6135	··X·····	$6143	XX······	$6151	X·······	$6159	X·X·X···
$6096	X·······	$6104	X·······	$6112	X···X···	$6120	X···X···	$6128	··X·····	$6136	X·X·····	$6144	X·X·····	$6152	X·······	$6160	X···X···
$6097	XXXX····	$6105	X·······	$6113	·XXX····	$6121	X···X···	$6129	XXXXX···	$6137	XXX·····	$6145	X··X····	$6153	XXXXX···	$6161	X···X···
$6098	········	$6106	········	$6114	········	$6122	········	$6130	········	$6138	········	$6146	········	$6154	········	$6162	········

$6163	········	$6171	········	$6179	········	$6187	········	$6195	········	$6203	········	$6211	········	$6219	········	$6227	········
$6164	········	$6172	········	$6180	········	$6188	········	$6196	········	$6204	········	$6212	········	$6220	········	$6228	········
$6165	X···X···	$6173	XXXXX···	$6181	XXXX····	$6189	XXXXX···	$6197	XXXXX···	$6205	·XXXX···	$6213	XXXXX···	$6221	X···X···	$6229	X···X···
$6166	XX··X···	$6174	X···X···	$6182	X···X···	$6190	X···X···	$6198	X···X···	$6206	X·······	$6214	··X·····	$6222	X···X···	$6230	X···X···
$6167	X·X·X···	$6175	X···X···	$6183	XXXX····	$6191	X·X·X···	$6199	XXXXX···	$6207	·XXX····	$6215	··X·····	$6223	X···X···	$6231	X··X····
$6168	X··XX···	$6176	X···X···	$6184	X·······	$6192	X··X····	$6200	X·X·····	$6208	····X···	$6216	··X·····	$6224	X···X···	$6232	X·X·····
$6169	X···X···	$6177	XXXXX···	$6185	X·······	$6193	XXX·····	$6201	X··X····	$6209	XXXX····	$6217	··X·····	$6225	·XXX····	$6233	·X······
$6170	········	$6178	········	$6186	········	$6194	········	$6202	········	$6210	········	$6218	········	$6226	········	$6234	········

$6235	········	$6243	········	$6251	········	$6259	········	$6267	··XXX···	$6275	·X······	$6283	XXX·····	$6291	·X······	$6299	X·X·X···
$6236	········	$6244	········	$6252	········	$6260	········	$6268	·X······	$6276	··X·····	$6284	···X····	$6292	X·X·X···	$6300	·X·X····
$6237	X···X···	$6245	X···X···	$6253	X···X···	$6261	XXXXX···	$6269	··X·····	$6277	···X····	$6285	··X·····	$6293	···X····	$6301	X·X·X···
$6238	X···X···	$6246	·XX·····	$6254	·X·X····	$6262	···X····	$6270	XX······	$6278	····X···	$6286	···XX···	$6294	········	$6302	·X·X····
$6239	X·X·X···	$6247	··X·····	$6255	··X·····	$6263	··X·····	$6271	··X·····	$6279	···X····	$6287	··X·····	$6295	········	$6303	X·X·X···
$6240	XX·XX···	$6248	·XX·····	$6256	··X·····	$6264	·X······	$6272	·X······	$6280	··X·····	$6288	···X····	$6296	········	$6304	·X·X····
$6241	X···X···	$6249	X···X···	$6257	··X·····	$6265	XXXXX···	$6273	··XXX···	$6281	·X······	$6289	XXX·····	$6297	········	$6305	X·X·X···
$6242	········	$6250	········	$6258	········	$6266	········	$6274	········	$6282	········	$6290	········	$6298	········	$6306	········

Table B-5. Character set extracted from alternate ColecoVision BIOS.

$5515	·XXXXXX·	$5523	···XXXXX	$5531	·X···X··	$5539	········	$5547	···XX···	$5555	··XX··XX	$5563	·XX··XX·	$5571	···XX···	$5579	·XX···X·
$5516	X······X	$5524	·····X··	$5532	·XX·XX··	$5540	········	$5548	···XX···	$5556	··XX··XX	$5564	·XX··XX·	$5572	··XXXXX·	$5580	·XX··XX·
$5517	X·XXXX·X	$5525	·····X··	$5533	·X·X·X··	$5541	········	$5549	···XX···	$5557	·XX··XX·	$5565	XXXXXXXX	$5573	·XX·····	$5581	····XX··
$5518	X·X····X	$5526	·····X··	$5534	·X·X·X··	$5542	········	$5550	···XX···	$5558	········	$5566	·XX··XX·	$5574	··XXXX··	$5582	···XX···
$5519	X·X····X	$5527	········	$5535	········	$5543	········	$5551	········	$5559	········	$5567	XXXXXXXX	$5575	·····XX·	$5583	··XX····
$5520	X·XXXX·X	$5528	········	$5536	········	$5544	········	$5552	···XX···	$5560	········	$5568	·XX··XX·	$5576	·XXXXX··	$5584	·XX··XX·
$5521	X······X	$5529	········	$5537	········	$5545	········	$5553	···XX···	$5561	········	$5569	·XX··XX·	$5577	···XX···	$5585	·X···XX·
$5522	·XXXXXX·	$5530	········	$5538	········	$5546	········	$5554	········	$5562	········	$5570	········	$5578	········	$5586	········
$5587	··XXXX··	$5595	····XX··	$5603	····XX··	$5611	··XX····	$5619	········	$5627	········	$5635	········	$5643	········	$5651	········
$5588	·XX··XX·	$5596	····XX··	$5604	···XX···	$5612	···XX···	$5620	·XX··XX·	$5628	···XX···	$5636	········	$5644	········	$5652	········
$5589	··XXXX··	$5597	···XX···	$5605	··XX····	$5613	····XX··	$5621	··XXXX··	$5629	···XX···	$5637	········	$5645	········	$5653	········
$5590	··XXX···	$5598	········	$5606	··XX····	$5614	····XX··	$5622	XXXXXXXX	$5630	·XXXXXX·	$5638	········	$5646	·XX·XXX·	$5654	········
$5591	·XX··XXX	$5599	········	$5607	··XX····	$5615	····XX··	$5623	··XXXX··	$5631	···XX···	$5639	········	$5647	··XXX·XX	$5655	········
$5592	·XX··XX·	$5600	········	$5608	···XX···	$5616	···XX···	$5624	·XX··XX·	$5632	···XX···	$5640	···XX···	$5648	········	$5656	···XX···
$5593	··XXXXXX	$5601	········	$5609	····XX··	$5617	··XX····	$5625	········	$5633	········	$5641	···XX···	$5649	········	$5657	···XX···
$5594	········	$5602	········	$5610	········	$5618	········	$5626	········	$5634	········	$5642	··XX····	$5650	········	$5658	········
$5659	········	$5667	··XXXXX·	$5675	····XX··	$5683	··XXXXX·	$5691	··XXXXX·	$5699	·····XX·	$5707	·XXXXXXX	$5715	··XXXXX·	$5723	·XXXXXXX
$5660	······XX	$5668	·XX···XX	$5676	···XXX··	$5684	·XX···XX	$5692	·XX···XX	$5700	····XXX·	$5708	·XX···XX	$5716	·XX···XX	$5724	·XX···XX
$5661	·····XX·	$5669	·XX··XXX	$5677	····XX··	$5685	·XX···XX	$5693	·XX···XX	$5701	···XXXX·	$5709	·XX·····	$5717	·XX·····	$5725	·····XX·
$5662	····XX··	$5670	·XX·X·XX	$5678	····XX··	$5686	····XXX·	$5694	····XXX·	$5702	··X··XX·	$5710	·XXXXXX·	$5718	·XXXXXX·	$5726	····XX··
$5663	···XX···	$5671	·XXX··XX	$5679	····XX··	$5687	··XXX···	$5695	·XX···XX	$5703	·XXXXXXX	$5711	······XX	$5719	·XX···XX	$5727	···XX···
$5664	··XX····	$5672	·XX···XX	$5680	····XX··	$5688	·XX···XX	$5696	·XX···XX	$5704	·····XX·	$5712	·XX···XX	$5720	·XX···XX	$5728	···XX···
$5665	·XX·····	$5673	··XXXXX·	$5681	··XXXXXX	$5689	·XXXXXXX	$5697	··XXXXX·	$5705	·····XX·	$5713	··XXXXX·	$5721	··XXXXX·	$5729	··XXXX··
$5666	········	$5674	········	$5682	········	$5690	········	$5698	········	$5706	········	$5714	········	$5722	········	$5730	········
$5731	··XXXXX·	$5739	··XXXXX·	$5747	········	$5755	········	$5763	····XXX·	$5771	········	$5779	·XXX····	$5787	·XXXXXX·	$5795	··XXXX··
$5732	·XX···XX	$5740	·XX···XX	$5748	···XX···	$5756	···XX···	$5764	···XX···	$5772	········	$5780	···XX···	$5788	·XX···XX	$5796	·XX··XX·
$5733	·XX···XX	$5741	·XX···XX	$5749	···XX···	$5757	···XX···	$5765	··XX····	$5773	·XXXXXX·	$5781	····XX··	$5789	······XX	$5797	·XX·XXX·
$5734	··XXXXX·	$5742	··XXXXXX	$5750	········	$5758	········	$5766	·XX·····	$5774	········	$5782	·····XX·	$5790	·····XX·	$5798	·XX·XXX·
$5735	·XX···XX	$5743	······XX	$5751	···XX···	$5759	···XX···	$5767	··XX····	$5775	·XXXXXX·	$5783	····XX··	$5791	···XXX··	$5799	·XX·····
$5736	·XX···XX	$5744	·XX···XX	$5752	···XX···	$5760	···XX···	$5768	···XX···	$5776	········	$5784	···XX···	$5792	········	$5800	·XX···X·
$5737	··XXXXX·	$5745	··XXXXX·	$5753	········	$5761	··XX····	$5769	····XXX·	$5777	········	$5785	·XXX····	$5793	···XX···	$5801	··XXXX··
$5738	········	$5746	········	$5754	········	$5762	········	$5770	········	$5778	········	$5786	········	$5794	···XX···	$5802	········
$5803	···XXX··	$5811	·XX·XXX·	$5819	···XXXX·	$5827	·XX·XX··	$5835	·XXXXXXX	$5843	·XXXXXXX	$5851	···XXXX·	$5859	·XX···XX	$5867	··XXXX··
$5804	··XX·XX·	$5812	·XXX··XX	$5820	··XX··XX	$5828	·XXX·XX·	$5836	··XX···X	$5844	··XX···X	$5852	··XX··XX	$5860	·XX···XX	$5868	···XX···
$5805	·XX···XX	$5813	·XX···XX	$5821	·XX·····	$5829	·XX···XX	$5837	··XX····	$5845	··XX····	$5853	·XX·····	$5861	·XX···XX	$5869	···XX···
$5806	·XXXXXXX	$5814	·XXXXXX·	$5822	·XX·····	$5830	·XX···XX	$5838	··XXXX··	$5846	··XXXX··	$5854	·XX··XXX	$5862	·XXXXXXX	$5870	···XX···
$5807	·XX···XX	$5815	·XX···XX	$5823	·XX·····	$5831	·XX···XX	$5839	··XX····	$5847	··XX····	$5855	·XX···XX	$5863	·XX···XX	$5871	···XX···
$5808	·XX···XX	$5816	·XX···XX	$5824	··XX··XX	$5832	·XX··XX·	$5840	··XX···X	$5848	··XX····	$5856	··XX·XXX	$5864	·XX···XX	$5872	···XX···
$5809	·XX···XX	$5817	·XXXXXX·	$5825	···XXXX·	$5833	·XXXXX··	$5841	·XXXXXXX	$5849	·XXXX···	$5857	···XXX·X	$5865	·XX···XX	$5873	··XXXX··
$5810	········	$5818	········	$5826	········	$5834	········	$5842	········	$5850	········	$5858	········	$5866	········	$5874	········

Table B-5 continued.

$5875	···XXXXX	$5883	·XX··XX·	$5891	·XXXX···	$5899	·XX···XX	$5907	·XX···XX	$5915	···XXX··	$5923	·XX·XXX·	$5931	···XXX··	$5939	·XX·XXX·
$5876	·····XX·	$5884	·XX··XX·	$5892	··XX····	$5900	·XXX·XXX	$5908	·XXX··XX	$5916	··XX·XX·	$5924	·XXX··XX	$5932	··XX·XX·	$5940	·XXX··XX
$5877	·····XX·	$5885	·XX·XX··	$5893	·XX·····	$5901	·XXXXXXX	$5909	·XXXX·XX	$5917	·XX···XX	$5925	·XX···XX	$5933	·XX···XX	$5941	·XX···XX
$5878	·····XX·	$5886	·XXXX···	$5894	·XX·····	$5902	·XX·X·XX	$5910	·XX·XXXX	$5918	·XX···XX	$5926	·XXXXXX·	$5934	·XX·X·XX	$5942	·XXXXXX·
$5879	·····XX·	$5887	·XX·XX··	$5895	·XX···XX	$5903	·XX···XX	$5911	·XX··XXX	$5919	·XX···XX	$5927	·XX·····	$5935	·XX··XXX	$5943	·XX·XX··
$5880	·XX··XX·	$5888	·XX··XXX	$5896	·XX···XX	$5904	·XX···XX	$5912	·XX···XX	$5920	··XX·XX·	$5928	·XX·····	$5936	··XX·XX·	$5944	·XX··XXX
$5881	··XXXX··	$5889	·XX···XX	$5897	·XXXXXX·	$5905	·XX···XX	$5913	·XX···XX	$5921	···XXX··	$5929	·XX·····	$5937	···XXX·X	$5945	·XX···XX
$5882	········	$5890	········	$5898	········	$5906	········	$5914	········	$5922	········	$5930	········	$5938	········	$5946	········
$5947	··XXXXX·	$5955	·XXXXXX·	$5963	·XXX··XX	$5971	·XXX··XX	$5979	·XXX··XX	$5987	·XX···XX	$5995	··XX··XX	$6003	·XXXXXXX	$6011	··XXXXXX
$5948	·XX···XX	$5956	·X·XX·X·	$5964	··XX··XX	$5972	··XX··XX	$5980	··XX··XX	$5988	·XX···XX	$5996	·XX···XX	$6004	·XX···XX	$6012	··XX····
$5949	·XX·····	$5957	···XX···	$5965	·XX···XX	$5973	·XX···XX	$5981	·XX···XX	$5989	··XX·XX·	$5997	·XX···XX	$6005	·····XX·	$6013	··XX····
$5950	··XXXXX·	$5958	···XX···	$5966	·XX···XX	$5974	·XX···XX	$5982	·XX·X·XX	$5990	···XXX··	$5998	··XX·XX·	$6006	···XXX··	$6014	··XX····
$5951	······XX	$5959	···XX···	$5967	·XX···XX	$5975	·XX··XX·	$5983	·XXXXXXX	$5991	··XX·XX·	$5999	···XXX··	$6007	··XX··XX	$6015	··XX····
$5952	·XX···XX	$5960	···XX···	$5968	·XXX·XX·	$5976	··XXXX··	$5984	·XXX·XXX	$5992	·XX···XX	$6000	·XXXX···	$6008	·XX···XX	$6016	··XX····
$5953	··XXXXX·	$5961	··XXXX··	$5969	··XXXX··	$5977	···XX···	$5985	·XX···XX	$5993	·XX···XX	$6001	·XXX····	$6009	·XXXXXX·	$6017	··XXXXXX
$5954	········	$5962	········	$5970	········	$5978	········	$5986	········	$5994	········	$6002	········	$6010	········	$6018	········
$6019	········	$6027	·XXXXXX·	$6035	···X····	$6043	········	$6051	··XX····	$6059	········	$6067	·XX·····	$6075	········	$6083	······XX
$6020	·XX·····	$6028	·····XX·	$6036	··X·X···	$6044	········	$6052	··XX····	$6060	········	$6068	·XX·····	$6076	········	$6084	······XX
$6021	··XX····	$6029	·····XX·	$6037	·X···X··	$6045	········	$6053	···XX···	$6061	··XXXXXX	$6069	·XX·XXX·	$6077	··XXXXX·	$6085	··XXX·XX
$6022	···XX···	$6030	·····XX·	$6038	········	$6046	········	$6054	········	$6062	·XX···XX	$6070	·XXX··XX	$6078	·XX···XX	$6086	·XX··XXX
$6023	····XX··	$6031	·····XX·	$6039	········	$6047	········	$6055	········	$6063	·XX···XX	$6071	·XX···XX	$6079	·XX·····	$6087	·XX···XX
$6024	·····XX·	$6032	·····XX·	$6040	········	$6048	········	$6056	········	$6064	·XX··XXX	$6072	·XX···XX	$6080	·XX···XX	$6088	·XX···XX
$6025	······XX	$6033	·XXXXXX·	$6041	········	$6049	XXXXXXXX	$6057	········	$6065	··XXX·XX	$6073	··XXXXX·	$6081	··XXXXX·	$6089	··XXXXX·
$6026	········	$6034	········	$6042	········	$6050	XXXXXXXX	$6058	········	$6066	········	$6074	········	$6082	········	$6090	········
$6091	········	$6099	····XXX·	$6107	········	$6115	·XX·····	$6123	········	$6131	········	$6139	·XX·····	$6147	···XXX··	$6155	········
$6092	········	$6100	···XX···	$6108	········	$6116	·XX·····	$6124	········	$6132	········	$6140	·XX·····	$6148	····XX··	$6156	········
$6093	··XXXXX·	$6101	···XX···	$6109	··XXXXX·	$6117	·XX·XXX·	$6125	···XXXX·	$6133	··XXXXXX	$6141	·XX··XX·	$6149	····XX··	$6157	·XX·XXX·
$6094	·XX····X	$6102	··XXXX··	$6110	·XX·····	$6118	·XXX··XX	$6126	····XX··	$6134	·····XX·	$6142	·XX·XXX·	$6150	····XX··	$6158	·XXXXXXX
$6095	·XXXXXXX	$6103	···XX···	$6111	·XX···XX	$6119	·XX···XX	$6127	····XX··	$6135	·····XX·	$6143	·XXXXX··	$6151	····XX··	$6159	·XX·X·XX
$6096	·XX·····	$6104	···XX···	$6112	·XX···XX	$6120	·XX··XX·	$6128	····XX··	$6136	·····XX·	$6144	·XX··XXX	$6152	····XX··	$6160	·XX···X·
$6097	··XXXXX·	$6105	··XXXX··	$6113	··XXXX·X	$6121	·XX··XXX	$6129	···XXXX·	$6137	·XX··XX·	$6145	·XX···XX	$6153	···XXXX·	$6161	·XX··XXX
$6098	········	$6106	········	$6114	········	$6122	········	$6130	········	$6138	··XXXX··	$6146	········	$6154	········	$6162	········
$6163	········	$6171	········	$6179	········	$6187	········	$6195	········	$6203	········	$6211	·····XX·	$6219	········	$6227	········
$6164	········	$6172	········	$6180	········	$6188	········	$6196	········	$6204	········	$6212	····XX··	$6220	········	$6228	········
$6165	·XX·XXX·	$6173	··XXXXX·	$6181	··XXXXX·	$6189	··XXXXX·	$6197	·XX·XXX·	$6205	··XXXXX·	$6213	··XXXXXX	$6221	·XXX··XX	$6229	·XXX··XX
$6166	·XXX··XX	$6174	·XX···XX	$6182	·XX···XX	$6190	·XX···XX	$6198	·XXX··XX	$6206	·XXX···X	$6214	···XX···	$6222	··XX··XX	$6230	··XX··XX
$6167	·XX···XX	$6175	·XX···XX	$6183	·XXX··XX	$6191	·XX··XXX	$6199	·XX···XX	$6207	···XXX··	$6215	···XX···	$6223	·XX···XX	$6231	·XX···XX
$6168	·XX··XX·	$6176	·XX···XX	$6184	·XX·XXX·	$6192	··XXX·XX	$6200	·XXXXXX·	$6208	·X···XXX	$6216	···XX·XX	$6224	·XX··XXX	$6232	·XX··XX·
$6169	·XX··XXX	$6177	··XXXXX·	$6185	·XX·····	$6193	······XX	$6201	·XX···XX	$6209	··XXXXX·	$6217	····XXX·	$6225	··XXX·XX	$6233	··XXXX··
$6170	········	$6178	········	$6186	·XX·····	$6194	······XX	$6202	········	$6210	········	$6218	········	$6226	········	$6234	········
$6235	········	$6243	········	$6251	········	$6259	········	$6267	····XXX·	$6275	··X·····	$6283	·XXX····	$6291	········	$6299	X·X·X·X·
$6236	········	$6244	········	$6252	········	$6260	········	$6268	···XX···	$6276	···X····	$6284	···XX···	$6292	·XX·XXX·	$6300	·X·X·X·X
$6237	·XX···XX	$6245	·XX···XX	$6253	··XX··XX	$6261	·XXXXXXX	$6269	···XX···	$6277	····X···	$6285	···XX···	$6293	··XXX·XX	$6301	X·X·X·X·
$6238	·XX·X·XX	$6246	··XX·XX·	$6254	·XX···XX	$6262	····XXX·	$6270	··XX····	$6278	·····X··	$6286	····XX··	$6294	········	$6302	·X·X·X·X
$6239	·XXXXXXX	$6247	···XXX··	$6255	·XX···XX	$6263	···XXX··	$6271	···XX···	$6279	····X···	$6287	···XX···	$6295	········	$6303	X·X·X·X·
$6240	·XXX·XXX	$6248	··XX·XX·	$6256	··XXXXXX	$6264	··XXX···	$6272	···XX···	$6280	···X····	$6288	···XX···	$6296	········	$6304	·X·X·X·X
$6241	·XX···XX	$6249	·XX···XX	$6257	······XX	$6265	·XXXXXXX	$6273	····XXX·	$6281	··X·····	$6289	·XXX····	$6297	········	$6305	X·X·X·X·
$6242	········	$6250	········	$6258	··XXXXX·	$6266	········	$6274	········	$6282	········	$6290	········	$6298	········	$6306	·X·X·X·X

Table B-6. Character set extracted from Atari 5200 BIOS ROM image.

$0000	········	$0008	········	$0010	········	$0018	········	$0020	···XX···	$0028	········	$0030	···XXX··	$0038	········	$0040	········
$0001	········	$0009	···XX···	$0011	·XX··XX·	$0019	·XX··XX·	$0021	··XXXXX·	$0029	·XX··XX·	$0031	··XX·XX·	$0039	···XX···	$0041	····XXX·
$0002	········	$000a	···XX···	$0012	·XX··XX·	$001a	XXXXXXXX	$0022	·XX·····	$002a	·XX·XX··	$0032	···XXX··	$003a	···XX···	$0042	···XXX··
$0003	········	$000b	···XX···	$0013	·XX··XX·	$001b	·XX··XX·	$0023	··XXXX··	$002b	···XX···	$0033	··XXX···	$003b	···XX···	$0043	···XX···
$0004	········	$000c	···XX···	$0014	········	$001c	·XX··XX·	$0024	·····XX·	$002c	··XX····	$0034	·XX·XXXX	$003c	········	$0044	···XX···
$0005	········	$000d	········	$0015	········	$001d	XXXXXXXX	$0025	·XXXXX··	$002d	·XX··XX·	$0035	·XX··XX·	$003d	········	$0045	···XXX··
$0006	········	$000e	···XX···	$0016	········	$001e	·XX··XX·	$0026	···XX···	$002e	·X···XX·	$0036	··XXX·XX	$003e	········	$0046	····XXX·
$0007	········	$000f	········	$0017	········	$001f	········	$0027	········	$002f	········	$0037	········	$003f	········	$0047	········

$0048	········	$0050	········	$0058	········	$0060	········	$0068	········	$0070	········	$0078	········	$0080	········	$0088	········
$0049	·XXX····	$0051	·XX··XX·	$0059	···XX···	$0061	········	$0069	········	$0071	········	$0079	·····XX·	$0081	··XXXX··	$0089	···XX···
$004a	··XXX···	$0052	··XXXX··	$005a	···XX···	$0062	········	$006a	········	$0072	········	$007a	····XX··	$0082	·XX··XX·	$008a	··XXX···
$004b	···XX···	$0053	XXXXXXXX	$005b	·XXXXXX·	$0063	········	$006b	·XXXXXX·	$0073	········	$007b	···XX···	$0083	·XX·XXX·	$008b	···XX···
$004c	···XX···	$0054	··XXXX··	$005c	···XX···	$0064	········	$006c	········	$0074	········	$007c	··XX····	$0084	·XXX·XX·	$008c	···XX···
$004d	··XXX···	$0055	·XX··XX·	$005d	···XX···	$0065	···XX···	$006d	········	$0075	···XX···	$007d	·XX·····	$0085	·XX··XX·	$008d	···XX···
$004e	·XXX····	$0056	········	$005e	········	$0066	···XX···	$006e	········	$0076	···XX···	$007e	·X······	$0086	··XXXX··	$008e	·XXXXXX·
$004f	········	$0057	········	$005f	········	$0067	··XX····	$006f	········	$0077	········	$007f	········	$0087	········	$008f	········

$0090	········	$0098	········	$00a0	········	$00a8	········	$00b0	········	$00b8	········	$00c0	········	$00c8	········	$00d0	········
$0091	··XXXX··	$0099	·XXXXXX·	$00a1	····XX··	$00a9	·XXXXXX·	$00b1	··XXXX··	$00b9	·XXXXXX·	$00c1	··XXXX··	$00c9	··XXXX··	$00d1	········
$0092	·XX··XX·	$009a	····XX··	$00a2	···XXX··	$00aa	·XX·····	$00b2	·XX·····	$00ba	·····XX·	$00c2	·XX··XX·	$00ca	·XX··XX·	$00d2	···XX···
$0093	····XX··	$009b	···XX···	$00a3	··XXXX··	$00ab	·XXXXX··	$00b3	·XXXXX··	$00bb	····XX··	$00c3	··XXXX··	$00cb	··XXXXX·	$00d3	···XX···
$0094	···XX···	$009c	····XX··	$00a4	·XX·XX··	$00ac	·····XX·	$00b4	·XX··XX·	$00bc	···XX···	$00c4	·XX··XX·	$00cc	·····XX·	$00d4	········
$0095	··XX····	$009d	·XX··XX·	$00a5	·XXXXXX·	$00ad	·XX··XX·	$00b5	·XX··XX·	$00bd	··XX····	$00c5	·XX··XX·	$00cd	····XX··	$00d5	···XX···
$0096	·XXXXXX·	$009e	··XXXX··	$00a6	····XX··	$00ae	··XXXX··	$00b6	··XXXX··	$00be	··XX····	$00c6	··XXXX··	$00ce	··XXX···	$00d6	···XX···
$0097	········	$009f	········	$00a7	········	$00af	········	$00b7	········	$00bf	········	$00c7	········	$00cf	········	$00d7	········

$00d8	········	$00e0	·····XX·	$00e8	········	$00f0	·XX·····	$00f8	········	$0100	········	$0108	········	$0110	········	$0118	········
$00d9	········	$00e1	····XX··	$00e9	········	$00f1	··XX····	$00f9	··XXXX··	$0101	··XXXX··	$0109	···XX···	$0111	·XXXXX··	$0119	··XXXX··
$00da	···XX···	$00e2	···XX···	$00ea	·XXXXXX·	$00f2	···XX···	$00fa	·XX··XX·	$0102	·XX··XX·	$010a	··XXXX··	$0112	·XX··XX·	$011a	·XX··XX·
$00db	···XX···	$00e3	··XX····	$00eb	········	$00f3	····XX··	$00fb	····XX··	$0103	·XX·XXX·	$010b	·XX··XX·	$0113	·XXXXX··	$011b	·XX·····
$00dc	········	$00e4	···XX···	$00ec	········	$00f4	···XX···	$00fc	···XX···	$0104	·XX·XXX·	$010c	·XX··XX·	$0114	·XX··XX·	$011c	·XX·····
$00dd	···XX···	$00e5	····XX··	$00ed	·XXXXXX·	$00f5	··XX····	$00fd	········	$0105	·XX·····	$010d	·XXXXXX·	$0115	·XX··XX·	$011d	·XX··XX·
$00de	···XX···	$00e6	·····XX·	$00ee	········	$00f6	·XX·····	$00fe	···XX···	$0106	··XXXXX·	$010e	·XX··XX·	$0116	·XXXXX··	$011e	··XXXX··
$00df	··XX····	$00e7	········	$00ef	········	$00f7	········	$00ff	········	$0107	········	$010f	········	$0117	········	$011f	········

$0120	········	$0128	········	$0130	········	$0138	········	$0140	········	$0148	········	$0150	········	$0158	········	$0160	········
$0121	·XXXX···	$0129	·XXXXXX·	$0131	·XXXXXX·	$0139	··XXXXX·	$0141	·XX··XX·	$0149	·XXXXXX·	$0151	·····XX·	$0159	·XX··XX·	$0161	·XX·····
$0122	·XX·XX··	$012a	·XX·····	$0132	·XX·····	$013a	·XX·····	$0142	·XX··XX·	$014a	···XX···	$0152	·····XX·	$015a	·XX·XX··	$0162	·XX·····
$0123	·XX··XX·	$012b	·XXXXX··	$0133	·XXXXX··	$013b	·XX·····	$0143	·XXXXXX·	$014b	···XX···	$0153	·····XX·	$015b	·XXXX···	$0163	·XX·····
$0124	·XX··XX·	$012c	·XX·····	$0134	·XX·····	$013c	·XX·XXX·	$0144	·XX··XX·	$014c	···XX···	$0154	·····XX·	$015c	·XXXX···	$0164	·XX·····
$0125	·XX·XX··	$012d	·XX·····	$0135	·XX·····	$013d	·XX··XX·	$0145	·XX··XX·	$014d	···XX···	$0155	·XX··XX·	$015d	·XX·XX··	$0165	·XX·····
$0126	·XXXX···	$012e	·XXXXXX·	$0136	·XX·····	$013e	··XXXXX·	$0146	·XX··XX·	$014e	·XXXXXX·	$0156	··XXXX··	$015e	·XX··XX·	$0166	·XXXXXX·
$0127	········	$012f	········	$0137	········	$013f	········	$0147	········	$014f	········	$0157	········	$015f	········	$0167	········

Table B-6 continued.

$0168	········	$0170	········	$0178	········	$0180	········	$0188	········	$0190	········	$0198	········	$01a0	········	$01a8	········
$0169	·XX···XX	$0171	·XX··XX·	$0179	··XXXX··	$0181	·XXXXX··	$0189	··XXXX··	$0191	·XXXXX··	$0199	··XXXX··	$01a1	·XXXXXX·	$01a9	·XX··XX·
$016a	·XXX·XXX	$0172	·XXX·XX·	$017a	·XX··XX·	$0182	·XX··XX·	$018a	·XX··XX·	$0192	·XX··XX·	$019a	·XX·····	$01a2	···XX···	$01aa	·XX··XX·
$016b	·XXXXXXX	$0173	·XXXXXX·	$017b	·XX··XX·	$0183	·XX··XX·	$018b	·XX··XX·	$0193	·XX··XX·	$019b	··XXXX··	$01a3	···XX···	$01ab	·XX··XX·
$016c	·XX·X·XX	$0174	·XXXXXX·	$017c	·XX··XX·	$0184	·XXXXX··	$018c	·XX··XX·	$0194	·XXXXX··	$019c	·····XX·	$01a4	···XX···	$01ac	·XX··XX·
$016d	·XX···XX	$0175	·XX·XXX·	$017d	·XX··XX·	$0185	·XX·····	$018d	·XX·XX··	$0195	·XX·XX··	$019d	·····XX·	$01a5	···XX···	$01ad	·XX··XX·
$016e	·XX···XX	$0176	·XX··XX·	$017e	··XXXX··	$0186	·XX·····	$018e	··XX·XX·	$0196	·XX··XX·	$019e	··XXXX··	$01a6	···XX···	$01ae	·XXXXXX·
$016f	········	$0177	········	$017f	········	$0187	········	$018f	········	$0197	········	$019f	········	$01a7	········	$01af	········

$01b0	········	$01b8	········	$01c0	········	$01c8	········	$01d0	········	$01d8	········	$01e0	········	$01e8	········	$01f0	········
$01b1	·XX··XX·	$01b9	·XX···XX	$01c1	·XX··XX·	$01c9	·XX··XX·	$01d1	·XXXXXX·	$01d9	···XXXX·	$01e1	·X······	$01e9	·XXXX···	$01f1	····X···
$01b2	·XX··XX·	$01ba	·XX···XX	$01c2	·XX··XX·	$01ca	·XX··XX·	$01d2	····XX··	$01da	···XX···	$01e2	·XX·····	$01ea	···XX···	$01f2	···XXX··
$01b3	·XX··XX·	$01bb	·XX·X·XX	$01c3	··XXXX··	$01cb	··XXXX··	$01d3	···XX···	$01db	···XX···	$01e3	··XX····	$01eb	···XX···	$01f3	··XX·XX·
$01b4	·XX··XX·	$01bc	·XXXXXXX	$01c4	··XXXX··	$01cc	···XX···	$01d4	··XX····	$01dc	···XX···	$01e4	···XX···	$01ec	···XX···	$01f4	·XX···XX
$01b5	··XXXX··	$01bd	·XXX·XXX	$01c5	·XX··XX·	$01cd	···XX···	$01d5	·XX·····	$01dd	···XX···	$01e5	····XX··	$01ed	···XX···	$01f5	········
$01b6	···XX···	$01be	·XX···XX	$01c6	·XX··XX·	$01ce	···XX···	$01d6	·XXXXXX·	$01de	···XXXX·	$01e6	·····XX·	$01ee	·XXXX···	$01f6	········
$01b7	········	$01bf	········	$01c7	········	$01cf	········	$01d7	········	$01df	········	$01e7	········	$01ef	········	$01f7	········

$01f8	········	$0200	········	$0208	···XX···	$0210	······XX	$0218	···XX···	$0220	···XX···	$0228	········	$0230	······XX	$0238	XX······
$01f9	········	$0201	··XX·XX·	$0209	···XX···	$0211	······XX	$0219	···XX···	$0221	···XX···	$0229	········	$0231	·····XXX	$0239	XXX·····
$01fa	········	$0202	·XXXXXXX	$020a	···XX···	$0212	······XX	$021a	···XX···	$0222	···XX···	$022a	········	$0232	····XXX·	$023a	·XXX····
$01fb	········	$0203	·XXXXXXX	$020b	···XXXXX	$0213	······XX	$021b	XXXXX···	$0223	XXXXX···	$022b	XXXXX···	$0233	···XXX··	$023b	··XXX···
$01fc	········	$0204	··XXXXX·	$020c	···XXXXX	$0214	······XX	$021c	XXXXX···	$0224	XXXXX···	$022c	XXXXX···	$0234	··XXX···	$023c	···XXX··
$01fd	········	$0205	···XXX··	$020d	···XX···	$0215	······XX	$021d	········	$0225	···XX···	$022d	···XX···	$0235	·XXX····	$023d	····XXX·
$01fe	XXXXXXXX	$0206	····X···	$020e	···XX···	$0216	······XX	$021e	········	$0226	···XX···	$022e	···XX···	$0236	XXX·····	$023e	·····XXX
$01ff	········	$0207	········	$020f	···XX···	$0217	······XX	$021f	········	$0227	···XX···	$022f	···XX···	$0237	XX······	$023f	······XX

$0240	·······X	$0248	········	$0250	X·······	$0258	····XXXX	$0260	XXXX····	$0268	XXXXXXXX	$0270	········	$0278	········	$0280	········
$0241	······XX	$0249	········	$0251	XX······	$0259	····XXXX	$0261	XXXX····	$0269	XXXXXXXX	$0271	········	$0279	········	$0281	···XXX··
$0242	·····XXX	$024a	········	$0252	XXX·····	$025a	····XXXX	$0262	XXXX····	$026a	········	$0272	········	$027a	········	$0282	···XXX··
$0243	····XXXX	$024b	········	$0253	XXXX····	$025b	····XXXX	$0263	XXXX····	$026b	········	$0273	········	$027b	········	$0283	·XXX·XXX
$0244	···XXXXX	$024c	····XXXX	$0254	XXXXX···	$025c	········	$0264	········	$026c	········	$0274	········	$027c	XXXX····	$0284	·XXX·XXX
$0245	··XXXXXX	$024d	····XXXX	$0255	XXXXXX··	$025d	········	$0265	········	$026d	········	$0275	········	$027d	XXXX····	$0285	····X···
$0246	·XXXXXXX	$024e	····XXXX	$0256	XXXXXXX·	$025e	········	$0266	········	$026e	········	$0276	XXXXXXXX	$027e	XXXX····	$0286	···XXX··
$0247	XXXXXXXX	$024f	····XXXX	$0257	XXXXXXXX	$025f	········	$0267	········	$026f	········	$0277	XXXXXXXX	$027f	XXXX····	$0287	········

$0288	········	$0290	········	$0298	···XX···	$02a0	········	$02a8	········	$02b0	XX······	$02b8	········	$02c0	···XX···	$02c8	XXXX····
$0289	········	$0291	········	$0299	···XX···	$02a1	········	$02a9	········	$02b1	XX······	$02b9	········	$02c1	···XX···	$02c9	XXXX····
$028a	········	$0292	········	$029a	···XX···	$02a2	··XXXX··	$02aa	········	$02b2	XX······	$02ba	········	$02c2	···XX···	$02ca	XXXX····
$028b	···XXXXX	$0293	XXXXXXXX	$029b	XXXXXXXX	$02a3	·XXXXXX·	$02ab	········	$02b3	XX······	$02bb	XXXXXXXX	$02c3	XXXXXXXX	$02cb	XXXX····
$028c	···XXXXX	$0294	XXXXXXXX	$029c	XXXXXXXX	$02a4	·XXXXXX·	$02ac	XXXXXXXX	$02b4	XX······	$02bc	XXXXXXXX	$02c4	XXXXXXXX	$02cc	XXXX····
$028d	···XX···	$0295	········	$029d	···XX···	$02a5	·XXXXXX·	$02ad	XXXXXXXX	$02b5	XX······	$02bd	···XX···	$02c5	········	$02cd	XXXX····
$028e	···XX···	$0296	········	$029e	···XX···	$02a6	··XXXX··	$02ae	XXXXXXXX	$02b6	XX······	$02be	···XX···	$02c6	········	$02ce	XXXX····
$028f	···XX···	$0297	········	$029f	···XX···	$02a7	········	$02af	XXXXXXXX	$02b7	XX······	$02bf	···XX···	$02c7	········	$02cf	XXXX····

Table B-6 continued.

$02d0	···XX···	$02d8	·XXXX···	$02e0	········	$02e8	········	$02f0	········	$02f8	········	$0300	········	$0308	········	$0310	········
$02d1	···XX···	$02d9	·XX·····	$02e1	···XX···	$02e9	···XX···	$02f1	···XX···	$02f9	···XX···	$0301	···XX···	$0309	········	$0311	·XX·····
$02d2	···XX···	$02da	·XXXX···	$02e2	··XXXX··	$02ea	···XX···	$02f2	··XX····	$02fa	····XX··	$0302	··XXXX··	$030a	··XXXX··	$0312	·XX·····
$02d3	···XXXXX	$02db	·XX·····	$02e3	·XXXXXX·	$02eb	···XX···	$02f3	·XXXXXX·	$02fb	·XXXXXX·	$0303	·XXXXXX·	$030b	·····XX·	$0313	·XXXXX··
$02d4	···XXXXX	$02dc	·XXXXXX·	$02e4	···XX···	$02ec	·XXXXXX·	$02f4	··XX····	$02fc	····XX··	$0304	·XXXXXX·	$030c	··XXXXX·	$0314	·XX··XX·
$02d5	········	$02dd	···XX···	$02e5	···XX···	$02ed	··XXXX··	$02f5	···XX···	$02fd	···XX···	$0305	··XXXX··	$030d	·XX··XX·	$0315	·XX··XX·
$02d6	········	$02de	···XXXX·	$02e6	···XX···	$02ee	···XX···	$02f6	········	$02fe	········	$0306	···XX···	$030e	··XXXXX·	$0316	·XXXXX··
$02d7	········	$02df	········	$02e7	········	$02ef	········	$02f7	········	$02ff	········	$0307	········	$030f	········	$0317	········

$0318	········	$0320	········	$0328	········	$0330	········	$0338	········	$0340	········	$0348	········	$0350	········	$0358	········
$0319	········	$0321	·····XX·	$0329	········	$0331	····XXX·	$0339	········	$0341	·XX·····	$0349	···XX···	$0351	·····XX·	$0359	·XX·····
$031a	··XXXX··	$0322	·····XX·	$032a	··XXXX··	$0332	···XX···	$033a	··XXXXX·	$0342	·XX·····	$034a	········	$0352	········	$035a	·XX·····
$031b	·XX·····	$0323	··XXXXX·	$032b	·XX··XX·	$0333	··XXXXX·	$033b	·XX··XX·	$0343	·XXXXX··	$034b	··XXX···	$0353	·····XX·	$035b	·XX·XX··
$031c	·XX·····	$0324	·XX··XX·	$032c	·XXXXXX·	$0334	···XX···	$033c	·XX··XX·	$0344	·XX··XX·	$034c	···XX···	$0354	·····XX·	$035c	·XXXX···
$031d	·XX·····	$0325	·XX··XX·	$032d	·XX·····	$0335	···XX···	$033d	··XXXXX·	$0345	·XX··XX·	$034d	···XX···	$0355	·····XX·	$035d	·XX·XX··
$031e	··XXXX··	$0326	··XXXXX·	$032e	··XXXX··	$0336	···XX···	$033e	·····XX·	$0346	·XX··XX·	$034e	··XXXX··	$0356	·····XX·	$035e	·XX··XX·
$031f	········	$0327	········	$032f	········	$0337	········	$033f	·XXXXX··	$0347	········	$034f	········	$0357	··XXXX··	$035f	········

$0360	········	$0368	········	$0370	········	$0378	········	$0380	········	$0388	········	$0390	········	$0398	········	$03a0	········
$0361	··XXX···	$0369	········	$0371	········	$0379	········	$0381	········	$0389	········	$0391	········	$0399	········	$03a1	···XX···
$0362	···XX···	$036a	·XX··XX·	$0372	·XXXXX··	$037a	··XXXX··	$0382	·XXXXX··	$038a	··XXXXX·	$0392	·XXXXX··	$039a	··XXXXX·	$03a2	·XXXXXX·
$0363	···XX···	$036b	·XXXXXXX	$0373	·XX··XX·	$037b	·XX··XX·	$0383	·XX··XX·	$038b	·XX··XX·	$0393	·XX··XX·	$039b	·XX·····	$03a3	···XX···
$0364	···XX···	$036c	·XXXXXXX	$0374	·XX··XX·	$037c	·XX··XX·	$0384	·XX··XX·	$038c	·XX··XX·	$0394	·XX·····	$039c	··XXXX··	$03a4	···XX···
$0365	···XX···	$036d	·XX·X·XX	$0375	·XX··XX·	$037d	·XX··XX·	$0385	·XXXXX··	$038d	··XXXXX·	$0395	·XX·····	$039d	·····XX·	$03a5	···XX···
$0366	··XXXX··	$036e	·XX···XX	$0376	·XX··XX·	$037e	··XXXX··	$0386	·XX·····	$038e	·····XX·	$0396	·XX·····	$039e	·XXXXX··	$03a6	····XXX·
$0367	········	$036f	········	$0377	········	$037f	········	$0387	·XX·····	$038f	·····XX·	$0397	········	$039f	········	$03a7	········

$03a8	········	$03b0	········	$03b8	········	$03c0	········	$03c8	········	$03d0	········	$03d8	········	$03e0	···XX···	$03e8	········
$03a9	········	$03b1	········	$03b9	········	$03c1	········	$03c9	········	$03d1	········	$03d9	···XX···	$03e1	···XX···	$03e9	·XXXXXX·
$03aa	·XX··XX·	$03b2	·XX··XX·	$03ba	·XX···XX	$03c2	·XX··XX·	$03ca	·XX··XX·	$03d2	·XXXXXX·	$03da	··XXXX··	$03e2	···XX···	$03ea	·XXXX···
$03ab	·XX··XX·	$03b3	·XX··XX·	$03bb	·XX·X·XX	$03c3	··XXXX··	$03cb	·XX··XX·	$03d3	····XX··	$03db	·XXXXXX·	$03e3	···XX···	$03eb	·XXXXX··
$03ac	·XX··XX·	$03b4	·XX··XX·	$03bc	·XXXXXXX	$03c4	···XX···	$03cc	·XX··XX·	$03d4	···XX···	$03dc	·XXXXXX·	$03e4	···XX···	$03ec	·XX·XXX·
$03ad	·XX··XX·	$03b5	··XXXX··	$03bd	··XXXXX·	$03c5	··XXXX··	$03cd	··XXXXX·	$03d5	··XX····	$03dd	···XX···	$03e5	···XX···	$03ed	·XX··XX·
$03ae	··XXXXX·	$03b6	···XX···	$03be	··XX·XX·	$03c6	·XX··XX·	$03ce	····XX··	$03d6	·XXXXXX·	$03de	··XXXX··	$03e6	···XX···	$03ee	·····XX·
$03af	········	$03b7	········	$03bf	········	$03c7	········	$03cf	·XXXX···	$03d7	········	$03df	········	$03e7	···XX···	$03ef	········

$03f0	····X···	$03f8	···X····
$03f1	···XX···	$03f9	···XX···
$03f2	··XXX···	$03fa	···XXX··
$03f3	·XXXX···	$03fb	···XXXX·
$03f4	··XXX···	$03fc	···XXX··
$03f5	···XX···	$03fd	···XX···
$03f6	····X···	$03fe	···X····
$03f7	········		

Nintendo Famicom / NES

Table B-7. Character set extracted from Nintendo Entertainment System (NES) BIOS ROM.

$0001	··XXX···	$0009	···XX···	$0011	·XXXXX··	$0019	·XXXXXX·	$0021	···XXX··	$0029	XXXXXX··	$0031	··XXXX··	$0039	XXXXXXX·	$0041	·XXXXX··
$0002	·X··XX··	$000a	··XXX···	$0012	XX···XX·	$001a	····XX··	$0022	··XXXX··	$002a	XX······	$0032	·XX·····	$003a	XX···XX·	$0042	XX···XX·
$0003	XX···XX·	$000b	···XX···	$0013	····XXX·	$001b	···XX···	$0023	·XX·XX··	$002b	XXXXXX··	$0033	XX······	$003b	····XX··	$0043	XX···XX·
$0004	XX···XX·	$000c	···XX···	$0014	··XXXX··	$001c	··XXXX··	$0024	XX··XX··	$002c	·····XX·	$0034	XXXXXX··	$003c	···XX···	$0044	·XXXXX··
$0005	XX···XX·	$000d	···XX···	$0015	·XXXX···	$001d	·····XX·	$0025	XXXXXXX·	$002d	·····XX·	$0035	XX···XX·	$003d	··XX····	$0045	XX···XX·
$0006	·XX··X··	$000e	···XX···	$0016	XXX·····	$001e	XX···XX·	$0026	····XX··	$002e	XX···XX·	$0036	XX···XX·	$003e	··XX····	$0046	XX···XX·
$0007	··XXX···	$000f	·XXXXXX·	$0017	XXXXXXX·	$001f	·XXXXX··	$0027	····XX··	$002f	·XXXXX··	$0037	·XXXXX··	$003f	··XX····	$0047	·XXXXX··
$0008	········	$0010	········	$0018	········	$0020	········	$0028	········	$0030	········	$0038	········	$0040	········	$0048	········
$0049	·XXXXX··	$0051	··XXX···	$0059	XXXXXX··	$0061	··XXXX··	$0069	XXXXX···	$0071	XXXXXXX·	$0079	XXXXXXX·	$0081	··XXXXX·	$0089	XX···XX·
$004a	XX···XX·	$0052	·XX·XX··	$005a	XX···XX·	$0062	·XX··XX·	$006a	XX··XX··	$0072	XX······	$007a	XX······	$0082	·XX·····	$008a	XX···XX·
$004b	XX···XX·	$0053	XX···XX·	$005b	XX···XX·	$0063	XX······	$006b	XX···XX·	$0073	XX······	$007b	XX······	$0083	XX······	$008b	XX···XX·
$004c	·XXXXXX·	$0054	XX···XX·	$005c	XXXXXX··	$0064	XX······	$006c	XX···XX·	$0074	XXXXXX··	$007c	XXXXXX··	$0084	XX·XXXX·	$008c	XXXXXXX·
$004d	·····XX·	$0055	XXXXXXX·	$005d	XX···XX·	$0065	XX······	$006d	XX···XX·	$0075	XX······	$007d	XX······	$0085	XX···XX·	$008d	XX···XX·
$004e	····XX··	$0056	XX···XX·	$005e	XX···XX·	$0066	·XX··XX·	$006e	XX··XX··	$0076	XX······	$007e	XX······	$0086	·XX··XX·	$008e	XX···XX·
$004f	·XXXX···	$0057	XX···XX·	$005f	XXXXXX··	$0067	··XXXX··	$006f	XXXXX···	$0077	XXXXXXX·	$007f	XX······	$0087	·XXXXXX·	$008f	XX···XX·
$0050	········	$0058	········	$0060	········	$0068	········	$0070	········	$0078	········	$0080	········	$0088	········	$0090	········
$0091	·XXXXXX·	$0099	···XXXX·	$00a1	XX···XX·	$00a9	·XX·····	$00b1	XX···XX·	$00b9	XX···XX·	$00c1	·XXXXX··	$00c9	XXXXXX··	$00d1	·XXXXX··
$0092	···XX···	$009a	·····XX·	$00a2	XX··XX··	$00aa	·XX·····	$00b2	XXX·XXX·	$00ba	XXX··XX·	$00c2	XX···XX·	$00ca	XX···XX·	$00d2	XX···XX·
$0093	···XX···	$009b	·····XX·	$00a3	XX·XX···	$00ab	·XX·····	$00b3	XXXXXXX·	$00bb	XXXX·XX·	$00c3	XX···XX·	$00cb	XX···XX·	$00d3	XX···XX·
$0094	···XX···	$009c	·····XX·	$00a4	XXXX····	$00ac	·XX·····	$00b4	XXXXXXX·	$00bc	XXXXXXX·	$00c4	XX···XX·	$00cc	XX···XX·	$00d4	XX···XX·
$0095	···XX···	$009d	XX···XX·	$00a5	XXXXX···	$00ad	·XX·····	$00b5	XX·X·XX·	$00bd	XX·XXXX·	$00c5	XX···XX·	$00cd	XXXXXX··	$00d5	XX·XXXX·
$0096	···XX···	$009e	XX···XX·	$00a6	XX·XXX··	$00ae	·XX·····	$00b6	XX···XX·	$00be	XX··XXX·	$00c6	XX···XX·	$00ce	XX······	$00d6	XX··XX··
$0097	·XXXXXX·	$009f	·XXXXX··	$00a7	XX··XXX·	$00af	·XXXXXX·	$00b7	XX···XX·	$00bf	XX···XX·	$00c7	·XXXXX··	$00cf	XX······	$00d7	·XXXX·X·
$0098	········	$00a0	········	$00a8	········	$00b0	········	$00b8	········	$00c0	········	$00c8	········	$00d0	········	$00d8	········
$00d9	XXXXXX··	$00e1	·XXXX···	$00e9	·XXXXXX·	$00f1	XX···XX·	$00f9	XX···XX·	$0101	XX···XX·	$0109	XX···XX·	$0111	·XX··XX·	$0119	XXXXXXX·
$00da	XX···XX·	$00e2	XX··XX··	$00ea	···XX···	$00f2	XX···XX·	$00fa	XX···XX·	$0102	XX···XX·	$010a	XXX·XXX·	$0112	·XX··XX·	$011a	····XXX·
$00db	XX···XX·	$00e3	XX······	$00eb	···XX···	$00f3	XX···XX·	$00fb	XX···XX·	$0103	XX·X·XX·	$010b	·XXXXX··	$0113	·XX··XX·	$011b	···XXX··
$00dc	XX··XXX·	$00e4	·XXXXX··	$00ec	···XX···	$00f4	XX···XX·	$00fc	XXX·XXX·	$0104	XXXXXXX·	$010c	··XXX···	$0114	··XXXX··	$011c	··XXX···
$00dd	XXXXX···	$00e5	·····XX·	$00ed	···XX···	$00f5	XX···XX·	$00fd	·XXXXX··	$0105	XXXXXXX·	$010d	·XXXXX··	$0115	···XX···	$011d	·XXX····
$00de	XX·XXX··	$00e6	XX···XX·	$00ee	···XX···	$00f6	XX···XX·	$00fe	··XXX···	$0106	XXX·XXX·	$010e	XXX·XXX·	$0116	···XX···	$011e	XXX·····
$00df	XX··XXX·	$00e7	·XXXXX··	$00ef	···XX···	$00f7	·XXXXX··	$00ff	···X····	$0107	XX···XX·	$010f	XX···XX·	$0117	···XX···	$011f	XXXXXXX·
$00e0	········	$00e8	········	$00f0	········	$00f8	········	$0100	········	$0108	········	$0110	········	$0118	········	$0120	········
$0121	········	$0129	········	$0131	········	$0139	········	$0141	··XXX···								
$0122	········	$012a	········	$0132	········	$013a	········	$0142	·X···X··								
$0123	········	$012b	········	$0133	········	$013b	········	$0143	X·XXX·X·								
$0124	········	$012c	········	$0134	········	$013c	········	$0144	X·X·X·X·								
$0125	········	$012d	··XX····	$0135	··XX····	$013d	·XX·XX··	$0145	X·XX··X·								
$0126	········	$012e	··XX····	$0136	··XX····	$013e	·XX·XX··	$0146	X·X·X·X·								
$0127	········	$012f	··X·····	$0137	········	$013f	····X···	$0147	·X···X··								
$0128	········	$0130	········	$0138	········	$0140	········	$0148	··XXX···								

Table B-8. Character set extracted from Sega Mega Drive (a.k.a. Sega Genesis) BIOS ROM.

$b000	········	$b008	···XX···	$b010	··XX·XX·	$b018	···X··X·	$b020	····X···	$b028	·XXX···X	$b030	···XX···	$b038	··XX····	$b040	····XX··	$b048	··XX····
$b001	········	$b009	···XX···	$b011	··XX·XX·	$b019	···X··X·	$b021	··XXXXXX	$b029	·X·X··X·	$b031	··X··X··	$b039	··XX····	$b041	···X····	$b049	····X···
$b002	········	$b00a	···XX···	$b012	·X··X···	$b01a	·XXXXXXX	$b022	·X··X···	$b02a	·XXX·X··	$b032	···XX···	$b03a	·X······	$b042	··X·····	$b04a	·····X··
$b003	········	$b00b	···XX···	$b013	········	$b01b	···X··X·	$b023	··XXXXX·	$b02b	····X···	$b033	··X·X··X	$b03b	········	$b043	··X·····	$b04b	·····X··
$b004	········	$b00c	········	$b014	········	$b01c	·XXXXXXX	$b024	····X··X	$b02c	···X·XXX	$b034	·X···X·X	$b03c	········	$b044	··X·····	$b04c	·····X··
$b005	········	$b00d	···XX···	$b015	········	$b01d	··X··X··	$b025	·XXXXXX·	$b02d	··X··X·X	$b035	·X···XX·	$b03d	········	$b045	···X····	$b04d	····X···
$b006	········	$b00e	···XX···	$b016	········	$b01e	··X··X··	$b026	····X···	$b02e	·X···XXX	$b036	··XXX··X	$b03e	········	$b046	····XX··	$b04e	··XX····
$b007	········	$b00f	········	$b017	········	$b01f	········	$b027	········	$b02f	········	$b037	········	$b03f	········	$b047	········	$b04f	········
$b050	········	$b058	····X···	$b060	········	$b068	········	$b070	········	$b078	·······X	$b080	···XXXX·	$b088	···XX···	$b090	··XXXXX·	$b098	··XXXXX·
$b051	····X···	$b059	····X···	$b061	········	$b069	········	$b071	········	$b079	······X·	$b081	··XX··XX	$b089	··XXX···	$b091	·XX···XX	$b099	·XX···XX
$b052	··X·X·X·	$b05a	····X···	$b062	········	$b06a	········	$b072	········	$b07a	·····X··	$b082	··XX··XX	$b08a	···XX···	$b092	·XX···XX	$b09a	······XX
$b053	···XXX··	$b05b	·XXXXXXX	$b063	········	$b06b	·XXXXXXX	$b073	········	$b07b	····X···	$b083	··XX··XX	$b08b	···XX···	$b093	····XXX·	$b09b	···XXXX·
$b054	··X·X·X·	$b05c	····X···	$b064	········	$b06c	········	$b074	········	$b07c	···X····	$b084	··XX··XX	$b08c	···XX···	$b094	··XXX···	$b09c	······XX
$b055	····X···	$b05d	····X···	$b065	··XX····	$b06d	········	$b075	··XX····	$b07d	··X·····	$b085	··XX··XX	$b08d	···XX···	$b095	·XX·····	$b09d	·XX···XX
$b056	········	$b05e	····X···	$b066	··XX····	$b06e	········	$b076	··XX····	$b07e	·X······	$b086	···XXXX·	$b08e	··XXXX··	$b096	·XXXXXXX	$b09e	··XXXXX·
$b057	········	$b05f	········	$b067	·X······	$b06f	········	$b077	········	$b07f	········	$b087	········	$b08f	········	$b097	········	$b09f	········
$b0a0	·····XX·	$b0a8	·XXXXXX·	$b0b0	··XXXXX·	$b0b8	··XXXXXX	$b0c0	··XXXXX·	$b0c8	··XXXXX·	$b0d0	········	$b0d8	········	$b0e0	······XX	$b0e8	········
$b0a1	····XXX·	$b0a9	·XX·····	$b0b1	·XX···XX	$b0b9	·XX···XX	$b0c1	·XX···XX	$b0c9	·XX···XX	$b0d1	···XX···	$b0d9	···XX···	$b0e1	····XX··	$b0e9	········
$b0a2	···XXXX·	$b0aa	·XXXXXX·	$b0b2	·XX·····	$b0ba	·····XX·	$b0c2	·XX···XX	$b0ca	·XX···XX	$b0d2	···XX···	$b0da	···XX···	$b0e2	··XX····	$b0ea	·XXXXXXX
$b0a3	··XX·XX·	$b0ab	·XX···XX	$b0b3	·XXXXXX·	$b0bb	·····XX·	$b0c3	··XXXXX·	$b0cb	··XXXXXX	$b0d3	········	$b0db	········	$b0e3	·X······	$b0eb	········
$b0a4	·XX··XX·	$b0ac	······XX	$b0b4	·XX···XX	$b0bc	····XX··	$b0c4	·XX···XX	$b0cc	······XX	$b0d4	········	$b0dc	········	$b0e4	··XX····	$b0ec	·XXXXXXX
$b0a5	·XXXXXXX	$b0ad	·XX···XX	$b0b5	·XX···XX	$b0bd	····XX··	$b0c5	·XX···XX	$b0cd	·XX···XX	$b0d5	···XX···	$b0dd	···XX···	$b0e5	····XX··	$b0ed	········
$b0a6	·····XX·	$b0ae	··XXXXX·	$b0b6	··XXXXX·	$b0be	···XX···	$b0c6	··XXXXX·	$b0ce	··XXXXX·	$b0d6	···XX···	$b0de	···XX···	$b0e6	······XX	$b0ee	········
$b0a7	········	$b0af	········	$b0b7	········	$b0bf	········	$b0c7	········	$b0cf	········	$b0d7	········	$b0df	··X·····	$b0e7	········	$b0ef	········
$b0f0	·XX·····	$b0f8	··XXXXX·	$b100	··XXXX··	$b108	···XXX··	$b110	·XXXXXX·	$b118	··XXXXX·	$b120	·XXXXXX·	$b128	··XXXXXX	$b130	··XXXXXX	$b138	··XXXXX·
$b0f1	···XX···	$b0f9	·XX···XX	$b101	·X····X·	$b109	···XXX··	$b111	·XX···XX	$b119	·XXX··XX	$b121	·XX···XX	$b129	··XX····	$b131	··XX····	$b139	·XXX··XX
$b0f2	·····XX·	$b0fa	······XX	$b102	··XXX··X	$b10a	··XX·XX·	$b112	·XX···XX	$b11a	·XX·····	$b122	·XX···XX	$b12a	··XX····	$b132	··XX····	$b13a	·XX·····
$b0f3	·······X	$b0fb	···XXXX·	$b103	·X··X··X	$b10b	··XX·XX·	$b113	·XXXXXX·	$b11b	·XX·····	$b123	·XX···XX	$b12b	··XXXXX·	$b133	··XXXXX·	$b13b	·XX··XXX
$b0f4	·····XX·	$b0fc	···XX···	$b104	·X··X··X	$b10c	·XXXXXXX	$b114	·XX···XX	$b11c	·XX·····	$b124	·XX···XX	$b12c	··XX····	$b134	··XX····	$b13c	·XX···XX
$b0f5	···XX···	$b0fd	········	$b105	·X··X··X	$b10d	·XX···XX	$b115	·XX···XX	$b11d	·XXX··XX	$b125	·XX···XX	$b12d	··XX····	$b135	··XX····	$b13d	·XXX··XX
$b0f6	·XX·····	$b0fe	···XX···	$b106	··XX·XX·	$b10e	·XX···XX	$b116	·XXXXXX·	$b11e	··XXXXX·	$b126	·XXXXXX·	$b12e	··XXXXXX	$b136	··XX····	$b13e	··XXXXX·
$b0f7	········	$b0ff	········	$b107	········	$b10f	········	$b117	········	$b11f	········	$b127	········	$b12f	········	$b137	········	$b13f	········
$b140	·XX··XX·	$b148	···XX···	$b150	····XX··	$b158	·XX···XX	$b160	·XX·····	$b168	·XX···XX	$b170	·XX···XX	$b178	··XXXXX·	$b180	·XXXXXX·	$b188	··XXXXX·
$b141	·XX··XX·	$b149	···XX···	$b151	····XX··	$b159	·XX··XX·	$b161	·XX·····	$b169	·XXX·XXX	$b171	·XXX··XX	$b179	·XX···XX	$b181	·XX···XX	$b189	·XX···XX
$b142	·XX··XX·	$b14a	···XX···	$b152	····XX··	$b15a	·XX·XX··	$b162	·XX·····	$b16a	·XXXXXXX	$b172	·XXXX·XX	$b17a	·XX···XX	$b182	·XX···XX	$b18a	·XX···XX
$b143	·XXXXXX·	$b14b	···XX···	$b153	····XX··	$b15b	·XXXX···	$b163	·XX·····	$b16b	·XX·X·XX	$b173	·XXXXXXX	$b17b	·XX···XX	$b183	·XXXXXX·	$b18b	·XX···XX
$b144	·XX··XX·	$b14c	···XX···	$b154	XX··XX··	$b15c	·XX·XX··	$b164	·XX·····	$b16c	·XX·X·XX	$b174	·XX·XXXX	$b17c	·XX···XX	$b184	·XX·····	$b18c	·XX·XXXX
$b145	·XX··XX·	$b14d	···XX···	$b155	XX··XX··	$b15d	·XX··XX·	$b165	·XX·····	$b16d	·XX···XX	$b175	·XX··XXX	$b17d	·XX···XX	$b185	·XX·····	$b18d	·XX···XX
$b146	·XX··XX·	$b14e	···XX···	$b156	·XXXX···	$b15e	·XX···XX	$b166	·XXXXXXX	$b16e	·XX···XX	$b176	·XX···XX	$b17e	··XXXXX·	$b186	·XX·····	$b18e	··XXXXXX
$b147	········	$b14f	········	$b157	········	$b15f	········	$b167	········	$b16f	········	$b177	········	$b17f	········	$b187	········	$b18f	········

Table B-8 continued.

$b190	·XXXXXX·	$b198	··XXXXX·	$b1a0	·XXXXXX·	$b1a8	·XX··XX·	$b1b0	·XX···XX	$b1b8	·XX·X·XX	$b1c0	·XX···XX	$b1c8	·XX··XX·	$b1d0	·XXXXXXX	$b1d8	··XXX···
$b191	·XX···XX	$b199	·XX···XX	$b1a1	···XX···	$b1a9	·XX··XX·	$b1b1	·XX···XX	$b1b9	·XX·X·XX	$b1c1	·XX···XX	$b1c9	·XX··XX·	$b1d1	·····XXX	$b1d9	··X·····
$b192	·XX···XX	$b19a	·XXX····	$b1a2	···XX···	$b1aa	·XX··XX·	$b1b2	·XX···XX	$b1ba	·XX·X·XX	$b1c2	··XX·XX·	$b1ca	·XX··XX·	$b1d2	····XXX·	$b1da	··X·····
$b193	·XXXXXX·	$b19b	··XXXXX·	$b1a3	···XX···	$b1ab	·XX··XX·	$b1b3	··XX·XX·	$b1bb	·XX·X·XX	$b1c3	···XXX··	$b1cb	··XXXX··	$b1d3	···XXX··	$b1db	··X·····
$b194	·XX·X···	$b19c	·····XXX	$b1a4	···XX···	$b1ac	·XX··XX·	$b1b4	··XX·XX·	$b1bc	·XX·X·XX	$b1c4	··XX·XX·	$b1cc	···XX···	$b1d4	··XXX···	$b1dc	··X·····
$b195	·XX··XX·	$b19d	·XX···XX	$b1a5	···XX···	$b1ad	·XX··XX·	$b1b5	···XXX··	$b1bd	·XXXXXXX	$b1c5	·XX···XX	$b1cd	···XX···	$b1d5	·XXX····	$b1dd	··X·····
$b196	·XX··XXX	$b19e	··XXXXX·	$b1a6	···XX···	$b1ae	··XXXX··	$b1b6	···XXX··	$b1be	··XX·XX·	$b1c6	·XX···XX	$b1ce	···XX···	$b1d6	·XXXXXXX	$b1de	··XXX···
$b197	········	$b19f	········	$b1a7	········	$b1af	········	$b1b7	········	$b1bf	········	$b1c7	········	$b1cf	········	$b1d7	········	$b1df	········

$b1e0	·XX···XX	$b1e8	···XXX··	$b1f0	········	$b1f8	········	$b200	········	$b208	········	$b210	·XX·····	$b218	········	$b220	·····XX·	$b228	········
$b1e1	··X···X·	$b1e9	·····X··	$b1f1	········	$b1f9	········	$b201	········	$b209	········	$b211	·XX·····	$b219	········	$b221	·····XX·	$b229	········
$b1e2	··XXXXX·	$b1ea	·····X··	$b1f2	····X···	$b1fa	········	$b202	········	$b20a	··XXXX··	$b212	·XX·XX··	$b21a	··XXXX··	$b222	··XXXXX·	$b22a	··XXXX··
$b1e3	····X···	$b1eb	·····X··	$b1f3	···X·X··	$b1fb	········	$b203	···XX···	$b20b	·X···XX·	$b213	·XXXXXX·	$b21b	·XX··XX·	$b223	·XX··XX·	$b22b	·XX··XX·
$b1e4	··XXXXX·	$b1ec	·····X··	$b1f4	··X···X·	$b1fc	········	$b204	···XX···	$b20c	··XXXXX·	$b214	·XX··XX·	$b21c	·XX·····	$b224	·XX··XX·	$b22c	·XXXXXX·
$b1e5	····X···	$b1ed	·····X··	$b1f5	·X·····X	$b1fd	········	$b205	········	$b20d	·XX··XX·	$b215	·XX··XX·	$b21d	·XX··XX·	$b225	·XXX·XX·	$b22d	·XX·····
$b1e6	···XXX··	$b1ee	···XXX··	$b1f6	········	$b1fe	·XXXXXXX	$b206	········	$b20e	··XXX·XX	$b216	·XXXXX··	$b21e	··XXXX··	$b226	··XXXXX·	$b22e	··XXXX··
$b1e7	········	$b1ef	········	$b1f7	········	$b1ff	········	$b207	········	$b20f	········	$b217	········	$b21f	········	$b227	········	$b22f	········

$b230	········	$b238	········	$b240	··XX····	$b248	···XX···	$b250	····XX··	$b258	··XX····	$b260	···XX···	$b268	········	$b270	········	$b278	········
$b231	···XXX··	$b239	········	$b241	··XX····	$b249	········	$b251	········	$b259	··XX··XX	$b261	···XX···	$b269	········	$b271	········	$b279	········
$b232	··XX·XX·	$b23a	··XXXX··	$b242	··XXXX··	$b24a	···XX···	$b252	····XX··	$b25a	··XX·XX·	$b262	···XX···	$b26a	·XXX·XX·	$b272	··XXXX··	$b27a	··XXXX··
$b233	·XXXX···	$b23b	·XX··XX·	$b243	··XX·XX·	$b24b	···XX···	$b253	····XX··	$b25b	··XXXX··	$b263	···XX···	$b26b	·XX·XX·X	$b273	··XX··X·	$b27b	·XX··XX·
$b234	··XX····	$b23c	·XX··XX·	$b244	··XX·XX·	$b24c	···XX···	$b254	····XX··	$b25c	··XXXX··	$b264	···XX···	$b26c	·XX·XX·X	$b274	··XX··X·	$b27c	·XX··XX·
$b235	··XX····	$b23d	··XXXXX·	$b245	··XX·XX·	$b24d	···XX···	$b255	·XX·XX··	$b25d	··XX·XX·	$b265	···XX···	$b26d	·XX·XX·X	$b275	··XX··X·	$b27d	·XX··XX·
$b236	··XX····	$b23e	·····XX·	$b246	··XX·XX·	$b24e	···XX···	$b256	·XXXXX··	$b25e	··XX··XX	$b266	···XX···	$b26e	·XX·XX·X	$b276	··XX··X·	$b27e	··XXXX··
$b237	········	$b23f	··XXXX··	$b247	········	$b24f	········	$b257	··XXX···	$b25f	········	$b267	········	$b26f	········	$b277	········	$b27f	········

$b280	········	$b288	········	$b290	········	$b298	········	$b2a0	···XX···	$b2a8	········	$b2b0	········	$b2b8	········	$b2c0	········	$b2c8	········
$b281	········	$b289	········	$b291	········	$b299	········	$b2a1	···XX···	$b2a9	········	$b2b1	········	$b2b9	········	$b2c1	········	$b2c9	········
$b282	·XXXXX··	$b28a	··XXXXX·	$b292	··XX·XX·	$b29a	···XXXX·	$b2a2	··XXXX··	$b2aa	·XX··XX·	$b2b2	·XX···XX	$b2ba	·XX·XX·X	$b2c2	·XX··XX·	$b2ca	·XX···XX
$b283	·XX··XX·	$b28b	·XX··XX·	$b293	··XXX·X·	$b29b	··XX····	$b2a3	···XX···	$b2ab	·XX··XX·	$b2b3	·XX···XX	$b2bb	·XX·XX·X	$b2c3	··XXXX··	$b2cb	·XX···XX
$b284	·XX··XX·	$b28c	·XX··XX·	$b294	··XX····	$b29c	···XXX··	$b2a4	···XX···	$b2ac	·XX··XX·	$b2b4	··XX·XX·	$b2bc	·XX·XX·X	$b2c4	···XX···	$b2cc	··XX·XX·
$b285	·XXX·XX·	$b28d	·XX·XXX·	$b295	··XX····	$b29d	·····XX·	$b2a5	···XX···	$b2ad	·XX··XX·	$b2b5	··XX·XX·	$b2bd	·XX·XX·X	$b2c5	··XXXX··	$b2cd	···XXX··
$b286	·XX·XX··	$b28e	··XX·XX·	$b296	··XX····	$b29e	··XXXX··	$b2a6	····XX··	$b2ae	··XXXX·X	$b2b6	···XXX··	$b2be	··XX·XX·	$b2c6	·XX··XX·	$b2ce	···XX···
$b287	·XX·····	$b28f	·····XX·	$b297	········	$b29f	········	$b2a7	········	$b2af	········	$b2b7	········	$b2bf	········	$b2c7	········	$b2cf	··XX····

$b2d0	········	$b2d8	···XX···	$b2e0	····X···	$b2e8	···XX···	$b2f0	········	$b2f8	··XXXX··
$b2d1	········	$b2d9	···X····	$b2e1	····X···	$b2e9	····X···	$b2f1	········	$b2f9	·X····X·
$b2d2	·XXXXXX·	$b2da	···X····	$b2e2	····X···	$b2ea	····X···	$b2f2	··XX···X	$b2fa	X··XX··X
$b2d3	····XX··	$b2db	··X·····	$b2e3	········	$b2eb	·····X··	$b2f3	·XXXXXXX	$b2fb	X··XXXXX
$b2d4	···XX···	$b2dc	···X····	$b2e4	····X···	$b2ec	····X···	$b2f4	·X··XXX·	$b2fc	X··XX··X
$b2d5	··XX····	$b2dd	···X····	$b2e5	····X···	$b2ed	····X···	$b2f5	········	$b2fd	·X····X·
$b2d6	·XXXXXX·	$b2de	···XX···	$b2e6	····X···	$b2ee	···XX···	$b2f6	········	$b2fe	··XXXX··
$b2d7	········	$b2df	········	$b2e7	········	$b2ef	········	$b2f7	········		

LIST OF REFERENCES

"8-Bit-Nirvana: Atari Pong." 31 May 2008 <http://www.zock.com/8-Bit/D_Pong.HTML>.

Aarseth, Espen. "Computer Game Studies, Year One." *Game Studies: The International Journal of Computer Game Research* 1.1 (2001). <http://www.gamestudies.org/0101/editorial.html>.

---. "Genre Trouble: Narrativism and the Art of Simulation." *First Person: New Media as Story, Performance, and Game*. Ed. Pat Harrigan & Noah Wardrip-Fruin. Cambridge, MA: The MIT Press, 2004. 48-55.

aluminum. "Researching 'Computer' Typefaces." *Typophile.com*. 4 Feb 2008 <http://typophile.com/node/35550>.

Apicella, Vincent, Joanna Pomeranz, and Nancy Wiatt. *The Concise Guide to Type Identification*. New York: Design Press, 1990.

Archer, Wesley, Director. "Itchy and Scratchy Land." *The Simpsons*. 2 Oct 1994.

Atari. *Atari VCS Poster Catalog*. Atari , 1982. <http://www.atariage.com/catalog_overview.html?CatalogID=62>.

Atkins, Barry. *More than a Game: The Computer Game as Fictional Form*. Manchester, UK: Manchester University Press, 2003.

Badham, John, Director. *WarGames*. (DVD) Metro-Goldwyn Mayer, 1983.

Baer, Ralph H. "Video game history." *Ralph H. Baer Consultants*. 4 Feb 2008 <http://www.ralphbaer.com/video_game_history.htm>.

Bailey, Kenneth D. *Typologies and Taxonomies: An Introduction to Classification Technique*. Thousand Oaks, CA: Sage Publications, 1994.

Bailey, Ruffin. "Hacks, Mods, Easter Eggs, and Fossils: Intentionality and Digitalism in the Video Game." *Playing the Past: History and Nostalgia in Video Games*. Ed. Zach Whalen & Laurie N. Taylor. Nashville, TN: Vanderbilt University Press, 2008.

Bank Management Commission, Technical Subcommittee. "The Common Machine Language for Mechanized Check Handling: Final Specifications and Guides to Implement the Program. No. 147." 1959.

Bass, Rudi. "The Development of CBS News 36." *The Journal of Typographic Research* 1.4 (1967): 357-372.

Bauermeister, Benjamin. *A Manual of Comparative Typography: The PANOSE System*. New York: Van Nostrand Reinhold, 1988.

Berry, John D. "The Mischievous Mind behind Microsoft's TrueType Fonts ." *CreativePro.com* 20 Apr 2001. <http://www.creativepro.com/story/feature/12965.html>.

“ Berzerk (Atari).” *AtariAge.com* 20 Oct 2007. 20 Oct 2007 <http://www.atariage.com/software_page.html?SoftwareLabelID=29>.

“Berzerk Videogame by Stern.” *The International Arcade Museum and the KLOV*. 16 Aug 2007 <http://www.klov.com/game_detail.php?game_id=7096>.

Blackwell, Lewis. *20th Century Type (Remix)*. Corte Madera, CA: Gingko Press, 2000.

Blotkamp, Carel. “Introduction.” *De Stijl: The Formative Years, 1917 - 1922*. Ed. Carel Blotkamp. Trans. Charlotte I. Loeb & Arthur L. Loeb. Cambridge, MA: MIT Press, 1986. viii – xi.

---. “Theo van Doesburg.” *De Stijl: The Formative Years, 1917 - 1922*. Ed. Carel Blotkamp. Trans. Charlotte I. Loeb & Arthur L. Loeb. Cambridge, MA: MIT Press, 1986. 77 - 122.

Bogost, Ian. *Unit Operations: An Approach to Videogame Criticism*. Cambridge, MA: The MIT Press, 2006.

---. “Videogames or Video games.” *Gameology.org*. 27 Jun 2007 <http://www.gameology.org/blog/videogames_or_video_games_what_are_we_talking_about#comment-1269>.

Bogost, Ian, and Nick Montfort. “Platform Studies, a book series published by MIT Press.” *Platform Studies.com* 2007. 17 Aug 2007 <http://www.platformstudies.com/levels.html>.

Broos, Kees. *Wim Crouwel – Alphabets*. Corte Madera, CA: Gingko Press, 2003.

Buchsbaum, Walter H., and Robert Mauro, eds. *Electronic Games: Design, Programming, and Troubleshooting*. New York: McGraw-Hill Book Company, 1979.

Buckingham, David. “Studying Computer Games.” *Computer Games: Text, Narrative and Play*. Cambridge, UK: Polity, 2006.

Burnham, Van, ed. *Supercade: A Visual History of the Videogame Age, 1971 - 1984*. Cambridge, MA: The MIT Press, 2001.

Bushnell, Nolan. “Computer Space Technical Manual.” *ComputerSpaceFan*.com. 9 Apr 2008 <http://www.computerspacefan.com/ComputerSpace.pdf>.

Campbell-Kelly, Martin. “Programming the EDSAC: Early Programming Activity at the University of Cambridge.” *Annals of the History of Computing* 2.1 (1980): 7-35.

Camper, Brett. “Shareware Games: Between Hobbyist and Professional.” *The Video Game Explosion: a History from PONG to Playstation and Beyond*. Ed. Mark J.P. Wolf. Westport, CT: Greenwood Press, 2008.

Carey, Earl John. *Retro Game Programming: Unleashed for the Masses*. Course Technology PTR, 2005.

Carr, Diane. “Un-Situated Play? Textual Analysis and Digital Games.” *DiGRA Hardcore* 6 Nov 2007. <http://www.digra.org/hardcore/hc18>.

Carr, Diane, David Buckingham, Andrew Burn, and Gareth Schott. *Computer Games: Text, Narrative and Play*. Cambridge, UK: Polity, 2006.

Cayley, John. “Literal Art: Neither Lines nor Pixels but Letters.” *First Person: New Media as Story, Performance, and Game*. Ed. Noah Wardrip-Fruin & Pat Harrigan. Cambridge, MA: The MIT Press, 2004.

Chapman, Matt, and Mike Chapman. “Butt IQ.” *HomeStarRunner.com* 30 Oct 2001. 17 Dec 2007 <http://www.homestarrunner.com/sbemail3.html>.

---. “Web Comics.” *HomeStarRunner.com* 8 Oct 2007. 17 Dec 2007 <http://www.homestarrunner.com/sbemail181.html>.

Colbert, Bob. “Distella Disassembler.” 14 Feb 2008 <http://members.cox.net/rcolbert/distella.htm>.

“Computer Font - Fonts.com.” *Fonts.com by Monotype Imaging*. 25 Nov 2007 <http://www.fonts.com/findfonts/detail.htm?pid=205071>.

Covell, Chris. “ROMSearcher.” 14 Feb 2008 <http://www.disgruntleddesigner.com/chrisc/romsearcher.html>.

---. “ROMSearcher README.” 17 Apr 2008 <http://ftp.icm.edu.pl/packages/aminet/misc/emu/ROMSearcher.readme>.

Crouwel, Wim. “Type Design for the Computer Age.” *The Journal of Typographic Research* 4.1 (1970): 51-59.

Daniels, Simon. “Re: question about Westminster.” 10 Dec 2007.

Deuel, Brian. “Al Alcorn Interview.” *BrianDeuel.com*. 16 Apr 2008 <http://briandeuel.swift-mail.com/briandeuel.com/aainterview.html>.

Dictionary, American Heritage. *High Definition: An A to Z Guide to Personal Technology*. Houghton MifflinReference Books, 2006.

Drucker, Johanna. “Intimations of Immateriality: Graphical Form, Textual Sense and the Electronic Environment.” *Reimagining Textuality: Textual Studies in the Lage Age of Print*. Ed. Neil Fraistat & Elizabeth Bergmann Loizeaux. Madison, WI: University of Wisconsin Press.

---. *The Alphabetic Labyrinth: The Letters in History and Imagination*. London: Thames and Hudson, 1999.

---. *The Visible Word: Experimental Typography and Modern Art, 1909 1923*. Chicago: University of Chicago Press, 1996.

Editors. “Change is Good.” *Wired.* Feb 2007: 26.

Eskelinen, Markku. “The Gaming Situation.” *Game Studies: The International Journal of Computer Game Research* 1.1 (2001). 17 Aug 2007 <http://www.gamestudies.org/0101/eskelinen/>.

---. “Towards Computer Game Studies.” *First Person: New Media as Story, Performance, and Game*. Cambridge, MA: The MIT Press, 2004.

Ex, Sjarel. “Vilmos Huszár.” *De Stijl: The Formative Years, 1917 - 1922*. Ed. Carel Blotkamp. Trans. Charlotte I. Loeb & Arthur L. Loeb. Cambridge, MA: MIT Press, 1986. 77 - 122.

Falk, Hugh. “Broderbund.” 19 Dec 2007 <http://gotcha.classicgaming.gamespy.com/broderbund.htm>.

Forster, Winnie. *The Encyclopedia of Game Machines*. Magdalena Gniatczynska, 2005.

Frutiger, Adrian. “Letterforms in Phototypography.” *The Journal of Typographic Research* 4.4 (1970): 41-46.

---. “OCR-B: A Standardized Character for Optical Recognition.” *The Journal of Typographic Research* 1.2 (1967): 137-146.

Galloway, Alexander R. *Gaming: Essays on Algorithmic Culture*. Minneapolis, MN: University of Minnesota Press, 2006.

Genette, Gérard. *Paratexts: Thresholds of Interpretation*. Trans. Jane E. Lewin. Cambridge University Press, 1997.

Gifford, Kevin. “‘Game Mag Weaseling:’ A Mere Blip on the Radar.” *Game Set Watch* 19 Aug 2006. 15 Aug 2007 <http://www.gamesetwatch.com/2006/08/column_game_mag_weaseling_a_me.php>.

Goldsmith, Jr., Thomas T., and Estle Ray Mann. “Cathode-Ray Tube Amusement Device.” US Patent 2455992. 1948.

Graetz, J.M. “The Origin of Spacewar!.” *Supercade! A Visual History of the Videogame Age, 1971 - 1984*. Cambridge, MA: The MIT Press, 2001.

‘graphic6’. “WTF Case “font please” : MyFonts.” 25 Nov 2007 <http://www.myfonts.com/WhatTheFont/forum/viewthread.php?threadid=33868>.

Hague, James. “The Giant List of Classic Game Programmers.” 24 Apr 2008 <http://www.dadgum.com/giantlist/list.html>.

Heiserman, David L. *How to Design & Build Your Own Custom TV Games*. Blue Ridge Summit, PA: TAB Books, 1978.

“History.” *Computer Space Fan* 2008. 10 Apr 2008 <http://www.computerspacefan.com/History.htm>.

Hustwit, Gary, Director. *Helvetica*. (DVD) Plexifilm, 2007.

International Botanical Congress. *International Code of Botanical Nomenclature: Saint Louis Code: Adopted by the Sixteenth International Botanical Congress, St Louis, Missouri, July-August 1999*. Königstein, Germany: Koeltz Scientific Books, 2000.

Israel, Kirk. “2600 101.” *AlienBill.com* 2004. 15 Aug 2007 <http://alienbill.com/2600/101/>.

Janzen, Mary. “Courses Redesigned to Improve Students’ Learning.” *ITS Academic Computing Newsletter (Pennsylvania State Univesity)* Spring 2005: 10-11.

Jaspert, W. Pincus. *The Encyclopaedia of type faces*. London: Blandford, 1970.

Jerz, Dennis G. “Somewhere Nearby is Colossal Cave: Examining Will Crowther’s Original “Adventure” in Code and in Kentucky.” *Digital Humanities Quarterly* 1.2 (2007). 24 May 2008 <http://www.digitalhumanities.org/dhq/vol/001/2/000009.html>.

Jones, Steven. *The Meaning of Video Games*. New York, NY: Routledge, 2008.

Juul, Jesper. *Half-Real: Video Games between Real Rules and Fictional Worlds*. Cambridge, MA: The MIT Press, 2005.

---. “Where the Action Is.” *Game Studies: The International Journal of Computer Game Research* 5.1 (2005). <http://www.gamestudies.org/0501/editorial>.

Kent, Steven L. *The Ultimate History of Video Games: From Pong to Pokemon--The Story Behind the Craze That Touched Our Lives and Changed the World*. New York: Three Rivers Press, 2001.

Kiesling, Stephen. “Death of a Video Gamer.” *Video Games* Oct 1982: 14-15.

King, Geoff, and Tanya Krzywinska. *Tomb Raiders and Space Invaders: Videogame Forms and Contexts*. London: I. B. Tauris, 2006.

Kirriemuir, John. “A History of Digital Games.” *Understanding Digital Games*. Ed. Rutter, Jason & Jo Bryce. Thousand Oaks, CA: Sage Publications, 2006. 21-35.

Kirschenbaum, Matthew G. *Mechanisms: New Media and the Forensic Imagination*. Cambridge, MA: The MIT Press, 2008.

---. “Word as Image in an Age of Digital Reproduction.” *Eloquent images : word and image in the age of new media*. Ed. Michelle R. Kendrick & Mary E. Hocks. Cambridge, MA: MIT Press, 2003.

Konzack, Lars. “Computer Game Criticism: A Method for Computer Game Analysis.” *CGDC Conference Proceedings*. Tampere, Finland: Tampere University Press, 2002. 89-100. 10 Aug 2007 <http://www.vrmedialab.dk/~konzack/tampere2002.pdf>.

Kuchera, Ben. “An NES, two clones, and the Virtual Console. Where does Mario look his best?.” *Ars Techncia* 14 Feb 2007. 26 Apr 2008 <http://arstechnica.com/journals/thumbs.ars/2007/02/14/7044>.

Kücklich, Julian. “Game Studies 2.0.” *DiGRA Hardcore*. 24 May 2008 <http://www.digra.org/hardcore/hc12>.

---. “Perspectives of Computer Game Philology.” *Game Studies* 3.1 (2003).

Larabie, Ray. “Joystix font family.” 27 Apr 2008 <http://www.myfonts.com/larabie/joystix/>.

Licko, Zuzana. “Lo-Res.” *Emigre* 59 (2001).

Longo, Robert, Director. *Johnny Mnemonic*. (DVD) Sony Pictures, 1997.

Macmillan, Neil. *An A-Z of Type Designers*. New Haven, CT: Yale University Press, 2006.

McCourt, Michael. "Haunted House (Magnavox Odyssey, 1972)." *Armchair Arcade*. 16 Apr 2008 <http://www.armchairarcade.com/neo/node/1250>.

McGann, Jerome. *Radiant Textuality: Literature after the World Wide Web*. New York: Palgrave, 2001.

---. *A Critique of Modern Textual Criticism*. Chicago: University of Chicago Press, 1983.

McKenney, James L. "Developing a Common Machine Language for Banking: The ABA Technical Subcommittee Story." *IEEE Annals of the History of Computing* 17.4 (1995): 61 - 75.

Microsoft. "Microsoft Typography - Font information." 29 Mar 2008 <http://www.microsoft.com/typography/fonts/font.aspx?FID=208&FNAME=Westminster&FVER=1.01>.

Montfort, Nick. "Combat in Context." *Game Studies* 6.1 (2006). <http://gamestudies.org/0601/articles/montfort>.

---. "Continuous Paper: The Early Materiality and Workings of Electronic Literature." Philadelphia, PA, 2004. <http://nickm.com/writing/essays/continuous_paper_mla.html>.

---. "How Stella Got Her Text Back." *Grand Text Auto.org* 5 Oct 2005. <http://grandtextauto.gatech.edu/2005/10/05/how-stella-got-her-text-back/>.

---. *Twisty Little Passages: An Approach to Interactive Fiction*. Cambridge, MA: The MIT Press, 2005.

Morgan, Rik. "Tomy Blip / Blip-o-Mat." *Hand Held Museum*. 4 Feb 2008 <http://www.handheldmuseum.com/Tomy/Blip.htm>.

Motorola. "MCM6570, MCM6571, MCM6571A, MCM6572 thru MCM6579." 1979.

Moulthrop, Stuart. "First Person Thread | A Riposte to Espen Aarseth." *Electronic Book Review* 2004. 17 Aug 2007 <http://www.electronicbookreview.com/thread/firstperson/insurgent>.

Newman, James. "The myth of the ergodic videogame." *Game Studies: The International Journal of Computer Game Research* 1.2 (2002). 25 Apr 2008 <http://www.gamestudies.org/0102/newman/>.

---. *Videogames*. New York: Routledge, 2004.

Nutter, David, Director. "Pilot." *Terminator: The Sarah Connor Chronicles.* 13 Jan 2008.

"OCR-B 10 BT : Style Details : MyFonts." 4 Feb 2008 <http://www.myfonts.com/fonts/bitstream/ocr-b/ocr-b/>.

Owens, Mark, and David Reinfurt. "Pure Data: Moments in a History of Machine-Readable Type." *Visual Communication* 4.6 (2005): 144-150.

"Pac-Mondrian." *Prize Budget for Boys* 13 Dec 2004. 17 Dec 2007 <http://pbfb.ca/pac-mondrian/>.

Panofsky, Ted. "RE: Galaxy Game Inquiry." (Personal E-mail) 19 Mar 2008.

"para-, prefix1." *Oxford English Dictionary Online*. 2007.

"para-, prefix2." *Oxford English Dictionary Online*. 2007.

Pewther, Judson. "Artifacting." *Compute!'s Second Book of Atari Graphics*. Compute! Books, 1983. 7 Dec 2007 <http://www.atariarchives.org/c2bag/page193.php>.

Phil's Photo, inc. *Homage to the Alphabet*. Rockport, Mass: Rockport Publishers, 1980.

Pitts, Bill. "The Galaxy Game." *Stanford Computer History Exhibits* 29 Oct 1997. 25 Feb 2008 <http://infolab.stanford.edu/pub/voy/museum/galaxy.html>.

Purvis, Alston W. *Dutch Graphic Design: 1918-1945*. New York: Van Nostrand Reinhold, 1992.

qtchicks. "NAMCO '70~'80's ARCADE VUDEOGAMES FONT MUSEUM." 23 Apr 2008 <http://qtchicks.hp.infoseek.co.jp/fonts.html>.

Rabinow, Jacob. "Whither OCR? And Whence?." *Datamation* Jul 1969: 38 - 42.

Riddle, Sean. "chanf.asm (Channel F BIOS Dissassembly)." 1976. <http://veswiki.com/Disassembly:Bios>.

---. "Channel F info." *Sean Riddle's Home Page*. 28 Feb 2008 <http://members.cox.net/seanriddle/chanf.html>.

Robinett, Warren. "Foreword." *The Video Game Theory Reader*. Ed. Mark J.P. Wolf & Bernard Perron. New York: Routledge, 2003.

---. "Re: Question about Basic Programming." (Personal E-mail) 24 April 2008.

Rutter, Jason, and Jo Bryce. "Editorial Comment.." *Information, Communication & Society* (2003): 5-11.

Ryan, Marie-Laure. *Avatars of Story*. Minneapolis, MN: University of Minnesota Press, 2006.

Scan-Data Corporation. "The Two-Faced 200." *Datamation* Jul 1969: 146-147.

Schantz, Herbert F. *The History of OCR: Optical Character Recognition*. Recognition Technologies Users Association, 1982.

Sherman, Nick. "Bell Centennial." *NickSherman.com* 2005. 14 Aug 2007 <http://www.nicksherman.com/articles/bellCentennial.html>.

Shiny Entertainment. *Enter the Matrix*. New York, NY: Atari/Warner Brothers Interactive.

Signetics. *64 x 8 x 5 Character Generator (2513)*. Signetics.

Spangler, Kathleen. “New Technologies Challenge Classical Typography.” *Technological Change in Printing and Publishing*. Ed. Lowell H. Hattery & George P. Bush. Rochelle Park, NJ: Hayden Book Company, Inc., 1973.

“Star Trek (1972).” *The Killer List of Videogames*. 7 Aug 2007 <http://www.klov.com/game_detail.php?letter=&game_id=9771>.

Sullivan, George. *Screen Play: The Story of Video Games*. New York, NY: Frederick Warne, 1983.

Taylor, Laurie N. “Networking power : videogame structure from concept art.” *Videogames and Art*. Ed. Andy Clarke & Grethe Mitchell. Chicago, IL: Intellect, 2007.

Tor. “Gargoyle : Interactive Fiction.” 20 Apr 2008 <http://ccxvii.net/gargoyle/>.

“USA Standard Character Set for Optical Character Recognition. USASI X3.17-1966.” 18 Jul 1966.

Vasvari, Zsolt, and Christopher Kirmse. “MAWS - showing resource: berzerk.” *Mame World*. 28 Apr 2008 <http://www.mameworld.net/maws/romset/berzerk/>.

“Video Game: Computer Space Ball, Nutting Associates.” *The Arcade Flyer Archive*. 8 Feb 2008 <http://www.arcadeflyers.com/index.php?page=flyer&db=videodb&id=1532&image=1>.

“Video Game: Elepong, Taito.” *Arcade Flyer Archive*. 8 Feb 2008 <http://www.arcadeflyers.com/?page=thumbs&db=videodb&id=351>.

“Vintage Computing and Gaming | Archive » Retro Scan of the Week: “So You Want to Be a Video Games Inventor”.” 24 Nov 2007 <http://www.vintagecomputing.com/index.php/archives/222>.

Wachowski, Larry, and Andy Wachowski, Directors. *The Matrix*. (DVD) Warner Home Video, 1999.

Wallis, Lawrence W. “The Monotype Chronicles, Timeline, 1960 - 1970.” *MonotypeImaging.com*. 15 Aug 2007 <http://www.monotypeimaging.com/aboutus/MT1960_1970.aspx>.

Wardrip-Fruin, Noah, and Nick Montfort, eds. *The New Media Reader*. Cambridge, MA: The MIT Press, 2003.

Welch, Thomas Ross. “Electric Sign.” US Patent 2290261. 1942. <http://www.google.com/patents?id=7WRRAAAAEBAJ>.

Whalen, Zach. “Game/Genre: A Critique of Generic Formulas in the Video Games in the Context of ‘The Real’.” *Works and Days* 22.43/44 (2004).

Wikipedia contributors. “Architype Van Doesburg.” *Wikipedia, The Free Encyclopedia*. Wikimedia Foundation, 2007. 10 Feb 2008 <http://en.wikipedia.org/w/index.php?title=Architype_Van_Doesburg&oldid=180587471>.

---. “ATASCII.” *Wikipedia, The Free Encyclopedia*. Wikimedia Foundation, 2007. 22 May 2008 <http://en.wikipedia.org/w/index.php?title=ATASCII&oldid=138218302>.

---. “Electromagnetic interference.” *Wikipedia, The Free Encyclopedia*. Wikimedia Foundation, 2007. 22 Aug 2007 <http://en.wikipedia.org/w/index.php?title=Electromagnetic_interference&oldid=152808575>.

Wikipedia contributors, and 83.117.23.63. “Holotype.” *Wikipedia, The Free Encyclopedia*. Wikimedia Foundation, 2005. 22 Jan 2008 <http://en.wikipedia.org/w/index.php?title=Holotype&oldid=23413089>.

Winter, David. “PONG in a Chip.” *PONG-Story*. 16 Apr 2008 <http://www.pong-story.com/gi.htm>.

Wolf, Mark J. P. “Video Game Genres.” *The Video Game Explosion: a History from PONG to Playstation and Beyond*. Ed. Mark J.P. Wolf. Westport, CT: Greenwood Press, 2008.

Wolf, Mark J.P. “Arcade Games of the 1970s.” *The Video Game Explosion: a History from PONG to Playstation and Beyond*. Ed. Mark J. P. Wolf. Westport, CT: Greenwood Press, 2008.

Wood, Frank W. “Illuminated Announcement and Display Signal.” US Patent 0974943. 8 Nov 1910. 26 Apr 2008 <http://www.google.com/patents?id=m4cPAAAAEBAJ>.

Woodcock, Gregg. “Vectrex FAQ 6.0.” *DigitPress.com* 1996. 5 Apr 2008 <http://www.digitpress.com/faq/vectrex.txt>.

Zadeh, Lotfi Asker. *Fuzzy Sets and Applications: Selected Papers*. New York: Wiley, 1987.

Zapf, Hermann. “The Changes in Letterforms Due to Technical Developments.” *The Journal of Typographic Research* 2.4: 351-368.

BIOGRAPHICAL SKETCH

Zach Whalen grew up in East Tennessee, where he pursued undergraduate education at Carson-Newman College. As an honors student at Carson-Newman, Zach initiated his interest in digital media with a senior project on Hypertext Fiction and Critical Theory. It is also at Carson-Newman that Zach, through both athletic and musical collaboration, met and soon fell in love with his wife Stacy, who moved with Zach to Gainesville in 2002 so that he could pursue a Master's Degree in English at the University of Florida.

Zach's Master's Thesis, an approach to analyzing videogame music, has lead to an article in the journal *Game Studies,* and it is being reprinted in a collection on music and multimedia due out in 2009. Zach has also published articles and book chapters related to music and videogames, genre in videogames, the videogame *Grand Theft Auto: San Andreas*, and the *CSI* franchise of television shows and spin-off videogames. Zach has co-edited, with Laurie N. Taylor, a collection of essays titled *Playing the Past: History and Nostalgia in Video Games*, which will be published by Vanderbilt University Press in Fall 2008.

For most of his tenure as a doctoral student, Zach has worked with the online journal *ImageTexT: Interdisciplinary Comics Studies*, the first (and still only), online peer-reviewed journal of comics scholarship. *ImageTexT* was founded by Donald Ault and is published by the Department of English. Zach has served as Webmaster for *ImageTexT*, and from 2006 – 2008, Zach held the position of Production Editor for the journal.

Zach is also the founder, editor, and administrator of the website *Gameology.org*, a group blog and resource for a community of videogame scholars.

In fall 2008, Zach will move to Virginia to take a position as Assistant Professor at the University of Mary Washington. Also, more importantly, he and Stacy will welcome their first child (a baby girl!) into the world.