Games & Culture

A Typographic Archaeology of *Berzerk*

Journal:	*Games and Culture*
Manuscript ID:	GAMES-07-0086
Manuscript Type:	Original Manuscript
Keywords:	videogames, Atari 2600, typography, materiality, platform, Berzerk, Vectrex, textual studies

A TYPOGRAPHIC ARCHAEOLOGY OF *BERZERK*

Abstract

Analyzing videogames as medially-layered cultural artifacts is an increasingly influential and relevant approach in the field of game studies. This article extends this approach with the metaphor of archaeology to argue that the “layered” or “leveled” analytical structures employed by Lars Konzack, Nick Montfort, Espen Aarseth and others can be extended by considering how the artifacts embedded in these layers reveal the dialectical or discursive relationships among them. Just as an archeologist infers relationships among time periods and cultures, I argue that the contextualization of medial layers in videogame texts can be elucidated with an understanding of the artifacts or disruptions which occur at the intersections of game layers. By analyzing different versions of the videogame *Berzerk*, this article demonstrates how typographic analysis can characterize the expressive capabilities of a platform and how alphanumeric character design can anticipate or depend on the often-overlooked layer of the television screen.

Keywords

videogames, typography, platform studies, ludology, Berzerk, Atari 2600, Vectrex, fuzzy, jaggy, textual studies

Studying videogames requires an archaeological approach – digging down through layers of culture, context, and platform to learn more about the artifacts that constitute the gaming phenomena. 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 article, 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. The typeface emulated in one instance of Berzerk (for the Atari 2600 console) models the form of a popular typeface of the 1960s, Countdown™, which is often associated (often 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.

This means of achieving identity is important because it mobilizes the archaeological metaphor in two senses. First, the interferences of the screen, radio frequency noise, and any other intrusions are typically referred to as *artifacts* because they impede image transmission or image quality. One common example of this sort of artifact manifests clearly in JPEG images (a so-called "lossy" format) that have been greatly compressed. 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 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 literary and linguistic studies, the status of the physical text (including page layout, typography, printing quality) as a component of the text's meaning has had a relatively recent resurgence. In criticizing prevailing views of textual criticism, particularly those dependent on the ideology of authorship as the sole locus of critical interpretation, Jerome McGann 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, 1983, p. 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, 1996, p. 1). This article 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, social aspects of videogames by way of the physical and material circumstances of their production and reception.

One way of indexing those circumstances 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 2). In the first image, the stair-stepped edges of the letters' diagonal edges 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 from screen artifacts.

These effects are far from accidental. For example, *Compute!'s Second Book of Atari Graphics* includes a chapter devoted to techniques for taking advantage of screen artifacts for improved visual effects.[1] Still, the point is not that either fuzzy or jaggy type is a better or more faithful rendition of any ideal design of a character or the designer's intended appearance. Rather, both 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 ability of fuzzy and jaggy suggests the importance of understanding the effects screen as part of any approach to videogame study, but in fact the screen is typically taken for granted in the existing literature, though its utility as a metaphor for computing is significant in this context.

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 (2007). 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 (2006). The two illustrations appear in figure 3. The first (3A) is from Montfort's *Game Studies* article where it retains the ordinals employed in the initial list he provides of the levels. The second (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 level – a fitting characterization for a foundational substratum. This is significant because 3B implies a degree of co-involvement and dependency that is missing from the discrete, padded cells and strict ordinals of 3A. In other words, figure 3A presents the five levels as being *jaggy*, whereas figure 3A illustrates the relationship in a way that is more *fuzzy*.

Besides responding specifically to Lars Konzack's (2002) seven-layer model for game analysis,[2] Montfort also addresses other approaches which stratify videogame content for the purposes of study. Julian Kücklich also deals with one such approach in his article, "Perspectives of Computer Game Philology," where Kücklich caricatures the much-maligned, so-called "narratological" approach[3] 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" (2003). 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 of 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.

It is also significant 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" (2003, my emphasis). As he is using it here, Kücklich means something synomymous 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 significant 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.

Another common stratification of game content worth noting is condensed into the following syllogism 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)" (2004, p. 48). This is the core taxonomy most clearly articulated by Jesper Juul4 and most aggressively defended by Markku Eskelinen.5 The advantage of this ludological approach is that it allows videogames to be analyzed a long history of games and play that is as ancient as culture itself. However, when in practice this formal separation privileges rules at the exclusion of semantic context (as Aarseth elaborates in his essay), Stuart Moulthroup seems correct in calling the insularity of this allegedly self-contained system, "alarmingly narrow" (2004, p. 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 category. In other words, 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 competing) 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 his is not necessarily a direct response to the ludology conversation, Alexander Galloway's approach introduced in *Gaming: Essays on Algorithmic Culture* (2006) demonstrates one way of understanding how the material/semiotic dimension of the gameworld can influences 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" (2006, p.

32). Furthermore, this sense in which game characters are embodiments of “math made visible” (Galloway, 2006, p. 32) is continuous with control structures that determine formal actions within the game since these are dictated by informatic principles that drive software. In a similar approach, Ian Bogost’s unit operations (2006) creates an analytical method from the object-oriented logic of software, identifying both software (rules) and the gameworld as systems constructed on the basis of separate but interdependent units. 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.”

The logic of platform studies offers much that informs the present typographic approach to videogame textuality. Because a significant characteristic of game typography is the extent to which it relies on and exploits the constraints of the game platform, the typographic approach presented in this article is very much in harmony with the overarching logic of platform studies. To illustrate this, I offer the following analysis of the videogame *Berzerk*.

Berzerk

The first 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, 1982, p. 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, 1983, p. 70). Significantly, Radecki's criticism here does not identify the act of killing as affectively realistic. 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. The graphical impetus for this moral panic is an important point for contextualizing the game in terms of genre – *Berzerk* is a prototypical third-person shoot-em-up – but it is also a relevant point of comparison among the other versions of the game as it was ported to multiple console systems.

Not including the arcade sequel Frenzy (Stern Electronics, 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.6 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. This co-involvement seems to follow Montfort's own thinking when he drops the ordinals in representing level order for introducing Platform Studies (figure 3A). That is, whereas listing platform as the "first" level and placing it at the bottom of the stack affords 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 versions of Berzerk illustrate what happens as programmers attempt to retain the same textual and typographic material while changing platforms. The result is a relationship between levels where contingencies and propensities of each platform have enforced compromises (or improvements) in the game form and interface that manifest at the level of reception and operation.

Figure 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. 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). 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 6) 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 of both the platform and the display, also affects game play and the experience or interpretation of the game text.

The Arabic numeral 3 provides reflects a more subtle variation that reveals how pervasive the interactions and influences are among the levels of the game object. Figure 7**Error! Reference source not found.** 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 7B 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 8 illustrates this compositional logic by highlighting and isolating one such pixel.

The highlighted pixel in figure 8A illustrates 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 – 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 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 8 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 model, the physical conditions of the display technology.

Berzerk on Atari 2600

Figure 9 illustrates the relationship between *Berzerk*'s binary data and the numeral 3 as generated by the emulator Stella. The Atari 2600 lacked a native text rendering engine, 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 that call up a specified segment of code which contains a graphic representation of the Arabic numeral "3." 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.

In figure 9A, I have converted the game's code into a visually accessible 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 on the right (and the similar preceding images), I used the emulator 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 9B, the bit positions in the ROM must be either on or off 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 10 where the image in 10B is a different capture of the same numeral 3 represented in figure 9B. 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 10B, 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 ideal, 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 (10B) emphasizes the effects of the screen layer, the screenshot image (9B) attempts to ignore it.

This is why the use of jaggy type is so significant. It is not reclaiming the aesthetic of the screen grid, as is often implied, but is instead foregrounding the logical matrix of the game's machine code. This is not to say that the jaggy numeral three in figure 9B 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 10 makes an allowance for the regime of the screen as a signifier of mediality, then figure 11 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 actual Atari 2600 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 2B, this image is inherently and unself-consciously fuzzy and 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, one possibility as to why this shape was chosen for the numeral rather than the form used for 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 12 performs the same numerical signification within the printed, simulated game as the 3's in both game images in figures 8 - 11.[7] This image appears in the instruction manual for Atari's adaptation of *Berzerk* for the 2600, and 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 12B. The line that was previously a short post comprising a single pixel and a single code bit has now become a gradually tapering point. This printed, fuzzy image of the numeral 3 is quite a departure from the jaggy 3 in figure 8, but if anything, it has a stronger resemblance to the fuzzy 3 in figure 11. From this association, it is possible to surmise that the anonymous graphic artist 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 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 literally 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 2600 *Berzerk*: the shapes of the numerals in *Berzerk* appear to be adapted from Countdown™, a typeface Colin Brignall designed in 1965 for Letraset.

Table 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 not enough of a similarity to call *Berzerk*'s typography an implementation of Countdown, the typeface's influence on *Berzerk*'s design is clear.[8]

With regard to the numeral 3, the introduction of a tapering point seems to strengthen this association with Countdown since one of the typeface's defining features is the way Brignall used rounded internal and external corners in a way that mimics the effects of halation and light buildup. Whether or not Brignall actually intended to simulate screen display technology of the mid 1960s, the letter forms 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 (" Berzerk (Atari)," 2007), 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.

Berzerk on the 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.[9] 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. Letter forms are, therefore, composed of 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 13).

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 14 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 lettershapes. 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 3's 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 felicities 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.

References

Berzerk (Atari). (2007). *AtariAge.com*. Retrieved October 20, 2007, from http://www.atariage.com/software_page.html?SoftwareLabelID=29

Aarseth, E. (2004). Genre Trouble: Narrativism and the Art of Simulation. In P. Harrigan & N. Wardrip-Fruin (Eds.), *First Person: New Media as Story, Performance, and Game* (pp. 48-55). Cambridge, MA: The MIT Press.

Berzerk Videogame by Stern. *The International Arcade Museum and the KLOV*. Retrieved August 16, 2007, from http://www.klov.com/game_detail.php?game_id=7096

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

Bogost, I., & Montfort, N. (2007). Platform Studies, a book series published by MIT Press. *Platform Studies.com*. Retrieved August 17, 2007, from http://www.platformstudies.com/levels.html

Countdown SH™ font family. *MyFonts*. Retrieved December 9, 2007, from http://www.myfonts.com/fonts/efscangraphic/countdown-sh/

Defender (Atari). *AtariAge.com*. Retrieved December 11, 2007, from http://www.atariage.com/software_page.html?SoftwareLabelID=128

Design It. *PlayVectrex.com*. Retrieved December 9, 2007, from http://www.playvectrex.com/designit_f.htm

Drucker, J. (1996). *The Visible Word: Experimental Typography and Modern Art, 1909–1923*. Chicago: University of Chicago Press.

Eskelinen, M. (2004). Towards Computer Game Studies. In *First Person: New Media as Story, Performance, and Game*. Cambridge, MA: The MIT Press.

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

Hamilton, E. (1992). JPEG File Interchange Format, Version 1.02. *Digicamsoft.com*. Retrieved December 10, 2007, from http://www.digicamsoft.com/jfif3/jfif3-1.html

Juul, J. (2001). Games Telling stories? -- A brief note on games and narratives. *Game Studies: The International Journal of Computer Game Research*, *1*(1). Retrieved August 17, 2007, from http://www.gamestudies.org/0101/juul-gts/

Juul, J. (2005). *Half-Real: Video Games between Real Rules and Fictional Worlds*. Cambridge, MA: The MIT Press.

Kiesling, S. (1982). Death of a Video Gamer. *Video Games*, 14-15. Retrieved August 13, 2007, from http://home.hiwaay.net/~lkseitz/cvg/death.html

Konzack, L. (2002). Computer Game Criticism: A Method for Computer Game Analysis. *CGDC Conference Proceedings*, 89-100. Retrieved August 10, 2007, from http://www.vrmedialab.dk/~konzack/tampere2002.pdf

Kücklich, J. (2003). Perspectives of Computer Game Philology. *Game Studies*, *3*(1). Retrieved from http://www.gamestudies.org/0301/kucklich/

McGann, J. J. (1983). *A Critique of Modern Textual Criticism*. Chicago: University of Chicago Press.

McNeil, A. *Berzerk*. Stern Electronics.

McNeil, A. *Frenzy*. Stern Electronics.

Montfort, N. (2006). Combat in Context. *Game Studies*, *6*(1). Retrieved from http://gamestudies.org/0601/articles/montfort

Moulthrop, S. (2004). First Person Thread | A Riposte to Espen Aarseth. *Electronic Book Review*. Retrieved August 17, 2007, from http://www.electronicbookreview.com/thread/firstperson/insurgent

Pewther, J. (1983). Artifacting. In *Compute!'s Second Book of Atari Graphics*. Compute! Books. Retrieved December 7, 2007, from http://www.atariarchives.org/c2bag/page193.php

Portable Network Graphics (PNG) Specification (Second Edition). (2003). *W3 Consortium*. Retrieved December 10, 2007, from http://www.w3.org/TR/PNG/

RealSports Volleyball (Atari). *AtariAge.com*. Retrieved December 11, 2007, from http://www.atariage.com/software_page.html?SoftwareLabelID=404

Sullivan, G. (1983). *Screen Play: The Story of Video Games*. New York, NY: Frederick Warne.

Figures

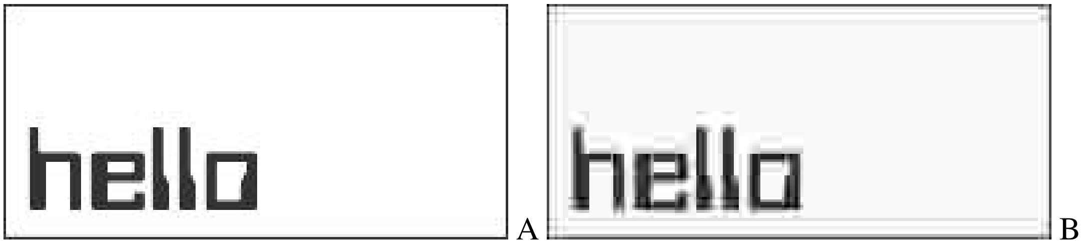


Figure 1. A) A JPEG image with little compression (95% quality). B) The same image with high compression (20% quality).

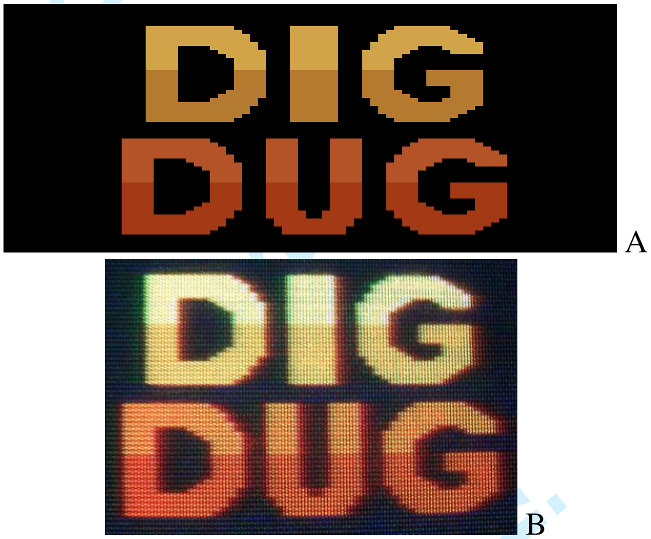


Figure 2. A) Screenshot of *Dig Dug* for Atari 2600 as played with the emulator Stella running on Windows XP. B) A photograph of *Dig Dug*'s title screen as played on an actual CRT television hooked up to an actual Atari 2600.

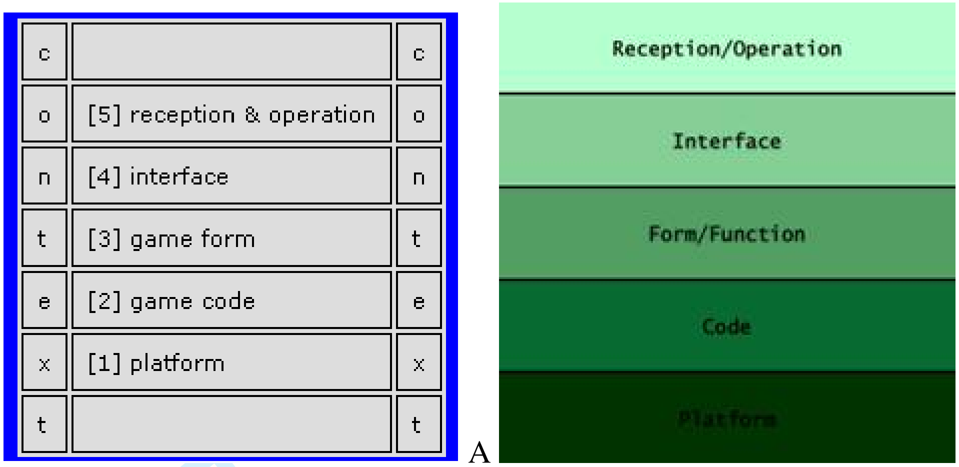


Figure 3. A) Montfort’s table of game levels for computer game analysis; from his article “Combat in Context.” <http://gamestudies.org/0601/articles/montfort> B) An alternate version (size adjusted) of Montfort’s table of five levels (with their ordinals removed) from the website introducing Montfort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

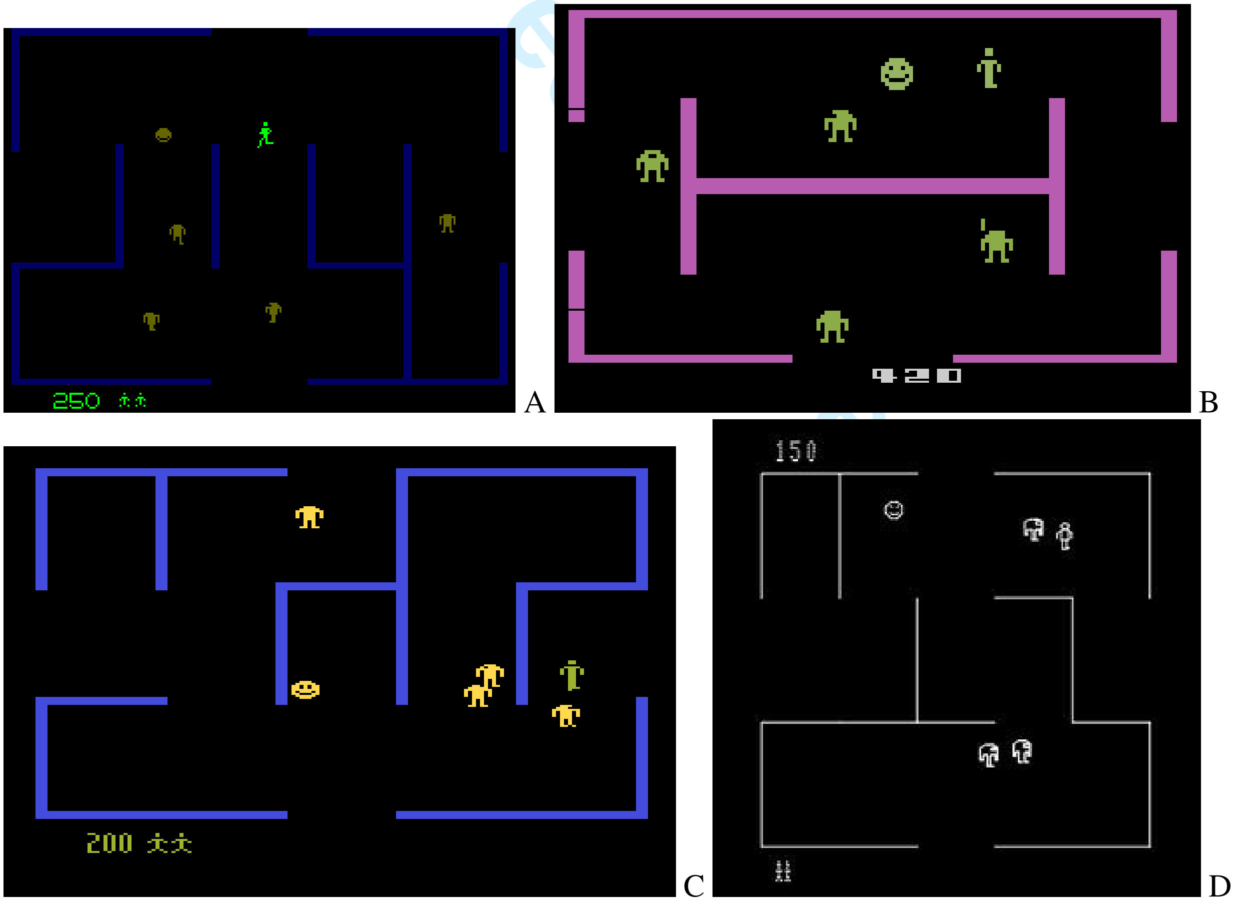


Figure 4. An array of the four licensed videogame versions of *Berzerk*. A) *Berzerk* the original arcade game [screenshot from MAME, scaled 25%]; B) *Berzerk* for Atari 2600 [screenshot from Stella, scaled 50%]; C) *Berzerk* for Atari 5200 [screenshot from MESS]. D) *Berzerk* for Vectrex [screenshot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

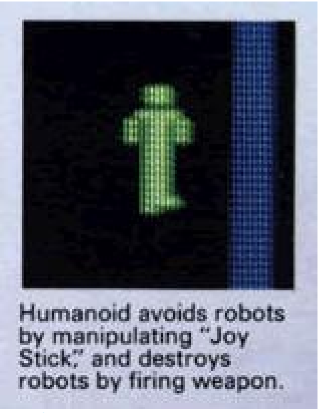

Humanoid avoids robots by manipulating "Joy Stick," and destroys robots by firing weapon.

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

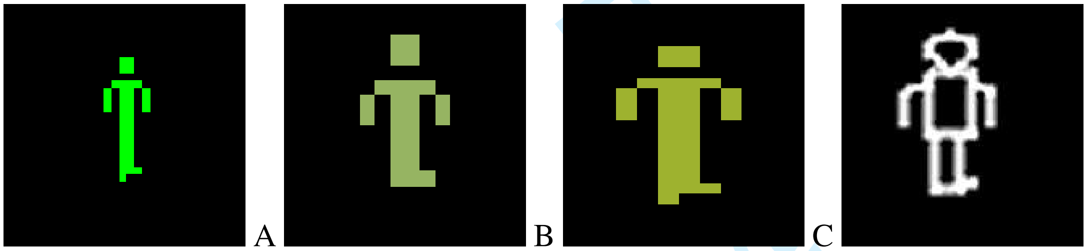


Figure 6. The different humanoid stick figures from the four videogame versions of *Berzerk*. The gap between the figure's head and shoulders appears in all but the Vectex version. A) Original arcade game [screenshot from MAME]; B) *Berzerk* for Atari VCS [screenshot from Stella]; C) *Berzerk* for Atari 5200 [screenshot from MESS]; D) *Berzerk* for Vectrex [screenshot from MESS].

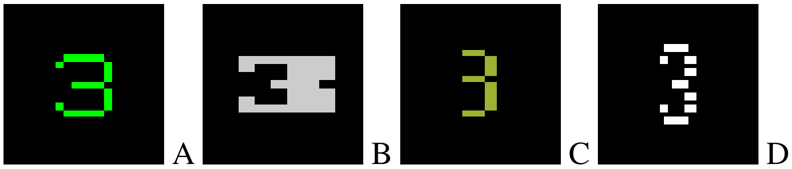


Figure 7. An array of different versions of the numeral 3 as depicted in different *Berzerk*s. A) *Berzerk* the original arcade game [screenshot from MAME]; B) *Berzerk* for Atari

VCS [screenshot from Stella]; C) *Berzerk* for Atari 5200 [screenshot from MESS]; D) *Berzerk* for Vectrex [screenshot from MESS].

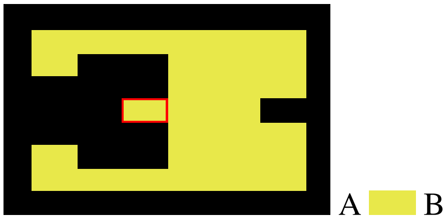
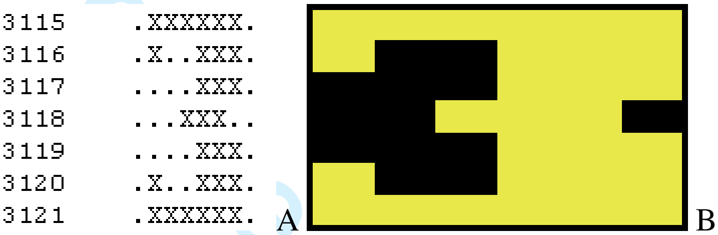
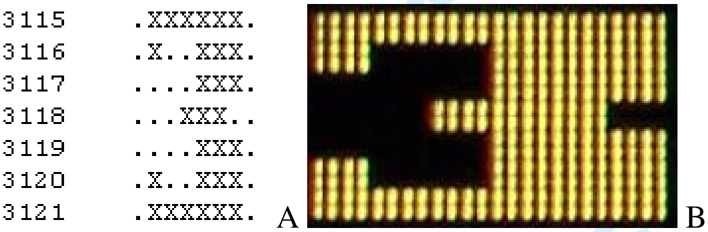


Figure 8. A) The numeral 3 from *Berzerk* for Atari VCS [screenshot from Stella]. The red outline highlights a single pixel. B) A solitary pixel from this numeral 3.



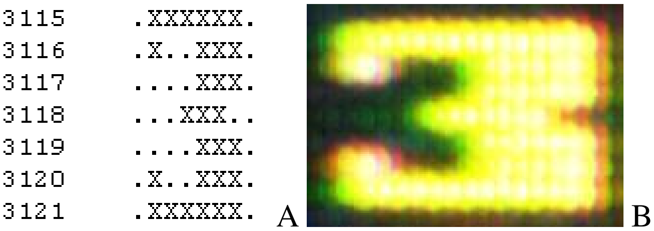
```
3115    .XXXXXX.
3116    .X..XXX.
3117    ....XXX.
3118    ...XXX..
3119    ....XXX.
3120    .X..XXX.
3121    .XXXXXX.
```

Figure 9. A) A representation a section of the machine code (ROM) which executes *Berzerk* for Atari.[10] 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.
```

Figure 10. A) Same code section as figure 9A. B) Photograph of numeral 3 generated by Stella on LCD monitor.



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

Figure 11. 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.

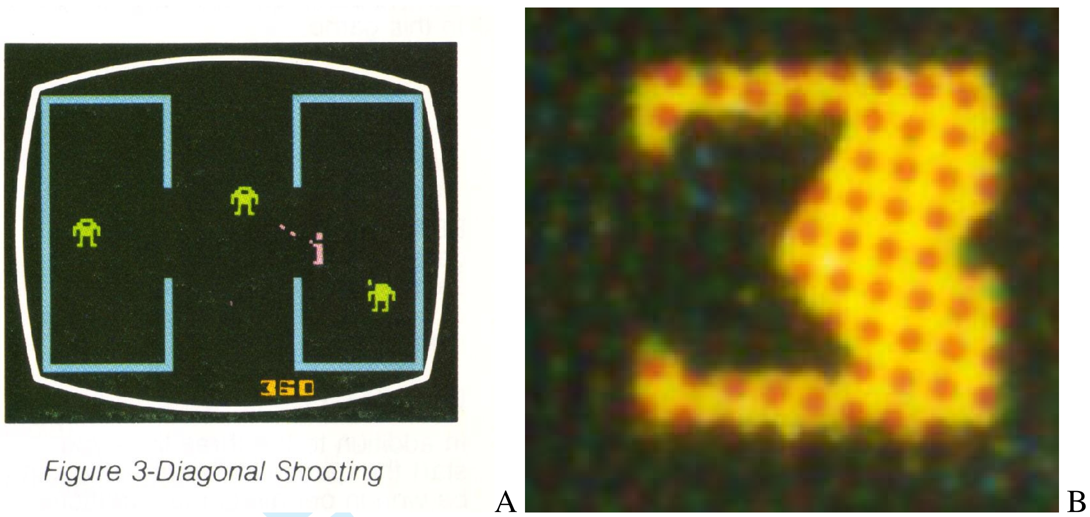


Figure 12. A print version of the same numeral 3 in *Berzerk* for Atari. A) Illustration of screen image (enlarged) from *Berzerk* for Atari instruction Manual. B) Greatly enlarged numeral 3 from 12A.

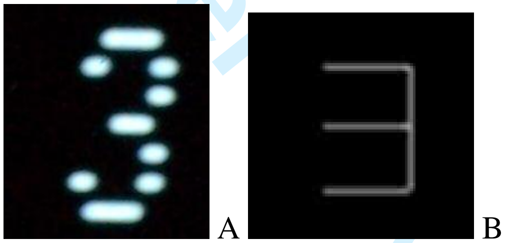
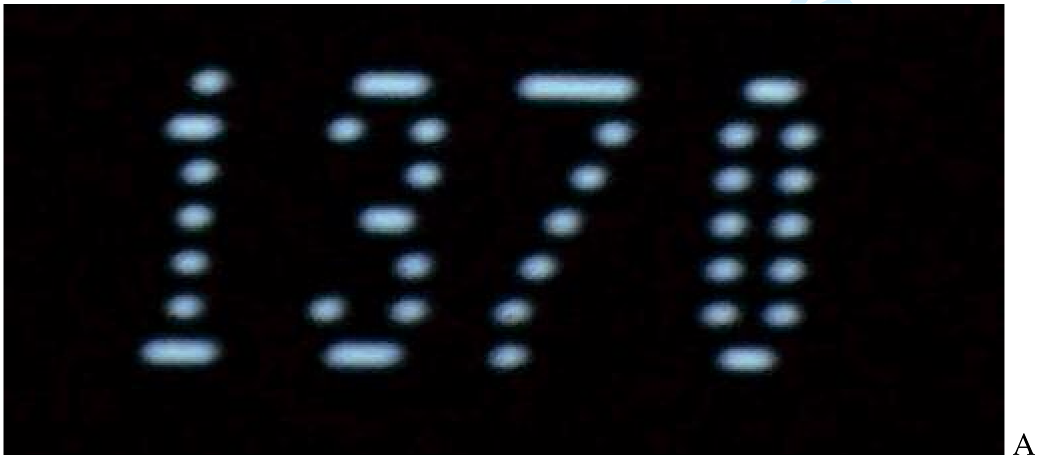
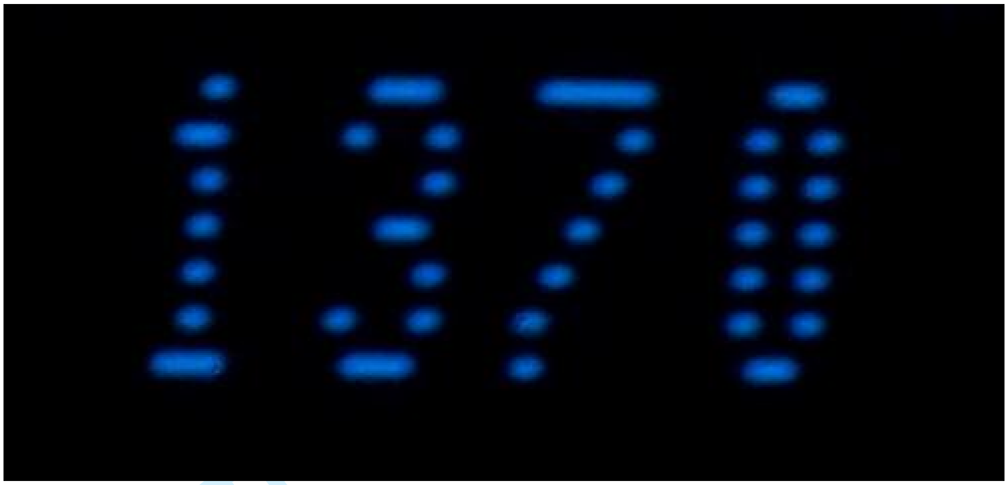


Figure 13. A) The letter B as rendered and displayed on a Vectrex (photographed with overlay). B) The letter B as displayed on an emulated Asteroids machine.



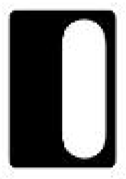
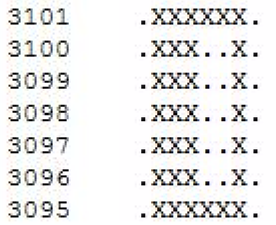
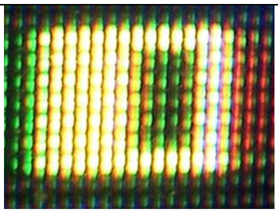
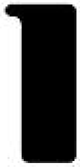
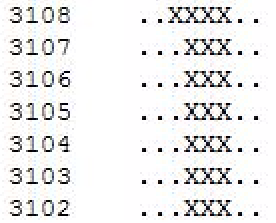
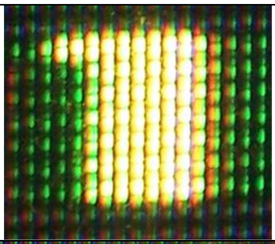
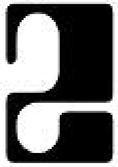
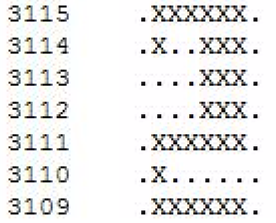
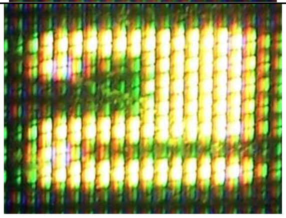
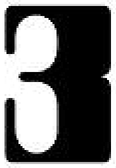
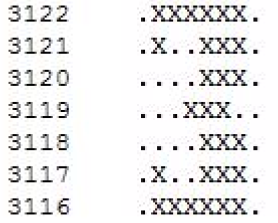
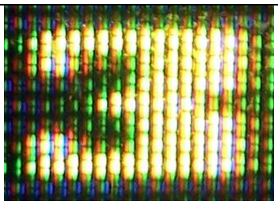
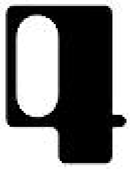
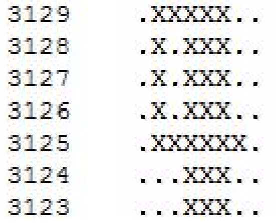
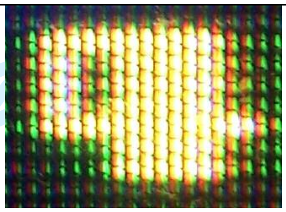
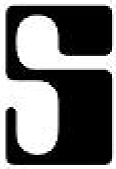
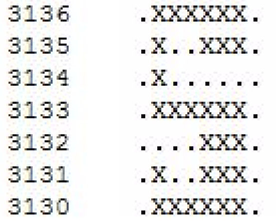
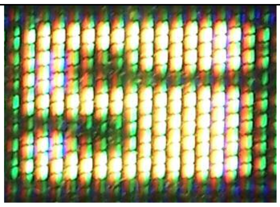


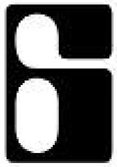
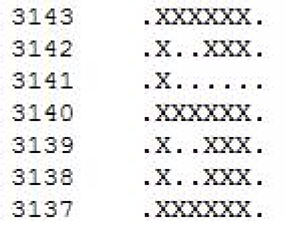
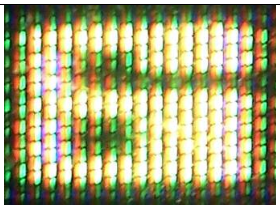
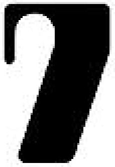
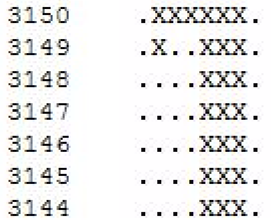
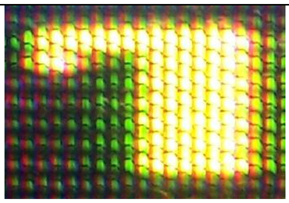
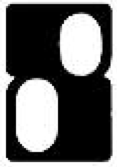
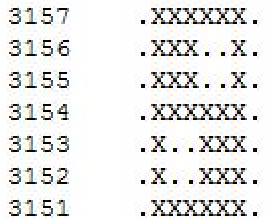
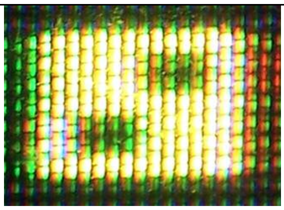
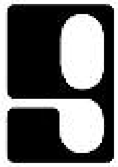
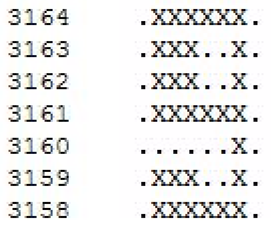
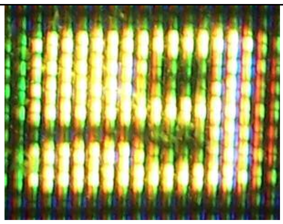
B

Figure 14. A) *Berzerk* score display photographed without overlay. B) Same score photographed with overlay.

Table

Table 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.	
	 3122 .XXXXXX. 3121 .X..XXX. 3120 ....XXX. 3119 ...XXX.. 3118 ....XXX. 3117 .X..XXX. 3116 .XXXXXX.	
	 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.	
	 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.

Notes

[1] 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, 1983).

[2] 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.).

[3] 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 which may or may not be arbitrary. I offer a broader survey of approaches to videogame textual analysis in another chapter.

[4] In this passage, Aarseth refers specifically to Juul’s article “Games Telling stories?” (2001) but his *Half-Real: Videogames between Real Rules and Fictional Worlds* (2005) addresses the topic directly and extensively.

[5] Aarseth directs readers to Eskelinen’s essay, “Towards Computer Game Studies” (2004, p. 36) in the same collection.

[6] 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 video game versions of *Berzerk*, the board game version is interesting for depicting Evil Otto with a body.

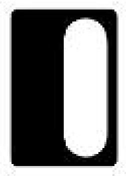
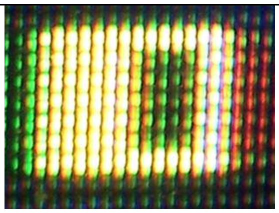
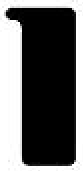
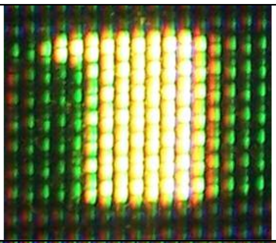
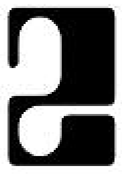
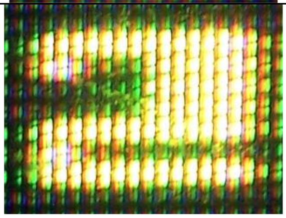
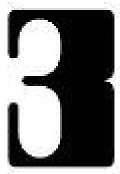
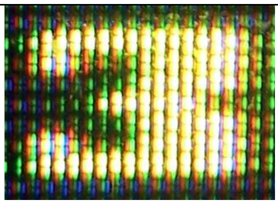
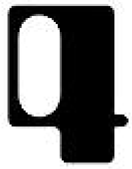
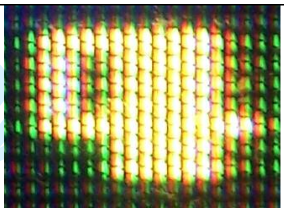
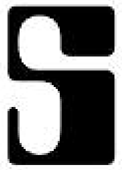
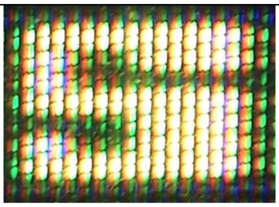
[7] Incidentally, achieving a score that contains a 3 in the first position is not that easy. In order to capture the images from Stella and the Atari, I had to reload the game until I started with an opening field of 3 robots so that, by killing them all, I received a bonus of 30 points. (The game randomly 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.

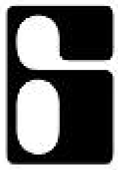
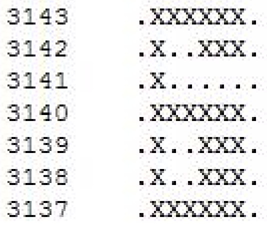
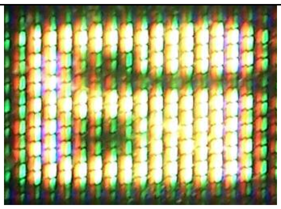
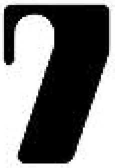
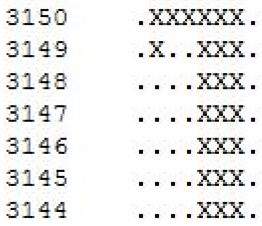
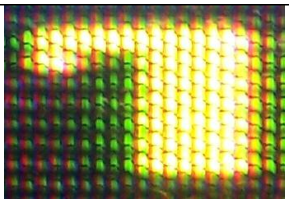
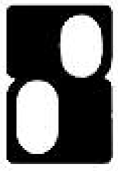
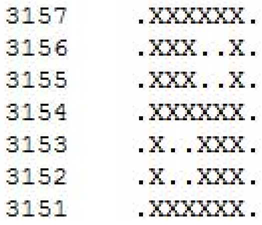
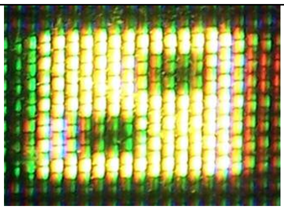
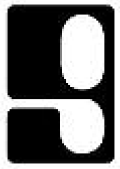
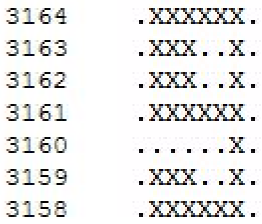
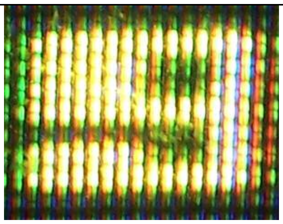
[8] An identical character set appears in *Defender* (1981), *RealSports Volleyball* (1982), and the prototypes *Bugs Bunny* and *Holey Moley*. According to the database at AtariAge.com, *Defender* and *RealSports Volleyball* were both programmed by Bob Polaro and graphically designed by Alan Murphy (“Defender (Atari)”; “RealSports Volleyball (Atari)”), suggesting that the two very different games shared a common codebase. A different implantation of a character set inspired by Countdown appears in *Demon Attack* (1982), *Marine Wars* (1983), and *Pooyan* (1982).

[9] 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>.

[10] 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.

Table 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.	
	3122 .XXXXXX. 3121 .X..XXX. 3120 ....XXX. 3119 ...XXX.. 3118 ....XXX. 3117 .X..XXX. 3116 .XXXXXX.	
	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.	
	 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.