Computing Makes the "Man": Programmer Creativity and the Platform Technology of the Atari Video Computer System

Ian Bogost[1] and Nick Montfort[2]

[1] School of Literature Communication and Culture
Georgia Institute of Technology
686 Cherry Street, Atlanta GA 30332 USA
ibogost@gatech.edu
[2] Program in Writing and Humanistic Studies
Massachusetts Institute of Technology
77 Massachusetts Avenue, Cambridge, MA 02139 USA
nickm@nickm.com

Abstract. Some of the cultural and technical forces that influenced the creation of the "man" (the player-controlled element) in two early home video games, *Pitfall!* and *Yars' Revenge,* are discussed. We find that the specific nature of the Atari Video Computer System (also known as the Atari VCS and Atari 2600) as a computing platform enables and constrains what can be done on the system, and that it also encourages and discourages certain types of creative work. For these reasons, understanding the platform is essential to understanding the development of games on this influential system and, by extension, the early history of video games.

Key words: Platform studies, Atari VCS, Atari 2600, video game development.

1 Two Cases from the Early Days of the Video Game

Creativity intersects with computing in many ways — not only when computers are used to model creativity, but also when they are used to enable and constrain the creative work of developers of digital media. We examine two cases where programmers of the Atari Video Computer System (also called the Atari VCS or Atari 2600) created a "man," an in-game representation controlled by the player, in a way that was sensitive to the capabilities of computing platform being used. The same icon would not have been created on a system with different capabilities; this helps to show how the surrounding games would also not have been developed in the same way if a different game system was being targeted. The cases considered are those of David Crane creating Pitfall Harry for the Activision game *Pitfall!* and Howard Scott Warshaw creating the fly-like Yar for the Atari game *Yars' Revenge*. They represent

some of the interesting interactions between programming and platform that we have written about in our forthcoming book, which takes a platform studies approach to the Atari VCS [1].

The Atari VCS was an extremely influential early home video game system. It was neither the first home system nor even the first cartridge-based system, but its success was so great that in the early 1980s, "Atari" became synonymous with "video game system" just as "Coke" often means "soft drink" and "Google" generically means "search engine" today. The creative work done on the system had an influence on the development of video game genres and interface conventions, among other things.

The claim in our title that "computing makes the 'man'" should be taken in only the most literal sense. It simply means that these two "man" objects, Pitfall Harry and the Yar, are generated by a particular computing system running particular programs — they do not exist without computing. This claim is not supposed to mean that the technology of a platform determines the creative output of programmers, or determines the programmers themselves, in any simplistic way. The relationship between creativity and platform is a complex one. We hope that our discussion here will shed some light on this complexity and help to show how the particularities of a computing platform are involved in the process of creating digital media.

2 Pitfall Harry

Pitfall! is an important early platformer and a predecessor to the side scroller, a form of video game which was made famous by *Super Mario Bros*. In this form, the "man" is seen from the side and typically moves from left to right as the background and structures continuously appear on the right and disappear on the left. With its side view, the ability of Pitfall Harry, the game's hero, to jump, swing, and climb and fall between different levels, and with the need to drive this character horizontally toward treasures, *Pitfall!* managed to do many of the essential things that a side scroller did even though it didn't smoothly scroll its environment.

Pitfall! arose from a combination of influences, technical and cultural. It started with the challenge of creating realistically animating graphics on the Atari VCS. The sprites in early games were static — one unmoving graphic comprises *Combat*'s planes, *Slot Racer*'s cars, even *Superman*'s human characters. *Pitfall!* creator David Crane had already experimented with simple animation to great effect in *Grand Prix*, in which the cars have wheels with tire treads that spin at different rates depending on the car's speed. But he had previously sketched out an idea for a realistically moving man. This became the basis for Pitfall Harry.

Because of the limitations of RAM, ROM, and processor cycles that were inherent to VCS programming, graphics like sprites were not considered external assets that could be dropped into a game. VCS programmers used quad-ruled paper to sketch out designs for sprites, considering not only the 8-bit wide patterns needed to render a

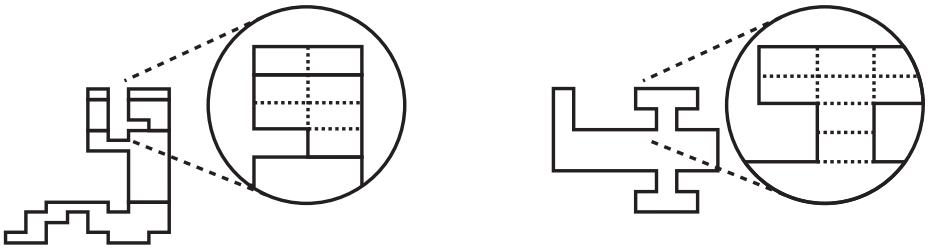

Figure 1. Pitfall Harry and the biplane from Combat are composed of the same divisions of the scan line, the wide pixels of the Atari VCS.

subject convincingly, but also how to design without changing sprite colors during a scan line and while accounting for the total size of a set of sprites in ROM. In some cases, the possible locations of a sprite on screen would dictate whether color changes were possible — for example, there might not be enough time to change sprite color and graphics values in addition to playfield graphics.

Another issue was the legibility of sprite graphics on-screen. The 8-bit width of VCS sprites do not provide a lot of room for detail, and some objects or creatures prove very difficult to render in such low resolution. Crane explained: "Early in my career at Atari, I designed a *Slot Machine* game. When I tried to draw traditional slot machine symbols — cherries, lemons, oranges, etc. — it became clear that there was no way to render those objects in 8 monochrome pixels. So I used cactus, cars and other angular objects that were easily recognizable when drawn with pixels" (Email to Bogost, 23 October 2007).

The smallest vertical unit in the VCS graphics system is a single scan line of the television screen. The smallest unit of horizontal space is a color clock, the amount of time it takes the machine's graphics chip, called Stella, to draw one color segment; Stella is configured to click away three color clocks for every cycle of the VCS's MOS Technologies 6502 processor. The detailed view of Pitfall Harry's head and neck drawn in Figure 1 on the left clearly shows that the "pixel" defined by these is rectangular, not square. This shape became a design feature; for example, the rectangular blocks help Pitfall Harry's legs appear longer than they would with square pixels. The *Combat* biplane sprite on the right appears to be comprised of square pixels because that program uses a two-line kernel, which only updates the sprite graphics every two scan lines.

The choice of the scorpion and cobra obstacles in *Pitfall!* evolved from a similar process, motivated more by how convincingly these opponents could be rendered than by any prior interest in those creatures.

Crane worked on the "little running man" animation for several months, refining its appearance and behavior. He walked deliberately around the office, trying to record his own leg and arm positions and to translate those movements onto pixel paper. However, Crane didn't do anything with the little running man right away. Each time

he finished a project, he would bring out the designs and think about a game that might make good use of it. Finally in 1982, a plan came together: "I sat down with a blank sheet of paper and drew a stick figure man in the center. I said, 'OK, I have a little running man… Let's put him on a path' (two more lines drawn on the paper). 'Where is the path?… Let's put it in a jungle' (draw some trees). 'Why is he running? … (draw treasures to collect, enemies to avoid, etc.) And *Pitfall!* was born. This entire process took about 10 minutes. About 1000 hours of programming later the game was complete.'" (Email to Bogost, 23 October 2007). The inspiration for *Pitfall!* wasn't the side-scrolling jungle adventure, but rather the running man. The adventure just gave him a reason to run.

Crane's technical innovation combined with several cultural influences to produce the "man" Pitfall Harry instead of just a generic running figure. Today, highly detailed videogame characters with complex backstories are common. Miyamoto's Jumpman (who later became Mario) and Iwatani's Pac-Man had become cultural icons before *Pitfall!* was released. But Pitfall Harry was the first popular video game character born on a home console system. He eventually spawned numerous sequels, licensed products, and even a television cartoon. The little running man was partly responsible, but the cultural references also helped to develop the game's fictional world.

The film *Raiders of the Lost Ark* was released in 1981. Crane acknowledges that the movie inspired the idea for an adventure in the jungle. But apart from that particular kind of wilderness setting and a guy who runs, little about the game resembles *Raiders*. (Howard Scott Warshaw's Atari-licensed Atari VCS *Raiders of the Lost Ark* cartridge takes considerable license with the film's character and the plot, but nevertheless has many more identifiable elements that can be read as related to the film.) Beyond the cinematic adventure of Indiana Jones, there were two important inspirations that contributed to Crane's design.

The first explains Pitfall Harry's ability to swing on a vine. This idea, of course, comes from Tarzan, the original vine-swinger, who was created by Edgar Rice Burroughs in 1912. Tarzan also inspired Taito's 1982 arcade game *Jungle Hunt*, although that game was developed independently of *Pitfall!*, with neither developer knowing about the other project. Perhaps jungle fever was in the air in that year.

The second explains the crocodiles in some of the *Pitfall!* ponds. From the 1940s through the mid-60s, Paul Terry's Terrytoons studio, best known for the character Mighty Mouse, released a theatrical cartoon series featuring two magpies named Heckle and Jeckle. The cartoons featured the typical amusing pranks, in which the two birds calmly outwitted a variety of foes. In one sequence, the two ran across the heads of crocodiles, deftly escaping their snapping jaws. Crane, who was born in the mid-1950s, remembered seeing the cartoons as a child. He speculated that this idea would make an interesting mechanic in an adventure game.

The result was interesting indeed, partly thanks to how it made the Heckle and Jeckle maneuver interactive. To the amateur player of *Pitfall!*, the screens with crocodile-filled ponds prove quite difficult. It is only possible to stand on the heads of

the crocs while their mouths are open, and a misstep lands Pitfall Harry in the water. As the player becomes more experienced, the player works up enough skill to jump quickly and deftly over the crocodiles, just like Heckle and Jeckle.

3 A Fly Named Yar

Howard Scott Warshaw's first assignment at Atari was the project that would eventually result in *Yars' Revenge*. Initially, he was to port the arcade game *Star Castle*, produced by Cinematronic, to the Atari VCS. As he told an interviewer, "I soon realized that a decent version couldn't be done, so I took what I thought were the top logical and geometric components of *Star Castle* and reorganized them in a way that would better suit the machine" [2]. Warshaw's comment reveals how the platform participates in the ecology of game development. The design of *Yars' Revenge* was not entirely determined by the Atari VCS, nor was it dropped on the platform by its programmer with no concern for how the system worked. The original idea was to imitate another game, but the capabilities and limitations of the VCS led the developer to create something different, a reorganization of *Star Castle*'s major components that recognized the differences between vector and raster graphics, exploited the abilities of the TIA, and was well-suited to home play.

It was a radical move for Atari to set aside *Star Castle*, which they had already arranged to license. An arcade game that was a hit would of course have a following already, one that might generate enthusiasm and an initial market. Even one that wasn't a huge success still contained a complete and fully implemented game design, one which had been tested on the playing (and paying) public.

Ironically, however, the hardware capabilities of an arcade machine — in terms of processing power, graphics, and controller setup — were always significantly different from those of the Atari VCS, so that having a well-tested and implemented game design implemented on an arcade platform didn't mean very much when it came to the home console's hardware. The most obvious difference between the underlying *Star Castle* computing system and the VCS was the arcade machine's vector graphics, which Atari called XY graphics. Atari's successful arcade games *Tempest*, *Battlezone*, *Asteroids*, and *Lunar Lander* all use this sort of graphics system, which employs a fundamentally different type of monitor. All early arcade games used a CRT, but the ones in vector graphics games are wired differently than are the ones in standard televisions. The electron beam does not sweep across the screen from left to right, being turned on and off as it makes its way down and then returns back to the top 60 times a second. Instead, the electron gun is pointed at a certain location, turned on, and moved from that (x, y) coordinate to another point in a straight line, where it is turned off again. (An oscilloscope functions in a similar way; it just uses a different method of deflecting the electron beam.) Because the beam can be made to move arbitrarily instead of progressively scanning along the screen, this way of

illuminating a phosphor-coated screen was also called "random scan."

Cinematronics was the first company to release an arcade game that used this display technology. The game was *Space Wars*, released in 1977. Like the first raster arcade game, Nolan Bushnell's pre-Atari *Computer Space*, it was a two-player game and an arcade implementation of the 1962 PDP–1 program *Spacewar*. *Star Castle* has significantly different gameplay, and was developed by a different person, but the space setting and control scheme show that it is clearly based on Cinematronics' earlier game *Space Wars*.

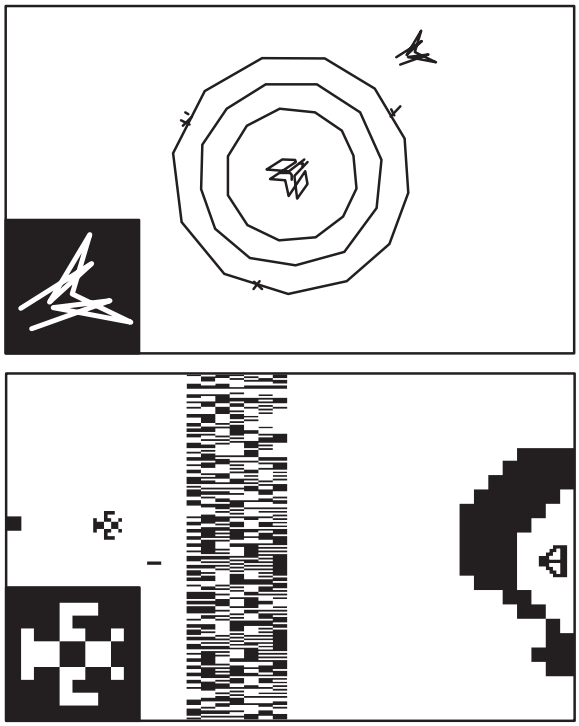

Figure 2. *Star Castle,* the vector-graphics arcade game shown at top, inspired *Yars' Revenge,* at bottom, which was made for the raster graphics of the Atari VCS. The *Star Castle* ship and *Yars' Revenge* "fly simulator," the Yar, are shown in detail in the lower left of each screen.

Vector graphics have certain advantages over raster graphics; at least, they did in the late 1970s and early 1980s. Specifically, it is much easier on a vector system to rotate shapes and to scale them up or down, as is needed when zooming in or out. For instance, to resize an object that is centered on (0,0) and consists of some set of lines, each endpoint's coordinates can simply be multiplied by the scaling factor. It does not get much trickier when the object is centered elsewhere or when there should be different scaling along the horizontal and vertical axes. It is also straightforward to shear a shape, keeping one axis constant while shifting the other. Rotation, scaling, and shear are the basic linear transformations, and any one can be accomplished by subjecting the coordinates of a shape's points to a single matrix multiplication. In combination with translation (the displacement of the shape in space, which just requires addition of the amount of the displacement), these allow for all the possible transformations that keep straight lines straight.

In a raster graphics system, particularly one with the limited computing power of the Atari VCS, the only feasible way to show a rotation of an object is to display a

different bitmap, a hard-coded image of the shape rotated by the desired amount. This is how tanks and planes in *Combat* are rotated, for instance. A simple form of scaling is supported in hardware, via the TIA's number-size registers, but smoother zooming has to be done by providing additional bitmap graphics. Even when the display hardware itself is not of the XY graphics sort, these benefits of vector graphics can be seen when comparing systems such as an early version of the bitmap-based Macromedia's *Director* and an early version of that company's vector-graphics *Flash* environment.

The object of *Star Castle* is to destroy the rotating cannon in the center of the screen repeatedly, with one's triangular, rotating ship, a vessel that looks and moves like the ones in *Asteroids* and *Space Wars*. The enemy cannon appears behind colored overlays and is surrounded by three concentric, rotating shields, each of which is made of line segments. The segments can be destroyed by fire from the player's ship, but whenever an entire ring is shot away, it regenerates. Whenever the player clears a path to the cannon, creating a chance to shoot at it to destroy it, the canon fires a large missile toward the player's ship. As the player is trying to break down the shield rings, three mines also move out and seek the player's ship. They can be avoided or shot, although shooting them does not increase the score and they soon reappear. After a central cannon is finally successfully destroyed, another one quickly appears with three intact rings around it.

In *Yars' Revenge*, the player's "ship" or "man" is the Yar, a "fly simulator" that is controlled with the joystick. *Yars' Revenge* does not have a ship that pivots clockwise or counterclockwise, something that was fairly easily implemented on the vector-graphics display system of *Star Castle*. Instead, the joystick is used to directly move the Yar in the standard eight directions — up, down, left, right, or diagonally. This is a form of movement that was fairly easy for the Atari VCS: translation while facing in one of eight directions. This type of mapping, a form of direct manipulation, wasn't exactly native to the system, though. Early games, including *Combat,* often used methods of joystick control that now seem far less obvious. The VCS cartridge *Asteroids,* a port of an arcade game that featured a rotating ship, used a rotate-and-thrust control scheme.

The Yar sprite is animated, requiring an additional frame for each direction, but its appearance facing right is a reflection of what it looks like facing left, allowing for some savings. Up/down reflection is not as straightforward as left/right reflection; the latter can be accomplished with a single write to a register on the VCS's graphics and sound interface board, Stella, while the former requires manually loading and storing new bitmap data. For this reason, the Yar sprites for up and down are both laid out in ROM. Switching between the two requires reading a different bitmap. The insect appearance of the Yar was simply based on what Warshaw could draw and animate in an interesting way within a player sprite. The name "Yar" has a more definite referent — it was devised by spelling Atari CEO Ray Kassar's first name backwards.

4 Standing on the Shoulders of Platforms

By choosing a platform, new media creators simplify development and delivery in many ways. Their work is supported and constrained by what this platform can do. Sometimes the influence is obvious: A monochrome platform can't display color, a video game console without a keyboard can't accept typed input. But there are many more subtle ways that platforms interact with creative production, due to the idioms of programming that a language supports or due to transistor-level decisions made in video and audio hardware. In addition to allowing certain developments and precluding others, platforms also encourage and discourage different sorts of expressive new media work with much more subtlety. In the case of *Pitfall!* and *Yars' Revenge*, the appearance, function, and behavior of each game's "man" was influenced by both cultural and technical factors. Pitfall Harry realized David Crane's attempt to realistically animate a running man. Crane used the platform's technical capabilities with virtuoso skill while also drawing on popular culture to create the character's context and program his behavior. The Yar was Warshaw's deft adaptation of a spaceship from a coin-op game with a totally different graphics system. The resulting figure went native on the home system, becoming an iconic character for the Atari VCS.

Particular platform studies may emphasize different technical or cultural aspects and draw on different critical and theoretical approaches, but to deal deeply with platforms and new media, these sorts of studies will all have to be technically rigorous. We hope our book-length consideration of the Atari VCS will be the first of many such studies. We have invited proposals for other books in the new Platform Studies series of The MIT Press, at <http://platformstudies.org>. We invite the consideration of all sorts of computing platforms, including not only video game consoles but also home computers of all sorts; important software systems that have been treated as platforms, from BASIC through HyperCard to Java; and minicomputer and mainframe systems such as PLATO and the PDP-10. The detailed analysis of hardware and code can connect to the experience of developers who created software for a platform and users who interacted with and will interact with programs on that platform. An approach that looks at a particular platform, or a closely related family of platforms, through time and in a variety of contexts of use is especially well-suited to uncovering new insights about digital media. The deep investigation of computing systems has the potential to reveal the interactions between platforms and creativity, design, expression, and culture.

Creators need to understand how the medium they work in produces meaning, aesthetics, and experiences. The same is true for the critic, who must be able, among other things, to grasp how an artifact signifies in relation to how it was created. Such principles apply to all expressive media, from paint to language to computation. Since the conventions and forms of media develop historically, it becomes increasingly important to understand the origins, trends, and alterations of creative expression. For

example, the modern poet benefits from an understanding of meter, which developed and changed throughout the millennia. Computational critics and creators should also be concerned with how form and material influence the way work is produced and received. Platform studies offers one inroad into such a practice, by focusing on the ways a platform's hardware and software interacts with processes of computational creation and reception.

References

[1] Montfort, N. & Bogost, I (2009). *Video Computer System: The Atari 2600 Platform*. Cambridge: MIT Press.

[2] Stilphen, S.. (2005). "Interview with Howard Scott Warshaw." April 23. <http://www.digitpress.com/archives/interview_warshaw.htm>.

Video Games Cited

Atari. (1979). *Asteroids*. Arcade. Developed by E. Logg and L. Rains.
Atari. (1981). *Asteroids*. Atari VCS. Programmed by B. Stewart.
Atari. (1980). *Battlezone*. Arcade. Programmed by E. Rotberg.
Atari. (1987). *Breakout*. Atari VCS. Programmed by B. Stewart.
Atari. (1977). *Combat*. Atari VCS. Programmed by J. Decuir and L. Wagner.
Atari. (1978). *Slot Racer*. Atari VCS. Programmed by W. Robinett.
Atari. (1979). *Superman*. Atari VCS. Programmed by J. Dunn.
Atari. (1980). *Tempest*. Arcade. Programmed by D. Theurer.
Atari. (1979). *Lunar Lander*. Arcade.
Atari. (1982). *Raiders of the Lost Ark*. Atari VCS. Programmed by H. S. Warshaw.
Atari. (1981). *Yars' Revenge*. Atari VCS. Programmed by H. S. Warshaw.
Bushnell, N. (1971). *Computer Space*. Arcade. Distributed by Nutting Associates.
Cinematronics. (1977). *Star Castle*. Arcade.
Cinematronics. (1977). *Space Wars*. Arcade.
Crane, D. (1982). *Pitfall!* Atari VCS. Activision,
Crane, D. (1982). *Grand Prix*. Atari VCS. Activision,
Nintendo. (1985). *Super Mario Bros*. NES. Designed by S. Miyamoto.
Taito. (1982). *Jungle Hunt*. Distributed by Midway.