



Project  
**MUSE**<sup>®</sup>

*Today's Research. Tomorrow's Inspiration.*

---

## Random and Raster: Display Technologies and the Development of Videogames

Nick Montfort  
Ian Bogost

IEEE Annals of the History of Computing, Volume 31, Number 3, July-September 2009, pp. 34-43 (Article)

Published by IEEE Computer Society



For additional information about this article

<http://muse.jhu.edu/journals/ahc/summary/v031/31.3.montfort.html>

# Random and Raster: Display Technologies and the Development of Videogames

Nick Montfort

*Massachusetts Institute of Technology*

Ian Bogost

*Georgia Institute of Technology*

Videogame developers have utilized many types of display technology, from oscilloscopes to Teletypes to high-definition LCD displays. Two significant early display technologies, raster scan and random scan CRTs, played a significant part in the history and evolution of videogames. A study of these technologies shows how the choice of one or the other, and the need to port games between the two, influenced game design and prompted developers to innovate.

---

Many forces have shaped the development of videogame displays and influenced the design and functionality of games, not just their appearance. The videogame display technologies that were put to use, and the extent to which they were used, had a great deal to do with convenience and the ability to source display components. Since early videogame implementations, resolutions have increased, bulky video picture tubes have given way to widescreen flat-panel LCD displays, and increased computing power has allowed real-time 3D graphics to replace 80-column text displays. Conventional wisdom might assume that such changes amount to simple technological improvements.

Looking back over the history of gaming, however, reveals that **different display technologies have exerted creative force on the development of specific videogames**. During the last half-century, very different display technologies have provided the “video” in videogames. For example, the very early game *Tennis for Two* used an oscilloscope, and early text adventures played on minicomputers often used a Teletype or other print terminal to print output on paper. Sometimes the display technology was imposed by the game play. For example, videogames played in the home typically can’t rely on expensive or unusual displays, so an ordinary television is usually the most sensible display choice. But a

programmer writing a game for a PDP-1 would have assumed that a text-output video terminal or Teletype machine would be ready at hand, and a coin-operated (coin-op) arcade game creator might have some flexibility in choosing a particular type or orientation of monitor in the cabinet. No matter the type, display technologies have exerted a strong influence on the evolution of videogames.

From the end of the 1970s through the 1980s, two fundamentally different display technologies were in use side by side in arcades. The **raster-scan display**, which worked similarly to the pre-flat-screen TV (an electron beam swept across and down the screen to draw the image), was used in *Space Invaders*, *Berzerk*, *Pac-Man*, *Donkey Kong*, *Robotron: 2084*, and many other coin-ops. Other systems used **random-scan displays** (also called **vector graphics**, stroke writing, and calligraphic), which arbitrarily pointed an electron gun. Atari used this technology, calling it XY graphics, when it developed *Tempest*, *Battlezone*, *Asteroids*, and *Lunar Lander*. (See the sidebar for more details about specific games.)

In this article, we describe some of the interaction between early videogame development and display technologies, **considering the influence of XY graphics in contrast to the more familiar raster-scan system**. In the 1970s, both random and raster displays were commonly used in many applications,

## Further Reading on Videogame History

We have not individually cited each well-known and well-documented historical fact (such as the release years of popular videogames) because such detail is outside the scope of this article. For general information on videogame history, we suggest several books.<sup>1–5</sup> There are also helpful resources easily found online, such as the longstanding Killer List of Videogames (<http://www.klov.com>) and the project MobyGames (<http://www.mobygames.com>). Information about the basic functioning of games can be easily determined by playing the games themselves.

Although certain information on the history of videogames, both online and in print, might be less than reliable, basic information about which companies and programmers developed popular games is not usually in dispute.

## References

1. V. Burnham, *Supercade: A Visual History of the Videogame Age 1971–1984*, MIT Press, 2003.
2. R. DeMaria and J. Wilson, *High Score!: The Illustrated History of Electronic Games*, Osborne/McGraw-Hill, 2002.
3. W. Forster, *The Encyclopedia of Game Machines*, Gameplan, 2005.
4. S.L. Kent, *The Ultimate History of Video Games*, Prima, 2001.
5. B. Weiss, *Classic Home Video Games, 1972–1984*, McFarland, 2007.

and XY graphics systems presented themselves as viable candidates for adoption in coin-op games like *Asteroids* and *Star Castle*. Often, random-scan types of displays offered a kind of high-resolution graphics unavailable in raster displays—for example, by smoothly scaling and rotating graphics in a way that would otherwise have been difficult, if not impossible.

Gaming brought both displays to a broad public and into contact with one another, along with new constraints. Differences between the two systems meant that developers porting popular random-scan coin-op titles were forced to deliberately evaluate game elements, sometimes devising entirely new approaches to similar high-level game concepts. When games that originally used vector displays were ported to raster-based home systems, the practical differences between the display technologies were revealed in a particularly interesting way—for example, this is the case with the home version of *Asteroids*, developed for the Atari VCS (a.k.a. Atari 2600), and the Atari VCS game *Yars' Revenge*, which was inspired by *Star Castle*. By looking at *Yars' Revenge* and *Asteroids* in detail, we can see how developers modified the games to account for the different display systems. Similarly, the home and arcade versions of *Battlezone* and *Tempest* show how entirely new ways of depicting the world on screen came about.

### Random and raster display technologies

Although all early arcade games used CRTs, those in vector-graphics games are wired differently than the ones in standard TVs.

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 each 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 similarly; it just deflects the electron beam using electrostatic force. XY monitors use magnetic force.

Just as a storage device that is arbitrarily accessible offers random access, random scan uses a beam that can be made to move anywhere. Such systems directly draw lines in an  $(x, y)$  coordinate system. Raster-graphics systems, on the other hand, draw a grid of data representing a pattern of pixels. If the correspondence between bits and pixels is one to one, the data is a bitmap; more generally, such data is a pixmap (p. 13).<sup>1</sup> As a result, it is possible to render detailed bitmapped images on a raster system, but a vector system makes it much easier to scale, shear, and rotate shapes that have been drawn with the primitives of that system.

In the 1980s, other differences between the two types of displays included the accessibility of color and black-and-white raster displays, the better brightness and resolution of vector displays, and the limited number of primitives that could be drawn on a vector display before it began to flicker. Videogame systems were not the first to explore the uses of either type of display. Both had been employed in computer graphics for a while before commercial videogaming came about, and by the mid-1970s, raster graphics had become dominant (p. 12).<sup>2</sup>

### The building blocks of *Star Castle*

Programmer Howard Scott Warshaw's first assignment at Atari was to port the 1980 vector-graphics arcade game *Star Castle*, produced by Cinematronics, to the Atari VCS. 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."<sup>3</sup>

Warshaw's comment reveals how the platform participates in the ecology of game development. The design of the resulting game, *Yars' Revenge* (1981), 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 Atari VCS led the developer to create something different. Specifically, the result was a reorganization of *Star Castle's* major components that recognized the differences between vector and raster graphics, exploited the abilities of the Atari VCS graphics system, and was well-suited to home play.

The Atari VCS has a graphics system that was unusual even for the time. To keep costs down, the unit lacked a frame buffer, a memory region in which the screen image could be set up and blitted across automatically. Instead, programmers had to write code that would set up each individual scan line for the television interface adapter (TIA). The registers for two player sprites, two one-pixel missiles, one one-pixel ball, and the playfield had to be ready on each line of the screen so the electron beam would have the appropriate objects ready to draw. For this reason, the challenge of programming the Atari VCS was sometimes referred to as "facing the beam."

Warshaw explained in another interview how radical it was for Atari to drop *Star Castle*, which the company had already arranged to license:

I did something no one else had ever done, I went to my boss and said that I had an idea for an original game that would use the same basic play principles of *Star Castle* but was designed to fit the VCS hardware so it wouldn't suck. And to their credit, they let me go with it. Think about that. They blew off a license to let me pursue an original concept with the promise of making a better game for the system. That would never happen today.<sup>4</sup>

A hit arcade game would have already had a following, one that might have generated enthusiasm and an initial market. Even in the early days of videogaming, when the arcade hits were *Pong* and *Tank* rather than *Pac-Man*, an arcade game's fan following was important when it came to the home console market. Given the differences in platform and the need to rewrite each game from scratch, a port to the Atari VCS had plenty of room to become a fairly involved adaptation rather than a straightforward re-creation. This could leave some arcade game fans disappointed, as happened with *Pac-Man*.

Cinematronics, the developer of *Star Castle*, was the first company to release an arcade game that used the XY graphics display technology: *Space Wars* (1977). Like the first raster arcade game, Nolan Bushnell's pre-Atari *Computer Space*, it was a two-player arcade implementation of the 1962 PDP-1 program *Spacewar*.<sup>2</sup> *Star Castle* had significantly different game play and was developed by a different person, but the space setting and control scheme showed a clear connection to *Space Wars*.<sup>5</sup>

Vector-graphics systems have certain advantages over raster-graphics systems—at least, they did in the late 1970s and early 1980s. Specifically, it is easier to rotate shapes and scale them up or down on a vector system, as is necessary 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 amounts of scaling in 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 all 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 only requires adding the distance to be translated), these allow for all the possible transformations that keep straight lines straight.

In a raster-graphics system, the only feasible way to show an object's rotation 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 the Atari VCS game *Combat* are rotated, for instance. A simple form of scaling is supported in

hardware, via the TIA's specialized number-size registers, but smoother zooming has to be done with new graphics. Even when XY graphics hardware is not used, these benefits of vector graphics can be seen when comparing software platforms, such as an early version of the bitmap-based Macromedia Director and an early version of Macromedia's vector-graphics Flash environment.

Vector-graphics games used the special capabilities of the display system to good effect. *Battlezone* is a classic example of a game that uses the scaling capability to make tanks larger, so that they appear to be approaching. *Asteroids*, on the other hand, makes more prominent use of the rotation facility.

### The smash hit *Asteroids*

The original *Asteroids* arcade game, developed by Ed Logg and based on a concept by Lyle Rains, was released in 1979. Other hit games that year were Atari's first vector-graphics game, *Lunar Lander*, and Namco's hit game *Galaxian*. The arcade *Asteroids* is interesting to compare to its Atari VCS port, which came out in 1981, the same year that *Yars' Revenge* came out for that console. This comparison demonstrates how a game can have a different appearance and function when it is created for a raster instead of a vector display.

Although the ship can move freely about the screen in *Asteroids*, the game is organized radially around the ship, which must be the player's center of attention and which can rotate and fire shots in all directions. With its space setting, its rotating and firing ship, and its controls, *Asteroids* also shares several fundamental qualities with *Spacewar*, *Space Wars*, and *Star Castle*.

Brad Stewart, who had earlier ported *Breakout* to the Atari VCS after winning the right to port the game by besting a fellow programmer in the arcade *Breakout*, went on to develop the VCS *Asteroids*. Both the VCS game and the coin-op game purport to be *Asteroids*, and they feature a rotating ship, rocks breaking up into smaller rocks, and flying saucer enemies. The arcade game uses the capabilities of the vector display beautifully and works well in the situation of the arcade, while the cartridge exhibits classic VCS qualities that connect to home gaming traditions and the TIA's affordances.

Players use five buttons to control the arcade *Asteroids*: a pair for rotating clockwise or counterclockwise, a pair to control thrust

---

## *Asteroids'* arcade original and Atari VCS port reveal how different display technologies predispose designers to develop different sorts of games.

---

and fire, and a hyperspace button that is set below the others—exactly the same five buttons as on *Space Wars*. *Star Castle* had the first four controls, but not the hyperspace button. In *Asteroids*, the monitor is oriented horizontally and the playing field's topology is that of a torus: objects wrap around vertically and horizontally. This is the same as in *Star Castle* and is one option that can be selected in *Space Wars*.

One difference between *Star Castle* and the arcade *Asteroids* is the color overlays, which place the central cannon in a circle of yellow and the protective shield around it in a ring of red. (Color overlays were not used exclusively on XY games; *Space Invaders* cabinets had them, too.) *Asteroids* lacked such colored transparencies. The game originally used a monochrome monitor, the Electrohome G05-801, a 19" monitor made by Wells-Gardner Electronics and previously used in *Lunar Lander* (p. 9).<sup>6</sup> Without overlays, an image on this monitor appears as white lines on black. The asteroids are drawn more dimly, and move about in arbitrary directions. The ship and flying saucers are drawn more brilliantly, and the shots fired from either are particularly luminous points. These aspects—along with high contrast, sharp lines, and a fast refresh rate—make *Asteroids* visually impressive in a way that is hard to imagine when looking at an emulator, screen shot, or diagram.

A player can immediately recognize the VCS *Asteroids* as a version of the arcade game. It boasts similar game play, visual appearance, and sound. The minor differences are telling, however. The asteroids are filled-in and sometimes flickering masses that are drawn in different colors rather than as monochrome outlines—assuming that the

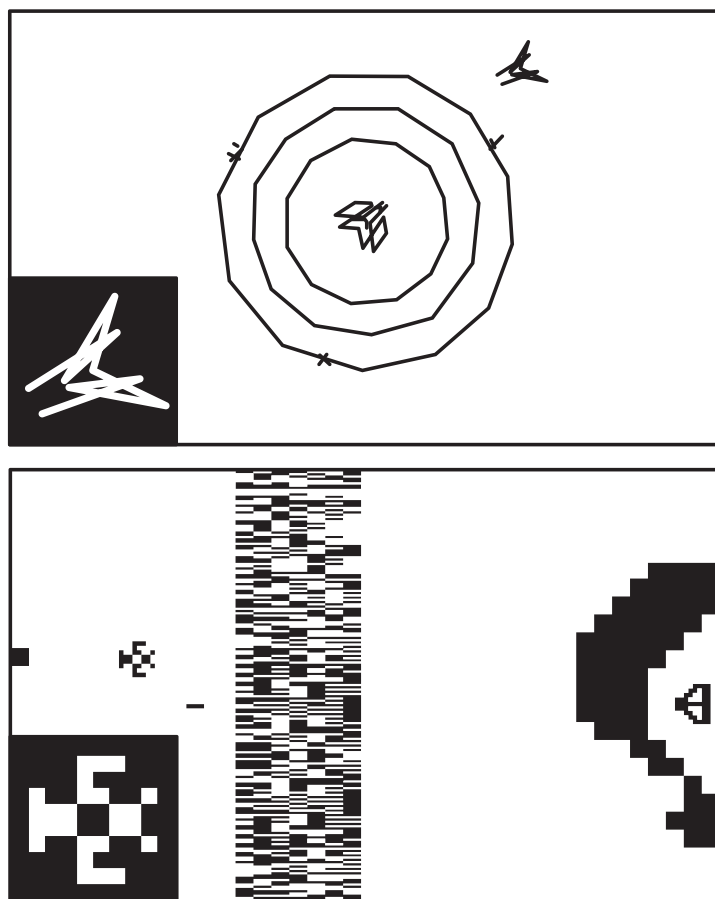


Figure 1. Screenshots from (top) *Star Castle* and (bottom) *Yars' Revenge*. The insets show details of the two ships and qualities of the different display technologies used.

Atari VCS is set to color and a color TV is used as a display. The ship is also solid, and it fires at most two shots at once—not the four that are possible in the arcade game. Although emulators have been able to make the VCS *Asteroids* conveniently available on computers, they use sharp, blocky pixels that were not seen in the same way by players in the early 1980s. CRT televisions blur pixels together resulting in something that was clearly different from a vector-graphics display, but which is also softer and fuzzier than an LCD display.<sup>7</sup>

The two versions of *Asteroids*, the arcade original and the Atari VCS port, reveal how different display technologies predispose designers to develop different sorts of games. Visually, the solid, colored, blocky rocks of the VCS version show that the TV is capable of color and that it is at least as easy to make a bitmapped sprite solid as to make it appear in outline—easier, actually. The different jags of the arcade *Asteroids*, defined only by lines, reveal what a vector-graphics system

can do in a straightforward way. The motion of the asteroids is also telling. On the random-scan display, their vectors of movement are random—any direction is possible. On the left-to-right sweeping raster display of the TV, there is a predisposition to up-or-down motion because the asteroids cannot move horizontally.

### From a port to *Yars' Revenge*

The object of *Star Castle* is to destroy the rotating cannon in the center of the screen repeatedly, using a triangular, rotating ship 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 cannon 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. *Star Castle* fared well among the several huge arcade hits released in 1980 (including *Pac-Man*, *Defender*, *Centipede*, and *Missile Command*), though it didn't attain as much fame.

In the Atari VCS game inspired by *Star Castle*, *Yars' Revenge*, the player's "ship" is the Yar, a "fly simulator" that the player controls with a joystick. The game replaces the pivoting of the ship about a point, which could easily be done by *Star Castle*'s vector-graphics display system, with movement in the standard eight directions—up, down, left, right, and diagonally. The latter form of movement was fairly easy for the Atari VCS: translation while facing in one of eight directions. 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 some savings. Because up/down reflection is not as straightforward as left/right reflection on the Atari VCS, the upward-facing and downward-facing Yar sprites are both laid out in ROM. Switching between the two requires reading a different bitmap. The insect appearance of the Yar (see Figure 1) was simply based on what Warshaw

could draw and animate in an appealing way within a player sprite. The name Yar was devised by spelling Atari CEO Ray Kassar's first name backwards.

The objective in *Yars' Revenge* is the Qotile, which moves up and down along the right side of the screen and is protected by a shield. All the levels are similar in form, but the first one (and all the odd-numbered levels) have a stationary, rounded shield around the Qotile, while the other levels (all the even-numbered ones) feature a block of shield with pieces that move left to right, down a space, right to left, down a space, and then right to left again through the block. The motion of the pieces mimics that of the CRT's electron gun as it sweeps across and back while moving down the screen.

Beyond the basic redesign of the game play and the graphical advances that were made, there were a few other innovations—minor, but telling—in *Yars' Revenge*. Some of these, such as the addition of a rudimentary pause command, show how game software could allow a fixed hardware platform to evolve and suit the needs of home players. Others demonstrate how some of the luminous effects of a vector display game were adapted, but not directly translated, into a raster-graphics display game.

*Yars' Revenge* features a large, four-part display routine that includes one part to draw the swelling explosion that ends a level; another to display the score; another to draw the shield for half the frames of the main sequence; and a final one to draw, in alternating frames during the main sequence, the multicolored neutral zone. Although the neutral zone and shield bears little resemblance to *Star Castle*'s spinning, concentric shapes, they create a striking effect that makes use of the console's different video system.

The neutral zone's random-looking patterns are not provided by a pseudorandom number generator—an intricate algorithm that, although deterministic, is complex enough to create a sequence that looks random. Such an algorithm can be implemented on the Atari VCS and was used in Activision's *Pitfall*, but this programming solution exacts a price in ROM (the code itself must be stored somewhere) and in cycles (the code must be run while the game is also carrying out the work of drawing the screen and updating the game state). The alternative is to lay out a random-looking pattern in ROM and simply load random-seeming bytes from this

small entropy pool, one adequate to create an appropriately irregular visual display. Such an approach takes fewer cycles, but it requires that some random-looking pattern, perhaps a fairly large one, be stored in ROM. If such a pattern were to be added, something else in the game would have had to be removed, and there was already less than a handful of bytes free in the finished *Yars' Revenge*.

Warshaw used the second technique, but he made use of a random-looking sequence of bytes that would already be laid out in ROM by the time the game was finished—the game's code itself—as shown in this 6502 assembly code obtained by disassembling the contents of cartridge ROM:<sup>8</sup>

```
EOR (neutralZonePtr),Y
AND neutralZoneMask
STA PF2
AND #$F7
STA COLUPF
```

In this part of the neutral zone kernel (the first instruction is located at \$F084), the values pointed to by `neutralZonePtr` are brought into the accumulator and masked against the contents of `neutralZoneMask`. This accumulator value is used first as the pattern that is loaded into a playfield register and, after it is masked again, as the playfield color. The label `neutralZonePtr` points to the same address as does another label, `gameTimer`. At this location, a count is stored that is continually incremented, once each line, and ranges over the addresses in cartridge ROM. This progression works its way through the code of the *Yars' Revenge* cartridge, with each byte of code being loaded, transformed, and displayed on the screen. The bytes in ROM end up being used in three contexts: as executable code, playfield graphics, and the playfield color. When the Qotile is hit, those bytes also supply the random-looking arrangement of the full-screen explosion. These effects certainly do not come from *Star Castle*, and there is no obvious way they could be duplicated on a vector-graphics display.

Warshaw has explained that he had planned to use this method from the start rather than working it out as a solution to a problem that came up: "It was just a cheap way to get the effect I wanted. I didn't have the time or space to do it any other way."<sup>5</sup> Still, by making the game's code into an important visual component in this raster-specific way, Warshaw showed how a functioning program could shine aesthetically. When players

---

**Vector art paved the way for modern 3D graphics, suggesting a way to represent solid objects in 3D space and animate them in real time.**

---

look at the neutral zone on the screen, they are literally looking at the code. *Yars' Revenge* may not have had any direct influence on the movie *Tron*, which brilliantly played with vector- and raster-graphics conventions, but that film was released the summer after Warsaw's game and features a multicolored, whirling master control program (MCP) that is cast in a different light by the neutral zone, actually drawn again and again by its own image, a code-and-data Janus.

Warsaw's "original" was a brilliant variation on *Star Castle*, played in virtuoso style on the Atari VCS. His skill in creating *Yars' Revenge* is not just seen in his programming chops and creativity, but also in his ability to innovate and improvise while building down from the top (an already completed vector-graphics coin-op game with a working game mechanic) and up from the bottom (a platform that offered a particular set of affordances and was used in the context of the home). While *Asteroids* shows that even a fairly direct arcade port encountered differences in display technologies, Warsaw's work exemplifies another way of navigating between display technologies and letting the capabilities of the display suggest design directions.

### Zooming through *Battlezone*

Ed Rotberg designed Atari's tank game *Battlezone* (1980) while he was working on another vector-graphics game, *Red Baron*, which let the player fly a biplane and was released soon after *Battlezone*. Although the plane game offered more degrees of freedom, it didn't match *Battlezone's* popularity as the first 3D game to reach a mass audience. At the time, tank games were not new; the early, raster-graphics arcade hit *Tank*, produced by Kee Games, offered one-on-one

tank fighting and set a precedent for *Battlezone*. It was also the inspiration for *Combat*, the tank game that came with the Atari VCS.<sup>9</sup>

Earlier, in 1974, two 3D networked games had been developed. *Maze* was developed by Greg Thompson, Dave Lebling, and others at MIT and based on work done at NASA Ames. That year, *Spasim* was also developed by Jim Bowery at the University of Illinois Urbana-Champaign. The former ran on an Imlac terminal with a vector-graphics display, and the latter used the PLATO-IV's plasma display panel. While impressive, these games required tremendous computing resources and did not reach a large number of players.

Although it had these illustrious ancestors, *Battlezone* was revolutionary in many ways. The game brought together 2D tank motion (along the flat plane of the ground) with a first-person view onto a 3D world. *Battlezone* not only featured an unusual perspective, it also had an unusual periscope cabinet configuration that let players (if they were tall enough) place their heads directly against a viewport while others could watch from either side. Besides the essential elements (3D opponents and sound effects), the landscape details included an erupting volcano.

*Battlezone* found popularity with the military, which was uncommon for the time. The US Army contracted with Atari to develop a modified game to train Bradley Infantry Fighting Vehicle gunners. This project was controversial inside Atari; apparently only two prototypes and a few units were made. Beyond this application, *Battlezone* also entranced a generation of players, some of whom went on to develop other 3D, first-person games. One young player who particularly enjoyed *Battlezone* was John Carmack, who went on to become the lead programmer at id Software, the company that pioneered the first-person shooter genre with *Wolfenstein 3-D* (1992), *Doom* (1993), and *Quake* (1996) (p. 20).<sup>10</sup>

Adaptations of *Battlezone* to the Atari VCS again reveal how game developers used raster affordances differently. Atari's port, also titled *Battlezone*, is an adequate game, but the 3D tank game that was developed by Activision, *Robot Tank* (1983), added weather, different types of partial damage, and day and night, all of which use raster-specific effects. When the arcade *Battlezone* tank gets hit, an irregular vector-graphics tree makes it appear as if the glass of the monitor has cracked. When the player's vehicle is destroyed in either Atari VCS game, gray, random shifting makes the TV screen look like it is losing



reception and starting to display snow. The vector-graphics visual signal for death is rasterized in these two tank games into something more suitable for that display and more easily accomplished with the console's graphics system.

The space trading strategy game *Elite*, released for BBC Micro personal computers in 1984, was notable for using wireframe 3D graphics with hidden line removal—the lines behind surfaces were not drawn so that structures appear solid. This technique evolved in other games, including the *Battlezone*-like title *Spectre* for Macintosh computers, released in 1991. The line-rendering technique quickly evolved from wireframes to filled surfaces to texture-mapped surfaces, which appeared in the early 1990s in *Wolfenstein 3-D* and *Ultima Underworld*. Vector art paved the way for modern 3D graphics, suggesting a way to represent solid objects in 3D space and allowing the development of feasible methods to animate such objects in real time. Today, 3D polygons are rasterized for display in pixel form on a modern computer or TV monitor, but the basic way that shapes appear in 3D space could already be seen by players zooming through *Battlezone's* vector landscape.

### Tempest turning

*Tempest*, designed by Atari's Dave Theurer and released in 1981, used a different sort of monitor: the Wells-Gardner 6100 Quadrascan, a color display with three beams for red, blue, and green. Along with Atari's *Space Duel*, *Tempest* was in the first wave of color vector-graphics games. The game also allowed players to choose the level where they wanted to begin; this poor man's continue feature let more capable players jump to more challenging levels, while novices could select an easier stage. In addition, *Tempest* offered differently shaped levels, so that the terrain and the challenges varied as the player progressed.

To allow the game's monsters to move around, *Tempest* took advantage of the ease with which objects can be rotated in an XY framework. It used scaling to make it appear that enemies were moving forward and growing larger. And as those enemies moved over uneven terrain, *Tempest* exploited the XY graphics hardware's ability to shear the creatures.

The much later *Tempest 2000*, released in 1994, uses raster and vector idioms. In this version, a tube's rasterized wireframe, representing the landscape of a level, rotates before the

player who is preparing to start the game. When a level is selected, that shape moves toward the player and the empty spaces between lines solidify, planes of color filling the tube in a raster-specific way. The player's ship and the early opponents look more or less like vectors, but this changes as the game reveals more powerful creatures. The typography is empathetically bitmapped and sometimes dissipates across the screen. Finally, the game includes a new type of level, the warp stage, that offers rings to fly through and a background that looks like nothing an XY graphics system could produce.

Although *Tempest 2000* transforms the arcade original into an updated raster game, it also hints at the earlier vector-graphics qualities of *Tempest*, both to make a connection to its arcade inspiration and as a way of evoking nostalgia for the random-scan tubes of years past. The game shows that although a TV cannot convincingly imitate a vector-graphics display, it can be made to remediate such graphics and remind the player of the original display system, rather than completely replacing the vector idiom with a raster one, as with *Yars' Revenge* and *Asteroids*.

### Raster and random, bitmap and vector

Videogames of the late 1970s and early 1980s are strongly associated with vector graphics. In recent years, we have seen a trend to re-create this "vector look" in contemporary games that are made to seem retro. New commercial games have been created that borrow the geometric appearance of vector graphics even though they are not played on XY displays. The most commercially successful of these titles is *Geometry Wars*, originally an Xbox shooter released in 2003 that combines the simple geometric shapes of games like *Asteroids* with psychedelic shader effects afforded by modern 3D graphics acceleration. In games like *Geometry Wars*, the use of a simple triangular ship and abstract geometric enemies references the game's vector predecessors. Other games, such as *Tempest 2000* developer Jeff Minter's Xbox 360 game *Space Giraffe*, released in 2008, borrow the tube shooter dynamics descended from *Tempest* but use vector-style line art as only a small part of the visuals, which include 3D shader and bloom effects.

Systems with vector-graphics displays provided developers with some capabilities that were not available a decade later. Thanks to the popularity of their vector games, Atari developed a general computing system to

aid in their creation. The **Atari Analogue Vector Generator** (AVG) is a special set of chips used to run the graphics systems of *Asteroids*, *Battlezone*, *Tempest*, *Gravitar*, and *Star Wars*. The AVG, which went through several versions, facilitated numerous useful operations for vector rendering. These included drawing a vector from the current cursor position to a new  $(x, y)$  position, positioning the cursor at the center of the screen, setting the color and intensity for new vectors, and program flow operations such as jumps and subroutines.<sup>2,5,11</sup> Of course, *Tempest 2000*, *Geometry Wars*, and other reimaginings of vector games were not able to use such hardware. **Those games required that new software subsystems be created to simulate the vector effect in raster- or polygon-based graphics systems.**

Vector art has persisted and evolved in computer graphics in other ways as well. Just as vector shapes allowed game developers to enact translations and scaling effects with ease, other forms of computer art benefited from mathematical representation. Vector displays have allowed higher-resolution graphics than would have been possible at the time of *Asteroids*.

**Bandwidth became a limitation in later eras. On the early Web, slow dial-up connections conflicted with users' increasing expectation for rich graphics and animations, a trend begun by GUI operating systems and multimedia CD-ROMs. Here vector graphics offered a solution as well. Because vector art is drawn based on a few parameters rather than being rendered as pixmaps, a PC can easily be made to draw many frames of an animation from instructions rather than data. These sets of instructions take up less memory and require that less data be sent over the network.**

Such a factor was primary in the design of **FutureSplash Animator** (1996), the program that became Macromedia Flash and then Adobe Flash. Macromedia's other popular animation product, Director, had been designed for use as CD-ROM and desktop software and uses raster graphics primarily to present photorealistic effects. Thanks to the widespread adoption of broadband Internet service, bitmap art is more frequently used in Flash compositions because it can be rendered fairly efficiently on today's computers. The distinction between raster- and vector-based representation is one that continues to affect not only what we see on the Web, but also how geographical data is abstracted in **geographic information systems** (GISs), where a debate between the two styles is ongoing and

conversion between them is often necessary. The importance of these two hardware display systems continues to extend beyond the commercial life of random-scan CRTs and the presence of XY graphics games in arcades.

### Conclusion

The random-scan display technology has faded, and even the once-ubiquitous raster-scan CRTs are rapidly yielding to flat-scan displays. Nevertheless, these displays left their traces directly and indirectly on many different games, including some of the most influential home-console and arcade titles. **And many of the important concepts behind the two different systems live on today in software.**

Today's modern techniques for 3D graphics descend partly from early vector wireframe techniques. **XY graphics display hardware is certainly not in widespread use, but XY (and XYZ) ways of thinking have become prevalent because of changing graphics capabilities and because of software systems such as Flash.**

Consideration of the two display systems is no doubt important for understanding the history of videogames. The way in which the different technologies supported and constrained game development sheds light on contemporary and past development practices and how they relate to computer hardware. **Beyond that, the healthy competition between random and raster, and the interesting challenges that prompted developers to innovate when moving between the two systems, suggest that standardization is not always ideal. A monoculture of videogame displays (or other hardware) might be convenient, but working on and across different technologies can provoke new, striking developments.**

### Acknowledgments

The text of this article from "The building blocks of *Star Castle*" through "Serving up *Yars' Revenge*" has been adapted from chapter five of our book *Racing the Beam* (MIT Press, 2009). This book is the first in the Platform Studies book series (<http://www.platformstudies.com>) the authors are coediting at the MIT Press.

### References and Notes

1. J.D. Foley et al., *Computer Graphics: Principles and Practice*, Addison-Wesley: 1996.
2. It's at least amusing to note that *Computer Space*, the first arcade game, was a raster adaptation of a game that ran on a Type 30 Precision CRT display, a random-scan display.

*Pong* was originally developed for the TV. Although there are probably more compelling reasons for *Pong's* commercial success, which greatly outstripped that of *Computer Space*, these two games do provide examples of one unsuccessful "port" from one display technology to another and one successful game that was done with a specific display in mind.

3. S. Stilphen, "Interview with Howard Scott Warshaw," Apr. 2005; [http://www.digitpress.com/archives/interview\\_warshaw.htm](http://www.digitpress.com/archives/interview_warshaw.htm).
4. H.S. Warshaw, interview by J. Weesner, Apr. 2007.
5. T. Skelley, "Tim Skelley's History of Cinematronics," June 1999; <http://www.dadgum.com/giantlist/archive/cinematronics.html>.
6. J. Novak, *Game Development Essentials: An Introduction*, Thompson Delmar, 2004.
7. Z. Whalen, "Lost in Emulation: World of Difference in Videogame Typography," *Proc. 3rd Ann. UF Game Studies Conf.*, 2007.
8. D. Debro, "YarsRevenge.asm," Disassembly of *Yars' Revenge* with labels and comments added, 21 Sept. 2005.
9. Although the 2D combat tank games corresponded to the 3D, first-person *Battlezone*, Rotberg's other game, *Red Baron*, can be seen as a 3D version of the combat biplane games.
10. D. Kushner, *Masters of Doom*, Random House, 2003.
11. See also [http://www.coinop.org/kb\\_dl.aspx/KB/gametech/atari\\_xy.html](http://www.coinop.org/kb_dl.aspx/KB/gametech/atari_xy.html), 2007.



**Nick Montfort** is an associate professor of digital media at the Massachusetts Institute of Technology. He collaborates on literary projects and writes poems, text generators, and interactive fiction such as *Book and Volume* and *Ad Verbum*. Montfort has a PhD in computer and information science from the University of Pennsylvania. Contact him at [nickm@nickm.com](mailto:nickm@nickm.com).



**Ian Bogost** is a videogame designer, critic, and researcher as well as an associate professor at the Georgia Institute of Technology and founding partner at Persuasive Games. His research considers videogames as an expressive medium, and his creative practice focuses on games about social and political issues. Bogost has a PhD in comparative literature from the University of California, Los Angeles. Contact him at [ibogost@gatech.edu](mailto:ibogost@gatech.edu).

For further information on this or any other computing topic, please visit our Digital Library at <http://computer.org/csdl>.

	
<p><b>PURPOSE:</b> The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.</p> <p><b>MEMBERSHIP:</b> Members receive the monthly magazine <i>Computer</i>, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.</p> <p><b>COMPUTER SOCIETY WEB SITE:</b> <a href="http://www.computer.org">www.computer.org</a></p> <p><b>OMBUDSMAN:</b> Email <a href="mailto:help@computer.org">help@computer.org</a>.</p> <p><b>Next Board Meeting:</b> 17 Nov. 2009, New Brunswick, NJ, USA</p>	
<p><b>EXECUTIVE COMMITTEE</b></p> <p><b>President:</b> Susan K. (Kathy) Land, CSDP*</p> <p><b>President-Elect:</b> James D. Isaak; *<b>Past President:</b> Rangachar Kasturi; *<b>Secretary:</b> David A. Grier; *<b>VP, Chapters Activities:</b> Sattupathu V. Sankaran; †<b>VP, Educational Activities:</b> Alan Clements (2nd VP); *<b>VP, Professional Activities:</b> James W. Moore; †<b>VP, Publications:</b> Sorel Reisman; †<b>VP, Standards Activities:</b> John Harauz; †<b>VP, Technical &amp; Conference Activities:</b> John W. Walz (1st VP); *<b>Treasurer:</b> Donald F. Shafer; * <b>2008–2009 IEEE Division V Director:</b> Deborah M. Cooper; † <b>2009–2010 IEEE Division VIII Director:</b> Stephen L. Diamond; † <b>2009 IEEE Division V Director-Elect:</b> Michael R. Williams; † <b>Computer Editor in Chief:</b> Carl K. Chang†</p> <p><small>*voting member of the Board of Governors    †nonvoting member of the Board of Governors</small></p>	
<p><b>BOARD OF GOVERNORS</b></p> <p><b>Term Expiring 2009:</b> Van L. Eden; Robert Dupuis; Frank E. Ferrante; Roger U. Fujii; Ann Q. Gates, CSDP; Juan E. Gilbert; Don F. Shafer</p> <p><b>Term Expiring 2010:</b> André Ivanov; Phillip A. Laplante; Itaru Mimura; Jon G. Rokne; Christina M. Schober; Ann E.K. Sobel; Jeffrey M. Voas</p> <p><b>Term Expiring 2011:</b> Elisa Bertino; George V. Cybenko; Ann DeMarle; David S. Ebert; David A. Grier; Hironori Kasahara; Steven L. Tanimoto</p>	
<p><b>EXECUTIVE STAFF</b></p> <p><b>Executive Director:</b> Angela R. Burgess; <b>Director, Business &amp; Product Development:</b> Ann Vu; <b>Director, Finance &amp; Accounting:</b> John Miller; <b>Director, Governance, &amp; Associate Executive Director:</b> Anne Marie Kelly; <b>Director, Information Technology &amp; Services:</b> Carl Scott; <b>Director, Membership Development:</b> Violet S. Doan; <b>Director, Products &amp; Services:</b> Evan Butterfield; <b>Director, Sales &amp; Marketing:</b> Dick Price</p> <p><b>COMPUTER SOCIETY OFFICES</b></p> <p><b>Washington, D.C.:</b> 2001 L St., Ste. 700, Washington, D.C. 20036  <b>Phone:</b> +1 202 371 0101; <b>Fax:</b> +1 202 728 9614; <b>Email:</b> <a href="mailto:hq.ofc@computer.org">hq.ofc@computer.org</a></p> <p><b>Los Alamitos:</b> 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314  <b>Phone:</b> +1 714 821 8380; <b>Email:</b> <a href="mailto:help@computer.org">help@computer.org</a></p> <p><b>Membership &amp; Publication Orders:</b>  <b>Phone:</b> +1 800 272 6657; <b>Fax:</b> +1 714 821 4641; <b>Email:</b> <a href="mailto:help@computer.org">help@computer.org</a></p> <p><b>Asia/Pacific:</b> Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan  <b>Phone:</b> +81 3 3408 3118 • <b>Fax:</b> +81 3 3408 3553  <b>Email:</b> <a href="mailto:tokyo.ofc@computer.org">tokyo.ofc@computer.org</a></p> <p><b>IEEE OFFICERS</b></p> <p><b>President:</b> John R. Vig; <b>President-Elect:</b> Pedro A. Ray; <b>Past President:</b> Lewis M. Terman; <b>Secretary:</b> Barry L. Shoop; <b>Treasurer:</b> Peter W. Staecker; <b>VP, Educational Activities:</b> Teofilo Ramos; <b>VP, Publication Services &amp; Products:</b> Jon G. Rokne; <b>VP, Membership &amp; Geographic Activities:</b> Joseph V. Lillie; <b>President, Standards Association Board of Governors:</b> W. Charlton Adams; <b>VP, Technical Activities:</b> Harold L. Fleischer; <b>IEEE Division V Director:</b> Deborah M. Cooper; <b>IEEE Division VIII Director:</b> Stephen L. Diamond; <b>President, IEEE-USA:</b> Gordon W. Day</p> <p style="text-align: right;">   <b>Celebrating 125 Years</b>  <i>of Engineering the Future</i> </p> <p>revised 1 May 2009</p>	