CHAPTER 2

Abstraction in the Video Game

MARK J. P. WOLF

Most writing on video games considers either narrative concerns or gameplay and what is usually referred to as "interactivity." Instead of focusing on the "game" aspects of the video game, this essay looks at the video game as "video," or rather, as a medium of visual imagery. The video game began with perhaps the harshest restrictions encountered by any nascent visual medium in regard to graphic representation. So limited were the graphics capabilities of the early games, that the medium was forced to remain relatively abstract for over a decade. Gradually as technology improved, designers strove for move representational graphics in game imagery, and today they still continue to pursue ever more detailed representations approximating the physical world. At the same time, video games have come to rely on conventions from film and television, allowing the depiction and navigation of their diegetic worlds to seem more intuitive and familiar to players. Yet by limiting themselves to conventions established in other media, game designers have neglected the realm of possibilities which abstraction has to offer. This great, untapped potential will only be mined by a deliberate move back into abstract design that takes into consideration the unique properties of the video game medium.

In order to get a better sense of how abstraction has been used in video games, we might first examine some of the ways in which the video game can be seen as an extension of abstract art, and the different types of abstraction that can be present within a video game.

Abstraction, Time, and Interaction

To *abstract* something is to simplify it, reducing it to a few essentials and basic forms instead of trying to reproduce it. *Representation,* which seeks to create resemblances and reproduce something, is the polar opposite of abstraction (and is sometimes conflated with *realism).* Most artwork falls in the spectrum between the two extremes, since even very representational artwork usually falls short of fully reproducing its subject. Abstraction has appeared throughout art history, from the earliest cave paintings, to mosaics, tilework, geometric patterns and ornamentation, to the works of Cézanne and the Impressionists, Cubists, and Surrealists, whose work inspired the Abstract Expressionist movement in New York during the 1940s and 1950s. Finally, as abstract art moved more and more into the conceptual realm, it began to raise questions as to what exactly constituted a work of art.

During the second phase of Abstract Expressionism, an approach to abstract art developed known as "action painting," of which Jackson Pollock is perhaps the best-known practitioner. For the action painters, the process of creating the art, including the elements of time and performance, became important. In his 1952 essay, "The American Action Painters," art critic Harold Rosenberg wrote,

> At a certain moment the canvas began to appear to one American painter after another as an arena in which to act, rather than a space in which to reproduce, redesign, analyze, or "express" an object, actual or imagined. What was to go on the canvas was not a picture, but an event.[1]

Rosenberg's description of the direction in which abstract art was developing, one that was time-based, interactive, and event-based, seems to anticipate the video game, which would appear on mainframe computers a decade after the essay.

Besides action painting, the idea that a work of art could be an event was found in areas of conceptual art and performance art of the late 1950s and 1960s, such as Allen Kaprow's Happenings, the Fluxus festival, Jordan Belson's Vortex Concerts, and the Japanese group Gutai Bijutsu Kyokai. Although many of these events centered around the performance of the artist, some began to incorporate the audience into the event, for example, John Cage's piece for piano, 4:33, which contained no notes at all and consisted solely of the ambiance of the auditorium, including sounds made by the audience. Audience participation varied from mere attendance as a spectator at art events to active engagement with interactive artwork such as Nam June Paik's closed-circuit video installations in which a live video image of the viewer appeared, or Myron Krueger's *VIDEOPLACE* (1970), which featured wall-projected video imagery that combined the user's live image with interactive computer graphics produced in real time. In some cases, the point or experience embodied in the artwork could only be understood through active interaction rather than passive spectatorship.

The video game, appearing when it did, took advantage of the interest in interactive art and the intersection of art and electronic technology, and it is perhaps no coincidence that many early game programmers were counterculturally engaged.[2] Because of the limitations of early video game technology, graphics were stark and minimalist for the most part, and rather abstract as a result, which happened to coincide with certain minimalist tendencies of the time. Interactivity was, of course, what the made the games interesting; like certain video installation pieces, one could affect the image onscreen in real time and watch it change.

When it left private mainframe computers, however, the video game met the public in the arcade and later in the home, not in the art museum. Video games also differed from interactive art because of their status as games, which meant that there was usually some motive or goal toward which the player's interaction was directed, whereas in art, the experience itself was the goal. The addition of a goal or challenge to the experience connected the video game to such things as pinball and other arcade games, table-top games, and board games, all of which would influence the shape the video game would take both aesthetically and commercially. But the video game, as an audio-visual medium, stood apart from other games, for its gameplay lacked the solidity of game pieces or the physicality of pinball action and the Newtonian collisions of pool balls. The video game took place within an image, whose interactivity required a new way of reading and understanding abstract imagery.

Reading the Abstract Image in the Video Game

Just as the understanding of cinematic conventions seems second nature to us today, the playing of video games has likewise come to feel natural and automatic. It is easy to forget that the video game interface, with its hand-eye coordination linked to onscreen action, was not always as intuitive as it now seems. The first arcade video game, Nolan Bushnell's *Computer Space* (1971), failed commercially, because players found its controls difficult to understand and use, and it was not until Bushnell's second effort, *PONG* (1972), which had a single control and simplified graphics, that the video game as a commercial entity finally found success.

In this sense, the video game required abstract imagery to be read in a new way. Since the substance of video games is simultaneously both *imagery* and *events,* their elements can be abstract in both *appearance* and *behavior.* Among the first tasks a player encounters while learning to play a game is the identifying of the different elements seen onscreen and understanding how they function and behave. Elements occurring in video games can be divided into four general categories: those indicating the player's presence in the game (the player-character); those indicating the computer's presence in the game (computer-controlled characters); objects that can be manipulated or used by game characters; and the background environment that generally serves as the setting and is not manipulated or altered by any of the characters during the game.

Among these elements, the most important is arguably the player-character. Player-characters, which represent the player's influence and effect on the game's diegetic world, can be either surrogate-based or implied. *Implied* player-characters are not visible onscreen. In these games, player interaction is indicated through onscreen events that occur as a result of the player's actions, including changes in the game's point of view (particularly in first-person games in which the point of view is controlled by the player), by changes in informational graphics that indicate what the player is doing (like the crosshairs in *Missile Command* [1980] and the radar in *Battlezone* [1980]), or by the direct manipulation of onscreen objects (such as is found in *Tetris* [1989] or *Video Chess* [1979]). In any of these games, the player-character is constructed outside of the screen, both *visually* (sometimes through perspective, the way a photograph constructs a point of view and implies a spectator) and *interactively* through the game interface and the coordinated onscreen action under the player's control.

Surrogate-based player-characters appear as onscreen graphics representing the player's character (see Figure 2.1). Since the 1980s, most onscreen player-surrogates are character-based (such as Pac-Man, Mario Mario, Lara Croft, Sonic the Hedgehog, and so on), but this has not always been the case. In the earliest video games, the player-surrogate was function-based; instead of an anthropomorphic character, the player was represented onscreen by a graphic of a tool or vehicle that the player controlled. Function-based players surrogates included spaceships, (as in numerous outer space games), tanks or planes (as in *Combat* [1977] and *Air-Sea Battle* [1977]), cars (in racing and driving games), or even a "paddle" (as in *PONG* [1972] or *Breakout* [1976]). One advantage of function-based player-surrogates was that they often represented rigid objects for which a minimum of animation was needed. When they moved, spaceships, cars, tanks, and paddles traveled in straight lines or turned in place, whereas characters with moving arms and legs required more animation and computing power. Atari's *Football* (1976), for example, used a series of "X"s and "O"s to represent the players, similar to football diagrams, instead of human figures.

As graphics improved, games began moving towards human figures as character-based player-surrogates, beginning with simple blocky forms such as those found in Midway's *Gun Fight* (1975), Project Support Engineering's

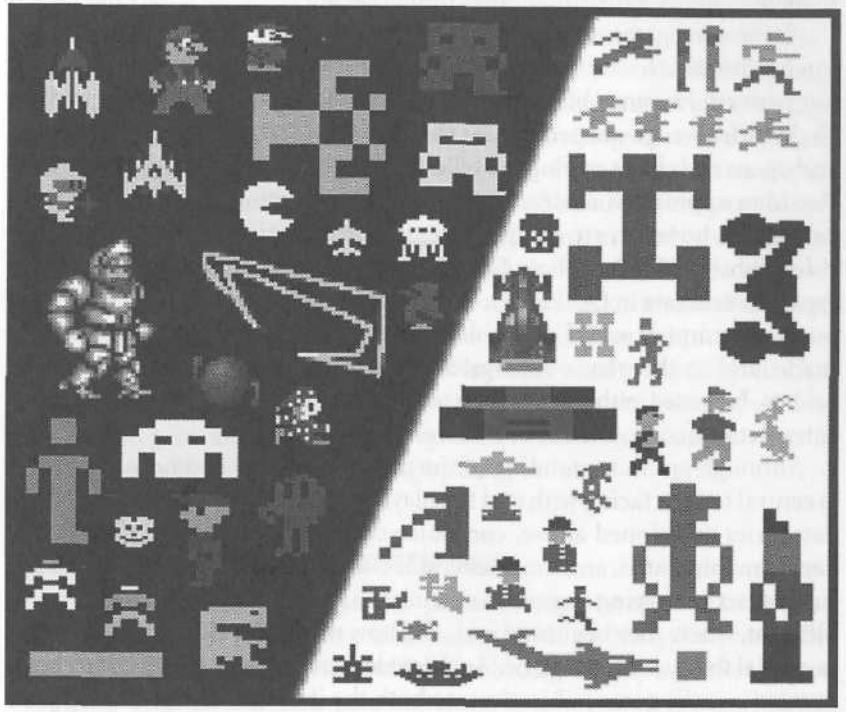

Figure 2.1. Examples of player-characters (avatars) in early video games. Characters are designed to appear on either dark or light backgrounds. Note the five different styles of race car icons.

Knights in Armor (1976), Atari's *Outlaw* (1976), *Homerun* (1978), and *Basketball* (1978), and Meadows's *Gypsy Juggler* (1978). Animals appeared as player-surrogates beginning with *Frogs* (1978). The player-surrogates in these games, however, were still functional in that they were differentiated only by color, and given no names or individual traits that made them different from each other. Only later in games such as *Pac-Man* (1980) and *Donkey Kong (1981)*, and games adapted from other media such as Atari's *Raiders of the Lost Ark* (1982) and *E.T.: The Extraterrestrial* (1982), did the player-surrogates have names and identities of their own apart from the player. From the early 1980s on, however, character-based player-surrogates were by far the most common form of player-character in video games, no doubt due in part to the stronger identification they could engender. Simple avatars are similar to cartoons, of which Scott McCloud writes, "The cartoon is a vacuum into which our identity and awareness are pulled, an empty shell that we inhabit which enables us to travel to another realm. We don't just observe the cartoon, we become it!"[3] Abstraction, then can become an aid to identification, rather than something that alienates.

Player-surrogates, whether they are function-based or character-based, can be abstract to varying degrees. In some cases, they are visually abstract but perform recognizable functions. Pac-Man, for example, is only a yellow circle with a wedge-shaped gap. As the gap faces forward as Pac-Man moves, and opens and closes erasing the yellow dots, it functions as a mouth, making Pac-Man appear less abstract and more like an eating head. Some player-surrogates, however, are completely nonrepresentational, like the square in *Adventure* (1978), the yellow C-shaped surrogate in *Tempest* (1981), or the "spark" surrogate in *Qix* (1981). The player controls each of these onscreen, but no attempt at visual resemblance with something outside the game is made, and so the player-surrogates in these three games cannot really be said to be based either on characters or functions without imposing an interpretation onto the abstract forms.

Although an understanding of the player-character and how it operates is central to interfacing with and the playing of a video game, the other three categories mentioned above, computer-controlled characters, objects that can be manipulated, and the unalterable background environment, also can be abstract to varying degrees. Knowing the role and function of each game element, where they begin and end, and how they affect the player-character is crucial to learning the game. As these elements grow more abstract, however, so can the game's objectives, as both the interface and gameplay grow less intuitive. Thus, another reason for making game elements representational are the default assumptions and diegetic structures that accompany them and make both the interface and gameplay more transparent and intuitive.

Games with representational graphics often rely on conventions from other audiovisual media, and increasingly, on conventions established in earlier video games, giving them a built-in familiarity that allows players to begin playing without having to learn the interface. Still, early games were abstract enough, even when they were representational, that an explanation of play had to be included, either as a demo running in between games, instructions printed on the game cabinet, or in an instruction manual in the case of home games. Such explanations were especially important for games with unusual goals or gameplay, or deliberately abstract games, like Qix (1981) and *Q*bert* (1983). Often the titles of games, such as *PONG, Asteroids, Tank, Football,* and so on, helped players to interpret what the graphics were meant to represent. Home games, purchased instead of played in an arcade, could afford to have more complex gameplay and even feature glossaries defining the games' low-resolution iconography. For example, *Raiders of the Lost Ark* (1982) for the Atari 2600 had over a dozen objects the player could use, which were represented in two different sets of graphic icons, one for the playfield and one for the inventory strip at the bottom of the screen (both sets were depicted and defined in the game's instruction manual).

The simplicity of the early games helped players to read and understand the images and action onscreen during a game. Later games built on earlier conventions, adding graphical complexity, spatial navigation, and a greater mixture of diegetic and nondiegetic graphics on the same screen, as players grew more sophisticated. Likewise, early games helped players grow accustomed to the hand-eye coordination and input devices used to control the action. *PONG* and games like it familiarized players with the paddle controller, while other games used firing buttons, joysticks, and trak-balls, all of which now seem elementary compared to controllers commonly in use today, like the multibutton two-handed controller used in the Sony PlayStation, Microsoft Xbox, or the 12-button controller for the Nintendo 64. As the medium grew and conventions began accumulating, video games grew from a novelty into an industry and popular pastime.

The Golden Age of Abstraction in the Video Game

The Golden Age of the video game is generally considered to be the period from the rise of the video game in the early 1970s to 1984, the year of the great video game industry crash, which was brought on by a market glutted with game systems and their pale imitators. In many ways this period was also the Golden Age of abstraction in the video game, due in large part to the technological limitations of the time.

Just as early film audiences thrilled to see anything move in the one-shot actualities of the 1890s, early video games, with their abstract and often minimalist graphics, had no competition and generally found success (a plethora *of PONG* imitations, based on the AY-3-8500 chip, testify to the high demand for home games). As games proliferated and the initial novelty of controlling objects onscreen wore thin, competition among systems increased, especially with the industry changeover from hard-wired consoles to those that used interchangeable cartridges. The number of games available for a given system was one consideration for system buyers, along with graphical complexity. Game graphics were, and to a large extent still are, the main criteria by which advancing video game technology is benchmarked by the buying public; thus representational graphics act as a means of visually benchmarking the computer's graphics against the visual experience of unmediated reality, while abstract graphics are unable to serve such a purpose.

Even the earliest games claimed to represent something, from space battles to ping-pong games. Market pressure for more representational and "realistic" graphics in games is certainly one of the forces that shaped the look of games and pushed the technology forward. Another was the game programmers themselves. According to Atari 2600 programmer Rob Fulop, whose games include *Demons to Diamonds, Cosmic Ark,* and *Missile Command,* as well as the 2600-ported versions of *Space Invaders* (1978) and *Night Driver* (1980), game complexity naturally evolved as programmers attempted to outdo one another. Certain types of material were chosen because they were easier to do; for example, outer space games required only a black background (usually with stars) and a few spaceships and laser bullets.[4]

Working against the desire for representation were the technological restrictions embodied in the programming of the games. Even today, video games' appearance and behavior are still subject to limitations regarding the number of polygons used, rendering speeds, moving objects and collision detection, and screen resolutions. Some of the limitations Atari programmers were confronted with can be found in the "Stella Programmer's Guide" of 1979 by Steve Wright ("Stella" was the in-house name for the 2600 at Atari). For example, the Atari 2600 had only 128 bytes of RAM (and no disk storage), and a graphics clock that ran at roughly 1.2 MHz.[5] Early cartridges for the Atari 2600 contained as little as 2 or 4 kilobytes of ROM. The television image produced for the games had only 192 lines vertically and 160 color clocks, or pixels, across.[6]

Because the microprocessor was so slow, most games used only half of the 192 lines available, and roughly half of the horizontal resolution as well. As the Guide describes it:

> The actual TV picture is drawn one line at a time by having the microprocessor enter the data for that line into the Television Interface Adaptor (TIA) chip, which then converts the data into video signals. The TIA can only have data in it that pertains to the line being currently drawn, so the microprocessor must be "one step ahead" of the electron beam on each line. Since one microprocessor machine cycle occurs every 3 clock counts, the programmer has only 76 machine cycles per line (228/3 = 76) to construct the actual picture (actually less because the microprocessor must be ahead of the raster). To allow more time for the software, it is customary (but not required) to update the TIA every two scan lines.[7]

In addition to the low resolution, the graphics making up the playing field (or playfield) laid over the solid-color background were drawn only on the left side of the screen and then duplicated on the right half of the screen, either as a repetition or a mirror image of the left side, accounting for much of the horizontal symmetry found in the early Atari games. This limitation is particularly noticeable in games with more complex graphics, for example, the mazes and castles of *Adventure* (1978), the city screens in *Superman* (1979), and the forest and buildings in *E.T.: The Extraterrestrial* (1982) (see Figure 2.2).

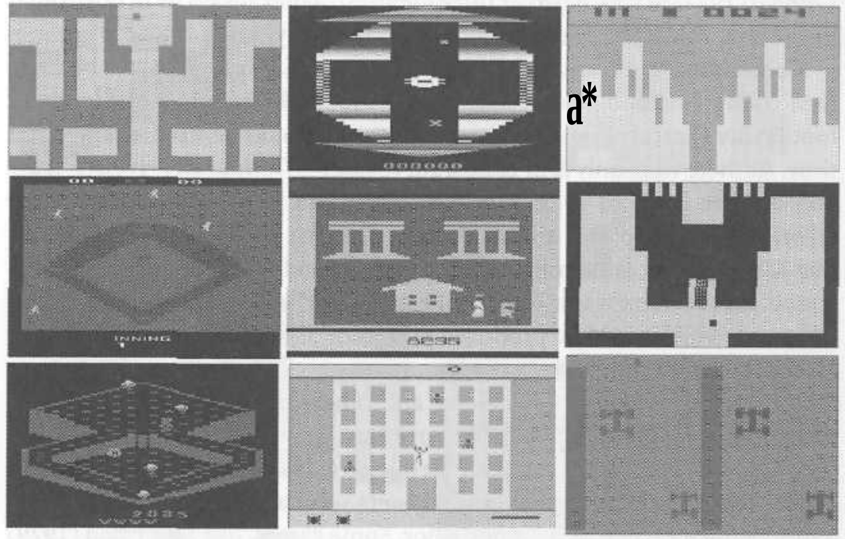

Figure 2.2. Because of the way the Atari 2600 was programmed, the right side of the background had to either be a duplication of the left side or a mirror image of it, resulting in symmetrical screens, a graphical limitation that had to be worked into the games.

Motion was also subject to restraints. Because of the way in which the raster scan occurs, horizontal and vertical movement had to be programmed in different ways. As the positioning of objects on the horizontal axis was limited by the machine cycle and writing of registers, horizontal motion had to be further fine-tuned through a separate register that would fill in the resultant gaps in movement.[8] In the end, horizontal motion was more difficult to achieve than vertical motion, resulting in a bias toward having action occur on the vertical axis.[9]

The playfield consisted of blocks of a single foreground color placed over a solid color background, and on top of the playfield were the "moveable objects graphics," which were limited to five objects: two players, two missiles (one for each player), and a ball. The players were the characters controlled by either the human game player or the computer, while the missiles and ball were objects used or fired by the players. Each player graphic was eight bits (pixels) wide, while the missiles and balls were only one bit wide, although the single bit could appear as 1,2,4, or 8 clock counts of horizontal line time (a full line being 160 clock counts). Player graphics also could be stretched horizontally, and up to three copies of the player could appear horizontally, though they would have the same attributes (color, width, movement, and missile action). An example of this appears in games 19-20 and 25-27 of *Combat* (1977) in which two or three identical planes always fly in horizontal formation, rotate together, and fire simultaneously.

Color, luminance, and sound were also limited. A programmer had sixteen colors (including transparency) to choose from, and eight levels of luminosity (brightness), but only four of each could be used at any given time. As there were only four color-lum (color and luminosity) registers, the objects were grouped so that the first player and its missile were the same color, the second player and its missile were the same color, the ball and playfield were the same color, and the background was a separate color. For sound effects, there was a "noise-tone generator" that created sixteen types of sound, which could be produced at sixteen different pitches and sixteen levels of volume (including silence).

These programming restrictions, and others, account for much of the style, graphics, and gameplay of the early Atari games. But more memory and programming tricks helped game developers overcome some limitations; for example, only four color-lum registers were available, meaning that a game character could only be one color. Some games, like *Superman* (1979) and *E. T.: The Extraterrestrial* (1982) got around this by changing the color-luminosity values on a line-by-line basis (see Figure 2.3), which allowed characters to be multiple colors, although any given horizontal line of pixels had to be the same color; a stylistic limitation due to the way the monitor scans the image onto the screen. Graphic complexity, then, was often a sign of programming prowess and graphics evolved as programmers tried to outdo each other.[10]

Advances in software and hardware allowed graphics to become more detailed and representational, but there was still some interest in the creation of purely abstract graphics. In 1978, Atari released the Atari Video Music C-240, which connected a stereo system to a television screen, translating an audio signal into abstract video graphics. The idea of machines to produce abstract graphics determined by sounds, or "visual music," can be traced back to the Ocular Harpsichord built by the Jesuit Father Louis Bertrand Castel in 1730, and extends through optical toys like the chromatrope slide for magic lanterns in the 1800s, to the early 1900s and the projected-light electrical "color organs" of A. Wallace Rimington, a British painter, and Italian futurists Arnaldo Ginna and Bruno Corra. The 1920s and 1930s saw the invention of several color organs, Thomas Wilfred's Clavilux, Mary Hallock Greenewalt's Sarabet, and Charles Dockum's MobilColor projectors, which in some ways were the forerunners of the video synthesizers that appeared in the 1960s and 1970s.[11]

Video synthesizers used analog electronics to generate a real-time video image from an audio signal, resulting in abstract graphics that were displayed

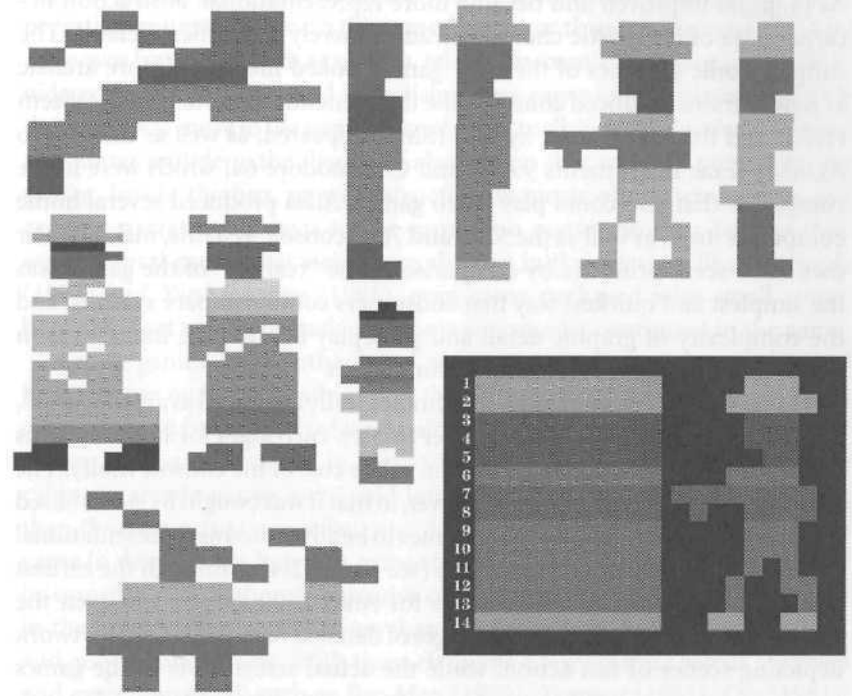

Figure 2.3. Another limitation of the Atari 2600 involved the color-luminance register. Onscreen characters have only one color-luminance register assigned to them, limiting the character to a single color. But if the register is changed for every scan line (see the fourteen scan lines in the image in the black box), a character made up of multiple colors can be displayed, although there will still be only one color on any given horizontal line of the character. The top row of characters and the helicopter on the lower left are from *Superman* (1979), and the three characters in the second row are from ***E.T.: The Extraterrestrial* (1982),**

on a television screen. While most video synthesizers were experimental units used by artists and were not available to the public, the Atari Video Music was simpler, mass-produced, and marketed to the owners of the Atari 2600. The Atari Video Music was not a game itself but its successor, the Atari Jaguar Virtual Light Machine (VLM) was built into the Atari Jaguar console.

The audience for abstraction and experimental graphics, however, was not as large as the audience for games and increasingly detailed representational graphics, and market pressure for increased realism moved game design away from abstraction as video game graphics improved in the mid-to-late 1980s.

The Decline of Abstraction

As graphics improved and became more representational, abstraction became more of an artistic choice instead of merely a technical default. The simple, iconic graphics of the early games looked more and more archaic as newer, more advanced consoles like the Nintendo Entertainment System (NES) and the Sega Master System (SMS) appeared, as well as the Coleco ADAM, Texas Instruments 99/4a, and Commodore 64, which were home computers that also could play video games. Atari produced several home computers itself, as well as the 5200 and 7800 console systems, making their own 2600 seem primitive by comparison. The "realism" of the games was the simplest and quickest way that consumers could compare systems, and the complexity of graphic detail and gameplay became the main areas in which home games would compete for players.

Home video games were not as technologically advanced as arcade games, and while arcade games cost a quarter to play, cartridges for home systems could cost $20 to $30 each {in addition to the cost of the console itself). The home game had one advantage, however, in that it was bought by stores based mainly on the packaging, allowing games to be advertised as representational despite their highly abstract graphics (see Figure 2.4). Although the earliest boxes featured only text, later boxes for Atari 2600 games, and even the cartridges themselves, featured collages of detailed representational artwork depicting scenes of fast action, while the actual screenshots of the games

Figure 2.4. Box claims versus actual game screens.

themselves were relegated to the backs of the boxes if they even appeared at all. The games, then, were sold based on their supposed connections to the narrative contexts shown on the games' boxes, or through their connections to known franchises such as movies, television, comic books, or even arcade video games. The boxes and advertising were eager to help players imagine that there was more to the games than there actually was, and actively worked to counter and deny the degree of abstraction that was still present in the games. Inside the box, game instruction manuals also attempted to add exciting narrative contexts to the games, no matter how far-fetched they were. Several games that were more abstract in their design, like *Centipede* (1982) and *Yars' Revenge* (1981), even came packaged with small comic books that set up the narrative that was supposedly continued in the game.

Arcade games, on the other hand, always remained several steps ahead of home games technologically and in their representational ability, although simplicity and fast action had to remain priorities if they were to keep a steady stream of quarters rolling in. Apart from the graphics featured on their cabinets, arcade games were sold less through packaging and advertising than through actual gameplay; one decided to play either by watching the game in demo mode between games or by watching the game being played by someone else. Elaborate narrative contexts, then, are often less important in the selling of an arcade game than are fast action, interesting graphics, and good sound effects. With these elements a few abstract games survived and even flourished, such as *Pac-Man* (1980), *Tempest* (1981), *Qix* (1981), *Q*bert(\982), Quantum* (1982), *Marble Madness* (1984), and *Tetris* (1988). A number of these games even inspired sequels (*Qix II: Tournament* [1981], *Super Qix[1987], Twin Qix*[1995], *Q*bert's Qubes* [1983], *Tetris Plus* [1990], *Hatris* [1990], *Tetris Plus 2* [1997], and *Marble Madness 2: Marble Man* [1991], and over half a dozen *Pac-Man* sequels[12]), a phenomenon more typical of games with narrative content.

Despite a few successes, however, game design leaned increasingly toward representational games and fewer abstract ones were produced. While technological improvements allowed games to become more representational, they cannot, in and of themselves, explain why pure abstraction was so overwhelmingly rejected. One reason may be found in the work of the art theorist Wilhelm Worringer, whose 1908 treatise, *Abstraction and Empathy*, remains a classic. Worringer suggested that there are two fundamental aesthetic impulses that are mutually exclusive, the desire for abstraction and the desire for empathy. He writes:

> In the forms of the work of art we enjoy ourselves. Aesthetic enjoyment is objectified self-enjoyment. The value of a line, of a form consists for us in the value of the life that it holds for us. It holds its beauty only through our own vital feeling, which, in some mysterious manner, we project into it.[13]

Thus, the urge toward representation is due to a desire for empathy, even though, as noted above and in Scott McCloud's work, abstraction can help to increase identification, the game's diegetic world is easier to enter into if it resembles the real world. But what happens when the real world is seen as an uncomfortable or threatening place? Of the opposing urge to the pole of abstraction, Worringer writes:

> Whereas the precondition for the urge to empathy is a happy pantheistic relationship of confidence between man and the phenomena of the external world, the urge to abstraction is the outcome of a greater inner unrest inspired in man by phenomena of the outside world; in a religious respect it corresponds to a strongly transcendental tinge to ail notions.[14]

The need for empathy does seem to explain the great popularity of representational art among the general public compared to that of abstract art, which seems to be more of an acquired taste. In most media, abstract works are relegated to a marginalized genre, created and seen by only a few. Abstraction comes closest to reaching a broad audience in music, but even there the majority of albums produced include the human voice as an empathic anchor and link to human emotional experience. Worringer sees the need for empathy as a kind of understanding of, or being able to relate to, the object or scene being depicted:

> … the process of empathy represents a self-affirmation, an affirmation of the general will to activity that is in us. 'We always have a need for self-activation. Indeed this is the basic need of our nature.' In empathising with this will to activity into another object, however, we *are* in the other object. We are delivered from our individual being as long as we are absorbed into an external object, an external form, with our inner urge to experience.[15]

The player-character surrogate in the video game is, in a very concrete sense, the external object into which the player is absorbed, which receives the player's will to activity. This may help explain why the majority of player-character surrogates in video games are character-based. Even in abstract games one can find them; *Pac-Man* and *Q*bert* and their sequels are character-based, although both are still very abstract and personify their characters through cabinet art, the cut scenes in *Pac-Man,* and embedded names that suggest personhood ("Man" and "bert"**).** As in many character-based games, a simple chase narrative is also present, making the games less abstract in function, if not in appearance. The fact that Pac-Man flees, eats with a moving mouth, and even "dies" further personifies what would otherwise be merely a yellow circle with a wedge missing. (In some sense, the condition of a player learning to control an onscreen surrogate and developing hand-eye coordination is similar to the "mirror stage," in which an infant learns to recognize and control his reflection in the mirror.)

During the 1980s, the video game also became more integrated with other forms of media, both commercially and aesthetically, and many of its conventions had codified. Although the iconic graphics of video games still retained a degree of abstraction, purely abstract games had become a minor genre, which, apart from *Tetris,* had relatively few hit games in the latter half of the 1980s. But during the early 1990s, as the technological advances of computer graphics began to spread from the laboratory and university to film, television, and video games, new possibilities and uses for abstraction developed.

New Possibilities for Abstraction

Besides purely abstract games, there have always been games that are abstract in that they are representations of abstract things, for example, games that are adaptations of board games, card games, table-top games, and pinball games. Versions of pencil-and-paper games (like Tic-tac-toe and Hangman) are not based on physical objects, yet each is a representation of something found outside of the video game, making them ontologically similar to the two-dimensional objects depicted in Jasper Johns's paintings of flags, targets, maps, letters, and numerals. Abstraction, then, can appear as an element within representational game graphics, or even as part of the subject matter, for example, the patterns and designs appearing as surface textures found on three-dimensional objects and settings in video games.

Although attempts at three-dimensional graphics can be found in earlier games like *NightDriver* (1976), it was not until the 1980s that games with real 3-D computation appeared, including the wireframe landscape of the vector arcade game *Battlezone* (1980), and the abstract but three-dimensional world of *I, Robot* (1983), the first 3-D game to feature filled-polygon graphics (as opposed to wireframe graphics). Computing and rendering speeds continued to increase, and the next advances in game graphics would include 3-D lighting and shadow effects and eventually texture mapping.

The idea behind texture mapping, which places a two-dimensional texture, pattern, or image onto a surface in a 3-D computer-generated object, first appeared in 1974 in the doctoral dissertation of Ed Catmull at the University of Utah.[16] As hardware and software grew more sophisticated, texture mapping spread from university computers to film during the late 1980s, most notably in Pixar's award-winning short films *Luxo Jr.* (1986), ***Red's Dream* (1987),** *Tin Toy* (1988), and *Knickknack* (1989), which vividly illustrate the advances taking place. With the release of Intel's 64-bit Pentium chip in 1993, home computers and game consoles were ready to incorporate texture mapping into video games. Two hit games also appeared in 1993 that incorporated texture mapping and changed the standards of computer games; *Doom* and *Myst.*

Doom featured texture-mapping in a 3-D environment rendered in real time during game play, while *Myst* had prerendered high-resolution images with finer textures and subtler lighting effects. Both games signaled the beginning of a new era in video game graphics, and *Myst's* sequel, *Riven* (1997), used larger and more detailed texture maps to achieve its photorealism. Although the objects the textures are mapped onto are representational, the texture maps display patterns of rust, cracks, grain, splatter, corrosion, and wear, which are often the very subject matter explored in works of abstract art. Ironically, it is through the application of these abstract texture maps that video game imagery achieves heightened realism and greater representational ability.

Even as representational games strive for greater photorealism, the abstract game has not vanished entirely. In January of 2002, Sega's game *Rez* was released for the Sony PlayStation 2. Graphically, the game is a nostalgic throwback to the days before texture mapping, and its luminous solid-colored polygonal figures with glowing vertices and vectors are at times reminiscent of the film *Tron* (1982) (see Figure 2.5). It also features "Wave Master club beats and a laser light show,"[17] which, along with its psychedelically-inspired graphics, also link it to the video synthesizer tradition. Despite its dazzling graphics, *Rez* is still given a clichéd save-the-world frame story that involves going into the cyberworld "inside" a computer, as in *Tron.* It features several different designs of player-characters through which the player advances, a number of which are anthropomorphic, as are some of the "bosses" or enemy characters in the game. Gameplay itself is relatively simple, and the only controls are a targeting cursor and a fire button. The narrative context, anthropomorphic figures, and simple play may indicate Sega's reluctance to think too far outside the box of video game conventions, but the game does reveal possibilities for video game design even if it does not embrace them entirely.

Although abstract video games may not be learned as intuitively as representational games, abstraction need not imply simplicity. With the sophistication of today's hardware and software, a vast amount of potential in the area of abstract gaming remains untapped. Complexity may even help bring about a rebirth of the abstract game. A study published in *Empirical Studies of the Arts* by Tsion Avital and Gerald C. Cupchik regarding the perceiving of hierarchical structures in abstract paintings found that "affective ratings were tied to interesting complexity, but not to disorder," and that "hierarchically complex artworks facilitate both affective orienting and an exploratory search for meaning."[18] Discussing experiments in which

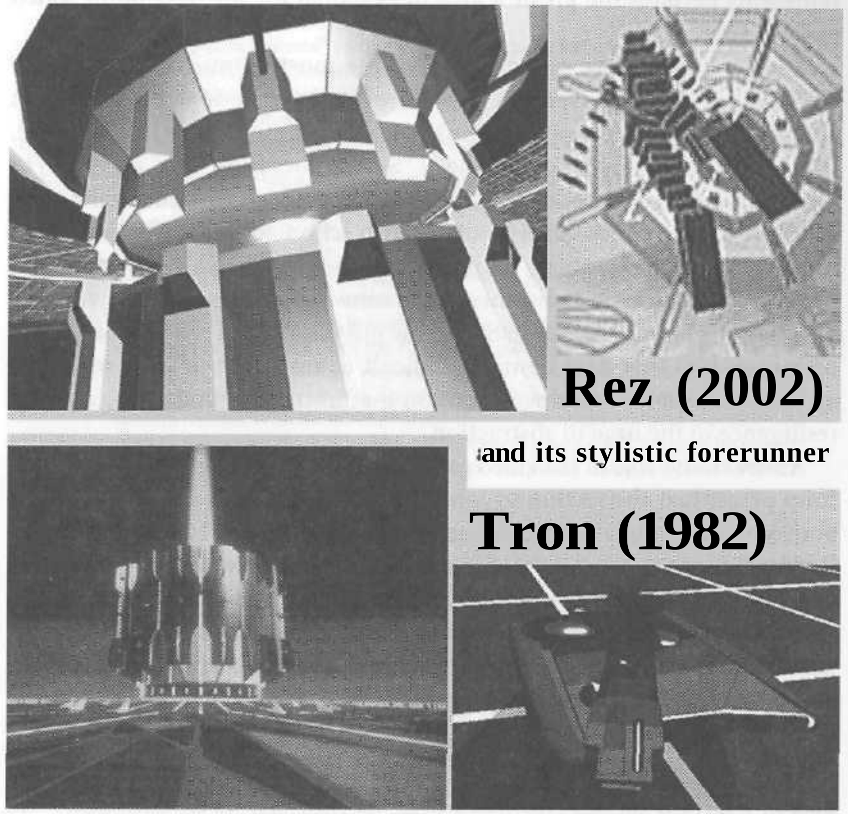

Figure 2.5. The PlayStation 2 game *Rez* (2002) and its stylistic forerunner, the film *Tron* (1982). *Rez* features an avatar (in the upper-right-hand corner) which has the same black-paneled, red-verticed look of *Tron's* Recognizers, as well as a large cylindrical object (in the upper-left-hand corner) reminiscent of *Tron's* MCP

the subjects viewed a series of paintings in sequence, the authors write, "serial presentation attracts the viewer's attention and stimulates affective response because it raises expectations and curiosity regarding the emergent structures."[19] The time and motion present in a video game, coupled with complex graphics, could add to the stimulation of attention and curiosity, and play with expectations in a variety of ways. Games could even be designed such that the rules by which they are played, and the ways actions and consequences are connected, could be varied from game to game, requiring a player to learn them anew every time. A variety of non-Euclidean spaces could be used, making navigation less intuitive. In any event, abstraction represents perhaps the greatest area of potential for the developing video game medium.

Finally, it should be noted that even the most representational games available, or perhaps even imaginable, will always be to some degree an abstraction ofthe things or situations they are trying to represent or simulate. Will Wright, the designer of the line of "Sim" games from *SimCity* (1989) to *The Sims* (2000), believes abstraction to be a key design element, in the same way that Japanese rock gardens are an abstraction of nature.[20] The player's mind is forced to complete or imagine game details, which engages and involves them more in the game. At the same time, the simplified versions of situations found in video games allow players to feel a sense of order and understanding that may be more difficult to find in their own real lives, an idea that echoes Worringer's notion that uncertain times bring about a resurgence in the urge to abstraction.

Abstraction's role in the video game medium has changed over the years, from perceptual abstraction to conceptual abstraction, but it appears to be both a necessary and inevitable part of the video game-playing experience. Rather than try to avoid or sublimate abstraction, game design can usefully incorporate abstraction, resulting in new gaming experiences and game conventions. Just as computer simulations and mathematical visualization have taken graphic design in directions other than photorealistic representation, abstraction can expand and explore the great potential that the video game medium has to offer.

Notes

1. Quoted in Marilyn Stokstad, *Art History*. Revised Edition. New York: Harry N. Abrams, Inc., 1999, 1114.
2. Many were "hippie programmers," according to Rob Fulop, a programmer who worked at Atari during the 1970s. (Rob Fulop, phone conversation with the author, March 5, 2002.) Atari programmer Bob Polaro also described the Atari workplace environment in a similar manner. See Backiel, Al. "Dinner with Bob Polaro : An interview by Al Backiel." Available online at <http://www.digitpress.com/archives/arc00054.htm>.
3. Scott McCloud, *Understanding Comics* (New York: Harper Collins, 1993), 36.
4. According to Rob Fulop, phone conversation with the author, March 5, 2002.
5. According to Warren Robinett, "the 6502's clock speed was the color clock divided by 3, which was roughly 1.2 MHz." (Warren Robinett, e-mail to author.) The "color clock" refers to the color sub-carrier signal, the frequency of which is 3.58 MHz.
6. The entire image produced was slightly larger, but portions of it were not visible onscreen:

 > The actual TV picture is drawn line by line from the top down 60 times a second, and actually consists of only a portion of the entire "frame".... A typical frame will consist of 3 vertical sync (VSYNC) lines, 37 vertical blank (VBLANK) lines, 192 picture lines, and 30 overscan lines. Atari's research has shown that this pattern will work on all types of TV sets. Each scan line starts with 68 clock counts of horizontal blank (not seen on the TV screen) followed by 160 clock counts to fully scan one line of TV picture. When the electron beam reaches the end of a scan line, it returns to the left side of the screen, waits for the 68 horizontal blank clock counts, and proceeds to draw the next line below.

 From the "Television Protocol" section at the beginning of Steve Wright, *Stella Programmer's Guide* (December 3,1979). Reconstructed online in 1993 by Charles Sinnett. Available online at <http://www.classic-games.com/atari2600/stella.html>.
7. Wright, *Stella Programmer's Guide.*
8. An object would be positioned, and then motion could be added to its relative position:

 > Since there are 3 color clocks per machine cycle, and it can take up to 5 machine cycles to write the register, the programmer is confined to positioning the objects at 15 color clock intervals across the screen. This coarse positioning is fine tuned by the Horizontal Motion___
 >
 > Horizontal Motion allows the programmer to move any of the 5 graphic objects relative to their current horizontal position. Each object has a 4 bit horizontal motion register... that can be loaded with a value in the range of +7 to —8___This motion is not executed until the HMOVE register is written to, at which time all motion registers move their respective objects. Objects can be moved repeatedly simply by executing HMOVE. Any object that is not to move must have a 0 in its motion register. With the horizontal positioning command confined to positioning objects at 15 color clock intervals, the motion registers fill in the gaps by moving objects +7 to —8 color clocks. Objects cannot be placed at any color clock position across the screen___
 >
 > There are timing constraints for the HMOVE command. The HMOVE command must immediately follow a WSYNC (Wait for SYNC) to insure the HMOVE operation occurs during horizontal blanking. This is to allow sufficient time for the motion registers to do their thing before the electron beam starts drawing the next scan line. Also, for mysterious internal hardware considerations, the motion registers should not be modified for at least 24 machine cycles after an HMOVE command. (From sections 7.0 and 8.0 of Wright, *Stella Programmer's Guide)*
9. According to Rob Fulop, phone conversation with the author, March 5, 2002.
10. Rob Fulop, phone conversation with the author, March 5, 2002.
11. The history of color organs is described in William Moritz, "The Dream of Color Music, And Machines That Made It Possible," *Animation World Magazine* 2, no 1 (April 1997). Available online at <http://www.awn.com/mag/issue2.l/articles/moritz2.l.html>.
12. For example, Ms. *Pac-Man* (1981), *Ms. Pac-Man* Plus(1981), *Baby Pac-Man* (1982), *Pac-Man Plus* (1982), *Jr. Pac-Man* (1983), *Pack Pal* (1983), *Pac-Land* (1984), *Pac-Mania* (1987), and *Hyper Pac-Man* (1994).
13. Wilhelm Worringer, *Abstraction and Empathy: A Contribution to the Psychology ofStyle*, trans. Michael Bullock (Chicago: Elephant Paperbacks, Ivan R. Dee, Inc., [1908] 1997), 14.
14. Worringer, *Abstraction and Empathy: A Contribution to the Psychology ofStyle*, 15.
15. Worringer, *Abstraction and Empathy: A Contribution to the Psychology ofStyle*, 24.
16. Ed Catmull, *A Subdivision Algorithm for Computer Display of Curved Surfaces* (Ph.D. thesis, University of Utah, 1974).
17. According to the game's official website, at <http://www.sega.com/games/post_gamegame.jhtml?PRODID=843>.Thanks to Keith Feinstein for alerting me about *Rez.*
18. Tsion Avital and Gerard C. Cupchik, "Perceiving Hierarchical Structures in Nonrepresentational Paintings," *Empirical Studies of the Arts* 16, no. 1 (1998): 59-70.
19. Avital and Cupchik, "Perceiving Hierarchical Structures in Nonrepresentational Paintings."
20. Wagner James Au, "Dispatches from the Future of Gaming : Page 3: Will Wright Speaks...To Arnold Schwarzenegger?" *Gameslice.com* (2001). Available online at <http://www.gameslice.com/features/gdc/index3.shtml>.