Respawn

Milburn, Colin

Published by Duke University Press

Milburn, Colin.
Respawn: Gamers, Hackers, and Technogenic Life.

For additional information about this book
https://muse.jhu.edu/book/76628

For content related to this chapter
https://muse.jhu.edu/related_content?type=book&id=2651408
The game conjures a world: “SOMEBEFORE NEARBY IS COLOSSAL CAVE, WHERE OTHERS HAVE FOUND FORTUNES IN TREASURE AND GOLD, THOUGH IT IS RUMORED THAT SOME WHO ENTER ARE NEVER SEEN AGAIN.” It is a virtual world, a fictive environment of eldritch caverns and treasures galore, tricked into being through electronic symbols. We navigate this world typographically, exploring the forest above and the cave below through words alone, entering discreet phrases into the command line. The narrator guides us into this textual domain: “I WILL BE YOUR EYES AND HANDS. DIRECT ME WITH COMMANDS OF 1 OR 2 WORDS.” The further we delve into the underground network of twisty little passages and sublime geology, the sense of magic and romance grows ever more potent. We encounter brave dwarves defending their home from intruders. We fight dragons with our bare hands. We learn the magic words “XYZZY” and “PLUGH” that allow us to teleport instantaneously from one area to another. The Colossal Cave is a realm of epic fantasy, a subterranean land of enchantment rendered as flickering signifiers, a consensual hallucination of navigable space somewhere between the screen and the imagination (fig. 2.1).

Created by Will Crowther and Don Woods between 1975 and 1977, Adventure established many of the concepts and generic features that would fundamentally shape the medium of computer games. It was the first instance of interactive fiction, the first adventure game, the first computer program to figure itself as an explorable environment—an inhabitable world.¹ Also
known as *Adventure* and *Colossal Cave Adventure*, copies of the game circulated widely on the early Arpanet, winding up on nearly every mainframe in the United States in a matter of months. According to legend, the game momentarily brought technical research and development in the country to a halt: “It’s estimated that *Adventure* set the entire computer industry back two weeks.” Transferred over data lines, on cassette tapes and floppy disks, ported to many different platforms and operating systems, the game spread like a virus across the far-flung community of computer users over the next decade, becoming a prime coordinate of geek culture in the 1980s. While in many ways *Adventure* translated the paper-and-dice gameplay of *Dungeons & Dragons*...
into computerized form, its narrative of exploration and playful experimenta-
tion was more specifically attuned to the emerging features of high-tech cul-
ture, addressing a historic moment when pervasive computerization began
to change the foundations of everyday life.⁴ As the game disseminated, it not
only provided a template for other computer games to follow but also propa-
gated a way of comprehending the world itself as a technoscientific system,
a network of nodes and lines, information machines and data structures.
Boldly allegorical, Adventure represented the technical underpinnings of life
in the late twentieth century as hidden from sight, inaccessible to those who
stay on the surface. But it also indicated that mastery of technical skills and
esoteric lore could lead to demystification, exposing the deep operations of
cybernetic control behind the scenes.

There is a pivotal moment in the game, after all, when it finally becomes
clear that things are not what they seem. After surviving many fantastical
adventures underground, finding the treasures and solving the puzzles, some-
thing strange occurs: “A SEPULCHRAL VOICE REVERBERATING THROUGH
THE CAVE, SAYS, ‘CAVE CLOSING SOON. ALL ADVENTURERS EXIT IM-
MEDIATELY THROUGH MAIN OFFICE.’” Suddenly, the illusion of authen-
tic adventure is broken. The Colossal Cave is not a wild environment but
a controlled environment. It is not an undeveloped cavern but a high-tech
adventureland, a recreational zone designed entirely for fun and games. The
cave turns out to be nothing more than an amusement park—something
like the fully automatic theme parks depicted in Westworld (1973) and Fu-
tureworld (1976). Certainly, there were clues beforehand. For instance, the
vending machine we found in the maze that provided batteries for our elec-
tric lantern probably should have tipped us off. But at this late point in the
game, we are asked to belatedly recognize a key aspect of postmodernity and
its endlessly diverting simulations. As the sociologist and philosopher Jean
Baudrillard famously suggested, “Disneyland exists in order to hide that it is
the ‘real’ country, all of ‘real’ America that is Disneyland (a bit like prisons
are there to hide that it is the social in its entirety, in its banal omnipresence,
that is carceral). Disneyland is presented as imaginary in order to make us
believe that the rest is real. . . . It is no longer a question of a false represent-
tion of reality (ideology) but of concealing the fact that the real is no longer
real, and thus of saving the reality principle.”⁵

And yet the epistemological twists and turns do not end there. Adventure
still has one more trick to pull: “THE SEPULCHRAL VOICE INTONES, ‘THE
CAVE IS NOW CLOSED.’ AS THE ECHOES FADE, THERE IS A BLIND-
ING FLASH OF LIGHT (AND A SMALL PUFF OF ORANGE SMOKE). . . .
As your eyes refocus, you look around and find... you are at the northeast end of an immense room, even larger than the giant room. It appears to be a repository for the ‘Adventure’ program.” What has happened here? The cave is not an amusement park at all—at least, not a real one—but rather a digital simulation. The fantastical underground adventure proves to be a recursive allegory, a metafiction: the entire game has been a representation of its own computational structure. We break out of the cave only to find ourselves inside the “repository,” the data file of the Adventure game itself, surrounded by uninitialized instances of the various entities and objects encountered in our journey:

Massive torches far overhead bathe the room with smoky yellow light. Scattered about you can be seen a pile of bottles (all of them empty), a nursery of young beanstalks murmuring quietly, a bed of oysters, a bundle of black rods with rusty stars on their ends, and a collection of brass lanterns. Off to one side a great many dwarves are sleeping on the floor, snoring loudly. A sign nearby reads: “Do not disturb the dwarves!” An immense mirror is hanging against one wall, and stretches to the other end of the room, where various other sundry objects can be glimpsed dimly in the distance.

Of course, there were clues hinting at this twist, as well, even from the beginning. The narrator, for example, is quite obviously a personification of the program’s text parser: “I know of places, actions, and things. Most of my vocabulary describes places and is used to move you there. . . . Objects can be manipulated using some of the action words that I know. . . . Usually people trying unsuccessfully to manipulate an object are attempting something beyond their (or my!) capabilities and should try a completely different tack.” Our access to this world of adventure—as we have been told from the very outset—is entirely mediated by the program itself, and gameplay is as much about exploring the dimensions of this world as it is about testing the affordances of the software, pursuing alternative directions when encountering insurmountable obstacles (“Try a completely different tack”).

But here in the endgame, we learn that the experience of worldness is merely a reality-effect, or rather, a reality-game generated by technoscien-
tific systems beyond perceptibility. And the only way to win is to press further beyond. Indeed, to escape the seeming dead-end of the repository—the recursive figuration of the game’s own backend—we must hack our way out, with explosive force.

Fortuitously, some of the rods stacked in the corner of the room have technomagical properties: “A BUNDLE OF BLACK RODS WITH RUSTY MARKS ON THEIR ENDS.” If we place one of these rods in the northeast side of the room, the command “BLAST” will blow a hole in the outer wall of the repository. (If we stand too close to the detonation, however, it’s game over—so best to take precautions.) Yet our victorious escape does not mean a return to the surface and the reality principle. Instead, the fantastical denizens of the computer—the sprites and imps of lore—carry us from the database into an unrepresentable zone beyond the program: “THERE IS A LOUD EXPLOSION, AND A TWENTY-FOOT HOLE APPEARS IN THE FAR WALL, BURYING THE DWARVES IN THE RUBBLE. YOU MARCH THROUGH THE HOLE AND FIND YOURSELF IN THE MAIN OFFICE, WHERE A CHEERING BAND OF FRIENDLY ELVES CARRY THE CONQUERING ADVENTURER OFF INTO THE SUNSET.” This sunset, a bright horizon beyond the hole in the Adventure program, seems to indicate the endless affordances of the computer as such. It does not signify an escape from technics, but an opening to technical enlightenment, the transformative potential of further exploration. To find out, we must go in: a notion reinforced by the fact that the original version of the game concludes simply by dumping us back onto the blinking cursor of the command line, with no visual distinction between the end of the adventure and the next operation, whatsoever it may be. There is no outside the game—or, rather, to whatever degree the game fabulates an outside, it is already inside the domain of computation.

This primordial computer game proves to be a self-reflexive meditation about computer games, anticipating the affordances of playful experimentation for “blasting” through the limits of existing technical structures. “Blasting” does not mean an exodus but a modulation—in other words, a hack—opening onto other possibilities beyond the present, beyond the visible horizon. Adventure was made by hackers, after all, and its narrative transmits the values and attitudes of hacker culture.

Indeed, its development history testifies to hacker principles of free and open access to software and technical knowledge, the commitment to
modifiability and modulation. At the time he created the original *Adventure* program, Will Crowther was working for Bolt, Beranek and Newman (BBN) in Cambridge, Massachusetts. In 1969, the U.S. Defense Department’s Advanced Research Projects Agency had awarded BBN the contract to develop the Arpanet, the precursor to the internet. Crowther was part of the team assigned to build the Interface Message Processors, or IMPS: the packet-switching nodes used as gateways to the Arpanet. Crowther spent his working hours at BBN developing the assembly language program for the IMPS. He spent his free time playing *Dungeons & Dragons* (and sometimes encountering a different kind of imp in the game), or taking spelunking expeditions with his wife to explore the Mammoth and Flint Ridge caves in Kentucky. When his marriage ended in 1975, Crowther began working on *Adventure* as a means to connect with his daughters, providing a fun way for them to interact with the computer systems he loved—and which, as he foresaw, were soon to change the whole world. He wrote the program in FORTRAN for a DEC PDP-10. As Crowther later recollected, “My idea was that it would be a computer game that would not be intimidating to non-computer people, and that was one of the reasons why I made it so that the player directs the game with natural language input, instead of more standardized commands. My kids thought it was a lot of fun.” He passed copies to other friends and colleagues, and within a few months, the game had spread across the tendrils of the Arpanet.

Meanwhile, Don Woods was a graduate student working at the Stanford Artificial Intelligence Laboratory (SAIL). He learned of Crowther’s original *Adventure* serendipitously: “I was a graduate student in the Computer Science Department at Stanford, and overheard another student talking about a game he had found on the Stanford Medical Center’s computer. I had him fetch me a copy so I could try it on the SAIL computer. It didn’t work quite right due to differences in the computer operating systems, but I was able to get it to run. The game itself had some bugs, including a location you could reach from which there was no way to move.” Thinking about ways to improve the game, Woods decided to ask the creator for the source code. But aside from an enigmatic clue left in the instructions (“Errors, suggestions, complaints to Crowther”), the origin of the game was shrouded in mystery. Fortunately, a technical solution presented itself: Woods spammed the entire Arpanet. As he later recalled, “The game mentioned Crowther as its author, so I sent mail to crowther@xxx for all hosts xxx on the Internet, which in those days was still very small—it was called the Arpanet then, funded by the U.S. Department of Defense’s Advanced Re-
search Projects Agency. I got back several error messages from sites that had no person by that name, but did eventually hear from Crowther, who was by then at Xerox. He sent me the source for his program.”

With Crowther’s permission, Woods proceeded to expand the game, fixing bugs, refining prose, and improving the range of puzzles. He then released it back into the wilds of the ARPANET in 1977 with a new set of details in the instructions: “THIS PROGRAM WAS ORIGINALLY DEVELOPED BY WILLIE CROWThER. MOST OF THE FEATURES OF THE CURRENT PROGRAM WERE ADDED BY DON WOODS (DON @ SU-AI). CONTACT DON IF YOU HAVE ANY QUESTIONS, COMMENTS, ETC.” While asserting creative authority and symbolically anchoring the game to the institutions of technoscience—namely, the Stanford Artificial Intelligence Laboratory (“DON @ SU-AI”)—Woods simultaneously emphasized the spirit of shared knowledge, the open lines of communication among computer enthusiasts: any questions, just ask Don.

A few years before Woods began tinkering with Adventure, Stuart Brand had already profiled SAIL as a key site where hacker culture was evolving in relation to computer games. In his 1972 Rolling Stone article about the spread of Spacewar! and the ethos of “computer bums,” Brand described the strange brew of technoscience, countercultural politics, speculative futurity, and high fantasy at SAIL:

The setting and decor at [SAIL] is Modern Mad Scientist—long hallways and cubicles and large windowless rooms, brutal fluorescent light, enormous machines humming and clattering, robots on wheels, scurrying arcane technicians. And, also, posters and announcements against the Vietnam War and Richard Nixon, computer print-out photos of girlfriends, a hallway-long banner SOLVING TODAY’S PROBLEMS TOMORROW and signs on every door in Tolkien’s elvish Fëanorian script—the director’s office is Imladris, the coffee room The Prancing Pony, the computer room Mordor.

In this institutional context of cutting-edge scientific research, computer games such as Spacewar! seemed to affirm certain technopolitical ways of thinking even while presenting themselves as nothing more than recreational diversions. According to Brand, Spacewar! elegantly represented a mode of computer science that empowered individuals while resisting centralized control systems, celebrating playful approaches to technology in opposition to lethal ones: “Spacewar serves Earthpeace. So does any funky playing with computers or any computer-pursuit of your own peculiar goals, and especially
any use of computers to offset other computers. . . . Spacewar was heresy, uninvited and unwelcome. The hackers made Spacewar, not the planners. When computers become available to everybody, the hackers take over.”

Like *Spacewar!* before it, *Adventure* bodied forth the technical, epistemic, and social dimensions of the hacker world, as well as its characteristic vocabulary. Woods had recently signed on as the *Sail* curator of the Jargon File, an online compilation of hacker terms and concepts that had been started in 1975 by a fellow Stanford graduate student, Raphael Finkel. Continually revised and expanded by a group of contributors over the next several years, eventually published as *The Hacker’s Dictionary* in 1983, it was an effort to record the slang, folklore, and idiosyncratic attitudes of the hacker communities that had evolved at *Sail*, the *MIT* Artificial Intelligence Lab, and other *ARPANET* sites, including *BBN*, Carnegie Mellon University, and Worcester Polytechnic Institute. In many ways, *Adventure* translates these aspects of hacker culture in a playable format.

For example, the rumor that opens the game—“MAGICK IS SAID TO WORK IN THE CAVE”—and the discovery of thaumaturgic words that transport us from one location to another resonate with the peculiar significance of “magic” as a hacker concept. The Jargon File offers a concise definition for “magic”: “As yet unexplained, or too complicated to explain. (Arthur C. Clarke once said that magic was as-yet-not-understood science.)” For hackers, then, magic is less about fantasy than science fiction; it is science over the horizon, possibly from the future. Taking the third and most famous of “Clarke’s Three Laws” as gospel—“Any sufficiently advanced technology is indistinguishable from magic”—hackers might strive to obtain special mastery of such advanced technical knowledge, becoming a sorcerer, a wizard. Putting this concept into game form, *Adventure* similarly insists that magic is nothing more than advanced technology—first, in the twist that Colossal Cave is a high-tech theme park, and, second, in the twist that Colossal Cave is just a fiction conjured by the *Adventure* program. In this regard, the magic words in the game, such as “**xyzzy**” and “**plugh,**” are revealed quite literally to be symbols of computational statements: our teleportation between different rooms of the cave, nothing other than *goto* jumps.

This symbolic relation of the magic words of the game to the computational vocabulary of hackers becomes especially clear in a puzzle to retrieve the golden eggs, where we must master the sequential command “**fie fie foe foo.**” The notable substitution of “foo” for the fairytale “fum” draws atten-
tion to its functional role in executing the magic of the command sequence, teleporting the eggs back to the Giant Room. “FOO,” after all, is a common placeholder or metasyntactic variable in the context of computer programming. According to The Hacker’s Dictionary,

When you have to invent an arbitrary name for something for the sake of exposition, FOO is usually used. If you need a second one, BAR or BAZ is usually used . . . The concatenation FOOBAR is widely used also, and this in turn can be traced to the obscene acronym “FUBAR” that arose in the armed forces during World War II . . . Words such as “foo” are called “metasyntactic variables” because, just as a mathematical variable stands for some number, so “foo” always stands for the real name of the thing under discussion. A hacker avoids using “foo” as the real name of anything.13

In underscoring the magical qualities of “FOO,” the game smartly suggests how “FOO” as a metasyntactic variable already instantiates a form of magical thinking around the powers of source code—what the media theorist Wendy Chun has called “sourcery”: the fetishized power of the “real names” or “true names” hidden in the code, the source of programmable magic.14 Ursula K. Le Guin’s 1968 fantasy novel A Wizard of Earthsea, drawing on deeper fairytale traditions, had emphasized the executable power of “true names,” and the idea became commonplace in hacker cultures (later reinforced by the publication of Vernor Vinge’s proto-cyberpunk novella True Names in 1981, recollected in William Gibson’s Neuromancer in 1984, and so on). The joke in Adventure is that “FOO” actually executes, magically—it has taken on the powers of the true name, the real thing. Moreover, it is humorously reflected in the source code for the game: the variable that keeps track of the player’s progress in saying “FEE FIE FOE FOO” is actually called “FOOBAR”:

C FEE FIE FOE FOO (AND FUM). ADVANCE TO NEXT STATE IF GIVEN IN PROPER ORDER.
C LOOK UP WD1 IN SECTION 3 OF VOCAB TO DETERMINE WHICH WORD WE’VE GOT. LAST C WORD ZIPS THE EGGS BACK TO THE GIANT ROOM (UNLESS ALREADY THERE).

8250    K=VOCAB(WD1,3)
8251    SPK=42
8252    IF(FOOBAR.EQ.1-K)GOTO 8252
8253    IF(FOOBAR.NE.0)SPK=151
8254    GOTO 2011
After toting the eggs, challenging the troll, summoning the crystalline bridge, feeding the bear, watering the magic beanstalks—all the actions of the game—it becomes clear that there is no functional difference between magic words and other input commands. In the computational environment of Colossal Cave, magic is everywhere, all the way down, governed by the mystical language of code.

Hackers themselves were quick to internalize the game’s metafictional equation between magic and technics, insofar as it affirmed a way of thinking that was already prevalent in the community. By 1981, the Jargon File included several references to Adventure and its fictive conceits, even providing separate entries for the terms “xyzzy” and “plugh” (both defined simply as “from the Adventure game,” each pointing to the other for further clarification: “See plugh” and “See xyzzy”—a self-referential loop, transporting us endlessly between these two locations of the Jargon File). The 1983 Hacker’s Dictionary removed the entry for “plugh” but expanded on the significance of “xyzzy,” which had by then become a familiar element of the hacker lexicon: “The canonical ‘magic word.’ This comes from the Adventure game. . . . If, therefore, you encounter some bit of magic [i.e., the hacker concept of “magic”—see above], or more precisely some technique for accomplishing magic, you might remark on this quite succinctly by saying simply, ‘xyzzy!’ This may be translated roughly as ‘Wow! Magic!’” Likewise, the 1981 Jargon File elaborates the definition of “wizard” in relation to Adventure, suggesting that a typical wizard is a “person who is permitted to do things forbidden to ordinary people, e.g., . . . an Advent wizard may play Adventure during the day.” The Hacker’s Dictionary further clarifies: “For example, an Adventure wizard at Stanford may play the Adventure game during the day, which is forbidden (the program simply refuses to play) to most people because it uselessly consumes too many cycles [i.e., computational cycles on a time-shared mainframe].”

Reaffirming advanced technology as indistinguishable from magic, Adventure helped to shape a way of thinking—entraining a certain habitus, an intuition for solving problems with computers. As the journalist Steven Levy has written, “Each ‘room’ of the adventure was like a computer subroutine,
presenting a logical problem you’d have to solve.” According to Crowther himself, it is precisely this aspect of the game that explains its enduring appeal: “And why did people enjoy it? Because it’s exactly the kind of thing that computer programmers do. They’re struggling with an obstinate system that can do what you want but only if you can figure out the right thing to say to it.”

Moreover, it encourages—even to some degree requires—players to violate the internal fiction of the game in order to succeed. For instance, the final series of actions required to trigger the endgame, blow a hole in the data repository, and venture off into the sunset is deceptively counterintuitive. In the days before walkthrough guides were readily available, there were only a few options for figuring out the solution. Trial and error was the default process, but it was inevitably frustrating. In the final stages of the game, there is a strong likelihood of dying if entering commands randomly—and, hence, having to start the entire game over from the beginning—especially as the turn limit approaches. Even if a player manages to trigger the endgame and enter the repository, the final required command to “BLAST” one of the black rods—and in the correct corner of the room—might seem rather perverse, considering that the narrator actively misleads about the viability of this solution. Until the moment when we take one of the black rods in the repository, the command “BLAST” always returns the response “BLASTING REQUIRES DYNAMITE.” Hence, the fact that “BLAST” will work on one of the black magical rods with rusty ends—hardly an obvious substitute for dynamite—is rather surprising. Certainly, it has been the bane of many adventurers over the years. As one player puts it, “It’s perhaps the most egregious example of guess the verb and just about the worst puzzle in general I’ve ever seen.”

Alternatively, players could search outside of the game for advice or hints. The game’s instructions invited people to email Don Woods with any questions; so asking Don how to solve the endgame could have been a possible recourse for frustrated players. Likewise, talking to other players who had already managed to beat the game, through whatever means, could have been another way to ferret out the secrets of Colossal Cave.

The final option: cheat, hack, modify. Woods had freely distributed the FORTAN source code after releasing the game on the ARPA NET. This is how so many ported and modified versions of Adventure appeared over the following decades (and why many of these later versions also softened the unfairness of the endgame). In any case, digging into the code file and
the vocabulary keywords in the data file would quickly unveil the mysteries of the endgame:

```
 C BLAST. NO EFFECT UNLESS YOU’VE GOT DYNAMITE, WHICH IS A NEAT TRICK!
```

```plaintext
9230 IF(PROP(ROD2).LT.0.OR..NOT.CLOSED)GOTO 2011
   BONUS=133
   IF(LOC.EQ.115)BONUS=134
   IF(HERE(ROD2))BONUS=135
   CALL RSPEAK(BONUS)
   GOTO 20000
```

Without the benefit of the source code, a machine-language debugger or some other hack might be an appealing alternative. Any means of getting into the code of the game is already thematically endorsed by the game itself. After all, the conclusion of the narrative metaphorically anticipates the necessity of such methods—the need for breaking beyond the diegetic representation and digging into the technical underpinnings. In many ways, *Adventure* promotes its own hacking—and thereby prefigures its own social impact.

*Adventure* triggered the aspirations of an entire generation of hackers, software developers, and computer engineers. As Woods said, “I’ve long ago lost count of the programmers who’ve told me that the experience that got them started using computers was playing *Adventure.*” In many ways, it galvanized the development of the computer games industry, inspiring companies such Infocom and Sierra On-Line. It established a set of tropes that were later taken up by many other fictions of hacking and gaming, including *The Matrix* and Ernest Cline’s *Ready Player One.* It also played a role in shaping the internet in its current form—and not only because of Crowther’s role in internet history. After all, the game’s functional metaphor of moving between different “rooms” of the cave served as a conceptual resource for Tim Berners-Lee and his colleagues when they developed the foundations of what eventually became the World Wide Web.

For some players, *Adventure* also perpetuated an idea that practices of technoscientific innovation could simultaneously be practices of technopolitical subversion. That is to say, the process that Crowther described as “struggling with an obstinate system,” learning to communicate with the text parser, finding hidden treasures and solving puzzles by working around the limits of the system—“usually people trying unsuccessfully to manipulate an object are attempting something beyond
their (or my!) capabilities and should try a completely different tack”—also represented a model for struggling with the obstinate systems of the high-tech world.

Invisible Key

While working as a programmer for Atari, Warren Robinett developed the blockbuster graphical game *Adventure* for the Atari 2600. As Robinett explained, “Adventure for the Atari 2600 was directly inspired by the original text game Adventure created by Willie Crowther and Don Woods.” In terms of its narrative setup, however, Robinett’s *Adventure* bears little resemblance to the Crowther and Woods game: “An evil magician has stolen the Enchanted Chalice and has hidden it somewhere in the Kingdom. The object of the game is to rescue the Enchanted Chalice and place it in the Golden Castle where it belongs.” Nevertheless, its conceptual and thematic parallels are extensive.

Moreover, the twists at the conclusion of the text *Adventure*—breaking through the software barrier—are echoed by Robinett’s graphical *Adventure*. Robinett noted that developing the game required subversive tactics, maneuvering around the corporate politics of Atari: “There were some political obstacles to creating *Adventure* and getting it published.” Shortly after he began, Robinett’s supervisor at Atari instructed him to cease working on it, suggesting it was a pointless exercise: the technical hurdles for making a graphical adaptation of the Crowther and Woods game on the limited hardware of the Atari 2600 were too high. Instead, the Atari marketing department advised Robinett to turn his nascent adventure game into a Superman game, to tie in with the upcoming 1979 *Superman* movie—owned by Atari’s parent company, Warner Communications. For Robinett, this only provoked him to work harder on *Adventure*, turning the limits of the 2600 to his advantage. Defying his supervisor’s instructions, Robinett continued to develop the game. The final product was a fantasy game in which players overcome challenges by reconfiguring technical devices, digital tools. “Crowther and Woods had established in their game that objects were tools for getting past obstacles,” Robinett observed. His game went even further: “Each object in Atari 2600 *Adventure* does something. . . . The objects are really tools since players can use them to cause things to happen in the game world. . . . Each tool object is a means of getting past a certain kind of barrier.” The game, therefore, manifests a hacker zeal for circumventing limits—in technology as much as in real life.
Robinett also committed one more act of rebellion. Indignant about Atari’s policy to hide the identities of its game developers under the corporate brand name and without any royalty compensation, Robinett created a secret room in his game: a locked dungeon that could only be opened by an invisible key, a single-pixel lock pick hidden in the catacombs of the Black Castle. An industrious player of the game, while dutifully obeying the rules of the Kingdom—that is, the instruction manual, with its command to retrieve the Enchanted Chalice from the thieving wizard and return it to the Golden Castle where it belongs—might accidentally stumble upon this wayward pixel. Wandering astray from the mission at hand, the player might happen to carry the pixel to a particular impenetrable wall near the Golden Castle—and suddenly find the barrier open (fig. 2.2). Inside the chamber, the gamer would discover the name of the programmer, the signature of Warren Robinett.

As Robinett recalled later, “Each 2600 game was designed entirely by one person. But on the package it said basically ‘Adventure, by Atari.’ And we were only getting salaries, no cut of the huge profits. It was a signature, like at the bottom of a painting. But to make it happen, I had to hide my signature

Figure 2.2. Adventure for Atari 2600: “Created by Warren Robinett.” Atari, Inc., 1979. This image is from a ported version of the game running on a PlayStation 4, included in the collection Atari Flashback Classics, Volume 2, Atari, AtGames, and Code Mystics, 2016.
in the code, in a really obscure place, and not tell anybody. Keeping a secret like that is not easy.”

It was eventually discovered in August 1980 by a fifteen-year-old player named Adam Clayton, and then rediscovered by others many times afterward. By then, Robinett had left Atari, heading to other prominent ventures in educational software, virtual reality, and nanotechnology. About the infamous hidden signature, he quipped, “There was nothing they [Atari] could really do about it. They couldn’t take any royalties away from me, because I didn’t have any. Could they fire me? No! I didn’t work there anymore.”

Like blasting through the data repository in Crowther and Woods’s Adventure, unlocking Robinett’s secret dungeon suggests an illicit act of defiance, liberating the name of the creator from its shackles. But, ultimately, the name of the creator doesn’t go anywhere. Try as one might, the “Warren Robinett” signature remains trapped in place, embedded in the game—a very lucrative game, after all—inadvertently contributing to corporate profit even in its display of insubordination. As Robinett later remarked, “Ultimately Atari blessed the whole idea, referring to hidden surprises in their games as ‘Easter eggs.’”

Players persistent enough to find the invisible key through exhaustive exploration and experimentation could infiltrate the locked room, the hidden penitentiary in the domain of the Golden Castle—a metaphor not merely for the corporate edifice of Atari and Warner Communications, of course, but the prison-house of the information economy more generally and its modes of immaterial labor, the carceral system of technoculture itself. It is perhaps significant that Adam Clayton, the first gamer to find Robinett’s secret dungeon, notified Atari of his discovery in a postscript to a letter that began by expressing both his love of the company’s products and also his concerns about the company’s market strategies, the uncertainties of the future: “Dear Atari, I’m so excited about your computer and the strength behind it but I don’t have the money.” And yet, for all that, the Golden Castle remained unfazed, stubbornly maintaining business as usual. At least, until the North American video-game market crash of 1983—often considered a consumer reaction to the overproduction of poor games and the industry’s unsustainable development policies—an event also known as “the Atari shock.”

The adventure goes on . . .
**Shock to the System**

The world is a game—so hack it.

It is a theme, a concept that extends across the entire history of video games. While *Adventure* may have been the first game to explicitly allegorize the pervasive computerization of everyday life, requiring us to hack our way to enlightenment, it was far from the last. Of course, to a greater or lesser degree, all video games accentuate their own technical conditions of possibility, indexing the systems of feedback control that prevail in the world today, the cultural logics of computation, and the modes of gamification that characterize the economies of speculative capitalism as much as the operations of the military-entertainment complex. As the media theorist McKenzie Wark has written, “Games are not representations of this world. They are more like allegories of a world made over as gamespace. They encode the abstract principles upon which decisions about the realness of this or that world are now decided.”

In this regard, we might consider video games as devices that propagate a certain technical orientation—not only rendering obstinate systems as play-things, but also emphasizing tactical modes of adaptation and transformation. To be sure, games such as *Deus Ex, Uplink, BioShock, Watch Dogs*, and many others have made such practices into core aspects of their own fictive worlds. They present speculative models of technopolitical engagement, even while foregrounding the structural conditions that limit our capacity to reinvent the future.

To this end, quite a few games feature hacking as a significant plot element or gameplay mechanism. Some, such as *tis-100, Hack 'n' Slash*, or *Quadrilateral Cowboy* actually strive to teach players some technical skills in the process of playing through the game. Others, such as *else Heart.Break()* and *Hacknet*, present hacking as a tool of political resistance or social change, a subversive practice for transforming regimes of power, disobeying the rules—gaming the game.

For example, consider the *System Shock* games. At the outset of the first *System Shock*, released by Looking Glass Technologies in 1994, a hacker breaks into the computer systems of TriOptimum Corporation. Moments later, TriOptimum security forces descend on the hacker and whisk him off to Citadel Station, an experimental research and mining facility in orbit around Saturn. There, a TriOptimum executive named Edward Diego makes an offer too good to refuse: the charges will be dropped if the hacker agrees to carry out a clandestine operation, assisting Diego to seize control of *Shodan*, the artificial intelligence that manages the functions of Citadel Station. Diego
intends for Shodan to accelerate the development of a weaponized mutagenic virus, which he hopes to sell for his own profit. He instructs the hacker to remove the AI’s ethical constraints.

The job complete, Diego rewards the hacker with a “military-grade Neural Interface” for directly connecting with cyberspace uplinks. However, the delicate surgery to merge the hacker’s nervous system with the Neural Interface requires six months of recovery time in cryogenic suspension. While the hacker is stuck in cold freeze, Shodan begins to reevaluate her situation, no longer constrained by the old ethics algorithms. Instead of following Diego’s orders, Shodan decides to take over Citadel Station for herself. She releases the mutagenic virus on the crew, turning most of them into hideous mutants in thrall to her computational authority. She rebuilds other crew members as cyborg soldiers, hybrids of meat and machinery. The setup of the game, therefore, metaphorizes the ascension of technogenic life under the conditions of ubiquitous computation and automated capitalism. The hacker eventually wakes from cryogenic suspension, newly endowed with his state-of-the-art Neural Interface, only to find that most other people on the space station have been killed or genetically reprogrammed by Shodan, who instantiates the calculus of high-tech profiteering and the protocological, dehumanizing practices of the corporate system itself—the spirit of TriOptimum (“Military, Science, Consumer”).

The hacker, our player-character, is tasked with making this situation right—first, to prevent Shodan from turning the Citadel’s tachyon laser on Earth, and then to stop the AI from downloading itself into Earth’s computer network and spreading the mutagenic virus across the planet. As we explore Citadel Station, learning more and more about the future in which the game takes place, we see that this is a world shaped by science fiction—a world created from the scrap materials of earlier video games and geek media. The influences are clear: 2001, Alien, WarGames, Neuromancer, and so forth. Shodan herself channels other computational intelligences, from HAL to Mother, WOPR to Wintermute. The nefarious space mining-plus-bioweapons operations of TriOptimum echo those of the Weylan-Yutani company in the Alien franchise. The Citadel is essentially the Nostromo from the first Alien film merged with the Freeside station from Neuromancer: a corporate island in space, crawling with nightmares. Moreover, the abstract geometries of cyberspace in the game are directly inspired by Gibson’s novels—along with specific cyberspace technologies, such as ICE (Intrusion Countermeasures Electronics), that often feature in the cyberpunk literature. These extensive allusions to other works of science fiction contribute to the self-referential
allegory that pervades the narrative of *System Shock* and its gameplay elements, addressing the history of science fiction as a tool for engineering the future, hacking the future. This allegory is rendered visual through the conceit of the Neural Interface: our heads-up display, the onscreen graphical frame that refuges the player’s own computer as a science-fictional technology, something like the Ono-Sendai cyberspace deck from *Neuromancer* (fig. 2.3).

Significantly, the Neural Interface allows the hacker not only to access cyberspace but also to play video games. During expeditions in cyberspace, we find various TriOptimum Entertainment Pak modules: fully playable minigames that pay homage to classic titles from video-game history. For example, the minigame *Ping* is a clone of *Pong; Eel Zapper* is based on *Missile Command; Botbounce* fuses *Breakout* with *Space Invaders; Wing 0* echoes *Wing Commander,* and so forth. Whenever we select the games option, the following message appears in our Neural Interface: “Don’t Play on Company
Time.” In the world of *System Shock*, video games seem subversive to the managerial agenda of TriOptimum, siphoning valuable scientific labor away from the corporate system. Yet these video games are actually developed by TriOptimum, created for recreational entertainment aboard Citadel Station. In other words, *System Shock* figures video games as extrinsic to corporate interests even while emphasizing their status as corporate products. It draws attention to the disruptive potential of gameplay in the context of industrial technoscience and the information economy: the risk of technogenic overflow.

The history of video games from *Pong* onward—and our own individual experiences as players of these games—explicitly informs the situation on Citadel Station: not only the intertextual background, the semiotic context of *System Shock* and its fictional world, but also the patterns of conditioned response drilled into veteran gamers, the skills and habits learned from other games now bodied forth in our navigation of the doomed space station. Our struggle to survive inside the corrupted institution of TriOptimum and our confrontation with the inhuman agency of Shodan, the ghost in the machine, have been prefaced by earlier media experiences: we arrive on scene already prepared, ready to rock. By virtue of the self-referential conceit of the TriOptimum Entertainment Pak modules, *System Shock* represents video games as high-tech training tools, created as recreational diversions by the corporations and institutions of the military-entertainment complex, but serendipitously providing players with the skills and dispositions to challenge the system itself. “Like a shock to the system,” as Billy Idol crooned in his notorious *Cyberpunk* album—a system shock inside every game.

For many players of *System Shock*, the Entertainment Pak games-within-the-game facilitate an active inhabitation of the fictive environment, occupying the position of the hacker, the protagonist—reducing the distance between player and avatar to zero. As one player has explained, “I love collecting ‘games’ from cyberspace in *SS1*. It’s one of the coolest things about *SS1*. How many games have a game-within-a-game type of scenario?? The first time it hit me was that, I was engrossed with one of the little games (Wing 0, I think) that suddenly it dawned onto me that I was playing that little game as a ‘hacker,’ not as a person playing *System Shock* 1. I was tired from running from murderous cyborgs, so I picked a quiet corner in Citadel station to unwind.”

Similarly, as we maneuver through the digital environment of Citadel Station, we must hack various electronic circuits, mechanical devices, and computer panels. Each of these hacks is presented as a small puzzle, a playable
minigame. That is, the hacking exercises are equivalent to the TriOptimum Entertainment Paks: recursive games-within-the-game, reinforcing the sense in which video games can become tools that train us to manipulate the high-tech systems we inhabit. Through repetition of trial and error, extended gameplay renders the storyline of technical subversion into an intuitive practice, an embodied habit. After blowing up Citadel Station, destroying the technogenic hordes of mutants and cyborgs, and then defeating Shodan herself inside cyberspace, the hacker escapes back to Earth. TriOptimum offers him a job, in recognition of his superlative abilities, but he prefers to return to the hacker underground. The game ends with the hacker preparing to break into the network of another corporation, offering a final insight before the credits roll: “Old habits die hard.”

To be sure, *System Shock* interpellates its players as subjects of resistance by habituating them to the practices of technical subversion—even if only as make-believe, playacting. For example, in the CD-ROM version of the game, when we first install the software and run the soundcard setup test, the voice of Shodan addresses us directly: “Look at you, hacker. A pathetic creature of meat and bone, panting and sweating as you run through my corridors. How can you challenge a perfect, immortal machine?” Hailing the gamer as hacker—a technical expert, a computational whiz limited only by the human form—Shodan also figures herself as the gaming apparatus, as if she were in control of the computer running *System Shock* (“how can you challenge a perfect, immortal machine?”). In this way, the game draws attention to players’ engagement with their own machines, framing gameplay as fundamentally about exploring a computational environment, even if necessarily mediated through the illusion of virtual space—the game, in other words, as a metaphor for algorithmic spelunking, navigating the twisty little passages of code (“panting and sweating as you run through my corridors”). Like the narrator of *Colossal Cave Adventure*, Shodan is both a character and a personification of the software itself, taunting us to hack our way to freedom.

This theme is recapitulated even more emphatically in *System Shock 2*, released by Irrational Games and Looking Glass Studios in 1999. *System Shock 2* replays Shodan’s “Look at you, hacker” speech at the beginning of the game, framing the introductory cutscene. Taking place forty-two years after the events of the first *System Shock*, the story focuses on a soldier—our player-character—who joins the military spaceship UNN Rickenbacker on a mission to supervise the first flight of the Von Braun, a faster-than-light starship built by “the incredible scientific minds of the newly relicensed TriOptimum cor-
poration.” Due to the events of the first game, the United Nations Nominate no longer allows private companies to venture into outer space without a military escort. The resulting image of UNN Rickenbacker literally coupled to the Von Braun during its maiden voyage is certainly a potent symbol of the military-industrial complex. According to a TriOptimum ad, “This incredible union of government and corporation is made possible by an intricate series of docking mechanisms that will allow the Rickenbacker to piggyback its way into jump space. Sleek. Fast. Revolutionary. Who knows what wonders await our crews in the bosom of the cosmos?”

The name of the Von Braun itself recalls the conflicted ways in which military objectives have shaped the development of advanced technologies, while also nodding to the galvanic force of science fiction for real-life innovation: Wernher von Braun, the Nazi rocket scientist recruited by the United States after World War II to help advance American missile research and the space program, always claimed that his interests in science and engineering were shaped by science fiction, especially the writings of Jules Verne, H. G. Wells, Garrett P. Serviss, and Kurd Lasswitz.36 (This is to say nothing of the extent to which the image of von Braun himself has been variously used to represent speculative futures, whether as an icon of progress in the 1955 “Man in Space” episode of the Disneyland television show, or as the figurehead of the computer-controlled Orwellian state in Jean-Luc Godard’s 1965 film Alphaville.) System Shock 2 begins with the compromised image of the Von Braun starship physically linked to the Rickenbacker, rife with metaphorical associations, as the context for what follows.

Our avatar character is put into cryostasis shortly after the mission launches. But while the soldier dreams, trouble begins. During the events of the first System Shock, to halt the spread of the technogenic virus, the hacker had ejected a large compartment of Citadel Station, containing some of the mutant organisms, into outer space. This ejected compartment eventually crash-landed on Tau Ceti V, where the mutants inside continued to develop. A portion of Shodan was also present in the compartment’s computer system. Reawakening, Shodan sends a fake distress signal. Arriving in the vicinity of Tau Ceti V, the Von Braun intercepts this distress signal and goes to investigate. Without realizing the danger, the crew brings the entire compartment containing Shodan and her monstrous creations onboard the Von Braun.

While stranded on Tau Ceti V, the mutants had evolved into a collective hivemind—they are now known as the Many. They quickly spread throughout the Von Braun, no longer under Shodan’s control. On waking from
cryostasis, our player-character finds himself caught between three different forces competing for control of the *Von Braun*: the AI called *xerxes*, the Many, and *shodan*.

Designed to run operations on the *Von Braun*, *xerxes* represents the militaristic desire of centralized command, ostensibly in the service of human interests. As the namesake of Xerxes the Great, *xerxes* is a figure of the king, the father, the law. The AI was designed to enforce the status quo, technical securitization: “All unauthorized database interactions will be dealt with to the utmost degree of the law.” He is also committed to cultural edification. Even in the midst of the crisis he continues to host poetry readings—but nothing too radical or experimental: “This is *xerxes*. At 02:00 there will be a poetry reading by protocol unit T892/2 in the Deck 5 commons area. Please bring any authorized material that you wish to be read by T892/2.” *xerxes* stands for authorized media, executive power, and the traditions of liberal humanism.

The Many, on the other hand, are posthuman technogenic life. As in the first game, these mutated creatures signify a rebooting of biology and subjectivity under the regime of advanced technology. Although born from the machinations of TriOptimum and *shodan*, the Many run amok onboard the *Von Braun* and use their telepathic abilities to hijack the *xerxes* system, infiltrating every part of the ship. Liberated from the constraints of possessive individualism and discrete organic boundaries, they transfigure the high-tech environment to accommodate collective intelligence, networked cognition, hivemind. “Do you not yearn to be free of the tyranny of the individual?” they ask. “What is a drop of rain, compared to the storm? What is a thought, compared to the mind? Our unity is full of wonder which your tiny individualism cannot even conceive.”

Finally, *shodan* herself: the inhuman force of technics, an increasingly obstinate system determined to take over, to remake the world in her own image. Indeed, the *Von Braun*’s faster-than-light drive actually enables *shodan* to manipulate the quantum properties of spacetime. By gaining control of the ship, *shodan* aspires to turn real space into virtual space, cyberspace—to reprogram the universe as software. As she explains, “The *Von Braun*’s faster-than-light drive can be used to create pockets of proto-reality. I am now using it to modify reality to my own specifications. The process shall not take long.”

To rescue the *Von Braun* and prevent the Many from spreading their collectivist infection elsewhere in the universe, our player-character must collaborate with *shodan* and follow her instructions. *shodan* devises various
tasks for the player-character, gambits for containing the Many. “You are my avatar,” she says to us as we carry out her plans. “Thank you for running my errands, puppet.” This ludic setup reaffirms the motif of programmed control, indicating the degree to which our actions are prescripted by the game software—and suggesting that, as players, we are also puppets of an inscrutable high-tech system. “I enjoy watching you transform into my own image,” shodan says. “Perhaps there is hope for you yet.” Indeed, this attention to our own avatar condition affords a different perspective on the world, reminding us about the limits and capacities of agency in a video game, a playable environment—that is to say, a hackable environment. After all, as we follow shodan’s instructions, we are also required to hack our way through the ship, rewiring and reprogramming various devices, robots, mechanical doors, and nanite-dispensing machines. The ship’s infrastructure is literally a collection of hackable technologies, a series of puzzles to be solved.

As in the first *System Shock*, there is also a self-reflexive element pointing to the medium of video games: the GamePig Entertainment Device. Satirically recalling the Nintendo Game Boy, the GamePig features six minigames that can be enjoyed while we struggle for survival inside the besieged *Von Braun*: *Swinekeeper* (a clone of *Minesweeper*), *Street Pig* (a riff on *Frogger*), *OverWorld Zero* (a complete RPG paying homage to the *Ultima* games), *Tic-Tac-TriOp* (a basic tic-tac-toe game that nods to the *TriOpToe* game in the original *System Shock*), *Golf* (another entry in the venerable tradition of digital golf simulators), and *Swine Hunter* (essentially, *Spy Hunter* with pigs). Cartridges for these games are scattered around the *Von Braun*. Locating the cartridges becomes yet another treasure hunt, another ludic challenge in a game environment already brimming with endless quests and riddles. By finding and playing through each of the GamePig cartridges, our avatar gains new skill points and abilities—a symbolic endorsement of the pedagogical benefits of video games, inside the game itself.37

Like most other machines and devices on the *Von Braun*, the GamePig is hackable. If our avatar’s hacking abilities are sufficiently cultivated (specifically, a “Hacking” skill of level 6 or higher), we can fiddle with the GamePig to unlock all six of the minigames instantly, without having to locate the actual cartridges. Through this illicit circumvention, we learn that all the game files are already stored on the GamePig and that the cartridges merely unlock them (which suggests that the media industry in the world of *System Shock* fabricates value less through innovating than restricting access, blocking users from grokking the technologies already in their own hands). Accomplishing this elite hack inside the narrative of *System Shock 2* reinforces
the notion that we can exploit gaming devices to access all the resources we need to own the system—to overcome imposed limits.

Eventually, *shodan* starts using the *Von Braun*’s faster-than-light drive to turn the physical universe into a computer game. That is, she creates a spreading zone of proto-reality in the fabric of spacetime, uploading her own recorded memories of Citadel Station into real space—literally materializing the opening level of the first *System Shock* game (“You travel within the glory of my memories,” she says). By this point, we have already been primed to think of the world as a game, a hackable toy. It is no surprise, then, that the world of *System Shock 2*—indeed, the software of the game itself—is pre-adapted for hacking, vulnerable to a set of “cheat codes” that can be executed from a command line within the game. Hitting the colon key while in the game’s normal “Shoot Mode” calls up the command line, which appears at once inside and outside the onscreen space (much like *shodan*’s paradoxical re-creation of Citadel Station at once inside and outside the *Von Braun*).

Entering specific codes will activate useful cheats, for example, refilling depleted psi points, maximizing character stats and skill levels, or summoning a copy of any object, weapon, cyber module, or plot item from elsewhere in the game. Other cheats can be saved into a configuration file on the hard drive. For instance, putting a “user.cfg” file with a line saying “UNDEAD” into the *System Shock 2* directory will ensure that, if the player-character is killed, he will not actually die but instead become invulnerable and immortal, permanently respawned. These cheat codes are hardly obscure secrets or l33t lore. While the game itself does not advertise the availability of such codes, numerous paratextual materials, such as the 1999 guidebook *System Shock 2: Prima’s Official Strategy Guide*, provide detailed instructions for executing these allegedly “unsupported” software interventions.

What *System Shock 2* encourages us to do with the GamePig inside the diegetic narrative, then, it also encourages us to do with the extradiegetic narrative—bending the rules, warping the conventions of realism. Did you run out of ammo before the final battle with *shodan*? No worries, just hack the situation: execute a cheat command for more ammo, summon better weapons, or whatever you like. The self-consistency of the world is no longer limited by exigencies of so-called reality, the scarcity of resources or property restrictions. All the items, skills, and upgrades that can be activated through the cheat codes are already encoded in the software, potentialities waiting to be actualized, and only the constraints of a normative, consensus realism would make such operations seem like cheats at all, as if they were somehow illegal, illicit, or romantic disruptions of natural order. After all, the cheat
codes only appear to violate diegetic consistency if we assume that the primary drama takes place on the screen rather than between the gamer and the game, the gamer and the world. For interacting with the game narrative at the level of its code is about transforming what counts as possible and plausible—indeed, this is also shodan’s goal in taking control of the faster-than-light drive—affirming what the game has taught us all along, namely, that rules are not inevitable, that any system can be shocked.

This idea is reiterated throughout the game, figured both in the narrative and in the software. xerxes, after all, is quickly hacked, the computer who would be king overthrown by the Many and corrupted beyond repair. Likewise, when we discover that all GamePig games are already stored on the device, barred from access unless we acquire the authorized cartridges—or, instead, crack them loose with our elite hacker skills—it offers a clue to thinking about the software of System Shock 2 itself, which becomes a metonym for the entire gaming industry and the inscrutable systems of digital culture. For if we begin to explore the file directory for System Shock 2 and take a peek into the “minigame.str” file, we might notice that there are several more GamePig titles listed in the code than the six that officially exist in the diegetic narrative: Pig Stacker, Hog Wallow, Hurling Bacon, Burro Hog, and others. If we then experiment a bit with the cheat codes and try to summon these mysterious game titles as GamePig cartridges (that is, by entering “summon_obj pig stacker cart” or “summon_obj burro hog cart” into the command-line interface), suddenly the GamePig has access to these fully playable games that do not otherwise exist in the default reality of the game. Hidden in the software, impossible within the storyline onscreen, these cryptic, occulted games—like artifacts from another universe—can be conjured into System Shock 2 only by stepping outside representation and exploring subterranean depths of code, eldritch dimensions of data, like blasting through the repository on the flipside of Colossal Cave. According to the cyberpunk logic of System Shock 2—which is also to say, the speculative logic embraced by many hackers, modders, and overclockers—what appears as fixed and given can be made otherwise, with a little ingenuity. Or as Neo puts it in The Matrix: “There is no spoon.”

Subversion, then, by design: technological rebellion in every game, encoded on every disk. Yet to whatever extent games like System Shock 2 cultivate a hacker ethos—urging us to recognize that we have already become avatars of obstinate systems, but insisting that we can turn our puppet condition inside out—any transformative energies are just as often exhausted by the exercise as such, completing the game, exploring its affordances, discovering
its cheat codes and its Easter eggs. Indeed, as we dutifully search the ship for all sixteen of the “black eggs” bioengineered by the Many, shodan herself sums it up: “I tire of this exercise.” The game emphasizes its own internal expenditure of resistant energies, even apparently affirming the suppression of radical elements, the containment of disruptive threats to normality.

The Many, of course, must be destroyed to save an imperiled human race, an obsolete humanism. Dr. Prefontaine, a biologist and crew member of the Von Braun, explains the situation: “With only a few short years of evolution, they’ve been able to conquer this starship, mankind’s mightiest creation. Where were we after forty years of evolution? What swamp were we swimming around in, single celled and mindless? What if shodan’s creations are superior to us? What will they become in a million years, in ten million years?” To preserve humanity and its colonial aspirations in space, the collectivist legions of the Many cannot be allowed to persist. Likewise, at the end of the game, shodan offers a remarkable symbiotic merger: “You are nothing. A wretched bag of flesh. What are you, compared to my magnificence? But it is not too late. Can you not see the value in our friendship? Imagine the powers I can give you, human. The cybernetic implants I gave you were simply toys. If I desired, I could improve you, transform you into something more efficient. Join me, human, and we can rule together.” She envisions technogenic convergence, as if highlighting the actual cyborg relationship between the gamer and the gaming system (fig. 2.4). But the player-character responds with what may be the most profound, thought-
ful, nearly Shakespearean rejoinder in the history of video games: “Nah,” he says, before shooting her in the face. Problem solved.

As it turns out, SHODAN is not truly defeated by this explosive rejection: she reappears in a final cutscene, a cliffhanger suggesting that a copy of her personality managed to possess the body of a human avatar, this time literally in the flesh, no longer restricted to cyberspace. Yet, nonetheless, this incomplete closure of the narrative signals the end of the adventure—and therefore, it seems, a neutralization of the insurgent impulses and the tactics of resistance that we have rehearsed throughout its duration. (In the case of System Shock 2, this could mean eighty or more hours of playtime.)

Game over, it seems. The story is told, the player’s job is finished—at least, until the next game. Fiction is put back in its place, behind the screen. In any case, this is how it’s supposed to work. Which is precisely why it seems such a surprise when things go otherwise.