Foreword from the Book
<u>Video Game Theory Reader</u>
edited by Mark Wolf and Bernard Perron

Video game history in was made in 1978 when the ideas behind the all-text game Adventure *by Will Crowther and Don Woods inspired Warren Robinett to write a graphical adventure program for the Atari 2600, which he also entitled* Adventure. *In several on-line polls, Robinett's* Adventure *has several times been voted the best cartridge for the 2600, despite the fact that it was among the earliest cartridges to appear. The game also has an important place in video game history as it contained a number of firsts, including the first Easter Egg (hidden feature) to appear in a game; first screen credit; first significant off-screen events; and the first use of multiple screens that cut from one to the next, with conservation of screen direction used to continue the action, making* Adventure *the first to take advantage of cinematic conventions to orient the player in the game's diegetic world. The game contained the first use of identically-shaped characters whose behavior differed; the three dragons were different colors, each had different duties, and moved at different speeds.* Adventure*'s Bat was the first computer-controlled character to have more than one behavioral state (agitated and not agitated). And finally,* Adventure *was the first game to feature graphical, portable objects on-screen (as opposed to being located in an inventory) that a player-character could pick up, use, and drop, and that could be carried by a computer-controlled character as well (the Bat). In this regard, Robinett is among the foremost innovators of the video game during the era when the medium was coming into its own, and one whose work shows an active concern for the theoretical aspects of the medium.[--Editors]*

Foreword

Warren Robinett

It is hard to say what ranks lower on the artistic food-chain than video games. Comic books? TV sit-coms? X-rated films? These rat-like vermin at the bottom scurry to avoid the thunderous footfalls of the towering behemoths of the art world. Everyone knows that:

- Violinists, conductors, and composers are real artists.
- Novelists, poets, and playwrights are real artists.
- Painters, photographers, and film-makers are real artists.

But video-game designers? Is that even art?

The established art forms have their prestigious awards (Grammies, Pulitzer Prizes, Academy Awards). They have academic departments devoted to their study (Music Departments, English Departments, Art Departments). They have God-like practitioners of the past to idolize (Beethoven, Shakespeare, Picasso), and, as arch-angels, the living masters (McCartney, Vonnegut, Spielberg). And for these living masters, has climbing to the top of the Food-Chain of Art benefited their wallets or their sex lives? The true answer to this may perhaps strain the imagination of all but the horniest of nerd-programmers, hacking at 2:00 a.m. on his 3-D monster.

But wait! Despair not, horny nerds! Sometimes rat-like vermin can triumph over towering dinosaurs. Things do change. New art forms do come into existence -- not often -- but, if you think back, there was a time when there were no novelists or poets or playwrights. Before the invention of writing, there was only story-telling and oral ballads. The written stories were a new art form spawned by a new technology (writing). Perhaps the oral balladeers scoffed at the time. But from our perspective in a literate culture, there is skill, technique, and yes, an art, to piling word on word to make a novel or a poem. Likewise, from a perspective slowly emerging, there is not only skill and technique, but also an art, to piling bit upon bit to make a video game.

There is a natural progression in the emergence of a new art form. Often there is an enabling technology that must first be invented and made to work. In the case of cinema, for example, this was the motion-picture camera, projector, and film. Then come the first works exploring the new medium. The pioneers are often clueless, from the point of view of later practitioners, about what you can do with the medium, and a great deal of experimentation occurs, with a few successes, and with some bizarre and interesting relics that soon disappear. If the public likes what they see, they continue to buy, and the medium has a chance to develop. The ideas that work give rise to genres, and all too quickly, the youthful efflorescence is over (*Alas, that Spring should vanish with the Rose/ That Youth's Sweet-Scented Manuscript should close.*) and the genres harden. Critics arise when there are enough works that the public needs help sorting out the good from the bad. And trying to analyze what separates the good from the bad naturally leads to a theory, or theories, about the medium. Throughout the progression outlined here, in a healthy and developing medium, there is a continuous competition among the practitioners making new works. The critics and theorists cannot get started without a body of works to winnow and analyze, and their work is meaningless without a stream of new works being created, presumably being informed (somewhat) by their efforts. The wolf keeps the caribou strong. The players, the designers, the critics, and the theorists are natural members of a healthy ecosystem.

This homespun theory of art-form emergence is based on what I have observed in the field of video games. (For a homespun art theory, there are no doubt theorists waiting, vulture-like, to claw out its eyes and pick its bones. But hey, vultures, I was there and you weren't.) The enabling technology (computers and computer graphics) became cheap enough to reach consumers in the 1970s. A great many video games were created in the first ten years, and genres came to be recognized (shooters, racing games, sports sims, side-scrollers, etc.). Critics were born when the game magazines began rating various games (C+, A-, etc.). As far as I know, this book is the first to focus exclusively on the theory of video games. There is a set of questions that fall to the theorists to address, but before trotting out the Big Questions, let's return briefly to the early days of the medium.

The ideas that we, the early video game designers, had about what video games were, or could be, was based on the games that had been created up to that time. Ivan Sutherland, who pioneered computer graphics in the 1960s, did not make games, but he made an interactive visual simulation (*Sketchpad*) demonstrating compelling possibilities -- for example, a bridge truss whose beams could be pulled with an input device, with a structural simulation running continuously to calculate restoring forces. *Spacewar!*, done by MIT students on expensive research computers, was a game in which a pair of spaceships (represented by small 2-D icons on the screen) shot missiles at each other, while orbiting a central star. Al Alcorn created the first commercial video game "hit" -- *PONG*, an abstraction of Ping-Pong, with a square "ball" bouncing back and forth between the rectangular "paddles" controlled by two players. In the early 1970s, the dominant form of video game was the "coin-op" (coin-operated) game, which was placed in bars and pinball-machine arcades, and paid for by a stream of quarters from the pockets of the players.

When home video game consoles, which could be purchased by individual consumers, became the dominant form in the late 1970s, a rapid change occurred in video games. This change was driven purely by economics. Coin-op games existed in order to suck quarters out of the pockets of players; the games had to exciting, yes, but they could not be long. Typical coin-op games lasted 2 or 3 minutes. But for a consumer who had already paid $200 for his Atari 2600 video-game console and $25 for a game cartridge, games that took an hour or more to complete were not only permissible, but good. The phrase was "hours of play-value." Most of the current video-game genres originated either in the coin-op era or the early console era. Shooters and racing games were feasible as exciting fast-action coin-op games, but the games that took more time, such as adventure games, were first done on home video-game consoles.

My vantage point was as a 26-year-old programmer working for Atari, the company that first popularized commercial video games, at the time when video game consoles first reached the consumer. At that time, Atari gave its young designers almost complete design freedom. Just as I was finishing my first video game on the Atari 2600 console, I got a chance, at a Stanford research lab, to play the original text adventure game, which was called *Adventure.* (Thank you, Don Woods and Willie Crowther.) I decided that this idea -- a journey through a network of rooms, with objects you could move from place to place, and obstacles and monsters to get past -- could work as a video game.

Woods and Crowther's wonderful game used no graphics at all -- it was entirely text. Text described where you were: `You are in a debris room filled with stuff washed in from the surface. A low wide passage with cobbles becomes plugged with mud and debris here, but an awkward canyon leads upward and west. A note on the wall says "MAGIC WORD XYZZY."` Text described objects you could carry: `A three-foot black rod with a rusty star on an end lies nearby.` And text commands were typed by the player to move around and do things: **`GO WEST`** or **`TAKE ROD`** or **`SAY XYZZY`**.

In spite of various practical and political obstacles that I faced -- severe memory limits, my boss telling me it was impossible, the Atari marketing department telling me to change it into a game about Superman --- the idea did work as a video game. *Adventure* for the Atari 2600 console was the first action-adventure game. (It ended up with the same name as Woods and Crowther's public-domain game.) It was also a commercial success, selling one million copies. Both versions of *Adventure* helped to define the genre for later adventure games, such as *Legend of Zelda* and the *Ultima* series.

I had to solve various problems in designing *Adventure*:

- How do you represent an adventure game "room" graphically*? Use the entire screen to depict a map-like view of the current room.*
- How do you represent movable objects graphically? *Use hardware sprites to show them as little icons.*
- How do you represent your "self" and where you are in the game world? *Use another little icon.*
- How do you move within a room? *Use the joystick to move your self icon.*
- How do you move from room to room? *Drive yourself off the edge of the screen into an "adjacent" room.*
- How do you constrain where you can go? *Interpret the graphics depicting each room as walls and passages. Walls constrain where you can go.*
- How do you show the "inventory" of objects you have picked up? *You don't: the player can only carry one object at a time. Show the object beside you as you carry it.*
- How do you pick up an object? *Drive yourself into it. Touching it picks it up.*
- How do you drop an object? *Use the joystick button. (The Atari 2600 joystick had only one button.)*
- What kind of obstacles can be created? *Maze walls, locked doors, dragons that eat you.*

- How do you get past these obstacles? *Create tool-objects: a bridge to cross walls, key to open doors, a sword to kill dragons.*
- How do you make autonomous creatures? *Associate a subroutine executed each frame with a graphical object. Control their motions (towards or away from other objects in the game) by giving them "desires" and "fears".*
- What's the goal of the game? *It's a quest: you must find and retrieve the Holy Grail.*
- How do you fit all this into 4096 bytes of ROM and 128 bytes of RAM*? A good data structure, and efficient coding.*
- How do you get public credit for your cool game when Atari's policy is anonymity of the game designers? *Create a secret room that's really hard to get to, and hide your signature in it. Don't tell anyone until they've manufactured a few hundred thousand cartridges and shipped them all over the world. (This was the first Easter Egg.)*

My website **http://www.WarrenRobinett.com/adventure** has more information on the design of *Adventure*.

These design problems and solutions for *Adventure* illustrate that the evolution of the video-game medium has a *history*. At the time that I write this (2002), the inventors of most of the genres are still alive, each with quirky design histories of their own. They probably still have their design notes somewhere, in a bookshelf or an attic. Back when video games were a non-artform, nobody seemed to care about the design history of early video games. But now, hey... maybe someone should interview these people while they are still alive. If you accept the idea that a new artform is emerging, then interviewing the genre-creators is equivalent, for a Classicist, to interviewing Homer; for an English professor, to interviewing Shakespeare. (I pause to dodge the rotten tomatoes thrown by Classicists and Shakespearean scholars.) The yellowed design notes in the attic are the equivalent of the literary manuscript. There may be other artifacts, as well. For example, I wrote a book on the design of *Adventure* in 1983, but my publisher went out of business just as I finished it, so I have a 20-year-old manuscript that was never published on my bookshelf. If you believe this stuff has value, realize that there are people, artifacts, and relics accessible now, but that after some period of time -- it's hard to say exact how long -- they will have vanished into the bit-bucket of time.

This is a book about video-game theory, and I think a reasonable person might well wonder why a theory is needed for video games. I'll give my answer here. In the video-game ecosystem I described earlier, the ecological niches were player, designer, critic, and theorist. The roles of designers and players seem obvious -- designers design games and players play them -- but what are the jobs of the critics and theorists? I believe that the critic's job is to ask "Which are the good games?" and the theorist's job is to ask "Why?" From those basic questions, many new ones spring up:

What makes a good game? What are the desirable qualities? Fun? Realistic? Playable? Winnable? Responsive?

What are the elements of video games? By "elements" I mean such things as the game world, objects, autonomous creatures ("AI"s), other human players, goals, obstacles, weapons, tools, user-interfaces, controllers, and sounds. For comparison, some elements from which novels are built are: plot, character, description, and dialogue; and for poems: alliteration, meter, and rhyme. Some elements from which music is built are melody, harmony, rhythm, repetition, tempo, dynamics, and phrasing. Can we identify the essential and fundamental elements of video games?

What are the genres of video games? People seem to naturally want to group works into classes according to their similarities. What are those classes? These would be analogous to the genres we find in music (classical, rock, jazz, country) and among novels (detective novels, romance novels, historical novels).

What are the principal forms of video games? (For comparison, some literary forms are: novels, poems, plays, and essays.) What are the powerful techniques? (Some literary techniques are suspense, humor, irony, metaphor, metonomy, and exaggeration.) If you compare turn-based games (such as online card games), branching-video games (such as the classic *Dragon's Lair*), and real-time simulations (such as *Doom*), these exemplify very different types of gaming experiences. What terms should we use to talk about these differences?

Of course, these are only my ideas about the questions that theorists ought to ask. The variety in this book illustrates that video-game theorists -- now enjoying their own youthful efflorescence -- currently get to write their own job descriptions.

***

For my part, I will now follow my own advice, and try to analyze *why* my own game *Adventure* succeeded.

My main breakthrough, I think, was figuring out how to adapt the adventure game idea from its birth medium (with the player reading text descriptions and typing text commands) to the video game medium (with color, motion, animation, sounds, and joystick controllers). There were three powerful ideas in *Adventure* that were, at the time, different from how most video games were done.

A large game world (in this case, the network of rooms) that was bigger than a single screen, and which could be explored by the player.

Objects that the player could pick up and move around, which functioned as tools to do things in the game world, and which interacted with one another.

Creatures which moved around on their own within the game world, initiating actions. ("AI"s, in modern terminology).

Putting these elements together in a video game produced what we now call an action-adventure game. This has shown itself, over the last few decades, to be a very fertile genre. At the time that *Adventure* was designed, game designers were very experimental and were trying to figure out what you could do that was cool with this new medium of interactive computer graphics. I think you could now say that I discovered one of the "sweet spots" in the medium.

Of course, to make a finished and playable game, the details were important -- what the objects and creatures did, and what they looked like, how the game world was laid out, and how the controls worked. Here is a distillation of my meditations on why *Adventure* worked so well.

Story
Simple, understandable story, theme, and goal. The story (quest for the Holy Grail) was consistent with the fantasy theme (Dungeons and Dragons) and the goal of the game (find and retrieve the Grail). The Atari marketing department confused this a bit by renaming the Holy Grail to be the Enchanted Chalice, but it still worked.

Nice-looking box which conveyed the theme. Good manual.

Controls
The controls for moving yourself were intuitive. I used the joystick for what it is best at -- moving an icon in 2D on the screen.

Square cursor (self-icon) and walls. Most games which followed *Adventure* have used an animated character as the user's avatar. An advantage to the square cursor is that it is easy to see when the cursor will run into a wall, and where the paths are. The visual feedback (jiggling cursor) when running into a wall was valuable. You could also slide along a wall when the joystick was attempting a diagonal movement. Since you spent most of your time in the game moving through mazes, it was important to have this interaction smooth, intuitive, and glitch-free.

The user interface for grabbing and dropping objects was very easy to learn and remember. There was something simple and satisfying about grabbing objects and carrying them around. The "picking-up" and "putting-down" sound effects used 3-note ascending and descending arpeggios to convey "up" and "down", which people seemed to readily understand.

Allowing only one object to be carried at a time was a good decision. It simplified the user-interface. It meant the game could stay always in real-time (never going to an inventory screen). It created strategic choices. (Should you carry the weapon or the treasure?)

Technical enablers
Efficient, pointer-based data structures for the room-list and object-list.
Treating all objects in a unified manner, including the square cursor (the player's self), tool-objects, and creatures. All these objects used the same routines for movement, room-to-room motion, and collision detection.

The chase/flee priority list data structure, which allowed me to efficiently specify complex behaviors for the creatures in the game.

The space-efficiency and time-efficiency of these data structures, together with the uniform treatment of all object types, is the main reason I could fit such a relatively complex game into the game cartridge's small (4K) ROM memory.

Game world
The game world was a good size: 30 rooms, divided into 8 regions. It was small enough that you could explore and become familiar with all of it.

Disjoint regions. The game world was partitioned into disjoint regions when the castle doors were locked. Also, two of the mazes consisted of two disjoint parts, requiring use of the bridge to get into the other part of the maze.

Mazes. The multi-screen, non-planar mazes were interesting.
Object permanence -- objects and creatures were never created or destroyed. Each one was always somewhere in the game world. The game simulation proceeded independently of whether you could see an object or creature, so that you sometimes saw the effects of things that happened off-screen, such as the Bat picking up a new object. This gave the game world a very believable feeling.

No randomness during game-play. Algorithmically-generated complex behavior is more interesting and understandable than just generating a new monster every so often based on a random number generator.

Objects as tools to get past obstacles. This meant that with a simple user interface (that let you move yourself and pick up one object at a time), you could do a lot of things. Each new tool-object in the game gave you a new capability (sword, key, magnet, bridge, dot) without complicating the user interface.

Problem-solving. If a needed object (say the Black Key) was behind an obstacle (say the Green Dragon), this spawned a sub-goal. Get the sword to get past the dragon; then use the key to get into the castle.

Object-object interaction was easy to understand. These interactions were triggered by overlap of object shapes (which were called “collisions”). For example, the dragon was killed by having the sword touched to it. This is like in the real world when two objects touch each other, they affect one another.

Creatures (objects that moved around on their own, initiating actions). The creatures in the game were similar to animals in real life -- they moved around, they did things, they had “motivations” which could be inferred from their behavior. Each creature was implemented by a subroutine that controlled what it did, which was executed 20 times per second. I came up with a data structure to represent a creature’s goals -- a prioritized list of objects and whether to go toward or away from that object (thus representing the creature’s “desires” and “fears”). The subroutine went down its priority list until it found an object on the list in the same room with it. Then it went towards the object, or away from it. (I called this chasing and fleeing.) In other words, it responded only to its highest priority object in the same room with it. Each creature had its own priority list, and so each creature had different behaviors. There were 4 creatures altogether -- three dragons and one bat. Since each of the three dragons had slightly different priority lists, they had different behaviors. This was a pretty good scheme, because it modeled limited perception (creatures couldn’t “see” across room boundaries), and it allowed a creature to “change its mind” when a new object came into the room.
Surprises. Bat steals your sword. Bat carries dragon. (This always got a laugh the first time someone saw it.)

Display
Using animation of the dragon graphics tied to its state in the game (Chasing, Biting, Swallowed-You, and Dead) worked extremely well.

Animation of the bat flapping its wings was effective.

Tying short sound effects to game events was effective.

The sound effects were good, given the Atari 2600 sound hardware. The sounds evolved over time, which was made them more interesting than the sound effects of most contemporaneous Atari 2600 games.

The color-cycling through the entire color palette for the Enchanted Chalice worked very well to show that it was important and magical.

(Flaw) The dragons looked like ducks.

(Flaw) The objects flickered when there were more than two of them in the same room. There wasn't much I could do about this, given the 2600 hardware, but it was still ugly. However, the flickering did provide a clue to the existence of the Gray Dot, which in turn provided a clue to the existence of the secret room.

Playability

Chase/bite cycle. The dragon would chase the square cursor that represented your self. If the dragon touched you, it emitted a roar, displayed the Biting graphics, and paused for a fraction of a second. If it was still touching you after the brief pause, then it ate you (moving the square that represented you into the dragon's stomach). If you had managed to recoil away from it, then it resumed chasing you. Thus, the interaction could go like this: Bite-Recoil-Bite-Recoil-Bite-Recoil-Escape. Or Bite-Recoil-Bite-Recoil-Swallowed. I felt that this was a more interesting and less frustrating interaction than having it kill you the first time you touched it.

Bat and dragons could go through walls, but you had to follow the maze paths. This created a good balance in the game since the player is actually smarter than the simple AI routines of the bat and dragons.

Reincarnating when you were eaten by the dragon. Simple and understandable. Leaving all the objects where they were was a good decision. This meant getting killed did not cause you to start over completely. But it did penalize you. Bringing all dead dragons back to life when the player reincarnated was analogous to being vulnerable in the game of bridge. The closer you got to winning, the more you had to lose.

I chose not to have a timer in the game to emphasize exploration and problem-solving, which I felt would not be enhanced by arbitrary time limits.

You can win the game. In many video games of the period (for example, *Space Invaders*), you just kept playing until you finally got killed. In such games, you could not win.

Progressive difficulty levels. Level 1 was designed for beginners, and Levels 2 and 3 were harder. Flipping the difficulty switches made the dragons significantly more challenging.

Variety. Random object placement at the start of Level 3 (analogous to shuffling the cards before a hand of bridge) gave the game much more variety. The bat, which moved around objects, kept the game from being *too* predictable. It had enough variety to not be a pure puzzle, which can be solved the same way every time.

Secret room

I created the secret room in order to hide my signature in the game. Even though each game cartridge for the Atari 2600 console was created by a single person, Atari was keeping us game designers anonymous, which I found irritating. Also, I was kind of proud of the game. It was hard to figure out at the conceptual level, and hard to implement, too. And I had to fight through my boss at Atari telling me that it was impossible and not to work on it (which I ignored). Then when I had it half-implemented, the marketing department saw it and liked it (allowing me to continue working on it), but they then told me to keep the rooms and objects, but change it into a game about Superman, because Atari's parent corporation, Warner Communications, also owned the first Superman movie, which was coming out a few months later. But I stubbornly kept to the Dungeons-and-Dragons theme (while my co-worker John Dunn took over a copy of the half-finished code and did the Superman game). I finished *Adventure*. No royalty, no bonus, no pat on the back, not even a pizza. Atari really didn't treat its game designers very well during that period. All in all, I was pretty satisfied with the little surprise I left hidden in the final game code I handed over to be manufactured. Then I quit.

Getting into the secret room was pretty hard. You had to first discover the Gray Dot hidden in an inaccessible part of one of the mazes, and then by trial and error, somehow figure out that the Gray Dot would get you through a certain wall into the secret room.

Once into the secret room, well hey, why hold back at that point? My name filled the screen like a throbbing, multi-colored movie marquee.

Also, it was a bit of an experiment. I wasn't sure any player would discover the secret room. It was pretty obscure. But it turned out that some kids mapped all the mazes, and therefore found the little chamber that contained the Gray Dot. Then they went crazy trying to figure out what it was for, refusing to let their parents turn off the video game or the TV for weeks... The Gray Dot was one pixel in size, the smallest possible object. It was my attempt at irony, I guess. Small and insignificant-looking, yet important. (Pay no attention to that little man behind the curtain.) At least a few kids did discover how to get into the secret room on their own, without any clues other than those in the game. And then the rumor mill started churning.

Having a really hard-to-find secret place in the game, that was so secret it was hidden even from Atari was kind of cool. It fueled a good rumor buzz. My model in creating the secret room had been the secret messages supposedly hidden in Beatle records in the late 1960s -- if you played a certain Beatle songs backwards, you could supposedly hear the words "I buried Paul" -- and then people went crazy, playing every Beatle song backwards, hunting for secret messages.

For the players, the secret room was the meta-level, the way to truly beat the game and get to the real conclusion. For me, it was the meta-game I was playing with Atari management. They had the power to keep my name off the box, but I had the power to put it on the screen.

***

Many things about video games and the culture that surrounds them have changed enormously since *Adventure* was released 23 years ago. Yet some other things have not changed at all.

Back at Atari in the late 1970s, each game cartridge for the Atari 2600 console was created by one person. You had the idea, wrote the program, created the graphics, did the sound effects, chased down bugs, tested the game on kids, revised it until you were satisfied, and wrote a draft of the game manual. This made sense at that time, because with only 4K of ROM memory available to hold the game program, it only took a few months of programming to fill up the ROM. And not only was the memory extremely limited, but the processing power was also very limited (the Atari 2600's processor was a 1.2 MHz, 8-bit processor). To top it off, the display, although flexible, was also extremely limited, providing only two decent hardware sprites for displaying moving objects on the screen. It was the designer/programmer's job to make the trade-offs and come up with an interesting game, given these resources. A finished game cost Atari 4 to 6 months salary for one programmer, which amounted to less than $10,000.

Nowadays, game consoles have thousands of times the memory capacity of the Atari 2600, and the processors are also thousands of times faster. Games are normally made by teams of several dozen people, with the work done by several types of specialists: designers, producers, writers, programmers, artists, sound-effects specialists, and musicians. Budgets of more than a million dollars for a game are common. The game world is usually 3-D in current games, not the 2-D of yesteryear.

However, our perceptual and motor systems have not changed. Thirty frames per second still produces the illusion of smooth motion. Human visual acuity is still one-sixtieth of a degree. Reaction time is still around 150 milliseconds.

And little boys haven't changed. (Not even the ones who are now 30 or 40.) They still like to blow up bad guys, fight scary monsters, and drive noisy, powerful vehicles.

Many billions of dollars and billions of hours are now spent each year on video games. We've tried quite a few things by now. Will another factor of 10, or 100, or 1000 in memory, or computing power, or textured polygons enable amazing new game experiences that are not yet possible? It's hard to say.

It's a good time to think and debate about what interactivity is, what it means, and what it could be. It's a good time for video-game theorists to analyze, to make us think, to question old dogmas, to formulate new principles, and to ask the questions the rest of us didn't even know enough to ask.

Warren Robinett
Chapel Hill, North Carolina
December 2002