

Chapter 6:

Interview: Ed Logg



Asteroids, **Centipede**, and **Gauntlet**. If there was ever an impressive track record for a game designer, that is it. Throw in some lesser-known classics such as **Super Breakout**, **Millipede**, **Gauntlet II**, and **Xybots** and you have a truly unequalled career. Ed Logg designed and developed all of those titles at Atari back in the heyday of the arcades. These days, designing games for the coin-op market seems to be a dying art form, with so much of the industry's attention shifted to the home market. Today Logg continues to work in game development, adapting popular Atari arcade games such as the **San Francisco Rush** series to consoles, including the Nintendo 64 and the Sega Dreamcast. To look at them, the classic arcade games seem quite simple, but it is that simplicity which forced their designers to refine them to the point of perfection. Logg's classic coin-op games remain some of the best computer games ever made, and the insight designers can gain from studying them is enormous.



What was it like working at Atari in the late '70s and early '80s?

We were young and energetic. I imagine it is very similar to the atmosphere at most Internet startups these days. We were a relatively small group in the Coin Operated Games Division. This allowed everyone to know everyone else. Ideas and pranks flowed freely. Since we were working on a new medium we could do anything and it would be “new.” Even games like *Lunar Lander*, done by Rich Moore, which had been done originally years before, were new to our audience.

Where did most of the ideas for the games come from?

The ideas came from many sources. For example, Owen Rubin, another engineer at Atari, told me Nolan Bushnell had suggested to him an extension of *Breakout*. I took his idea and added many of my own to create *Super Breakout*, my first commercial success. The idea for *Asteroids* came from Lyle Rains, who was in charge of engineer-



Asteroids

ing at the time. He got the idea from a previous coin-op game. *Xybots* came from a challenge by Doug Snyder, a hardware engineer at Atari. We wanted to do a multi-player *Castle Wolfenstein*-like game but we had no “bit-map” hardware. So I created an algorithm based on 8x8 stamps and he did the hardware. *Centipede* came from a list of brainstorming ideas. Atari would go off-site each year to think up new ideas. One of those ideas was “Bug Shooter” which was used as a starting point for *Centipede*.

Management had reviews where they would come in and play the game and give feedback. Sometimes the consensus was negative and a game could be killed. Most often it would continue until it could be “field tested.” This meant it was left to the players to determine how much and for how long the game earned. However, sometimes good suggestions came from these reviews. The most important one of all was a suggestion made by Dan Van Elderen, who was in charge of engineering. He asked me why we could not shoot the mushrooms in *Centipede*. Yes, the



mushrooms were originally static. It was his suggestion that led to the breakthrough that made this game fun.

Were you excited to get into game development at Atari?

Actually, I had been doing games for many years on the side, while in high school, at Berkeley in the '60s and also at my first job at Control Data Corp. I ported *Star Trek* and the original *Dungeon* game between Stanford's and CDC's computers.

I had built a home computer a year or two before joining Atari, just to create and play games. I had been to a Pizza Time Theater and played *Pong* and *Breakout*, so I was well aware of the coin-op business. I had also played games and was very inspired by a prototype of the Atari VCS (2600) at a Christmas party in 1977. So the change in employment seemed natural for me. At the time I thought it was great for them to pay me to create and play games.

***Dirt Bike* was your first game for Atari, but I understand it didn't make it into production. What sort of game was it?**

This game was started by Dennis Koble who went on to do many consumer titles. It was a game similar to *Sprint* except you drove a dirt bike and the control was a set of handlebars that could be used to steer the bike instead of a steering wheel.

We field tested the game and it earned enough money to make it good enough not to kill outright but not good enough to make it into production. However, I had made *Super Breakout* at the same time I was working on *Dirt Bike*. No one at Atari had ever worked on two games at once before. *Super Breakout* had earned a large amount of money, and this probably led to the decision not to build *Dirt Bike*. I was not disappointed considering the success of *Super Breakout*.

What was the genesis of *Super Breakout*?

The original idea included six variations on *Breakout*. I envisioned three released games with two variations in each game. However, in actual play there was one overall favorite, Progressive Breakout. In the end we put three variations in one game: Progressive, Double, and Cavity Breakout. The variations that did not make it were more vertically oriented and I had to agree they were not as fun.

Were you given a lot of creative freedom on *Super Breakout*, or were you constrained since it was a sequel to a previous hit?

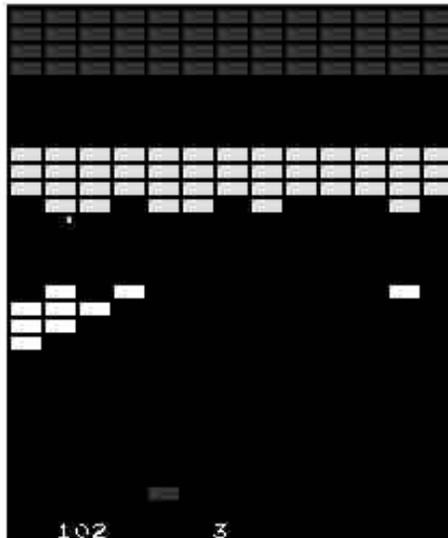
To me, *Super Breakout* was not a sequel. Remember the original game was not done in software. The code had to be created from scratch and the gameplay was completely different from the original even though we used the same controls.



I was given freedom because I was doing the title without any official sanction. It was not the last time I would do that, either. Games could be done in a short time in those days, which meant you could make something fun before anyone even noticed you were doing anything different.

Maybe I should explain how we were developing games in those days. We had one main Digital computer which had the cross assembler for our 6502 based games. We had several gals who would enter our handwritten pages into our programs and give us back a computer printout and a paper tape. Yes, you heard that right. We would then feed the paper tape through our

development system into the RAM replacing the game ROM on the PCB. We would debug this using primitive tools and a hardware analyzer and write our changes on the paper printout. Since this process left time between the debug session and the next version, I used this time to develop a second game. I would just swap the graphics PROM (yes, we created the graphics by hand ourselves), and load the new paper tape.



Super Breakout

That's really astonishing that you ever developed a game using such primitive methods. How did you manage to fine-tune your game with such a long time between versions?

Well, actually, I was very good at just patching RAM with new instructions, so it was easy to see what small changes did to the game. We also had an HP analyzer that we could use to trap on many conditions, which allowed us to find many bugs that many development systems cannot even do today. Actually it was possible to do some new coding while you were waiting for your last changes to be made, so less time was lost than you think.

But you would certainly agree that modern development tools have made game development easier?

There are several issues here. First, back then we often knew everything about the target hardware, which made it easier to see what was going wrong. Today, the target hardware is often hidden from us and there are several layers of software which can make debugging or doing what we really want to do difficult. So in this sense it is much harder now. Also these modern software or hardware layers are



often not documented, documented incorrectly, or just getting in our way. Second, the hardware has gotten very complex with interactions between the many bytes causing all sorts of problems. Third, the processors have become very complex, causing all sorts of debugging nightmares, especially in dealing with the caches. Fourth, today there are many programmers working on a game and it is easy to mess up one of your coworkers.

Surprisingly, the development environment has not gotten any faster over the past few years despite the great increases in the computing power and RAM. As an example, some of my files on my 25 MHz Mac IICI with 6 MB of RAM compile and link in the same time or faster than files on a 550 MHz PC under NT with 512 MB of RAM. Even the same project on my 150 MHz Indy builds faster than my 550 MHz PC. I firmly believe that every tool developer should be given the slowest possible system to use to develop their software! Otherwise, we are doomed to continue to run no faster with each new upgrade.

The modern tools are so much better than the old method, it is hard to imagine how I could have done so well, but you mustn't forget how much time is spent learning each new software tool, processor, and operating system these days. In addition, the amount of time wasted chasing after bugs on new systems because I did not understand some other hardware or software is quite large. But I would not want to go back to the old tools unless the processors, hardware, software, game concepts, and team sizes were much simpler.

I've never seen your next game, *Video Pinball*. How did it play?

It simulated pinball by using a half-silvered mirror with a monitor below the mirror and the graphics for the play-field above the mirror. The monitor would show the flippers and ball, which gave the impression the white ball was on the play-field. The play-field actually had LEDs controlled by the program which simulated lit targets. In addition, the control panel was hinged, which allowed the player to "nudge" the cabinet to give the ball some English. I did not think this game up. I believe it was Dave Stubben's idea.

How did you hope to convince players to play *Video Pinball* instead of the real thing?

I did not believe *Video Pinball* would be successful and I was asking that exact question. However, there were places video games could go that a large pinball game could not. In the end, the game earned more than I had expected and it was a commercial success. I must say I was wrong on my first impressions, and that does not happen often.



Was it hard to work on a project that you did not think would be any fun? Did the final game turn out to be entertaining?

The gameplay was fun but no comparison to a real pinball game. I was surprised that it sold as well as it did. Yes, it was hard to work on an idea that I did not think would work well. But I was young and motivated . . . What else can I say?

Where did the idea for *Asteroids* come from?

Lyle Rains had suggested to me the idea of a game where the player could shoot asteroids because there had been an earlier coin-op game with an indestructible asteroid that the players kept shooting instead of pursuing the intended goal. I told Lyle we would need a saucer to force the player to shoot the asteroids instead of



Asteroids

wasting time. I also suggested breaking the rocks up into pieces to give the players some strategy instead of just shooting the larger rocks first.

Lyle gave me the idea. People often attribute the success to one or the other of us. I would probably not have come up with the idea on my own and if someone else had done the game it would most likely have been totally different. So in truth, we should both be given credit for this idea. Come to think of it, without the vector hardware, *Asteroids* would not have been a success either. So there are many people and events that led to its success. I am very glad to have been there at that time and place.

The game changed very little in development from the original idea. I did make two saucers, one dumb and one smart. I made one fundamental change near the end of the project that had far-reaching implications. Originally, the saucer would shoot as soon as the player entered the screen. Players complained, and I agreed, this seemed unfair. Often the saucer was not visible just off the edge and if it started next to your ship you had no defense. So I added a delay before his first shot. This, of course, led to the “lurking” strategy. While testing, I had actually tried to lurk at one point and decided it was not going to work, which shows you how well the



game designer can play his own game.

Were you surprised by *Asteroids*' success?

I was not surprised by its success. It sounded like a fun game when I played it in my mind. Even after the first few weeks, people would come by and ask when they could play. That was a sign your game was fun!

Even when we field tested the game for the very first time, I saw a player start a game and die three times within 20 seconds. He proceeded to put another quarter in. This tells me the player felt it was his fault he died and he was convinced he could do better. This is one of the primary goals a game designer tries to achieve and it was clear to me *Asteroids* had "it."

Back there you mentioned that you played the game out "in your mind." Do you find that to be an effective technique for predicting whether a game will be fun or not?

It is a skill which I find works well for me. I also play devil's advocate with my ideas: I ask myself "what can go wrong?" or "will players be confused by what I am presenting?" I find that some designers often are so married to their ideas that they will not accept the concept that maybe it just won't work. I cannot tell you the number of great ideas I have had that I "played out" in my mind that turned out to be bad ideas.

I am one of the few designers I have ever met that has actually killed many of his own games. I think this is a good trait. Why waste another year to two if the gameplay does not play like you expected?

Did you work on the sequel, *Asteroids Deluxe*?

I did not do *Asteroids Deluxe*. It was done by Dave Shepperd. I was promoted around that time into a supervisor role. I believe I was also leading the four-player *Football* project. So I was busy. I have no problems doing sequels if that is the best course of action. I had some new ideas, so I wanted to do *Millipede*. *Gauntlet II* was a logical choice since Bob Flanagan, my co-programmer, and I knew the code and this was the best game concept we came up with.

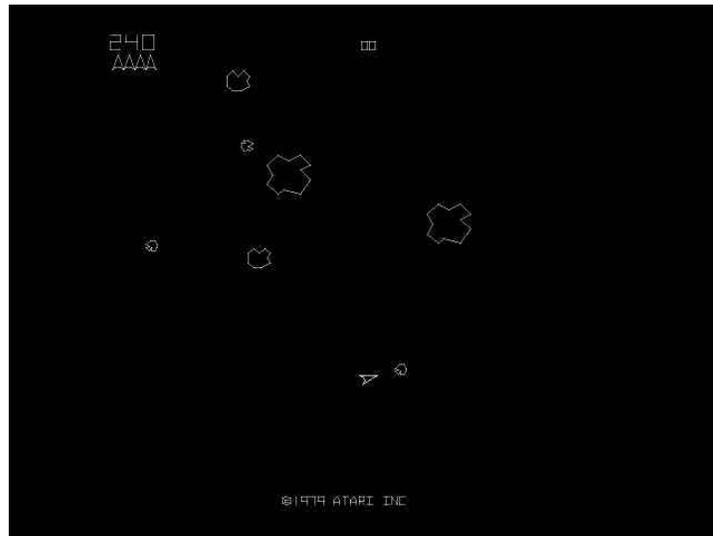
After *Asteroids* you didn't make another vector-based game. Did you not like working with the hardware?

Actually, I loved vector hardware for the reason it allowed me to put up high-resolution 768 by 1024 pictures. However, the industry was just moving over to color monitors at the time. Dave Theurer did do *Tempest* as a color vector game, but the color mask on color monitors did not permit high resolution. Besides, you could not fill the screen with color on vector-based games, so that medium died with the advance of color games.



Wasn't *Asteroids* the first Atari game to have a high-score table?

Actually, *Asteroids* was not the first game; there was another game that used it just prior. I thought the idea was a great way to preserve your score and identity for the world to see. So I added it to *Asteroids*. I see it as filling the role of graffiti. Now it is standard, of course, and the industry has added battery-backed RAM or EEROM to save it permanently.



Asteroids

Around this time you created the *Othello* cartridge for the Atari 2600. I understand you studied AI while at Stanford. Did the *Othello* project grow out of your interest in AI?

No, actually *Asteroids* showed more influence from my Stanford experience. While I was at the Stanford AI Lab, I had played *Space War* on their PDP machines. I had also played a coin-op version of this in the Student Forum coffee shop. In my mind, this was the first video game. *Pong* certainly was the first commercial video game. Anyway, the spaceship design in *Asteroids* was a copy of the original *Space War* ship.

I had played *Othello* as a board game and I was intrigued by possible strategies. So I worked on this game at home and developed an idea that the game could be played by pattern matching without any AI. In other words, the computer does not look ahead at your replies to any of its moves, which was the standard AI approach at the time. So really the *Othello* game I did had no AI. It was good enough for the beginner and average player. It was not an advanced game by any means. Besides, the 2600 had only 128 bytes of RAM so there was not much space to look ahead.

In fact, Carol Shaw had done the hard part by providing me the kernel which drew the pieces on a checkerboard. The 2600 was extremely difficult to do anything complex on. It was intended to do *Pong*-style games. You spent all of active video counting cycles to draw the screen. This left Vblank to do any thinking or other work. There was limited RAM so nothing complex could be saved in RAM. *Othello*



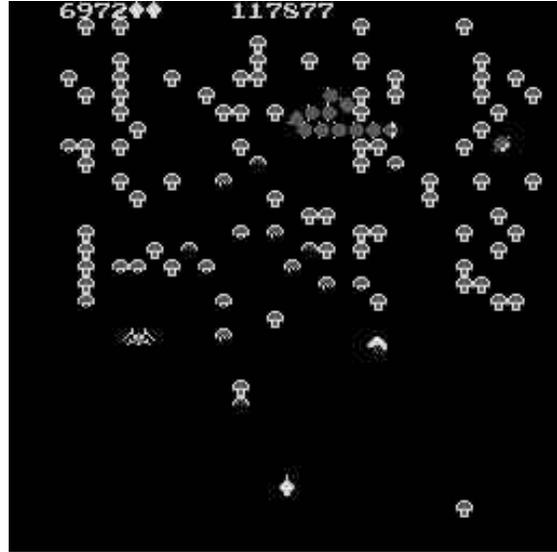
was 2,048 bytes. Most of this was the kernel. So I often spent time trying to eliminate a few bytes to add something new.

Was *Centipede* your next game?

No, as I mentioned I was a supervisor at the time. I was project leader on four-player *Football* and a kit to upgrade the plays on the original *Football* game.

On *Centipede*, I thought up the idea of the centipede segments and the way the legs moved. I do not believe it was mentioned in the original “Bug Shooter” brainstorming idea. In fact, no one has ever stepped forward to claim “Bug Shooter” as their idea. Maybe it was due to the finished product being so much different from the original idea. I had assigned a new programmer,

Donna Bailey, to do the programming on *Centipede*. Partway through the project, I quit being a supervisor (I didn’t like the job and it took me away from doing games) and spent time working on *Centipede*.



Centipede

So Bailey was pretty important to the game’s development?

I would guess she did about half the programming. The game design was left to me because she was working on her first project.

It seems that *Centipede* appeals to women more than most arcade games. Do you think Bailey had something to do with that?

I wish I knew the answer to that question. Someone could point out that no other game I have done appeals to women as much as *Centipede*.

Many theories have been suggested. One is that it was created by a woman. Another is that destroying insects fits well with a woman’s psyche. I believe this game appeals to women because it is not gender biased like fighting games or RPGs or sports games. Other examples like *Pac-Man* and *Tetris* are notable.

I do know *Centipede* fits the basic criterion for a game that appeals to a wide audience. It has a new, appealing look (to get players to try it), an obvious goal (shoot anything), clear rules, an easy set of controls, a sense of accomplishment (kill the entire centipede before he gets you), dynamic strategies abound (trap the



centipede and kill spiders or the blob strategy or channel the centipede or just plain straight-up play), enough randomness to make the game different each time, a goal to keep you going (a new life every 12,000 points), a clear sense of getting better with more play, and a sense that any death was the player's fault.

So you mentioned that *Centipede* grew out of a brainstorming idea. How did the brainstorming process work at Atari?

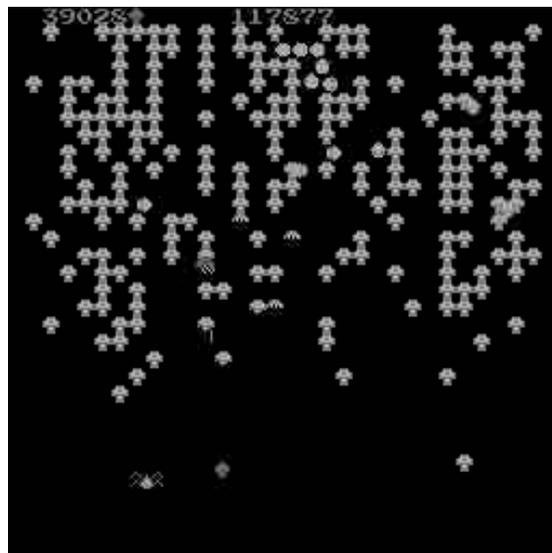
The brainstorming ideas came from anyone in the company. They were usually gathered weeks before the actual meeting which was held off-site, away from Atari. Often the ideas were just a theme. Most submittals had sort of a sketch or art to give the reader a little more info. Occasionally a full game description was submitted which explained the hardware, controls, art, and gameplay.

During the brainstorming session, each idea would be presented and then suggestions would be made for improving it. In addition, marketing would give a rundown of what was selling and the state of the industry. We would also break into smaller groups to discuss a specific type of game or talk about specific games themselves. In the end we would meet again to present any additional ideas from these smaller meetings and vote for the popular ideas. I would say we would get a majority from programmers and designers, but there were a significant number of ideas from artists and others in the company. I found many of the ideas needed a lot of work so it was not uncommon for the original brainstorming idea to get a major overhaul.

Atari Games Corp., now Midway Games West, still uses this process each year. But quite honestly, many of the recent coin-op games are just remakes of older games. For example, more versions of *Rush* or *Cruisin'*. The reason is often market driven: these are the games that have done well in the past and the company does not often want to risk taking a chance on a new theme.

How did *Centipede* change over the course of the game's development?

I mentioned that Dan Van Elderen asked why the player could not shoot mushrooms. I realized early I would need some means to create new mushrooms. This led to one being left when a centipede segment was shot. I also



Centipede



created the flea which left a trail of them when he dropped to create more randomness in the pattern. In other words, I did not want the player to create the only pattern of mushrooms. The spider was always planned to be my “*Asteroids* saucer” which kept the player moving; the spider also had to eat mushrooms to keep the player area somewhat free of mushrooms. The scorpion was added to add a randomness to the centipede pattern and create a sense of panic when the segments would come rushing to the bottom of the screen.

Do you try to create games which allow different players to use different strategies to succeed?

I do strive to give the players as much freedom to create as many strategies as possible. So in a sense, yes, I guess I do encourage players to experiment and try different strategies. I do try to make sure that none of them work all the time or make the game too easy. But I want to leave the player with the impression that if he was only a little bit better he could pull it off.

Why did you choose to use the trackball for *Centipede*?

I believe we used the trackball from the start. I had experience with the trackball on *Football* but I wanted something that was not as heavy and physical to move around. That is how the *Centipede* trackball came about. The trackball, just like the computer mouse, provides a means for inputting arbitrary direction as well as speed. No other controller comes close. It was the clear winner for player controllability.

In my opinion, *Centipede* is one of the best balanced games ever. Was there a lot of experimentation to achieve such a balance?

I would not use the term experimentation in this case because nothing was tried and discarded. There was a grasshopper that we intended to add to hop onto the player, but the spider was sufficient in forcing the player to move so the grasshopper was never even tried. Of course, you can still see the graphics for the grasshopper if you look at the self-test graphics.

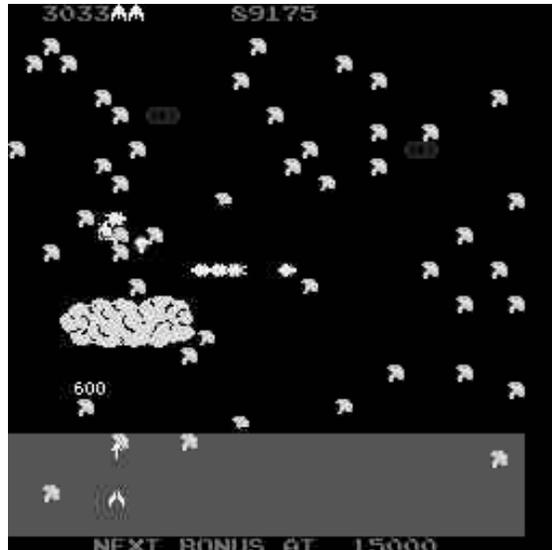
There certainly was a lot of tuning. The timing and speed of when things happened certainly was changed over the course of the project. The balance comes from the inherent rules of the game and the art of knowing when to leave the play alone and when to change something. This art is something that some people have and others just don't. I cannot define it other than to use the term “game sense.”

Were you given freedom to do whatever you wanted for *Millipede*?

With my past record I was given more freedom than anyone else. Something most people do not understand is that half of the games I started did not make it into production. No one ever hears about the failures. Some of the games I actually



killed myself. That's something I believe no one else at Atari did. Of course, there are a few I tried to kill but was not allowed to that eventually died. These days you would probably see them come out in the consumer market anyway just to get back some of the development cost. But in the coin-op market there is no chance to sell anything that isn't a clear winner.



Millipede

***Millipede* allowed players to start farther into the game, at 45,000 points, for example. Was this an effort to shorten the games of the expert players?**

It was a way to increase the cash box. It allowed the good players to start at a higher score where the gameplay was on a difficulty level that was probably just above his level of skill. This often meant shorter game times but would allow higher scores. In a sense I was doing this for marketing reasons. This was not a first for *Millipede*. *Tempest* had this feature back in 1981.

I particularly like the “growth” of the extra mushrooms in *Millipede*. Was this done using a “life” algorithm?

Yes, it is based on the game of life where two or three neighbors would create a new mushroom and anything more or less would kill the mushroom. This has an interesting history. Mark Cerny asked why I didn't do a life algorithm on the mushrooms. I told him I was busy but if he wanted to add it to the game he could. Of course, Mark, being the sharp guy he is, looked at my code and quickly created this feature. He also added the attract mode to demonstrate all the creatures.

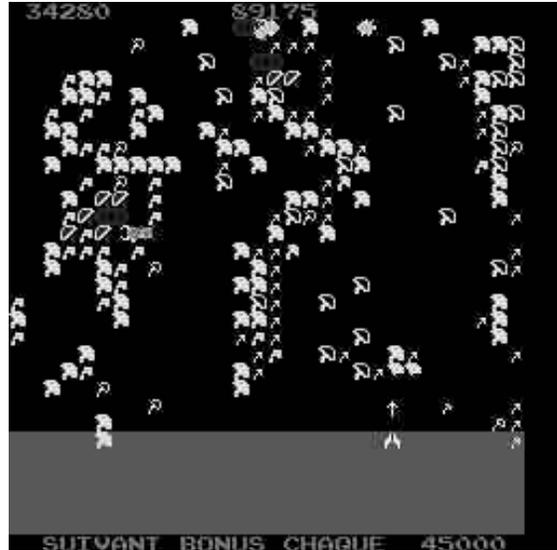
During the *Asteroids* to *Millipede* period, almost all your games were being ported to a wide variety of systems: the 2600, the Apple II, and so forth. How did you feel about these conversions?

It was good business for the company so it made business sense. Of course it always made me proud to see my game in many new places. I did have some concerns about several of the ports. I understand the limitations of some of the systems but I wanted to make sure the company released the best possible conversion. In



many cases I was involved in making sure it had all the features but unfortunately not often enough.

Some of the conversions made improvements that were not possible in the coin-op market. For example, in *Gauntlet* they made a quest mode with a limited amount of health. This would not be possible in coin-op where the object is to get more money added on a regular basis. Another example would be to look at the number of variations of *Pong* included on the Atari 2600 cartridge. It just makes good sense to add value for a consumer title.



Millipede

Was *Maze Invaders* the next game you worked on after *Millipede*? I know it never went into production.

It was a cute puzzle-like game. I was not sad it didn't make it; it did not earn enough on field test. My son loved the game though and I still have one of the two prototypes in my garage. The other was purchased by an operator in Texas, I believe. He loved the game so much he talked Atari into selling it to him.

I believe I mentioned earlier that nearly half of my games did not make it into production. There were engineers that had a higher percentage, Dave Theurer in particular. But there were others who never had a game in production.

The name *Maze Invaders* suggests perhaps something inspired by *Pac-Man*. Was it?

Yes, in a way. It was a maze-like game but the maze changed dynamically. The main character was very *Pac-Man* like; he was cute. There were some parts that I found frustrating, such as when the maze would temporarily block me off. I could not resolve this frustrating aspect, which is probably why it failed.

I understand in 1983 you also worked on a *Road Runner* laser disk game. Was it based on the Warner Bros. cartoon character?

Yes, it was based on Road Runner created by Chuck Jones. The player played the part of the Road Runner who would try to have Wile E. Coyote fall prey to some trap. I had Time Warner send me all of the Road Runner cartoons. I watched every one and selected the best shorts to be included on a laser disk. So when you



succeeded in getting Wile E. destroyed, the game would cut from the action to a similar scene from a cartoon where Wile E. met his usual fate.

I always loved the Road Runner and I thought I could bring him to a video game. When I started I had a vision of something unique. The game certainly met that criterion but it was not as fun as I had hoped. I certainly enjoying seeing all the old cartoons and meeting Chuck Jones but . . .

So the game was killed?

Laser disk games were failing in the coin-op world because of reliability problems. The game actually earned enough to warrant interest but not as a laser disk game. So when they asked me to port it to their new “System I” hardware, I declined, saying I had another idea I wanted to pursue. I am glad they let me pursue this new idea because this idea became *Gauntlet*. *Road Runner* was converted over to System I and actually was released.

Did *Gauntlet* follow your initial vision fairly closely, or did it change a lot in development?

I went back recently and looked at the original game design document and I was surprised how closely the graphics and gameplay matched the finished product. Of course, what did change during development was the hardware. I created an algorithm which would allow me to deal with 1,000 objects with-

out burdening the processor or slowing down the frame rate. I asked Pat McCarthy, the electrical engineer, if he could extend the existing hardware and he found a way to do this which would allow me to display all the objects I needed. In the end there were five patents issued for *Gauntlet*.

Because of the size of the PCB and the restrictions on PCB size for Japanese kits, we decided to use a four-layer PCB for *Gauntlet*. Atari had never laid out such a board nor had they ever used traces as small as we required. But in the end we



Gauntlet



paved the way for all future PCBs at Atari. So besides the success of the game in the industry, *Gauntlet* also made a giant leap in the way we did engineering and manufacturing at Atari.

To my memory of arcades in 1985, *Gauntlet* seemed to be one of the first action games to allow four players to play at once.

This was the first multi-player game which allowed players to end or leave at any time and the screen scrolling was controlled by their actions. This was not the first game to have multi-players. *Tank 8* allowed eight players on one monitor. But all the players had to start at the same time. The idea of using four players was designed into *Gauntlet* from the start. I suspect it was due to the fact that I could only put four players around an upright monitor.

I believe *Gauntlet* was the first game that allowed the player to buy in any time he wanted. I did not want the players to wait, like in *Tank 8*, for everyone to coin-up at the same time. The only solution was to have players come and go at will. Health was always planned from the start. I believe this idea came from *Dungeons & Dragons*, which was very popular at the time. So it was logical that money just bought more health. Since it is every coin-op designer's wish to have the players put as much money as they can into their game, I saw no reason why I would not have the players just increase their health with each coin. In hindsight, this is a wonderful idea because losing 2000 health was not as painful psychologically as inserting another quarter. Besides, the players would not need to reach into their pocket to find another quarter to insert before their character was lost.

Where did the idea to have the game say things like “Red Warrior needs food, badly” come from?

I do not remember. I suspect it was not my idea. It may have come from my co-programmer Bob Flanagan or from someone else at Atari. In any case we had a large list of phrases we wanted the “Dungeon Master” to say to taunt the player. There are several phrases that seem to stick in everyone's mind. My favorite is “the Wizard (me) seems to be eating all the food lately.”

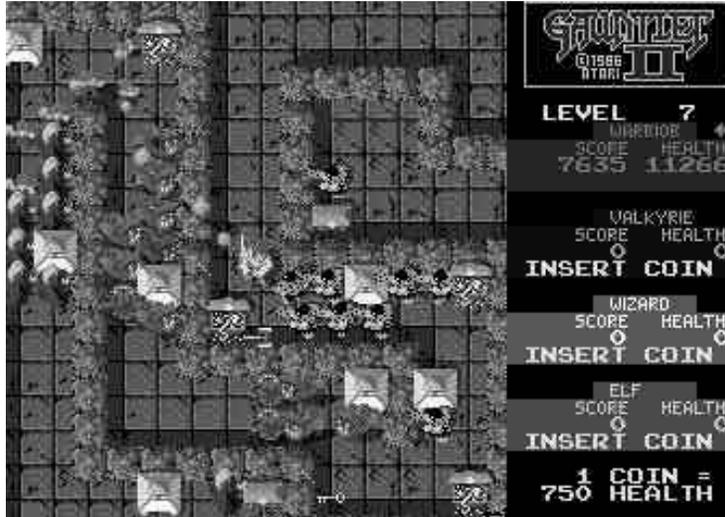
Many think the Valkyrie was the most powerful of the four characters.

Actually, the Hulk or the Wizard could be used to play forever. This was demonstrated first by players in Japan playing a one-player game. This was fixed later by reducing the amount of food on subsequent levels if the player had not lost enough health during the last level. The Valkyrie was designed to be the most balanced of the characters but shot power, shot speed, and strength proved to be more important than other attributes. This is why the Hulk and Wizard seemed to be the most powerful. Of course, the Elf was fun to play with for many players because you could always get more food or treasure than the other players.



***Gauntlet II* allowed four players to all be playing Valkyries, or Elves, or whatever combination they wanted. Did this mean the character classes had to be more equal than in the first game?**

No, we actually did very little that I can recall to equalize the characters. This feature was added because some players wanted to play a particular character and I did not want them to wait until the desired position was open. So in essence I eliminated another reason for not entering the game right away.



Gauntlet II

Was *Xybots* your next project after *Gauntlet II*?

Bob Flanagan and I actually started another game which I quickly killed after the initial gameplay turned out to be less fun than I had expected.

Xybots, as I mentioned earlier, started out as an idea to do *Castle Wolfenstein*. I started the game as a two-player split-screen *Gauntlet III*. Partway through marketing said they wanted something other than *Gauntlet*. So I changed the characters and enemies to be more like *Major Havoc*. I still regret changing the theme and wish I had kept my original game concept.

Was it a great engineering challenge to create the game's 3D look?

I developed a very interesting algorithm for doing the 3D rotation using just 8x8 pixel stamps, as we call them. I don't know how to explain how this worked without getting my original sketches to visually demonstrate it. I could have had the player rotate other than in 90-degree increments, but it made the gameplay simpler to just allow only 90-degree rotations.

If I recall, the game had interesting and unique controls.

The controller was very unique because it provided the standard eight-way joystick as well as a knob on top which could turn left or right to indicate a rotation. This control made the game more difficult, which is often the kiss of death in the



coin-op market. As with any 3D game, players could not easily visualize where they were despite the map available to them. In addition, it was possible to get shot in the back, which added to the frustration factor.



Xybots

How did you get involved working on the Atari *Tetris*?

I played a version of *Tetris* and was quickly addicted. I asked our legal counsel, Dennis Wood, to get the rights. Since I had just worked on reverse engineering the Nintendo Family Computer, which soon became the Nintendo Entertainment System in the U.S., I decided to create a version on the FC and NES and sell it through Tengen, which was Atari's consumer publisher. Dennis Wood got the rights and we showed *Tetris* first at the June Consumer Electronics Show. It was decided to improve the game so I redid the visuals and we released it at the following CES in January.

I should point out that I was working on another game at the time I was doing this, so I could not devote all my time to the *Tetris* project. It was this fact that made me need to turn over *Tetris* to Greg Rivera and Norm Avellar for the coin-op market. I did get my original code to run on the coin-op hardware before going back to my project. This is why my name appears on the credits of the coin-op version.

What did you like so much about *Tetris*?

It was just so addicting I knew we had to have it. In hindsight, I could explain why this game worked so well but I am not sure that would prove anything. Besides, the real question is "Why didn't I think of this idea?"

Was Tengen *Tetris* your only NES project?

I had *Centipede* and *Millipede* running on the FC before the lawsuit with Atari Corp. resulted in the ruling that they owned the rights to all our games prior to the sale of Atari to Tramiel by Time Warner. So we had to drop the work I did. So my previous work made *Tetris* very easy to do on the NES. I also added the two-player



simultaneous feature which made this game better than all the other versions. Later you would see Tengen versions selling for \$150 or more.

Why was Tengen *Tetris* eventually withdrawn from circulation?

You can read several versions of the story but I suspect the bottom line is the Hungarian who had the rights did a poor job of covering all the bases. The Russians accepted money from Nintendo when Nintendo created a new category of rights. Despite the fact we had the rights to computer systems, Nintendo claimed their Family Computer was not a computer even though they sold Basic and a keyboard and other services in Japan just like any other computer. I was certainly disappointed to see my work lost.

Why did you want to work on conversions of someone else's game?

As with many of my games, this was the best idea I could think of at the time. However, in this case, because I enjoyed it so much, it was an easy decision. What better way to play the game you like so much and make sure it comes out the way you like?

What did you work on next?

I eventually killed the game I was working on during the “*Tetris* Affair.” I believe *Steel Talons* was my next project. I wanted to do a 3D Red Baron flying/shooting game but marketing thought World War I planes were not cool enough for teens, who were the prime coin-op target audience. Marketing wanted jets and I thought that was a dumb idea because who wants to see dots at a distance shooting at each other. I wanted something close where you can see the detail of the enemy you are shooting at. Helicopters were the logical choice.

Wasn't *Steel Talons* a fairly authentic helicopter simulator?

Steel Talons had all the regular helicopter controls: a rudder, a collective for controlling height, and a stick for turning. Of course flying a helicopter is difficult without some assistance, so I had computer assist just like real military helicopters. I added automatic collective control so the player would maintain level flight and any landing would be smooth. It would also increase height if the ground was sloping in front of the height. The “real” mode just disabled this helping code and increased the player's acceleration to compensate. This was a unique feature and Atari was issued a patent on this idea.

The game had another interesting feature that had never been used on a video game before. We installed a pinball thumper, often used to indicate a free game, under the seat. This was used whenever the player's helicopter was hit by enemy fire. During the first field test, the voltage for this thumper was higher than it should have been and the first players to use it nearly jumped out of their seats when it



fired. The noise could be heard over the entire arcade.

The first field test also introduced a new problem that we never had before. I went out to check on collections and I tried to remove the coin box. If you have ever seen *Steel Talons*, you will see that the coin box is located at a strange angle requiring the operator to lift the box with his arms fully extended. Not the easiest position to lift any weight. Well back to the story. I tried to lift the box out but could not budge it. I thought it was jammed. I soon discovered that the box was so full and was so heavy it was nearly impossible to remove. This led to the strange instructions in the manual asking the operators to empty the coin box every couple of days.

On *Steel Talons*, didn't you work with *Battlezone* creator Ed Rotberg?

Yes I did. He was at Atari during the golden days of *Battlezone*, *Asteroids*, et cetera. He left Atari to do a start-up called Sente, before returning to Atari a few years later. He had just finished working on a *Tube Chase*-like game using the same 3D hardware that *Steel Talons* used. This hardware was a cost reduced version of the *Hard Drivin'* PCBs. So it was natural for Ed to work with me on this project. Another interesting feature of this game was fog. The original *Hard Drivin'* team did not believe me when I told them I could add fog to the world. I am still proud of this effect and they were surprised that it worked.

How did the *Space Lords* project come about?

I wanted to continue my ideas of multi-player play that I started on *Gauntlet*, and then continued on *Xybots* and *Steel Talons*. So I chose a 3D space environment with up to four cabinets linked together. Each cabinet had two monitors similar to *Cyberball*. I tried to keep the cost down by using Atari's "growth motion object" hardware which was cheaper by far than the 3D hardware used on *Steel Talons*. It could not draw 3D polygons, but it could grow or shrink flat textures.

I understand *Space Lords* did not do too well financially.

Space Lords had some strange earning patterns. At some arcades it earned more than \$1,000 per week for two double cabinets. But at some small arcades it earned only \$75 as a single cabinet. The bottom line is we had a difficult time selling it because of its cost and the limited number of locations it could be sold into. It was definitely hard to make a coin-op game using the concept of one player per monitor. Even though I added a second player as a gunner at half price, it was felt by many to be not as fun as being the pilot.

And *Space Lords* came out right around the time the fighting games were taking off.

The fighting games made *Space Lords* difficult to sell because they were often "kits," which sold much cheaper than a large dedicated upright. *Street Fighter II* had



great earnings and continued to earn good money for a long time.

In fact, since the early '90s most arcade games have been in one of a very few, limited genres. What do you think of many of the arcade games that come out these days?

You are right, the coin-op market seems to be all driving, fighting, and shooting with an occasional sports title, like golf. There are reasons for this. Driving has universal appeal and usually earns for long periods. So it is often the most accepted game theme. Besides, most home units do not have steering wheels and gas pedals or give you the feel of being inside a car. So you cannot get this experience in the home. Fighting games are now difficult to sell in the arcades and I believe this is because you can get the same experience on most advanced consoles. At the time they were cheap and earned big bucks. Shooting games are still viable because guns are not the standard controller on consoles or PCs. So the only way a game player can get this experience is in the arcade.

So the bottom line is, most arcade games these days are not unique and fit very limited categories. I don't think the arcades are completely dead but they are not the destination places they used to be.

Did *Space Lords* turn out to be your last coin-op?

I was working on a shooting game prior to my departure from Atari. That game died but the gun was used later on *Area 51*. I joined Electronic Arts who were trying to start up their own coin-op group. My intention was to start doing consumer games. But EA had some old Atari friends and I decided to join them. I had done one puzzle game which I killed and was working on a shooting game when they decided to drop out of the coin-op market. Then I was even more determined to enter the consumer games business.

How did you come to start doing N64 programming?

I was looking for a project to work on, so I contacted many companies to see what they had to offer. I was planning to work with another programmer from EA but he decided to join some friends to start up a new company. Atari wanted the coin-op *Wayne Gretzky 3D Hockey* done on the N64 and I was looking forward to doing something on that platform. This was partly because the game promised to look better than the PSX but also because it looked like we could be the first hockey title available. So I joined a group at Atari and we started work on *Wayne Gretzky 3D Hockey*. This turned out to be more work than I expected partly due to the state of N64 development systems but also due to the fact the coin-op was not going to be done until just before we released.



As you mentioned, a lot of the appeal of playing an arcade game like *San Francisco Rush* seems to be sitting in the chair, having the gearshift, the steering wheel, the force feedback, and so forth. How do you try to capture that for the N64, which has none of these niceties?

You are right.

The home does not have the environment of the arcade cabinets but we can do things on the home games we can never do in the arcade. We can provide more choices for the player, more tracks for them to learn, and more things to discover.

I try to keep the basic play the same but I always try to add value to the

product. This is one thing I made clear when I joined Atari. Atari wanted me to just do a straight port. That had always worked for them in the past. I did not believe this would work and told them I would be adding additional “stuff.” For example, on *Gretzky* we added a full-sized rink, a new AI, instant replay, more players, full seasons, etc. In general, home games require considerably more work. I also believe we can do different games for the home market that we could never do in the arcade. So for me, this opens up new possibilities.

Arcade pieces must be easy to learn with rules that are obvious and provide entertainment that lasts ninety seconds. The home market is not bound by these rules. Instead you must provide more life for your product. Often this means it takes the player longer to “finish” the game. Even when the player has finished it, there must be reasons why he will want to go back to do it all over again.

Do you like the engineering challenges of doing home conversions?

I actually enjoy the “old style” of trying to get everything to fit. I also enjoy adding tricks to get the frame rate as high as possible. It was very interesting to get all of *SF Rush* into 8 MB, which includes around 3 MB of audio and all the graphics.



San Francisco Rush: Extreme Racing for the Nintendo 64

**Do you miss doing original designs?**

Yes, I do miss the old game designs. 2D worlds are so much easier for the player to understand. I also like the idea of creating a game with a fixed set of rules and enough randomness so that the player can create different play-styles and their own strategies.

I am not sure I could sell a game with an “old design.” Players have different expectations now. They would expect 3D designs or Internet play or high-resolution textures and pre-rendered movies or highly developed characters . . . Besides, just about anything I do now will just elicit comments like “It is just a twist on game xxx with a little of game zzz.” For the record, many of the old designs were based on previous game ideas. Remember, *Asteroids* came from a previous game with a little of *Space War* thrown in, even though many thought of this as an original design.

You have been working with Atari for more than twenty years now, so you must really like it there.

Yes, Atari has been very good to me. I have a deep sense of loyalty to the company and the people I work with. Besides, I like what I am doing, so I see little reason to leave. I think the loyalty is mostly due to heredity. Longevity comes from doing what I like.

Working on games requires something which many people do not have. Many cannot take the constant pressure to perform, the long hours, and the thought that their “baby” that they have been working on may get killed after eighteen months of hard labor. Others are programmers or artists who have found more interesting things to do.

I must admit I have often thought of doing something else. I just have not found anything else I want to do more than what I am doing now. That could change or I may find myself doing games until I retire.

In the last few years, *Asteroids*, *Centipede*, and *Gauntlet* have all been remade.**How do you feel about the remakes?**

Many are doomed to fail just like most game ideas. *Gauntlet* was a good case of a remake that worked very well. *Arkanoid* was a remake of *Breakout* that worked very well. So remakes can work, but it is difficult.

The real failure comes from comparing the gameplay to the original. For example, making a 3D version of *Centipede* makes the gameplay harder because the 3D information is not as easy for the player to process. Remember, designers have had twenty years to play these old games and come up with a new twist to make a new great game. The fact that they haven’t done it yet seems to indicate that it is unlikely. Not impossible, but unlikely.



Which one of your games might you want to remake?

If I had the answer to that, and if I believed it was the best idea I had, I would be working on it. Besides, if I told you, then someone else would be doing it now, wouldn't they? In other words, I don't have any idea how to take some old classics and make them new and interesting in today's market.



Gauntlet Legends

How has the game development industry changed over the years?

The games industry has definitely changed, but it is still a video game industry. Video games were not a \$7 billion industry when I started. With big business comes big money and that invariably brings with it control over how it is spent. So there is definitely more politics at the corporate level. The interference from management comes from their need to control the costs, but the real reason, I believe, is due to the evolution of the games themselves. By that I mean, we could design and program a game in three months in the early years. In three months you did not spend enough money for them to interfere. Games have evolved to the point where you cannot do a game with just one person in a realistic amount of time. It takes several programmers, several artists, an audio specialist, and someone to manage the project over a period from twelve to twenty-four months. The console market has changed too. You did not need to spend \$1 billion to launch a new console in the early days, but it costs that much now. So with evolution comes longer periods for development and higher costs to produce a product. With the higher costs comes more money and hence more control (i.e., interference) over how it is spent.

For your original designs, you served as both designer and lead programmer. Do you enjoy working in both capacities?

Working as game designer and programmer is a good idea if you can pull it off. There are very few people who are good at both. So it is not a strategy I recommend today. For example, for today's complex multi-character and multi-level games, I



am not as good a designer as I would be on other styles of games. So I would be willing to give up this role to someone else.

The programmer has to implement the design and if the designer's ideas are not communicated well enough, then the game is programmed differently than the designer expected. I believe it is often the programmer who can make or break the "feel" of a game.

You seem to have missed one point. I was also project leader on many projects. This is a role I am very good at but receive no acknowledgment. My projects are almost always on time and if there are problems, management is often told well in advance. No one outside Atari probably is aware of this. Unfortunately, I do not enjoy this role so I try to spend as little time as possible actually managing a project.

You even served as artist on your early games, didn't you?

Early on it was a good idea. There is no reason to train an artist to create a rock on graph paper and provide me with the coordinates so I could enter them into my game. When there was so little in the way of graphics or audio required, it makes no sense to have another specialized person doing this. Today, it is an entirely different matter. Today it is absolutely required.



Asteroids

Do you feel that any of your games are underappreciated?

As a game designer, no, I do not feel I have any games that were underappreciated. If the game design works, then the gameplay is fun and the game sells. As a programmer, yes, there are probably some game ideas or algorithms or programming speed which are underappreciated. Many programming tricks I do for personal enjoyment so I am not looking for external recognition.



In the early days you were pretty limited by the technology available to you. Did the technology limitations foster creativity?

Yes, I would have to agree. There were many times I spent thinking about how to do something on a given hardware and that turned into a game. *Xybots* was certainly one of those games. On *Gauntlet* we created new hardware to make the gameplay possible.

When working with an original game design, where do you start?

First, I try to come up with the game and then look at all the aspects of the play. From the market perspective: will it sell, is the timing right, licensing requirements, competition, et cetera. From the player's perspective: what makes this game fun and what is unique that will make it interesting. From the development side: what will it take to do this game in terms of people and equipment and will it be fun to do. Ideas themselves come from just about every possible source. I have mentioned how some come from previous games, brainstorming ideas, technical challenges, and other people's suggestions.

So, once you have your idea, do you start coding right away, or do you spend a lot of time thinking it through ahead of time?

With the large budgets and large teams these days, it is necessary to do a game design document and technical design document before the game gets too far into development. However, I try to start work on some critical aspect while the design documents are being drawn up. I believe it is extremely important to work on the aspect of the game that will make or break the concept. The front-end movies, story line, front and back end screens can all wait until the gameplay has been proven. Sometimes this prototyping phase is quick but often it can take several months.

Once you have proven the gameplay concept in a prototype, how does the rest of development progress?

Games go through four phases for me. The high at the beginning of a project of doing something new and the feeling that this will really be a great game. The project often makes giant leaps in short periods. The middle part of the project is mundane. The concept has been proven but there is often so much work to do and the game does not appear to change much for all your effort. The third phase is often full of panic and stress. This is the part just before release when you just want the project to end. The fourth phase is one of satisfaction after the game has been released.

With the current long projects I often feel I am getting diminishing returns for my effort, so I am happy to have the game end. In my case, almost everything I had planned for my game has been implemented, so I am happy to call it done. Except for finding those irritating last-minute bugs . . .



So after the prototype is functional, you don't really enjoy the development process?

Yes, I would say the bulk of the game is done after the core game concept has been proven. However, there are often parts that prove rewarding during the long development before the game is finished. But after doing so many games over the past thirty years, working on, say, the user interface just does not get me all excited.

No, I would like to do a prototype and leave it to someone else to finish. But I feel I still have the vision for the gameplay and I do not believe another person or group would continue the gameplay as I envision it. So in the end I would feel that the game was not what I expected, not mine anymore. I would always have the feeling that if I had worked on it to the finish, the game would be better than what anyone else could have done. I guess I would feel differently if I had not been as successful as I have.

What role do you think AI plays in games?

In the old games AI had no involvement. Often the enemy would follow a fixed set of rules with some randomness thrown in if necessary. These days it is entirely a different matter. It is becoming very important for modern games. Some people have recommended that, when appropriate, each project have one specially trained person dedicated to doing the game AI. And for some games, I would agree.

Why do you think the games require more sophisticated AI now?

I believe the theme and gameplay of most new games require more AI. The sim games, the shooters, et cetera, all try to give the real sense of intelligent life competing against you. If games do not try to mimic real life then a set of rules may do just fine.

How important do you think it is to make the AI in a game "real"? That is, to provide the AI only with the information the player would have in the AI agent's position?

It is not necessary but may lead to more believable enemy AI, so I would recommend it in some cases. For example, in *Steel Talons*, the enemy gunners would not turn or fire until they could see you visually. If there was a hill in the way or you were hugging the ground at the end of their range, then they did not see you. This is one case where it was necessary.

Lately, a lot of attention is being given to combining games and stories. Many arcade coin-ops, perhaps as part of their nature, have almost no story. What do you think about telling a story within a game?

I have never been high on stories. I feel it is absolutely necessary to have the player grasp the theme: setting, ambience, and goals. Sometimes stories help to



make the goals easier to understand. Some games are made like a movie, so a story makes good sense: the player feels he is the main character that he is controlling. In a coin-op game, a story makes no sense unless it is shown in the attract mode. We do not want the player wasting his time watching something when he could be playing or putting in more money.

You mentioned before that you specifically wanted to get into doing games for the home market. Why was this?

I wanted to do home games instead of coin-op games because I saw more opportunity to do something new in the home market.

Do you not see any future for coin-op arcade games?

I suspect coin-op games in the arcades will tend toward cheaper simulation rides (physical movement or encompassing environment), just like you see now. They provide something you cannot get at home and are cheaper than the rides at Disneyland. I believe the coin-op arcade market is already there. The coin-op street mar-



The arcade version of *San Francisco Rush 2049*

ket will always need to be inexpensive. So I see a consumer platform in a coin-op box or cheap PCBs with simple games that do not require long development times.

I believe the consumer market already dominates over the coin-op industry. I do not have the numbers, but it is clear to me by looking at sales numbers of hit games and the dollars they represent. It is sad to see the changes in the coin-op industry. I am sure glad I was a part of the industry. I feel I was definitely in the right place at the right time.



Ed Logg Gameography

Super Breakout, 1977

Video Pinball, 1979

Asteroids, 1979

Othello (for Atari 2600), 1979

Football (4-player conversion), 1979

Centipede, 1981

Millipede, 1982

Gauntlet, 1985

Gauntlet II, 1986

Xybots, 1987

Tetris (conversion to NES), 1988

Steel Talons, 1991

Space Lords, 1992

Wayne Gretzky 3D Hockey (conversion to N64), 1996

San Francisco Rush (conversion to N64), 1997

San Francisco Rush 2 (conversion to N64), 1999

San Francisco Rush 2049 (conversion to N64 and Dreamcast), 2000

