THE COMPUTER MUSEUM REPORT

VOLUME 22 SPRING 1988

Operating Grants

Corporate Membership

Sustainer—Over $10,000

Digital Equipment Corp.
Hewlett-Packard
IBM
Massachusetts Council for the Arts and Humanities
Sequent Computer Corp.
Sun Microsystems
Symbolics, Inc.
Xerox Corporation

Benefactor—$10,000

Aetna Life & Casualty
Kurzweil Music Systems
MASSCOMP
NEC Corporation
Raytheon Company
Tandy/Radio Shack Corp.

Patron—$3,000

Amdahl Corporation
BankLink, Inc.
Boris Color Labs, Inc.
Coopers & Lybrand
DECUS
Dentsu, Inc.
Draper Laboratories
Fenwick Partners
Gaston & Snow
Gourmet Caterers, Inc.
Honeywell Bull
Lotus
Manufacturers Hanover
Maxell Corp. of America
McGraw-Hill, Inc.
Phoenix Technologies
Prime Computer, Inc.
Raster Technologies
Ropes & Gray
Stratus Computer
Sun Microsystems
The Travelers Companies
UNISYS
Zenith Data Corporation
Ziff-Davis

Contributor—$1,000

Adage, Inc.
Adelie Corp.
Analog Devices, Inc.
Apple Computer
Arthur Andersen, Inc.
Arthur D. Little, Inc.
Bachman Information Systems
Bank of Boston
Bank of New England
Banyan Systems
BASF
Baybanks
Bolt Beranek & Newman
Boston Consulting Group
Boston Safe Deposit
Cambridge Institute for Information Systems
Cincom Systems
Cognition, Inc.
CompuServe
Dane, Falb, Stone & Co.
Data Architects, Inc.
Data Translation
Deloitte Haskins & Sells
Dragon Systems
Eastech Management
E.I. DuPont de Nemours Co.
E.M.C. Corporation
Festive Foods Caterers
Gillette Company
Grafik Communications Ltd.
Goldman, Sachs & Co.
Greater Boston Chamber of Commerce
GTE Laboratories, Inc.
Hub Mail Advertising
Hyams Trust
Index Technology
Infocom
Infomart
Intel Corporation
Interface Group
Interleaf
Mentor Graphics
Merrill Pickard Anderson & Eyre
Microcom
Micro Control Systems
National Semiconductor
NYNEX Service Company
Orchard Computer
Palmer & Dodge
Peat, Marwick & Main
Pell, Rudman & Co., Inc.
Phase3 Systems
Rossin, Greenberg, Seronick & Hill
Russell Reynolds Associates
Schubert Associates
Scitex America
Shawmut Corporation
Sterling Hager
Technical Data
International
Teknowledge
Telectronics
Teleprocessing, Inc.
Teradyne
Testa, Hurwitz & Thibeault
The Technology Research Group
USIR
VM Software
Wollongong Group
XRE Corporation

Contributing Members

Donor—$500 or more

Rolland Arndt
Burlington Public Library
Michael Callahan
David Cutler
James B. Deaderick
Nick De Wolf
David Dinkle
Ray Duncan
Tom Eggers
Ed Feignebaum
Branko Gerovac
Tom and Rosemarie Hall
Martin Kirkpatrick
Raymond Kurzweil
Lawrence Public Library
Thomas Marill
Christopher Morgan
Linda J. Phillips
Paul R. Pierce
Audrey Reith
Linda & David Rodgers
Curtis Scott
O. & K. Selfridge
Sharon Public Library
Thomas F. Stephenson
Seth Stern
William Steul
Chuck Woodward

Sponsor—$250

Timothy Anderson
Anonymous
Isaac Auerbach
Dawn R. Banks & Robert M. Praetorius
John Barstow
Jordan Baruch
Graham Beachum
G. C. Belden, Jr.
James Bell
Gordon S. Brown
Roger M. Guoy
Peter Capek
George Chamberlain
George Champine
Richard J. Clayton
Clement Cole
Richard Corben
Phil Dorn
Dick Dunnington
Lucian J. Endicott
Bob O. Evans
Neil Faiman
Andrew Feit
Henry Fuchs
Kevin and Judith Galvin
Walter Gamble
James Harvison
J. Milton Hutson
W. Richard Johnson
Bob King
Alex Koller
Sheldon Laube
Paul Leach
John R. Levine
Eli Lipcon
Carl Machover
Barry Margolin
Andrew H. Mason
Charles Minter
Joseph M. Newcomer
Anthony Oettinger
PCK Technology Division
James N. Porter
John F. Pries
C. M. Riggle
Michael Roberge
Linda & David Rodgers
Michael J. Samek
Michael Thompson
G. Michael Uhler
Christian Walker
Allan L. Wallack
S.W. Welch
Andrew Wilson
William Wulf

Friend—$100

Kenneth R. Adcock
R.J. Alexander
Robert Allio & Assoc.
Professor Arvind
Robert Z. Bachrach
Richard G. Bahr
Mario Barbacci
Steve F. Barneby
Lois J. Bizak
David Bonner
Bontronics
John Brackett
Daniel S. Bricklin
Robert E. Briney
Fred & Nancy Brooks
Lawrence Brown
Brown University
Erik Bruhn
David Bryant
Gerald A. Bujaucius
David Bunnell
W. Carlson
Maria L. Carr
Julia Christianson
Joel Clinkenbeard
James F. Cody
Howard I. Cohen
Howard E. Cox, Jr.
Ronald Creamer
Michael Cronin
James Cudmore
Paul Curran
Charles Dana
John Davis
Randall Davis
Donald R. Daykin
Delta Management
F. Demeo
David R. Dick
Kevin W. Dick
Mary Dietz
Lucien Dimino
Michel Doreau
John Dove
Robert Doyle
Theodora Drapos
Esther Dyson
Joseph J. Eachus
Richard J. Eckhouse and Ruth Maulucci
William T. Elliott
P.J. Evans
Andrew Farber
Geoffrey Feldman and Sharon Lipp
Larry Forte
Lloyd Fosdick
Philip Fox
Robert France & Ruth Doan
J. Thomas Franklin
Clark Frazier
Daniel Freedman
Alan E. Frisbie
Bruce Gee
Roger Glovsky
Steven Gold
Steve Golson
Edward Gordon
Ross Gould
Roger H. Goun
J. M. Shag Graetz
Kenneth Grammann
William Graustein
Stephen Gross
Paul C. Guglietta
Jerrier A. Haddad
J. Scott Hamilton
Keith W. Hare
Frank E. Heart
Paulyn Heinmiller
Ted Hess
Grace M. Hopper
Nancy S. Horie
Mark Horowitz
Robert Ingham
Jeffrey Ives
Roy C. Johnson
Lieman Jones
Charles A. Jortberg
JSI, Inc.
Richard M. Karoff
Paul Katz
Gary Kessler
Richard King
Robert King
Mark Koretz
Stanley Kugell
Marvin Kurtti
Robert Laman
John Langell
Curtis Larock
Tsvi Lavi
Grace Leahy
Margaret L. Leiby
John V. Levy
Joseph L. Levy
Jon Liebman
John D.C. Little
Reed Little
George Logemann
Carl D. Lowenstein
Arthur Luehrmann
Michael Mahone
Julius Marcus
Melvin J. Mason
Richard O. Mason
Sibyl Masquelier
J. Stevadson Massey
Peter Masters
Robert Mayer, Jr
Robert McCormick
F. Warren McFarlan
Douglas McKenna
William McLean
Todd Medlock
James Meehan
R.W. Meister
Micro-Net Data
Jeanette Middour
Allen Moulton
Rayman Mustafa
Isaac R. Nassi
Steven Nevins
Landon Noll
Bernard J. Nordmann
David Novak
H. Edward Nyce
Thomas P. Oberst
James N. O'Boyle, Jr.
Martin O'Donnell
Lawrence & Pauline Oliva
Robert Otnes
Leon M. Padell
Joseph A. Pallone
Kenneth D. Patterson
James V. Patton
George Peo
Ernest Petrides
James H. Philip
Michael Pique
Michael Poe
John Poppen
John H. Porter
David Potter
Richard Powell
Printonic Corp. of America
Brendan Reilly
Steven Reilly
Nicholas Reinhardt
Terry Robinson
Ann Roe-Hafer
Thomas S. Roy II
Stephen Russell
Sabaki Corporation
Howard Salwen
Michael Sand
Robert Santy
Robert Schapp, Jr.
Stanley Schultz
Bennett Schwartz
Andrew Scott
Curtis A. Scott
Aaron Seidman
R. &. M. Shamel
Benson Shapiro
W. Lee Shevel
Dan Siewiorek
Diane J. Skiba
Phil Slaughter
Allen G. Sneider
Gerald Steinback
Max J. Steinmann
Bruce G. Stewart
Hugh W. Stewart
Robert and Diane Stewart
J. Michael Storie
Charles A. Stott
Joel Sugg
Steve Swerling
David Tarabar
John Tartar
Paul Tedesco
John Terrey
William R. Thompson
Warren G. Tisdale
Michael Tomasic
H. Dean Toombs
F. & M. Trapnell
Robert Trudel
David Tweed
Noah E. VanDenburgh
Charles Volkstorf
David P. Wheatland
John D. Wick
Hugh Wilkinson III
James Williams
Kevin Willoughby
Roy Wilson
Richard T. Witek
Frederic G. Withington
John Woodward
D. L. Wyse
Leo Yochim
Hermann Zapf

Capital Campaign

64K or more

Apollo Computer, Inc.
C. Gordon Bell
Control Data Corporation
Cullinet Software, Inc.
Data General
Gardner Hendrie
Mitchell Kapor
Carver Mead
Kenneth Olsen
John William Poduska, Sr.
Wang Laboratories

32K or more

American Telephone & Telegraph
Bank of America
Henry Burkhardt
ComputerLand
Kenneth Fisher
International Data Group
Burgess Jamieson
Mitre Corporation
Russell Noftsker
Robert Noyce
Robert M. Metcalfe
Fontaine K. Richardson

16K or more

American Management Systems
Anonymous
Clark Baker
Robert Berkowitz
The Boston Globe
Howard Cannon
David Donaldson
Doug Drane
David K. Dyer
Bruce Edwards
Index Systems
August Klein
Thomas Knight
James L. McKenney
NEC Corporation
David & Pat Nelson
Paul & Katherine Severino
Ron Smart
William Spencer
Charles E. Sporck
Ivan Sutherland
System One
Chris Terman
Minoru Tonai
Stephen Watson
Daniel Weinreb

8K or more

AFIPS
Harlan E. & Lois Anderson
Erich & Renee Bloch
Winston R. Hindle, Jr.
Theodore & Ruth Johnson
John Allen Jones
J. Carl Masi
Stan Olsen
Russell Planitzer
Doug Ross
Grant & Dorrit Saviers
Edward Schwartz
Software Results Corp.
Erwin Tomash
William Wolfson
Xerox Foundation

4K or more

Charles & Connie Bachman
Gene Brewer
Roger and Mary Cady
R. Steve Cheheyl
Robert & Eleanor Chinn
Pat Clark
Robert G. Claussen
Stephen Coit
William Congleton
Alex d'Arbeloff
Arnaud de Vitry
Robert Everett
Ford Motor Company
Jay W. Forrester
William Foster
General Systems Group
Gaston & Snow
Peter Hirshberg
Houghton Mifflin Company
ICL
Jeffrey Kalb
Allan Kent
Jack S. Kilby
Andrew C. Knowles III
David J. A. Koogler
Edward and Debbie Kramer
John W. and Edna W. Lacey
Liberty Mutual Insurance Company
Ralph and Linda Linsalata
Robert Lucky
John Norris Maguire
Richard D. Mallery
Thomas and Marian Marill
Daniel D. McCracken
Thomas and Elizabeth McWilliams
Meditech
Allen Michels
Microsoft
Robert M. Morrill
Laura & Ken Morse
New York Air, Inc.
Nolan Norton Company
John L. Payne
Nicholas A. Pettinella
Edward G. Perkins
Robert M. Price
Regis McKenna, Inc.
Ben Robelen
Jean E. Sammet
Sequent Computer Systems Corp.
Hal Shear
Alan F. Shugart
Richard L. Sites
Stratus Computer Corp.
Del Thorndike and Steve Teicher
3Com Corporation
Jean De Valpine
Charles & Angela Waite
Harvey W. Wiggins, Jr.

The Computer Museum
Museum Wharf
300 Congress Street
Boston, MA 02210

25th Anniversary of Computer Games Weekend

Jerry Rabinowitz

Mike Harrison chuckles while Ken and Laura Morse crash their "F-15 Strike Eagle."

Twenty-five years or more of computer games? That may be hard for people raised on "Pong" or "Asteroids" to believe. It seems like just yesterday video games invaded the arcades of America with their blinking lights, blips and beeps.

But for the flock of game aficionados and members of the press who descended upon The Computer Museum November 6-8, twenty-five years sounded just right.

Why? Because three of the inventors of the world's first interactive computer game were there. In 1962 a group of M.I.T. hackers working on the school's recently-acquired PDP-1 computer collaborated to create the game, known as Spacewar! It was perhaps as humble in its origins as it was powerful in its impact. With a multitude of computer and video games now solidly in place in homes, offices, schools, bars and arcades throughout the country, it seemed entirely appropriate for The Computer Museum to host an anniversary celebration.

While the weekend's spotlight focused first on Spacewar! and the historical side of computer games, it then went on to highlight a range of other events. Panel discussions on both the past and future of games, micromouse robot demonstrations, a birthday party, Core War tournament, and lots of representative computer games — all were ingredients of the Anniversary Weekend.

The weekend lifted off with a Gala Birthday Party, Friday night, when three of the original Spacewar! inventors — Steve "Slug" Russell, Alan Kotok and Martin "Shag" Graetz — were reunited. They were joined by a number of other prominent game inventors and experts who reminisced about computer game history and explored some of the industry's latest trends. Mingling with 150 of the Museum's guests and game devotees, these pioneers added to the catered dinner's general mood of festivity and nostalgia.

In addition, almost two dozen games were located around the Museum's fifth floor galleries for public use the entire weekend. They gave active testimony to the evolution of computer games: from Spacewar! and its unwieldy PDP-1 mainframe (part of the Museum's permanent collection) to the three-dimensional colored sights and stereo sounds of "Marble Madness" or "The Halley Project" on an Amiga personal computer.

Table of Contents

Cover: *Close-up of the back of Roe Adams's jacket at Computer Games Weekend. Photo: J. David Bohl*

Saturday featured a two-part symposium entitled "The Coming of Age of Computer Games." The morning session addressed the history of computer games, while the afternoon discussion focused upon the present and future of the games industry (see separate articles). Both sessions drew interested crowds to the Museum's auditorium.

Sunday's events were perhaps the most unusual of the weekend. David Otten and his team from M.I.T. made the most of a ten foot by ten foot maze to run their world champion MITEE micromouse through its paces. MITEE and a companion mouse dazzled the audience as they used infrared sight, computer memory and impressive acceleration to track the fastest route from start to center point in the maze. Micromouse teams from West Point Military Academy and Northeastern University were also on hand to learn some pointers and see the champ in action.


Jerry Rabinowitz

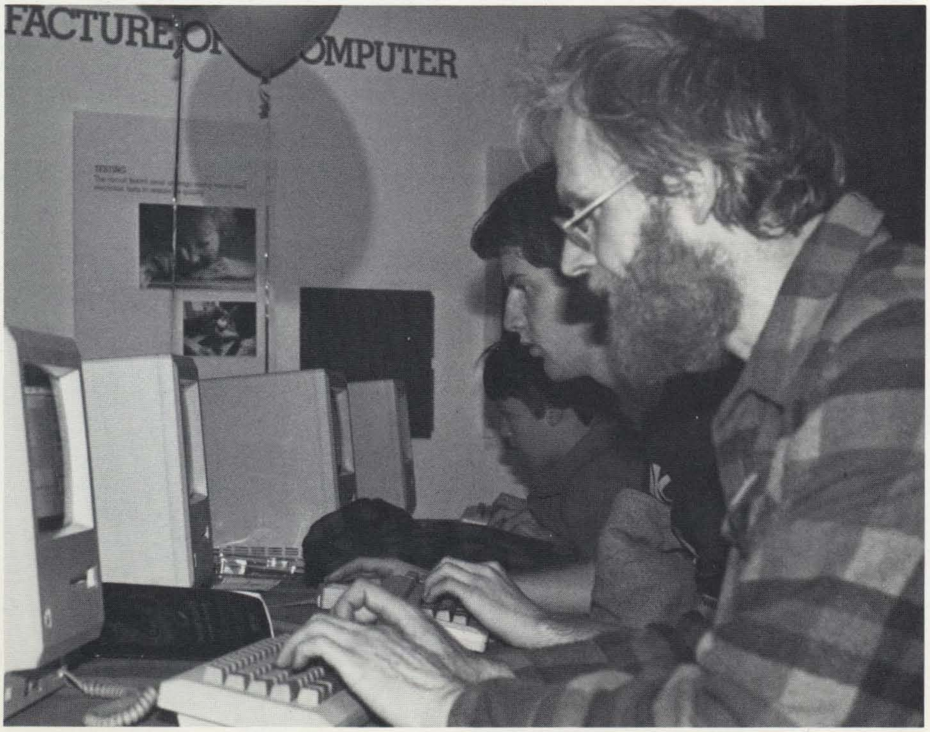
Charlie and Connie Bachman try their luck at "Spacewar!"

Running concurrently with the micromouse demonstration was a Core War teach-in, followed by the Second International Core War Competition. Twenty-five to thirty people took part in the informative session, led by Core War pioneer A.K. Dewdney and Core War Society Chairman Mark Clarkson. The participants learned about some of the more successful strategies and how to design their own Core War program. A Core War pits two programs, one against the other, in an attempt to gain control of a computer's memory. The eight quarterfinalists in this year's contest were narrowed down from a field of 130 and included the two finalists from a similar competition in Japan. The round robin style eliminations eventually trimmed the entries down to two — Ron Paludan's PLAGUE, and FERRET by Robert Reed. FERRET proved victorious in the best of five series, so that Reed succeeded last year's winner Chip Wendell on the Core War throne.

In addition to providing a festive opportunity for both the serious and the lighthearted gamester to enjoy a favorite subject, the weekend prompted national and international media coverage (from "Entertainment Tonight," Cable News Network, USA Today and stories by the AP, UPI and Reuters, to extensive features in the Boston Globe, Boston Herald and other local news outlets).

Coordinated by David Havlick, the weekend brought action to all aspects of the Museum. Our collections grew, members had a good time, and ideas for future games activities were generated. Watch out for Computer Games Month next November!

Players concentrate to stay alive in "Mazewars" and "Wizardry."

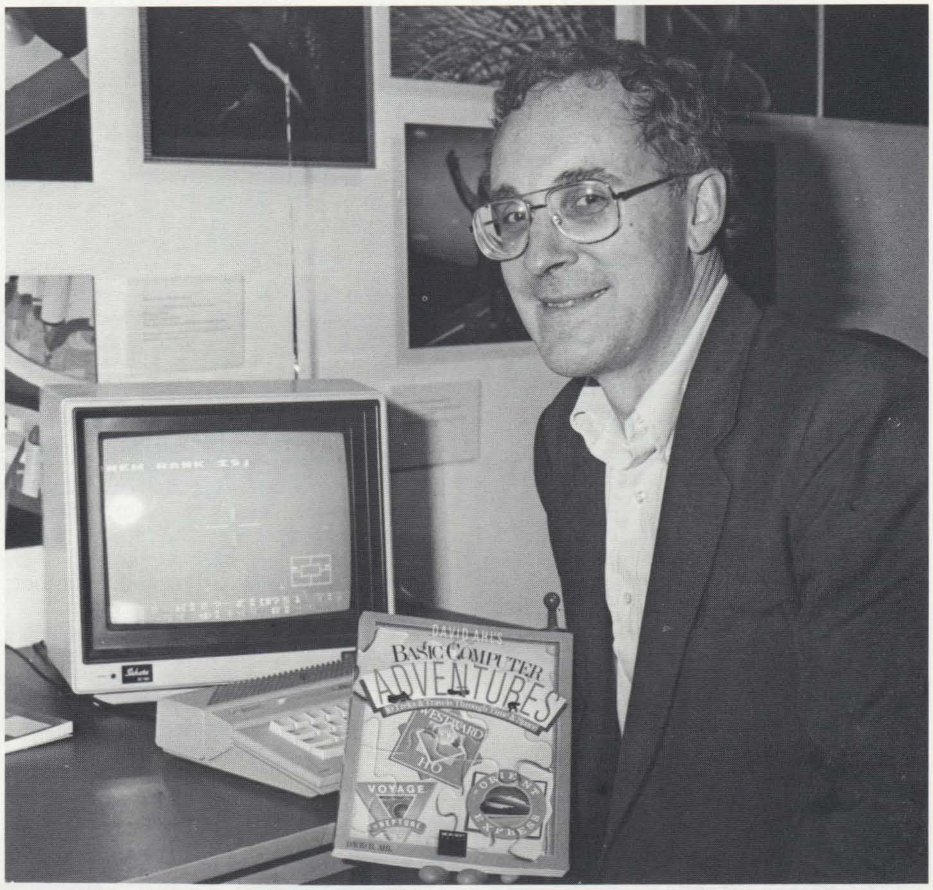


Linda Holekamp

The Beginnings of Computer Games

David Ahl

This is adapted from a keynote talk at The Computer Museum's Computer Games Weekend, November 6-8, 1987. David Ahl is the founder of Creative Computing, *the first magazine that focused on all the uses of the personal computer from games to science and home business.*



Jerry Rabinowitz

David Ahl brings fun and computing together with his books and magazines on computer gaming.

What Makes a Good Computer Game?

It takes many elements on several levels, skillfully combined, to make a good computer game. For example, good computer games are easy to learn, but not easy to beat. They are a challenge to expert players, but accessible to novices. They have elements of fantasy, but do not totally abandon reality. They are fun and keep us coming back for more.

One way of thinking of the world of computer games is as a Venn diagram of games, puzzles, and simulations (Figure 1). Simulations are representations of real-world processes such as a journey over the Oregon Trail, the landing of a lunar capsule, or a game of blackjack. Puzzles are problems with a baffling quality or great intricacy that require substantial mental ingenuity to solve such as the Chinese ring problem, the Lady and the Tiger, or even tic-tac-toe. And games, we know, can range from fantasy to shoot 'em up to Pacman.

Although thousands of computer games have come and gone, only a handful, such as Spacewar!, will be considered classics. I believe, in general, these classics will fall in the middle area of the Venn diagram. They will have some elements of fantasy, of simulation of real-world processes and people, and of puzzlement. While graphics may add to the visual presentation, they aren't really necessary. For example, the text adventure games from Infocom and others have elements of fantasy, simulation, and puzzlement which provide many layers of interest and challenge to a wide variety of players.

The First Computer Game.

Not only are we celebrating the twenty-fifth anniversary of Spacewar!, but in 1987, the thirtieth anniversary of computer games themselves.

The first computer game was developed in 1957 by Willy Higinbotham at Brookhaven National Laboratory. This is not widely known, and has not been widely written up, but I do know that some of the current games writers saw it and were influenced by it.

In the late fifties, people thought of computers as magic. At Brookhaven National Laboratories, one of the centers of atomic energy research, tours were held to educate the general public. Higinbotham noted that the visitors really couldn't relate to any of the machinery. He took a five-inch oscilloscope and devised a game. He used potentiometers to adjust the angle of little paddles in the bottom two corners. He put a line that represented a net in the middle and had a blip that bounced back and forth over the net, thus devising a simple game of tennis. The player adjusted the angle of the paddle to hit the ball higher or lower. You actually couldn't see the paddles but had to guess, based on turning the nobs of the potentiometer. One nice feature was that you always hit the ball if it came over the net. If you hit it into the net or over your head you lost. It wasn't a tremendously challenging game, but in 1957, it represented something that was "neat" and fun. I was a senior in high school, saw it and thought that it was spectacular. That was the first computer game even though it involved some special electronics and a mainframe with the capability of a small Atari today.

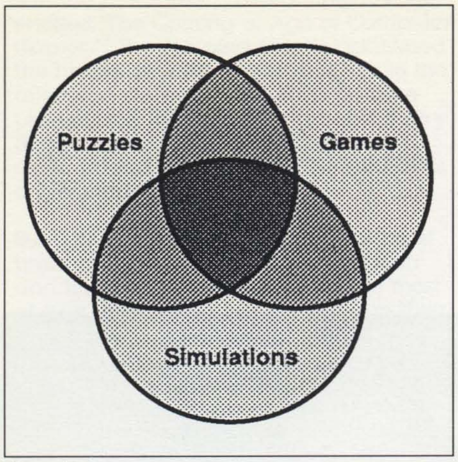


Figure 1

The First Widely Used Computer Simulation.
In the early sixties, the faculty of the business school at Carnegie started to build a monstrous business simulation known simply as "the management game," which in a form is still being used. The concept was set down in the late '50s to devise a simulation of the detergent industry, to allow students to take the role of companies and compete against each other, with a week equalling a year and play continuing for twenty years. What started as a simple marketing game then became more and more complex as other modules were added. In 1961 and '62, as the concept developed, additional modules were made for different areas such as research and production. A major challenge was getting these all to work with each other. It started to become a truly interactive simulation even though we had to feed the machine 3000 punched cards a week to run the model.

The original game was written in a language called GATE on a Bendix G-15 computer. In my second year at the Graduate School of Industrial Administration, I had a job to convert the program into the new language called FORTRAN. (I got the job because at the time I was one of the few people who knew FORTRAN, having learned it working at Grumman Aircraft on an IBM 704 simulating the cockpit controls of jet fighters.)

The PDP-8 Educational Simulations.
In 1969 when I joined DEC there really wasn't an educational market. The PDP-8s spoke machine language and FOCAL, an interactive language modelled on ALGOL written at DEC by Rick Merrill. It was a very interesting and powerful language that, in hindsight, could have been the generic language if DEC had made it widely available. Then BASIC would not have had a chance.

Rick Merrill also developed some simulation games — which is what interacting with a computer is all about. In one of these, Hammurabi, students manage a little city-state where they buy and sell land, feed their subjects, protect grain warehouses from rats, save grain for planting next year's crop, and deal with lots of little interacting variables. We fit both FOCAL and the program into the 4K memory available on the PDP-8. The original program was about 700 bytes. Since the world was not beating a path to DEC's door to buy FOCAL machines, we contracted with others to write BASIC for the PDP-8.

The BASIC Interpreter for a stand-alone $8500 4K PDP-8 with a teletype Model 33 used 3.6K of the memory. This left 400 bytes for the program. One of the first programs we managed to jam into this little machine was Hammurabi, which was soon followed by Lunar Lander — a game derivative of Spacewar!.

Level two of selling machines to schools was to sell time-shared systems. But these were hard to explain so we developed a demonstration. When we brought this to the Brockton School System they wanted to schedule it in the auditorium so that the citizens could come and approve this major expenditure for the school. The first problem was finding the nearest telephone and running a cord down the hallway to the auditorium. We brought our ASR 33 teletype and set it up onstage. A pamphlet explaining a scenario of interactions on Hammurabi was distributed to the audience. Then Jim Bailey dialed the computer at Digital. He heard the tone and it spelled out, "Logon please." He entered an account number and it replied "Logon please." After several iterations he realized the system was down. Since he was up on the stage, Jim said, "Hammurabi has just come back and said, 'How much do you want to plant?' No matter what key he pressed, the computer replied "Logon please." When the demo was over, Jim crumpled up the paper and put it in his pocket. The bottom line: Brockton bought the $58,000 system — the first Time-Shared 8 in a New England school.

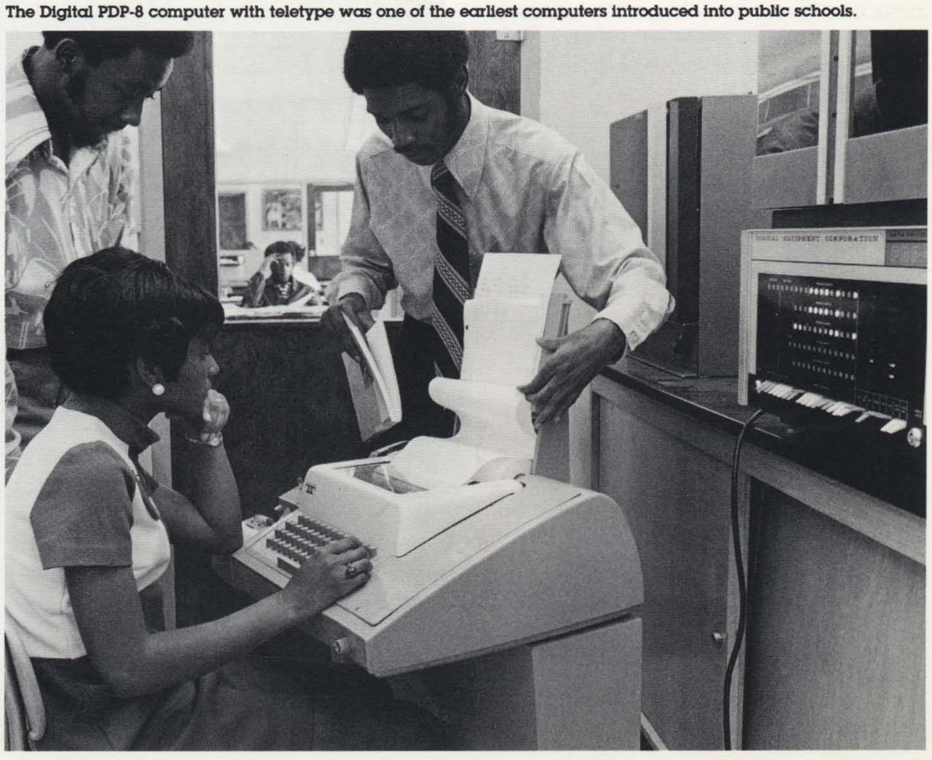
BASIC Computer Games.
At DEC there was little enthusiasm for publishing or distributing computer games. I was convinced they were of interest to our users. Because there was no support to publish BASIC Computer Games, I said 'I'll just do it. It won't cost anything. I'll type it in and do the layout myself.' It wound up costing DEC next to nothing and surprised everyone, even me, by selling out of the first printing of 10,000 in three months. In 1979, it became the first million selling computing book, in a version based on Microsoft BASIC under the Creative Computing label.

Its sequel, *More Computer Games,* did well, but the third book in the series, *Big Computer Games,* was printed but not distributed by Ziff Davis. My most recent book, *Basic Computer Adventures,* published by Microsoft Press in 1986, has ten simulations of real adventures such as the travels of Marco Polo and Amelia Earhart with a few puzzles built in.

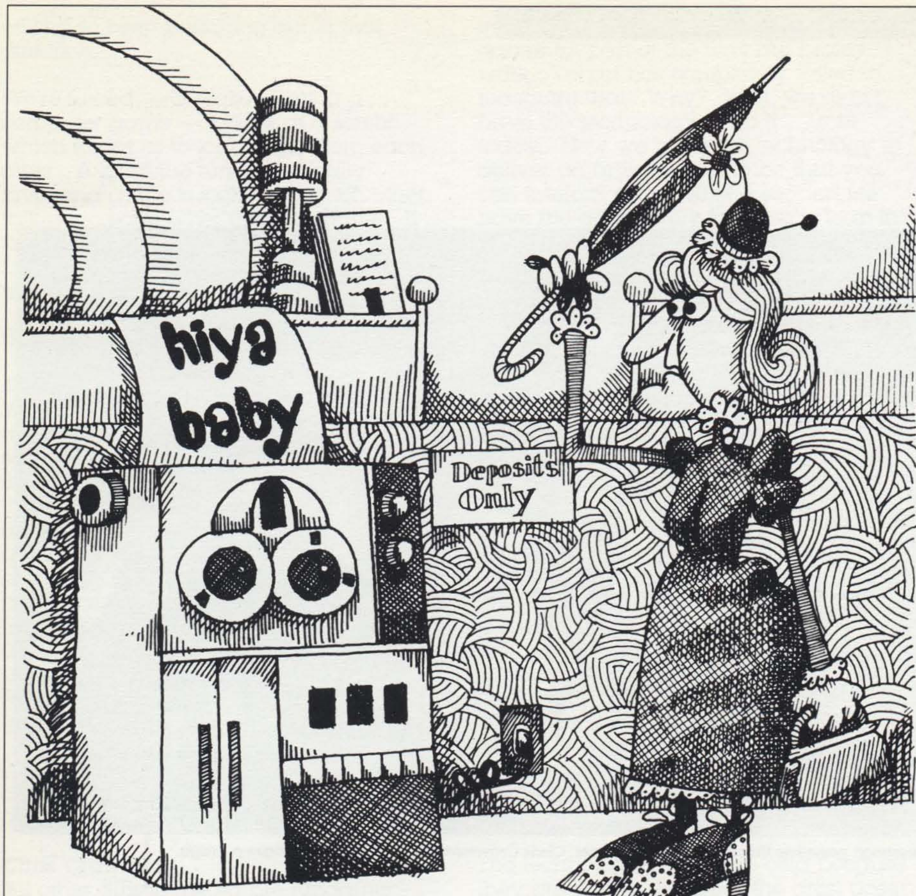
The First Personal Computing Magazine.
In November 1974, the first issue of *Creative Computing* came out, devoted to the idea that computers can be fun, not just business.

Nolan Bushnell's Second Game.
His first game was Computer Space, very much like Spacewar!. Unfortunately, it was distributed in the coin-op environment, bars and taverns, where the guy with a beer in one hand and a joystick in

The Digital PDP-8 computer with teletype was one of the earliest computers introduced into public schools.



Digital Equipment Corporation



Here is how the program looks:

```
01.05 SET RATE=.085
01.10 TYPE !!!!!,"HI THERE, GOOD LOOKING.  HOW MUCH MONEY DO YOU WANT",!
01.20 TYPE "TO BORROW ?";ASK PRINCIPAL
01.30 TYPE "THANK YOU DEAR.  HOW LONG DO YOU WANT TO BORROW THE ",!
01.40 TYPE "MONEY FOR ?";ASK TERM
01.50 SET INTEREST=PRINCIPAL*RATE*TERM
01.60 TYPE ,!,"SWEETS, THE GOING RATE OF INTEREST IS 8.5%.  IT WILL",!
01.70 TYPE "COST YOU",INTEREST," DOLLARS TO BORROW",PRINCIPAL
01.80 TYPE " DOLLARS",!,"FOR ",TERM," YEARS.  YOU DO UNDERSTAND"
01.85 TYPE ", OF COURSE,",!,"THAT THIS IS SIMPLE INTEREST.",!!
01.90 TYPE "STEP RIGHT UP TO OUR TELLER AND HE WILL BE GLAD TO",!

02.10 TYPE "HELP YOU.",!!
02.20 TYPE "NICE TALKING WITH YOU.  DO STOP IN AGAIN.  BYE,BYE,NOW.",!
02.30 GOTO 1.1
*
```

From Digital Equipment Corporation's FOCAL brochure. Gift of Karl West.

Digital's conversational programming language, FOCAL, may have had great potential for the PDP-8, but was soon overshadowed by the popularity of BASIC.

another wasn't up to learning the complexities of Spacewar!. Atari produced about 2,000 units but it never really was a big success.

Pong, a very simple and clever game, was a runaway hit. The story is that the first Pong game was put in a bar near Sunnyvale. Several days later Bushnell got a call asking him to take the game out because it didn't work. He took a look at the game and found that the breadpan of quarters was so full that the coins were jamming the mechanism. When the quarters were emptied once a day, it worked well. Eventually game designers built large coin receptacles eight inches deep under the whole machine.

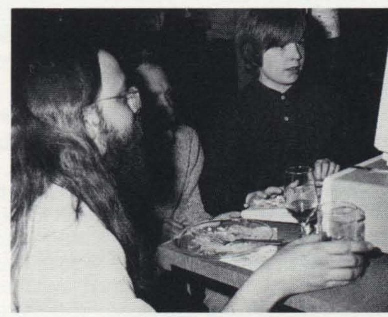
The Video Computer System (VCS).
There was no one device more responsible for getting computers and games into people's homes than Atari's VCS (called the 2600 today). First announced in 1978, it sold by the millions and got people thinking about games and computers.

Computer Games Overdose.
By 1982, over 6 billion dollars of quarters per year were being put into the slots of coin-op games alone, making that segment of the industry bigger than the rest of the sports industry combined, including football, the Indy 500, World Cup Soccer, and the Olympics. Hundreds of new games were announced and the life of a game went from over one year to less than two months. Less than one year later, boom turned to bust as manufacturers slashed prices and flooded the market with "me-too" products. Players got disgusted, and manufacturers, retailers and arcade operators started to go "belly up." The boom ended, but the games will go on forever.

The Beginnings of Rogue

Ken Arnold, the co-designer of Rogue, spoke about how he co-invented it less than ten years ago at Berkeley.

Since I'm less than thirty, I'm awed that I'm part of a history section. When I was first an undergraduate at Berkeley, the terminal room had ADM machines where you could only move the cursor down the page. This limited us to text games like Adventure and Rogue for the people who had ARPAnet accounts. Then came the dumb terminals where the cursor could move anywhere on the screen. That was really a boon to gaming. Then, people started to CRT hack....that is, draw pictures on the screen and move them around. For about two months that seemed to be entertaining. Some people decided that this was the way to start writing games.



Jerry Rabinowitz

Ken Arnold and "Rogue," a program that took "a billion and a half dollars of compute time."

Rogue was developed by Michael Toy at Santa Cruz. He then came to Berkeley when the game had no real magic, such as potions. I had written some utilities to use the cursor on the terminal and so he came to me to help me. Having a lot of recommendations to change the game that I was now addicted to, we started to work together.

Michael set four goals that were unique at the time. First was to move away from text-only adventure games that are essentially mazes with the player as the mouse.

Second, Michael wanted to write a game that would be different for the player every time and interesting for the writer to play, the innovation was to use a random number generator to create new landscapes each time.

The third decision was to make a game that was impossible to win. Without a couple of forms of cheating, Rogue is only possible to win one out of every hundred thousand times.

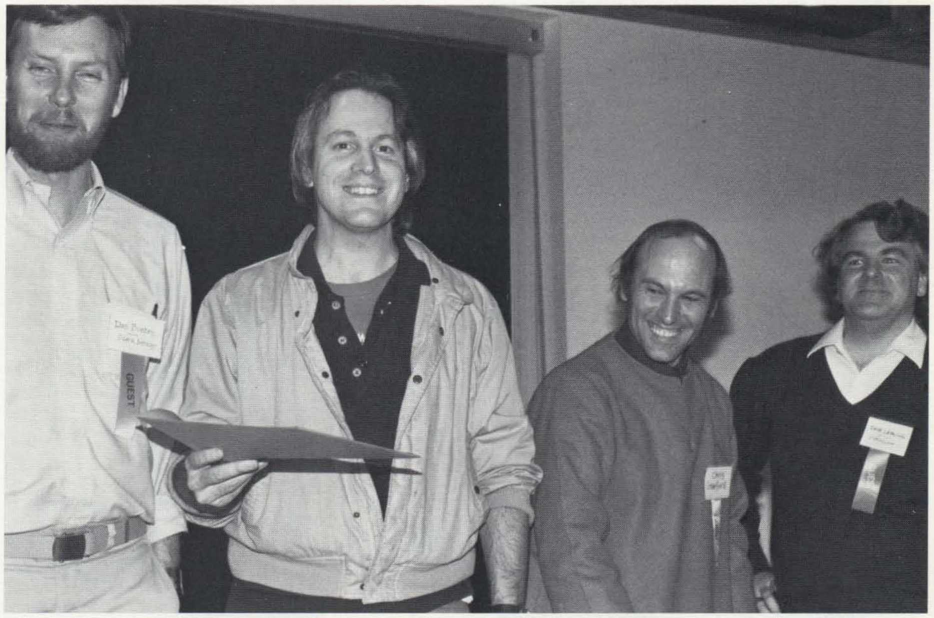
Finally, Rogue was designed as a long game — taking two or three hours to play and thus it never became appropriate for an arcade.

Rogue is one of the most copied games; after royalties the second most sincere form of flattery. After three months at Berkeley, the game used more compute cycles than any other program. Two years after Michael and I released Rogue, we calculated on the back of an envelope that we had used about a billion and a half dollars of compute time in Silicon Valley.

The Future of Computer Games

Panel: Dan Bunten, Chris Crawford, Dave Lebling, Tom Snyder. A. K. Dewdney, moderator

These are some highlights, adapted from a panel discussion at The Computer Museum's "Computer Games Weekend," November 7, 1987. ***Dan Bunten*** *is the designer of the award-winning games "M.U.L.E." and "Seven Cities of Gold."* ***Chris Crawford*** *designed "Balance of Power" and wrote* The Art of Computer Game Design. ***Dave Lebling*** *works for Infocom; his games include "Zork" and "The Lurking Horror."* ***Tom Snyder*** *produces educational software such as "Puppy Love" and "Snooper Troops."* Scientific American *columnist* ***A.K. Dewdney*** *pioneered the Core War computer program competition.*



Linda Holekamp

Afternoon panelists Dan Bunten, Tom Snyder, Chris Crawford and Dave Lebling share a laugh.

A.K. Dewdney:
We are all aware of the general view of computer games as mindless spinal recreations involving nothing higher than the cerebellum, that little mass of gray matter above your neck that helps you to play the piano, tennis and also to shoot hostile aliens. There are some who understand that there's a lot more to some computer games than that. I would say the intellectual content of games bears watching. A key question is: what is going to happen to that intellectual content in general? Will these games become more demanding at the cerebral level than at the cerebellar?

Sometimes to be educational, a computer game gives up recreational content. At the same time, it almost seems that the more recreational a game, the lower a common denominator it demands. Another important theme is the single player versus multi-player issue. I submit that there are no four people more competent to describe the current place of computer games and their future prospects than the four game designers [on this panel].

Beyond Nerddom: Multi-Player Games

Tom Snyder:
In 1962, I was introduced to nerddom. I found a book on computer relay circuitry written by Bell Telephone. And I designed a couple of binary coded decimal computers. I thought I invented digital electronics. My father told me I ought to send my paper plans to IBM, which I did. I was twelve at the time and I knew it was cute, not important. About a month later, when I came home from school, there was at least $10,000 worth of computer equipment on my front lawn with a note from the president of IBM saying, 'Remember us when you get older.' (joke)

I proceeded to go off the deep end at that point and told my parents that I'd like to make this computer I'd designed because I had the parts to do it. So all I did was computers because they were the one thing I could control in my life. When I was sixteen I gave them up because I had basically no social skills and found I couldn't get along with people. Since then, I've had a healthy respect for how uncool computers are for adolescents.

Dan Bunten:
This generation of computer owners doesn't feel comfortable about owning their computers. It's a bit of a sin. It's something we hide in our back rooms. We don't let our families in on it — there are a few inside friends we might tell, 'Yeah, I got a computer, but it's back in my office.' But you don't bring them all back there and say, 'Hey, we got this great game, why don't we all play it?' You know, it's not part of our social acceptability somehow. That's one of our problems.

I want to reach some level of success that says that now we're communicating with people other than nerds like ourselves.

Snyder:
At a baseball game you do two things — talk about what's going to happen and go to the bathroom. But it's great. There's something extremely social about these sports — it's the talking about it.

Intellivision's two biggest games were baseball and football. I had the problem of finding somebody to play with — I mean grown men don't invite each other over to do trivial things.

Dan Bunten did one of the few four-people games. There were quite a few two-people computer games out there but Dan really pushed the limit. Four people is better than two — that's a real great party. There's a lot of talking, kidding and social context, a lot of self-handicapping. People learn some rules about society when they're playing games. You don't learn rules about society playing with yourself.

Bunten:
Go back to what games were about. They're about people interacting with each other. They're about having fun with your friends. I have to say we're having trouble with that one, but I'm

willing to keep punching out in that direction.

We're forced — if we're playing a computer game — to look at a screen, which forces us to look away from each other. A lot of the fun in a socially involving game is looking at each other, talking to each other over the game. If the computer gets too good at being the focus of attention, then we've lost what we came here for.

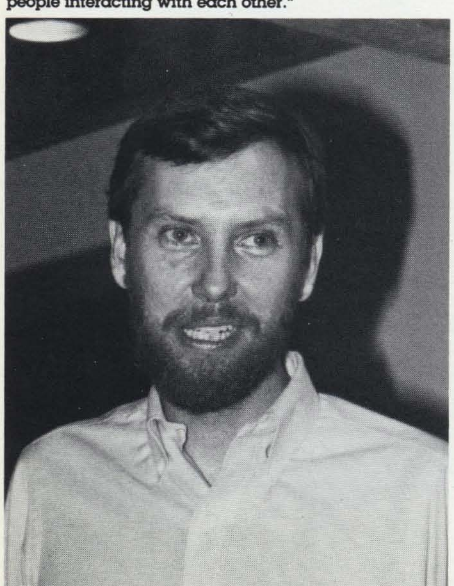
Games as Interactive Art

Dave Lebling:
What Dan said is true. The fun is not so much in playing the game but in the social interaction of four people playing the game. The fact that they have to sit there staring at the screen is really a drawback. I agree that multi-player games are really important; I'm not so certain that the technology is there to make them a big market yet. But I'm hoping that the things that will push it along — the way "Lotus 1,2,3" did for personal computers — will come about. At some future date, I think multi-player games will exist and be very good.

I think we're working in a pulp medium and we are working for what is in effect a pulp audience.

Chris Crawford:
All other artistic media are fundamentally non-interactive. Basically what you do with every art form is sit on your butt and absorb it. So we play wonderful music and what do you do? Sit back and listen. We paint a beautiful painting and you look at it. We write a great book and you read the book. But what do you actually do in all this? Nothing. You're passive. And that's a fundamental failure because the human mind is not a passive receptacle. You don't just open up the top of the skull and pour stuff in. The human mind works best when it gets to take the butterfly and tear the wings off it and play with it and

Dan Bunten: "Go back to what games were about... people interacting with each other."



Linda Holekamp

interact with it. That's an absolutely fundamental part of the way our brains work. Yet art has completely failed to recognize that. Why? Well, we didn't have the technology to do it.....until today. Now we have the technology to deliver an artistic experience that you can interact with. All of a sudden we have the opportunity for a large leap in the amount of human involvement in the artistic experience.

You can move technology forward on a timescale of months or years, but art? Art takes wisdom, and that takes a long time — decades or centuries.

Artificial personality is an artistic medium or regime dedicated to the capture of human nature through the medium of the algorithm. Now that may strike you as a little sick. Algorithms are cold mathematical equations. Somehow I'm going to try to express human personality through a cold medium like an algorithm? That may sound sick to you, but let me remind you that stone is cold. Look what somebody did with it when they made a statue called The Piéta. What about cat gut? Let's take the insides of a cat, cut him open and stretch out his insides. What are we going to do with that? We're going to play Beethoven's violin concerto.

The technologies of art are cold because they are things. It's what the artist does with the technology that breathes life and warmth into it. There's nothing intrinsically cold about algorithms. It's how much art you bring to them. The fact that so far algorithms have been exclusively in the hands of scientists and programmers is only an indication of how little artistic effort we've made so far.

Snyder:
Movies, books, records have common elements: love, sex, greed, sorrow, happiness, plot development, character development, people caring about each other, people getting angry, people killing themselves, people killing because they're in love.

Bunten:
We can't engage people by making better landing gear. At some level the vast majority of the human race cares about other people more than they care about things. To me, one of the best ways to let them manifest that care in relation to computer games is to make computer games that let people interact with each other and not with emulated, imagined or supposed characters inside a computer, no matter how good they are.

We may think that if we get a great new resolution or great sound shifts, we're going to suddenly have people saying, 'Hey, this is just like TV.' Well so what? TV's already here — we don't need something like TV.

Linda Holekamp

Chris Crawford: "Characters are what we care about."

Interactive Fiction

Snyder:
Interactive fiction is one of the headiest concepts of the '80s. It's also one of the most problematic entertainment forms of the '80s. That isn't to say we shouldn't develop it. Most entertainment software is missing some incredibly important elements that entertainment is all about, some kind of identification and caring about the character.

Crawford:
The single thing I identify as our biggest failure is not putting any characters into our games. That won't solve our problems, but we haven't even reached square one until we have characters. Imagine movies without characters in them. Imagine literature with no characters. Theater with no actors. Take the movie "Star Wars," and take out Darth Vader, Luke Skywalker, Han Solo, Princess Leia, R2D2, C3PO, Obi Wan Kenobi, and what do you have left? 'Da daa da da da daa daaa, zap zap boom. OK, roll the credits...' that's what you've got — nothing.

We really don't have any characters in our computer games. The characters we do have are fake. The best character I've ever seen in any computer game is Floyd, the robot from "Planetfall." Floyd is a cute guy who does funny things and then dies. But you see, if you walk up to Floyd and say, 'Floyd, I hate your stinking guts,' well, then Floyd is still a cute guy who does funny things and then dies. Because you see, Floyd isn't real. He's a fake. He doesn't have any personality. He doesn't feel anything. He doesn't even know you exist. He is a Potemkin Village. And he's the best we've got. In all of computer gamedom, we don't have a single character as rich, as subtle, as complex as Gilligan from "Gilligan's Island." We have yet to climb up to the level of television. It will be an artistic milestone when we get a game as good as the "A-Team" or "Dukes of

Hazzard." So we're in a terrible situation right now.

I say characters are what we care about. When we watched "Star Wars," we didn't care about the spaceship or about the zapping and all of that; what we cared about was the people. We need to put people into our games.

Lebling:
I really want to see games where I sit down and say, 'Floyd, let's not play hucka bucka beanstalk. Let's read Tolstoy,' and Floyd says, 'Ooh, I love Tolstoy,' and you go off into this completely different story. In "Pacman" if you had wanted to learn to coexist with the ghosts, wouldn't it have been wonderful if the author had had that in mind and handled it?

That begins to shade into interactive fiction where we always like to say you are in control of the story. Now we all know that that's really a lie because you really aren't. But wouldn't it be wonderful if you were?

Bunten:
I think that if a story is really important, it can't have a bunch of different endings. If somebody's that excited about this particular story, it's got one ending.

'What ifs' are interesting but they're not the same thing as a compelling story, well told, that involves you and brings you in.

Lebling:
In the real world you get one run through. One of the things about reading books and seeing movies is that it's like getting another run through in that world. You see somebody else's run through and maybe it helps you do yours a little better. If you could do many, many runs in the same interesting world, it might help you even more.

Snyder:
Character development is the key issue that's really holding us back. It's difficult to find an author who wants to write 9 million contingencies; most of them have a vision as Shakespeare did — that there's a character who's going to learn something, who's going to grow because of a sequence of events. Those are the kinds of authors who have existed for the past 2500 years, since Homer's time. They have a personal investment in themselves as artists creating an experience we're going to have. They don't give a damn about what my notion is about the order in which their story ought to take place.

You (Chris Crawford) continue to say that what's important to a good story is characters; I continue to say what's important to a good story is character development. If you just have free-floating characters, it could be interesting, it could be junk. I'm not interested in that.

Lebling:
Even if Floyd is the best character in the world, it wouldn't advance the story. The characters in popular fiction aren't that complicated. What's important is empathy.

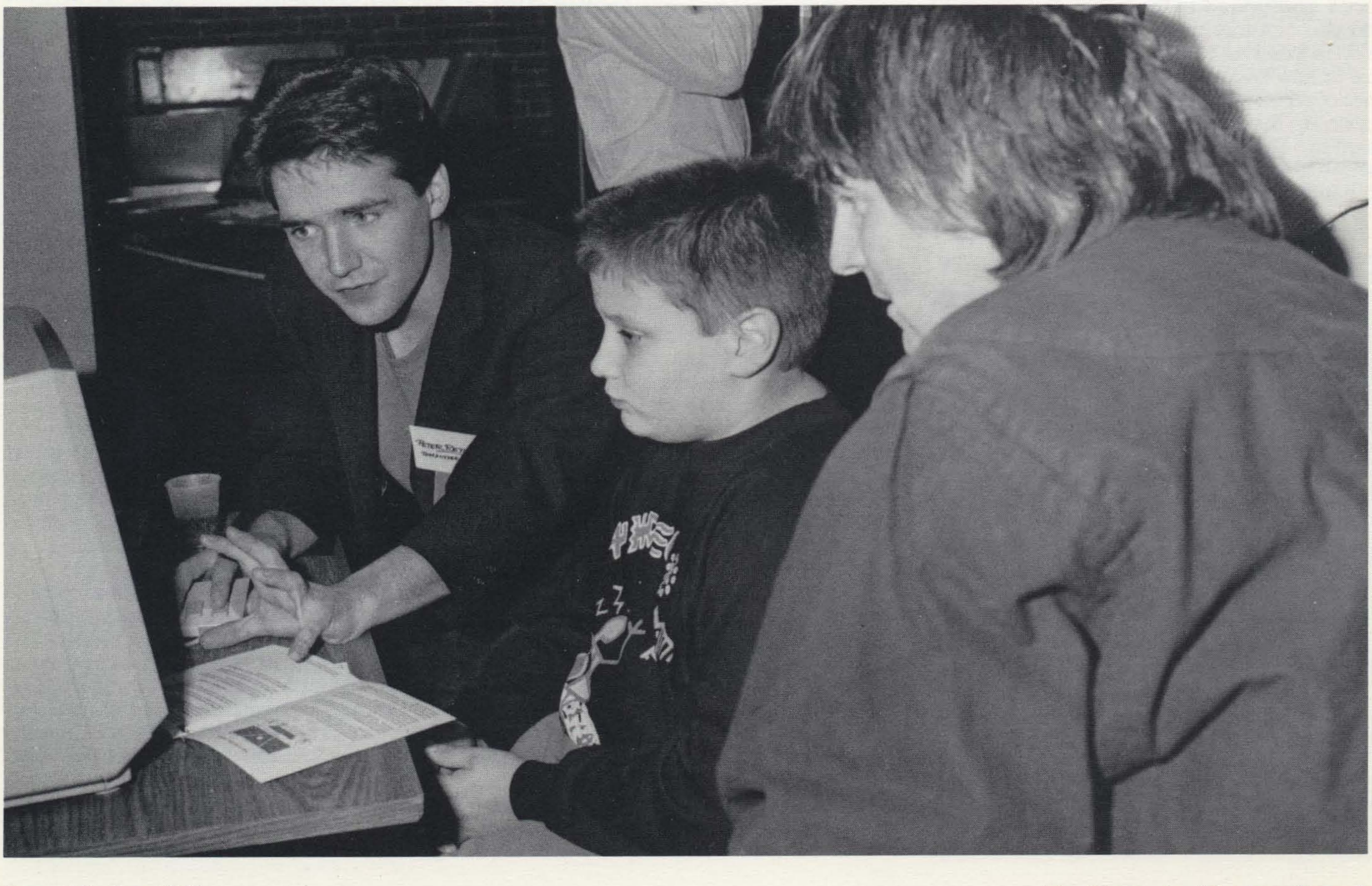
Bunten:
I understand the problem of building characters into a computer, and I sympathize with it. In fact, I would bow out of that problem and say, 'OK, we don't want characters in a computer. We want environments, worlds where I can be the character, the guy who runs out and does the neat things.'

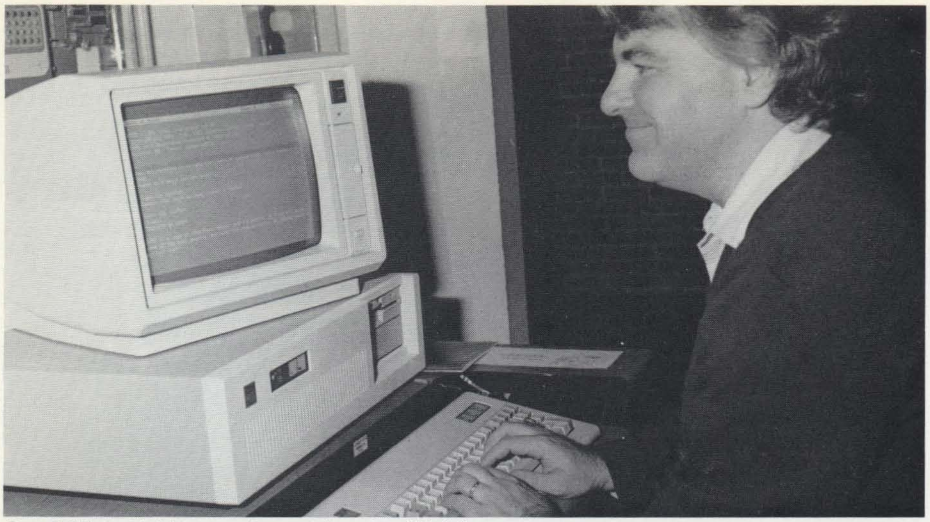
Snyder:
I don't want to be sexist, but I think it's an interesting statistic that more than 50% of all purchase and rental decisions on books and movies are made by women. Do you think for software it's anywhere near 50%? But I don't think we ought to bring women into this just to make the market bigger. We're not going to be happening if we just add another 50% of the population. I'm talking about women having the same kind of synergy that exists around books and records, where the pop culture explodes and grows.

My mother will learn to use any machine if it has those elements of personal emotional identification that are so important to her. The things that are important to mom are stories. She loves to program her VCR because there is content in there that makes a difference to her.

I think our industry has to stop and rephrase some things. There's a kind of

Peter Reynolds and Tom Snyder coax a young voyager into "The Halley Project."


Jerry Rabinowitz



Jerry Rabinowitz

Dave Lebling — in his own underground empire of "Zork."

looking down our noses, putting down the general public who refuses to 'interact.' Let's blame the forms of interactivity rather than our willingness to do interactivity.

Language

Crawford:
If you create a character inside a computer, then you have to be able to interact with him. The primary way human beings interact with each other is through language. You've got to talk to this person. How are you going to do that? There's an easy answer most people think of: use English. Talk to them in a regular language. Good luck. I can tell you right now you're not going to be able to talk to anybody in a computer in this century. A lot of people grossly underestimate the problems of getting natural language working on a computer. There are three major problems. The first is vocabulary, the second is syntax and the third, context; context is the killer. Vocabulary is a trivial problem. You just take the words, stuff them in memory, no big deal. There are only 600,000 words in the English language — a few megabytes of storage. Trivial. You can do perfectly well with a working vocabulary of 5000 words. You can say almost anything you want to say with the 5000 most commonly used words in the English language. How much storage would that take? A few K, no big deal. OK, second problem — syntax, grammar. You've got to store all the rules of the English language. Again, no big problem. A few years or decades of programming, but that's a solvable problem. You just start writing in the codes for all the weird rules in the English language. It'll take a lot of time, but it's manageable.

The killer is context. You see, language does not exist in isolation from reality. It mirrors reality. A word is not just something that sits in a dictionary or a look-up table in RAM. A word means something. And if you're going to understand its meaning, then you're going to have to understand the universe to which it refers. Let me give you an example of just how hairy this can get. Consider the following sentence: 'Computer, do not forward John Doe's personnel file to Mary Smith because I saw him sneaking out of her house this morning at 6 am.' Now think about the amount of knowledge you have to have about the world and human behavior to understand what that sentence means. Then think about putting it inside a computer. That's the killer. If you're going to put English inside a computer, you're going to have to put the whole universe in there too. That will take a little while.

Lebling:
How can we do English? That's a good question. We need to figure out how to expand that part of the universe which we simulate. The fallacy is that we have to do everything. There are 600,000 words in English. But even [MIT linguist] Noam Chomsky doesn't know what the grammar of English is. The meanings and the context are incredible, but only if you want to do everything.

But the key is: let's do a bit. A little box somewhere. Let's do that box really well. Then, let's define the boundaries of that box unambiguously so the person who's interacting knows where the boundaries are and doesn't get surprised because he can't wander off into a completely different geography from the one he thinks he's in. Let's just build that box a little bigger every time, get those boxes linked up right, and then we can do as much of the universe as is necessary to make good stories.

The Computer Games Weekend was made possible by generous support from:

David Bunnell, PC World Communications Inc.
Martin "Shag" Graetz
Christopher Morgan
Origin Systems Inc.

Special thanks also to the following:

Addison-Wesley Publishing Company Inc.
Apple Computers Inc.
Cynthia Barker
Bel Canto Restaurants
Michael Bergman
Allison Block, Cabot Public Relations
John Broder, Giardini/Russell
Broderbund Software
Jack Burness
Andrea Coville, Giardini/Russell
Chris Crawford
Paul Dworkin
Electronic Arts
Epyx
Alan Frisbie
Wendy Germain, Germain DRK Inc.
Infocom
The Lafayette Hotel
MacroMind Inc.
Charles Mann
Microprose Software
Mindscape
Sheryl Schultz, Sterling Hager
Stan Schulz
Josh Shapiro, Cabot Public Relations
Sir-Tech Software
Roxanne St. Clair, Hill and Knowlton
Katherine Stimpson
Jeffrey Tarter
Tech Plus Inc.
The Learning Company
Union Office Interiors

Whirlwind's Genesis and Descendants

"Worldwind's Genesis and Descendants" was the theme of a symposium held at The Computer Museum October 18, 1987. This was part of a weekend reunion of the Whirlwind group organized by David Israel. The symposium was recorded at the Museum and transcribed by Judy Clapp of the MITRE Corporation. Responsibility for the accuracy of the following adaptations of the talks belongs to The Computer Museum.

Jay Forrester, T. K. Finletter, and F. Wheeler Loomis visit the Whirlwind in November, 1951.

The MITRE Corporation Archives

Whirlwind's Success

Jay Forrester

Jay Forrester is Germeshausen Professor of Management and Director of the Systems Dynamics Group at MIT. He was the leader of the Whirlwind group at MIT from the late forties until 1956.

Why did Whirlwind succeed? Why did more technical innovations out of Whirlwind persist into the present time than from any other of the early computers? The reason revolves around several things: the vision of the future direction of computing, a dedication to excellence, and the organizational environment.

Project Whirlwind's Future Vision

The vision in Whirlwind reached well beyond the uses of computation and hand-calculating machines at that time. Our work quickly became identified with the field of real-time control and reliability.

The dedication to real-time control started well before Whirlwind first operated. In October 1947, when we were still determining the logical structure of the machine, two reports were written in the MIT Computer Laboratory suggesting that the Navy could use digital computers as Combat Information Centers for coordinating an anti-submarine task force. This meant coordinating the air, the surface, and the subsurface pictures to get an understanding of the totality of what was going on.

Building Reliable Systems

Reliability was important because you can't go back and do things over again in military applications. In 1948, before Whirlwind operated, Karl Compton, then President of MIT and also Chairman of the Research and Development Board, asked that we prepare a memorandum for him on the future use of computers in the military. Bob Everett, Hugh Boyd, Harris Fahnestock and I took two or three weeks to answer that question. The report culminated in a chart listing vertically about twelve wide-ranging areas of computer use in the military, such as logistics, scientific computation, air defense and anti-ballistic missile control. On the other axis were 15 years from 1948 to 1963.

That report is quite an interesting document in historical perspective. At each intersection in each square in the table, we estimated the condition of the field at that time, how much money would be spent yearly in research, engineering and production, and what the condition of the field would be relative to those end uses 15 years into the future. These estimates were made when no high speed general purpose computer had yet functioned.

The estimates are percentage-wise as good as and maybe better than most estimates made today for the time and cost of the next computer to be put into production. This was because we paid a great deal of attention to the political as well as the technological side. The cost estimates were arrived at by subdividing tasks to no more than 30 people working a calendar quarter and by deciding all the things that would have to be done. It was not necessarily correct in detail but it was a logically complete scenario including how long it would take for people to believe the results of the previous year, and how long it would take to get funding for the next step. The chart showed a total of $2 billion to be spent in research and development alone over the 15-year period. We went into a Navy conference with this. They thought the agenda involved whether we could have the next $100,000. There was a communication gap in that meeting.

Dedication to excellence

Many people in the Whirlwind group had had the World War II experience of going from theory through research to production design, then to manufacturing and into the battlefield, fixing their own mistakes at every stage. They understood how the decisions at the research stage really affect what happens later.

In my own early background, I had already started down that road, having grown up on a cattle ranch where you learned that if you did a sloppy job of fixing a tractor or a well, you would suffer the consequences very soon, have to do it over, and do it right. Part of the manifestation of that viewpoint showed up, of course, in our improving vacuum tubes. Until the 1950s, vacuum tubes primarily had been used for radios. Radio engineers were not concerned that the life of a vacuum tube was about 500 hours. But computer engineers, considering the use of many thousands of vacuum tubes, easily estimated that with such a short life, the machine would run no more than a few minutes between failures. One of the achievements of our group was determining the cause of failure of vacuum tubes. It turned out to be one thing. After removing that cause in the design, the life of vacuum tubes was increased, in one design step, from 500 hours to 100,000 hours or longer.

Excellence also meant thorough testing of components. We built a five-digit multiplier for the simple purpose of finding out whether an electronic device running continuously would be trouble-free or not. There was uncertainty about things that people now thoroughly understand.

One important issue was our uncertainty about thermal noise. We didn't know if random spikes of thermally generated noise were big enough to trigger our robust computing circuits. We wondered whether thermal noise would intrude itself often enough to be devastating to accurate computation. To test for this, the five-digit multiplier was run continuously. Every multiplication was checked against a reference number. Sure enough, it didn't compute reliably all the time. It had a great tendency to make mistakes at 3 a.m. This was traced to the janitor in the building next door, who would start the freight elevator at about that time, upsetting the power circuits enough to produce a computation error. As a result, a rotating motor generator with enough inertia to carry through that kind of transient noise was installed on both Whirlwind and the SAGE Air Defense machines. It was an expensive solution but a very effective one.

A lot of time was spent writing test programs to find out the source of a failed component. Occasionally, a visitor was asked to go any place in the computer racks, pull out a vacuum tube and bring it back to the control desk. When he got back, the location of the empty socket would have been typed out by the machine itself. Finding solid, existing, reliable errors, like a tube pulled out of its socket, was not nearly good enough.

Other means of determining reliability were also essential, which we discovered in various ways. I remember one Saturday, during one of many annual reviews, our inquisitor asked, "What are you going to do about the electronic components that are drifting gradually and are on the edge of causing mistakes? Any little random fluctuation in power, or streetcars going by, will cause circuits to sometimes work and sometimes not." This was a very important and powerful question that, frankly, we had done nothing about. It was such a pointed question and obviously such an important one that I felt an immediate answer was essential. I said to him, "Well, we could lower the voltage on a tube and convert it from a marginal to a permanent failure and then it would be easy to find." He thought it was a good solution and so did we, so the next Monday we started designing it into the computer. The marginal checking system in Whirlwind carried over into the SAGE Air Defense system, adding another factor of ten to the reliability.

Many of you may not know the statistics on the SAGE system's reliability. There were 30 or more SAGE Centers. Each building was about 160 feet square, four stories high, with upwards of 60,000 vacuum tubes in it. The question is: what percentage of the time do you think such a center would operate reliably? The answers I get from an audience today tend to run from 15% to 60 or 70%. They're really quite overwhelmed when they're told the historical statistics on the SAGE Air Defense system. It was installed in the late 1950s and operated for 25 years, until 1983. According to the data that Bob Everett was able to find, the uptime was 99.8%, which is really quite remarkable. In fact, you will have trouble finding anything equal to that, even when it has been designed with more modern components.

The attitude about the SAGE performance was that it must work reliably. To achieve high reliability, one must be a devout believer in Murphy's Laws — that if anything can go wrong it will. Every possible failure must be identified and forestalled. This attitude is the difference between something that is strikingly successful and disaster. In almost any major disaster, whether a technological or a social one, an ample number of people knew that it was likely to happen and knew in advance why it was going to happen. The information was there, and either they did not take any action, or they tried, and in the social circumstances of their environment, were not able to get any results. A warning is almost always present ahead of the trouble and the problem comes in getting any kind of action or acceptance of the threat.

The Organizational Environment

Another part of the success of the Whirlwind group came from the organizational environment within which we were operating. MIT in those days was a free enterprise society in which someone who had a vision and could raise the money for it could do what he thought was important.

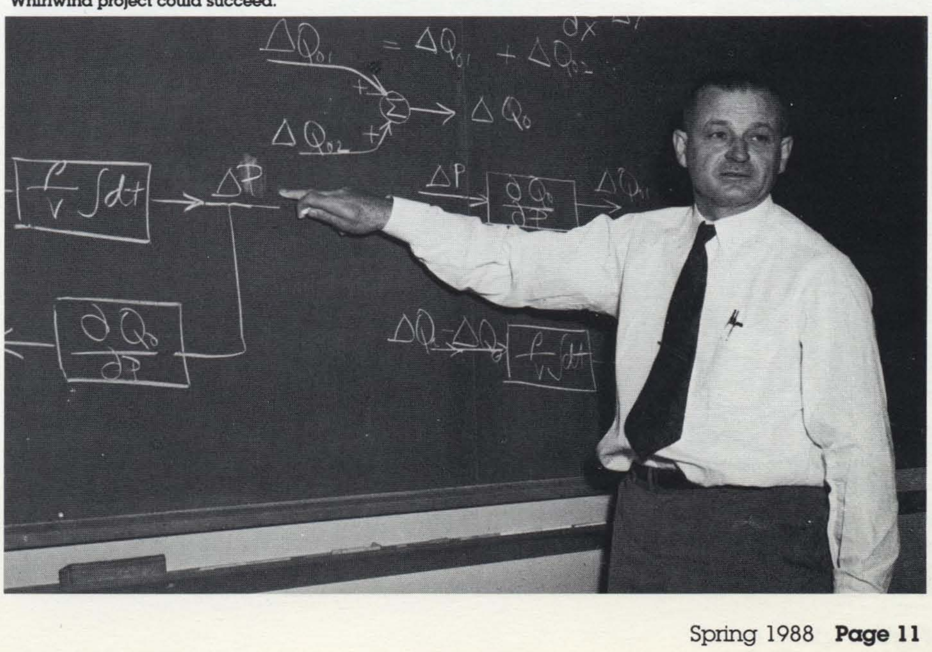
The Leaders

Within our immediate environment, two people conspicuously stand out as having made it possible for us to operate the way we did. One was Nathaniel (Nat) Sage, Director of the Division of Industrial Cooperation, under which outside funding came into MIT and the other was Gordon S. Brown. In addition, there were two promoters, in the best sense of that word, people who shared the vision and who spent their time building up the outside constituency to support the work. These were Perry Crawford and George Valley.

Sage, a civil engineer by training, was the son of an Army officer and grew up in Army camps around the world. Somewhere in that experience, he developed into a very good and self-confident judge of people. There were people at MIT that he trusted implicitly, and there were others that he wouldn't trust any farther than he could see them. Sage trusted Gordon Brown, Stark Draper, of the Draper Laboratory, and I think I can claim that he trusted me. He had confidence in us, lent great support to us, and would do rather remarkable things for us. I remember when someone chartered an airplane to come back from somewhere because it was a sensible thing to do to get home for the weekend. That caused an explosion in the Military Contracting Office where they thought this was not an appropriate use of funds. The contracting officer went to Nat Sage as the senior person. Sage would listen to them, nod, sympathize with them and say, "That really is too bad." Then he would put the whole thing in his desk drawer. He would never even tell us that the question had been raised, because he believed it probably was a proper thing to do.

Gordon Brown, my mentor at MIT, and director of the Servomechanisms Laboratory under which the Computer Laboratory operated, was a person who threw a great deal of responsibility onto young staff members, even as research assistants in the Electrical Engineering De-

Dr. Gordon S. Brown, who was Director of the Servomechanisms Laboratory, helped create an atmosphere in which the Whirlwind project could succeed.



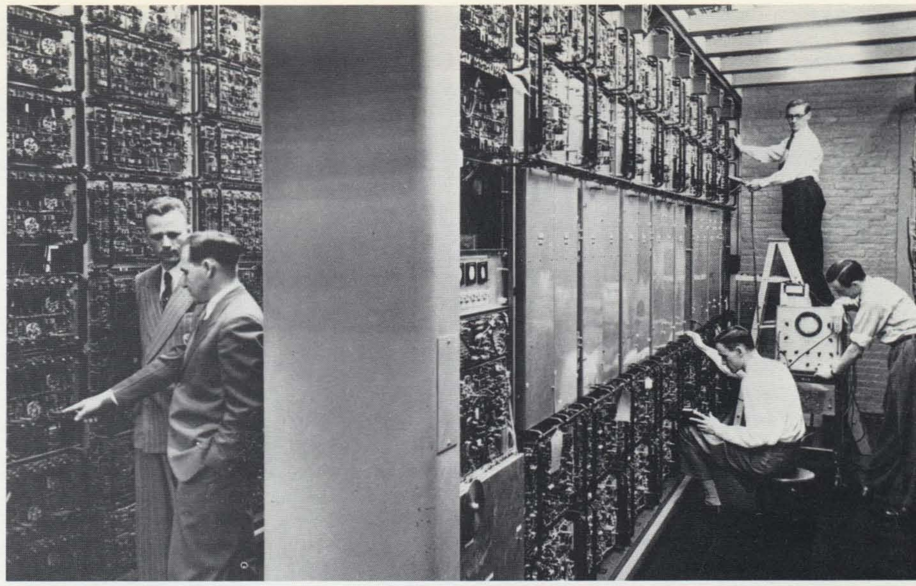
The MIT Museum

partment. He provided an environment in which people developed very rapidly, and in which they could attach themselves to some important and overriding goal. To him, goes much of the credit for making the environment where the Whirlwind computer project could flourish.

In 1939, Perry Crawford did his MIT Master's thesis on digital computation, which meant developing a ten-stage ring counter to compute with decimal numbers, but never carrying it beyond some individual computing circuits. He is a philosophical, looking-into-the-future type of person. By the time we made contact with him, he was in the Special Devices Center of the Navy in Port Washington, Long Island.

Perry Crawford is the person who first called my attention to the possibility of digital computation. We were standing on the front steps of 77 Massachusetts Avenue one afternoon when we were still working on analog computers in the Servomechanisms Lab. He began to tell me about the work on the Harvard Mark I computer, and about the ENIAC computer which was then under construction. He was a very uninhibited, unbureaucratic type and would circulate freely right up to the Naval Chief of Operations even though he was a civilian far, far down in the organization. He moved through the Navy selling the idea that digital computers had a future as Combat Information Centers. He had several computer projects under his direction that he raised money for. He is also the person who gave Whirlwind and other projects their names. All of them were named after air movements: Hurricane, Zephyr, Typhoon and Whirlwind.

The other promoter to whom we owe a great deal is George Valley, a professor of physics. He was on a committee of the Air Force looking into air defense. In the later stages of our work that led into



The MITRE Corporation Archives

From left to right: Jay Forrester, Norman H. Taylor, John A. O'Brien, Charles L. Corderman, and Norman H. Daggert inspect the open, high voltage Arithmetic and Electrostatic Storage Racks characteristic of computer equipment in the early 1950s.

Lincoln Laboratory, he was the person who would call up generals in the middle of the night, tell them what they should do, and ask for support. He did all those things you read exposés about in books on the politics of technology, but which are necessary to keep the program coordination running smoothly.

The Organization
Sometimes you have people in an organization, each of them with an IQ of 130, and come out with an organization whose IQ is 70. What you get is the least common denominator rather than the best of the participants. I'm not sure how one creates the opposite environment, but there is great power in a tightly knit organization that has the capability of using the strengths of each person and compensating for the weaknesses of each.

Every person has strengths and weaknesses. You need a team in which there are such things as a vision of the future, a sensitivity to political matters, the capability of developing people, technical competence, the courage to transcend adversity, salesmanship, integrity, and putting long-range goals ahead of the short term. We had those characteristics well represented, scattered throughout our group. No person had all of them. For every person there would be, perhaps, a glaring hole in one of those dimensions. Yet, it was a group that understood each other well enough to use people in situations where their strengths prevailed rather than their weaknesses. Out of that came an organization that was able to be much more effective than most of those we see around us in technology and in most corporations at the present time. It is still an unsolved challenge to understand how that sort of spirit and unity can be created.

The Hostile World
Another thing that helped us, but that we resented, was the hostility towards innovation. There was little outside understanding of our subject, the objectives, or the methods for building pioneering computers. Funds were almost always inadequate. Reviews and investigations required us to defend our position and to face the weaknesses that other people were pointing out. We benefited from the distractions caused by the periodic reviews in which everything was questioned. Why were we using so much money? Why were we running late? Why were we designing the machine the way we were?

The matter of cost was one of the things that the outside world understood least. Whirlwind was being judged in the context of mathematical research, in which the salary of a professor and a research assistant was the standard by which projects were measured. We were spending way beyond that level, and

The Whirlwind console room in 1951 with the marginal checking and toggle-switch test control panels on the left. Stephen Dodd, sitting at an input device, is being watched by Jay Forrester and Bob Everett. Ramona Ferenz is seated at the prototype display to monitor the Cape Cod system, the prototype for SAGE.

The MITRE Corporation Archives

were seen as running a "gold-plated operation." Although the gold plating was occasionally excessive, in retrospect, I think there was reason for it.

An organization can't run with two contradictory standards. If you're going to have high performance and high quality in the things that matter, it is very difficult to have low quality and low performance in the things that, perhaps, don't matter. For example, at an early demonstration for important people, we didn't want them sticking their fingers into the high voltage in all those racks of Whirlwind. I asked somebody to get rope to put along the aisles so visitors wouldn't walk among the racks of vacuum tubes. A nice-looking white nylon rope was procured and installed. During the demonstration, I saw some of our critics fingering this beautiful rope and looking at one another knowingly as if to say, "That's what you would expect here." It may not have cost any more than hemp rope, but it reinforced that impression of an extravagant operation. Another example was the Cape Cod display scopes built into plywood cabinets faced with mahogany. Although our cabinetmaker made these quite inexpensively, people looking at those mahogany cabinets, were reinforced in thinking we were extravagant. Eventually we solved this problem by spending additional money and painting the cabinets gray.

Whirlwind's Technology

Making the decision to build Whirlwind I with a 16 binary digit register length was tremendously hard for us. The mathematicians were up in arms. They thought it was too short to be of any possible use. We defended it at that time on the basis that it was a demonstration of feasibility and we would build a 32 or a 36 bit computer when the right time came. Many of today's desktop computers are still 16 bits and only now moving to 32 bits. Selecting 16 bits was not a useless register length for computing, only a serious short term political problem.

The objectives of a computer at that time dominated the kind of high-speed internal memory to be chosen. Since Whirlwind was for demonstrating a very high speed computation for real-time applications, we chose electrostatic storage tubes rather than any of the more reliable kinds of serial memories. Each electrostatic storage tube with 1024 binary digits cost us about $1000 and had a one month lifetime. That meant that the upkeep on a storage tube, just its replacement, cost about $1 per binary digit per month. If you were to spend that on your two-megabyte personal computer, it would cost you $24 million per year just to maintain computer storage. The improvement has been perhaps a million-fold since that time in cost. That's about a factor of two every two years in the intervening 40 years. The high cost of storage tubes was the major incentive for inventing and perfecting coincident-current, random-access magnetic memory.

The economy necessary in programming was quite remarkable by today's standards. We demonstrated a military combat information center with one real bomber, one real fighter, and a radar set to generate data, with the computer receiving radar data by telephone line, analyzing it, throwing away the noise, averaging and smoothing and predicting the track, doing the same for the fighter, computing the intercept heading for the fighter, and then transmitting instructions to the autopilot automatically. If we today asked a programmer how much computer memory would be necessary for such a program, the programmer would probably guess a million bytes, minimum. The task was done on Whirlwind with 650 bytes of memory, not megabytes, just plain bytes. It was a time when the costs favored cutting programs to the minimum and using, if necessary, a lot of time, a lot of manpower, to reduce the programs.

Contributions of Whirlwind

In spite of the sense of extravagant expenditure, the entire Whirlwind project totaled about $4,500,000. That doesn't seem like much in today's computer world. Out of that came the first parallel, high-speed, clock-driven computer, magnetic core memory, cathode ray tube displays driven by a computer, an interactive light gun connecting a person to the computer, and many other innovations that are still important today.

We thought we had a good view of the future and we did for the succeeding 15 years, but I must say that our view of the future did falter if you were to extend it beyond that time. I gave a talk in the mid-1950s to a computer convention in which I pointed out that the cost of computation had been falling by a factor of two every two years from 1940 to 1956. I said, "Of course that can't go on for very much longer." But, of course it did, and is still going on.

Becoming a User

After 1956, I went more into the use of computers, using the ideas of feedback systems that Gordon Brown had originally pioneered and applying the methodologies and concepts to understanding the behavior of social systems. My present work is focused on the way in which the policies of a corporation produce its successes and failures and the way in which the policies embedded in the private and governmental sectors produce the behavior of the national economy.

My present work is focused on understanding the so-called economic long wave, the great rise and fall of economic activity with peaks every 45 to 60 years. This behavior has produced the great depressions of the 1830s, the 1890s, and the 1930s. We believe that the present economic cross-currents are the beginnings of another such major downturn. Working on behavior of social and economic systems is now especially timely. Just as the frontier of physical science opened up in the 1800s, the frontier of understanding our social systems now lies immediately ahead.

The Whirlwind project had shown that a reliable real-time computer could be constructed and that aircraft could be tracked and intercepted. Robert Everett is shown here on the Control Force Demonstrator in 1947.

The MITRE Corporation Archives

Discovering a "New World" of Computing

Robert R. Everett

Robert R. Everett is the former president of MITRE Corporation.

In 1947, the first work on how to use a general purpose digital computer for tracking aircraft was carried out at MIT. The project accounts for many firsts, because we were the first to ever have those problems. It was like Columbus and his crew discovering a new world. Jay was our Columbus and we discovered many strange and wonderful things. The computer business has grown to be like the original 13 colonies, with a vast, beckoning wilderness we have yet to explore.

The Whirlwind project proved that a real-time computer reliable enough to work could be built and that aircraft could be tracked and intercepted. But translating this experimental knowledge into an operational countrywide system was a major activity. Both technical and "organizational design" were needed.

The Birth of Lincoln Lab
The first step toward SAGE was the formation of Lincoln Laboratory by MIT, where we had a strong organization and excellent experimental verification and demonstrations. When the Air Force decided to go ahead with SAGE, Lincoln Lab was given the technical responsibility. An Air Force project office was set up in New York, supported by Western Electric. Bell Telephone Laboratories played a role in designing tests and criticizing what went on. IBM was chosen to build the central machine and Burroughs, to build some of the radar processors.

Lincoln was able to stay on top of SAGE because the group had done the planning backed by real experiments and demonstrations. Jake Jacobs created a systems office. Coordination meetings were held in which people from dozens of organizations, hundreds of people at a time, would get together. The group from Lincoln defined the problems, defined the options for solving those problems, and proposed decisions. We would present all this, and then everybody was faced with the option of either agreeing or taking some responsibility to do something else. They never wanted to do the work necessary for a new plan, so we always got our way.

The Role of IBM
The choice of IBM to build the central machine was made by Jay Forrester, with some help from Bob Wieser, Norm Taylor, and me. We visited the possible contractors and chose IBM because it was a very successful organization with strong sales and clean factories.

IBM had a series of machines in production and their own set of strongly held opinions about technology, standards, and organization. In the beginning we said, "This is our business. We know what to do. You are here to manufacture it." They said, "We built computers long before you." We even argued about how to make the frames. They made frames out of square steel. We said, "You don't want to do that, it might rust on the inside and it won't last more than a few thousand years. You ought to use L-shape aluminum like we do." Over time, I think we came to understand each other.

We had to learn about communicating with IBM. Next to my office we put in a Teletype machine to communicate with Poughkeepsie. I arrived in the morning and just stared, fascinated, at this machine. I finally figured out why. I had always looked at Teletype machines or typewriters connected to computers that said dull things like "23" or "fault" or "redo." This machine said, "Good morning, it's a lovely morning in Poughkeepsie."

One lesson, I recall, involved working on the core memory. We built some 32 by 32 bit planes, and we knew we needed bigger ones than those but weren't sure we could handle the nonselect noise. Someone suggested we divide it up into quadrants and put a sense amplifier on each quadrant, which meant four sense amplifiers. Coming back from Poughkeepsie one night, I realized it only took two. I thought,"Wouldn't it be funny if we all died in a car accident and SAGE had four sense amplifiers?" The next morning, I rushed into work ready to tell everybody about the two sense amplifiers. On my desk was a memo from Bill Papian's organization that said, "By the way, you only need two sense amplifiers." You had to be careful not to assume you were the only person who might think of something.

About 200 staff at Lincoln tried to stay on top of the project by turning the jobs over to other people as fast as possible. We didn't have the resources to do the design ourselves. Some of the troops at Lincoln didn't want to give up design because they felt strongly about what they were doing and weren't sure they trusted some "Johnny-come-lately" like IBM or Burroughs to build things properly. Fortunately, IBM wanted to take the job over as much, if not more, than we wanted to get rid of it.

It was a lot more difficult with the software. We had by then written the Cape Cod programs and had some feeling for the difficulty. We tried to get IBM interested in it and they said, "No, we sell equipment." So we tried AT&T who declined. Finally, Systems Development Corporation, spun off from the Rand Corporation, was created for this purpose.

The Air Force Partnership
The software turned out to take thousands of people. Jay set up a recruiting operation, and we hired hundreds of people off the street, unemployed mathematics teachers and so on. The Lincoln group hired hundreds of people for SDC.

Once the Air Force committed itself to building SAGE, they gave us complete support. For example, when we needed more computer time, we just bought it. The problem was that there weren't many computers around. Somebody had the bright idea that the machines in production in Kingston on the test floor were only being run two shifts. We needed time. IBM seemed willing. So we sent one of our fellows to IBM to negotiate it. He returned knowing it would cost a lot of money. Months later, Harris Fahnestock came into my office, white and shaking, with a bill from IBM for a million dollars. I said, "Now don't get flustered, Harris. I know we should have told you, but you would've had to agree with it anyway so why don't you just pay the bill and go away?" And he did. You can't imagine that happening today. We probably all would have gone to jail. The Air Force never complained. They understood. They knew the computer time was needed. They knew it would cost money, and they paid the bill.

The way the Combat Center program was written involved getting Walter Attridge and busloads of SDC programmers to Syracuse, where the center was being put together. They wrote the Combat Center program at the site. Although it was a year late with a big overrun, it worked and worked well.

When the first SAGE center went operational on July 1, 1958, MIT's commitment was over. That fall, MIT spun off their Lincoln Lab SAGE people to MITRE, which has been working on similar problems ever since.

About 20 centers were built. The ICBM put an end to the high priority that air defense has had, but the system ran for quite a while through the early 1980s. When the last centers went down a couple of years ago, they were still running well and reliably.

We had come to the end of the first part of the journey. I went to MITRE and Jay Forrester stayed at MIT. He was our Columbus, the first boss for many of us, the best boss for all of us, the creator of Whirlwind and SAGE, Jay Forrester.

From World War II Radar Systems to SAGE

C. Robert Wieser

C. Robert Wieser is Director of Engineering at Science Applications International Corporation in Newport Beach, CA.

The 1949 detonation of a Soviet nuclear bomb was way ahead of the United States' time schedule for that event. Over night, the requirements for the air defense system changed drastically. The US air defense, patterned on the system used in the Battle of Britain, resulted in a five percent attrition rate for incoming bombers, i.e., 95% of the planes got through. With nuclear weapons, this rate was unacceptable. A chill went through the air of the defense community. Something had to be done. George Valley, Professor of Physics at MIT, understood that the existing system could not just be incrementally improved.

Improving the Radar System

Three major areas of the air defense system were identified that needed changing. The ground control intercept station that got information from a single, large, long-range radar, was dependent on the maintenance of a single station and only worked for aircraft targets at medium or high altitudes. If planes flew at low altitudes, long-range detection was impossible because radar follows line of sight, not the earth's curvature.

The second problem was that all of the processing of the radar data was manual. The detection of aircraft was done by men looking at oscilloscopes. Tracking was done by a grease pencil to mark successive radar blips on the scope. Vectoring instructions were done by approximation, the observer figuring out the right course to get to the right place and assigning a target time. Unreliable high-frequency radio was used to track radar from one station to the next. The time delays in the transmission spoiled matching up the tracks.

Finally, jet aircraft were just being introduced, aggravating the deficiencies of the system. Since the aircraft went much faster, it was harder for an operator to do intercept computations in his head and tell a fighter pilot where to find the target.

George Valley began to search for radically different new ideas needed to solve these problems. The first idea was to substitute commercial telephone lines for high-frequency radio. That was a social innovation because the military believed that its communication system should be completely independent of the communication system used by civilians regardless of their effectiveness.

George found that Jack Harrington, head of research at the Air Force Cambridge Research Center, was working on ways to reduce the bandwidth of radar data so that the radar picture could be transmitted over voice telephone lines. An experimental apparatus was working, hooked up to an old microwave early warning (MEW) radar at the Bedford Airport, now Hanscom Field. George understood that such a system would allow the integration of data from many radars into one network. Hooking radars together, a region the size of New England could be covered, and by including short-range radars that filled in the low altitude gaps, the coverage could be extended down to about 500 feet above the ground. These were powerful new ideas made possible by new technologies.

The next thing that George discovered was the existence of the Whirlwind project. Jay Forrester and Bob Everett told him about their earlier work foreshadowing automatic control. George saw the possibility of automating the radar surveillance data for whole regions of the country.

Real-time Control

At this time, I was working on the first program attempting to apply the digital computer to real-time air traffic control. My bright group of graduate students, called "Boy's Town," included Dave Israel, Bob Walquist, Jack Arnow, Howard Kirschner, and others. The group was too inexperienced to be overawed by our task. Overnight we converted from air traffic control to air defense.

The group followed an empirical, experimental approach, taking on the real world as fast as we could. Remote radar data came into the Barta building where Whirlwind I was under contruction. At the time, Whirlwind had no electrostatic storage. Random access memory was five flip-flop registers and 32 toggle switch registers that could be read by the machine. We got the radar data inserted into the machine and displayed. After this happened we came face-to-face with some problems.

First, radars see a lot of things that aren't airplanes. That tends to load up the transmission system. Second, telephone lines were not perfected for data transrmission. For example, dialing clicks came in as false targets. The progress in fixing those problems was very rapid because we didn't have to plead for permission. We just got the job done.

The next big event was when Whirlwind got one bank of electrostatic storage tubes with 256 registers. That was when we began to learn about the romance of computer programming. The word "software" had not been invented at the time. All of the programming was done in machine language because there wasn't anything else. With 256 registers, we extended the capability to simultaneously track-while-scanning ten airplanes. Alternatively, two airplanes could be tracked with vectoring instructions to indicate collision courses.

Preparations began to try the real thing, an interception of two airplanes. We made friends with people in the Air National Guard and persuaded one pilot,

From left to right: C. R. Wieser, Bob Everett, and Jay Forrester gather at Forrester's retirement party in June, 1956.

The MITRE Corporation Archives

The MIT Museum

On January 16, 1956, the SAGE system of continental air defense was introduced to the press at Lincoln Laboratory. From left to right: Edward L. Cochrane, Vice President for Industrial and Governmental Relations; George E. Valley, Jr., Associate Director of Lincoln Lab; Major General Raymond Maude; Colonel D.E. Newton, Jr., Commander and Vice - Commander, respectively, of Air Force Cambridge Research Center.

who was flying a small twin-engine Beechcraft to be the target. Another pilot with a T-6, single piston pilot trainer, was asked to be the interceptor. To run the system, we had to communicate with the interceptor pilot and pass the computed instructions to him by voice telephone. That was Howard Kirschner's job. With no digital displays on the computer, Howard, with the wonderful wiring in his brain, could read the indicator lights off the registers, convert them to decimal, and send instructions to the pilot. In April, 1951, we ran the first successful interception. The T-6 came within a thousand feet of the C-45. The impact of this accomplishment was so powerful that, three days later, the decision was made to build the Cape Cod System.

The Cape Cod System

This functional prototype of the air defense system was to be based on digital computation and remote transmission of radar data. Since it would be inappropriate to copy the hardware, Cape Cod was a functional prototype to test all the ideas for replication. Furthermore, it was a demonstration to ourselves, our friends, our skeptics, and our adversaries that this was more than intellectual nonsense.

The specifications for the Cape Cod System included doing air surveillance, automatically generating tracks, following the tracks, and generating vectoring instructions to interceptors. A group of Air Force enlisted men and officers were to carry out the project in two and a half years, from the spring of 1951 to the fall of 1953.

The system was completed on time with full functionality. Many engineering difficulties were encountered in building the pieces and putting them together because it was a new concept, made from new equipment, and new technology. Toward the end of the test period, the first core memory storage was installed on Whirlwind. The system went from 256 registers to several thousand, and the reliability was vastly improved.

In 1954, the system was expanded by increasing the radar network. The radars were located in Brunswick, Maine, Truro, Massachusetts, and Montauk Point, New York, and the interceptors included aircraft at Hanscom Field, bases on Long Island, and south of New York. Live exercises were run diverting Strategic Air Commmand bombers that were used as targets. Everything worked. A new development was the automatic ground-air data link so that Howard Kirschner did not have to read all those lights on the computer. It also foreshadowed the coming of missiles like the Bomarc which had no pilot.

The first ground-air data link experiments were interesting. Doc Draper of the Instrumentation Lab had a light test facility out at one end of Hanscom Field. Chip Collins, his chief pilot, discovered that one of the aircraft, a World War II B-26, Martin Marauder, had an autopilot that could take digital input. The radio frequencies were set up to send vectoring instructions directly to the autopilot. On the test we head Chip Collins say, "Let George do it," which meant switch to autopilot. A little while later, when we traced it on the scopes, he said, "Tallyho," as he sighted the target. Someone dubbed that "The Immaculate Interception."

With today's DOD guildines, no such experiment could be carried out. In two and a half years, we wouldn't have been able to agree on an operational requirement, get an acquisition plan together, set up the RFP, the Source Evaluation Board, the Source Evaluation Advisory Council, the Source Evaluation Executive, and all the other groups, and still negotiate a DOD contract. At that time, we just did the job that was expected of us.

From Cape Cod to SAGE

The decision to build the SAGE System did not fall out of building and demonstrating the Cape Cod System. Competing schemes existed and there was a lot of missionary work to do to get our ideas accepted.

The burden of selling "electro-theology" fell on Jay Forrester and George Valley. Jay commissioned us to write Technical Note 20, a master plan for the development and installation of the "Lincoln Transition System." (The name "SAGE" had not yet been invented. George Valley brought in General Gordon Saville of the Air Force. He was about five and a half feet tall, feisty, had a strong voice and understood his own opinions. After he read TN 20, he came back, went to the head of the table, threw it down and said, "You're the worst damn salesmen I ever met. This report is stinko profundo. What you ought to do is start all over again, and maybe if you worked real hard, you might work your way up to medium sorry." We listened to him carefully and began to understand that it's one thing to explain something that lies outside a person's experience and yet another thing to explain something that lies outside his imagination. The latter is much harder, but it has to be done.

A Once-in-a-Life Experience

Sometimes I ask myself why this was such an interesting experience, the like of which I haven't had since. There are a couple of reasons. We were saved from the day-to-day frustrations of butting heads with the bureaucracy. We could invest all of our engineering skills in the task we had to do.

An important reason is that we had the engineer's dream: a nationally important problem that was interesting and difficult but not impossible to solve. These are the best kind. We were in a day-to-day contest with Mother Nature. The odds were bad, but we always had a chance to win, and we won all the battles that led up to SAGE. We also won the cause for digital computation. If there's anyone who thinks we didn't win, just go to Radio Shack and try to buy an analog computer.

Calendar *Spring 1988*

March 6 Sunday 3 PM
Beyond Nature: Computer Graphic Simulations of Life
Peter Oppenheimer of The Computer Graphics Laboratory, New York Institute of Technology, will introduce and discuss his computer-generated experiments that create surreal forms of life captured on video.

March 13 Sunday 3 PM
Intelligent Machines of Today and Tomorrow
Raymond Kurzweil, inventor of a reading machine for the blind and other computer-based devices, will talk about artificial intelligence and introduce the special film the Kurzweil Foundation produced, "The Age of Intelligent Machines."

March 19 and 20 Sat. and Sunday 12:30 and 3:00 PM
SIGGRAPH Electronic Theatre 1987 Part 1
Four showings over the weekend of the edited tapes from SIGGRAPH 1987 with commentary by an authority. SIGGRAPH sponsors the annual "Academy Awards" for the international computer graphics community.

March 26 and 27 Sat. and Sunday 12:30 and 3:00 PM
SIGGRAPH Electronic Theatre 1987 Part 2
Four showings over the weekend of the edited tapes from SIGGRAPH 1987 with commentary by an authority.

April 1-30 10 AM-5 PM
Awesome Adventures
As part of Boston's Museum Goers Month, The Computer Museum presents a month-long interactive exhibit of such "awesome adventures" as maze exploration, flight simulation and 3-D animation.

Become a Computer Museum Member

☐ **YES!** I want to become a Member of The Computer Museum.

☐ **$20 Student** ☐ **$ 45 Family** ☐ **$250 Sponsor**
☐ **$30 Individual** ☐ **$100 Donor** ☐ **$500 Patron**

Members receive free admission for one full year, invitations to exhibit previews, advance notice of exhibitions and lectures, invitations to members-only events, a subscription to our quarterly magazine, The Computer Museum Report, and a 10% discount in the Museum Store.

Enclosed is my check made payable to **The Computer Museum** for $__________

or charge to: ☐ VISA ☐ MasterCard ☐ American Express

Card # __________ Expiration Date__________

Signature __________

Name __________

Address __________

City __________ State __________ Zip __________

Mail to: Membership Coordinator at The Computer Museum, 300 Congress Street, Boston, MA 02210.

The Computer Museum

The Computer Museum is a non-profit 501(c)3 foundation that chronicles the evolution of information processing through exhibitions, archives, publications, research and programs.

Museum Hours:

Summer: Open daily 10 – 5, Friday 10 - 9. *Winter:* Open Tuesday – Sunday 10 - 5, Friday 10 - 9. Open Mondays during Boston school vacation weeks, 10 - 5. Closed Thanksgiving, Christmas, and New Years Day. Hours are subject to change.

The End Bit

00000000000000001

Computer Space was the first coin-operated video game. It was developed by Nolan Bushnell in 1971. While "Computer Space" was a modest failure and only sold about 2,000 units, Bushnell's next game, "Pong," was a tremendous hit that ushered in the era of video arcades and home game machines.

Produced while Bushnell was with Nutting Associates, the "Computer Space" flyer describes the game's "BEAUTIFUL SPACE-AGE CABINET" and "the reality of controlling your own rocket ship in gravity-free outer space." In fact, "Computer Space" was very near "Spacewar!" in terms of the action that it offered. The game's original instructions conclude with the offer, "If I can help answer any question concerning this machine, please do not hesitate to call me personally. Nolan K. Bushnell, Chief Engineer, Nutting Associates, Inc." The following year, 1972, Bushnell started a game company of his own — Atari.

This photo is from the game's advertising and instruction brochure printed in 1971. It was donated to The Computer Museum by Alan Frisbie.

Address Correction Requested

Non-Profit
Organization
U.S. Postage
PAID
PERMIT NO. 110
FITCHBURG, MA
01420

The Computer Museum 300 Congress Street Boston, Massachusetts 02210