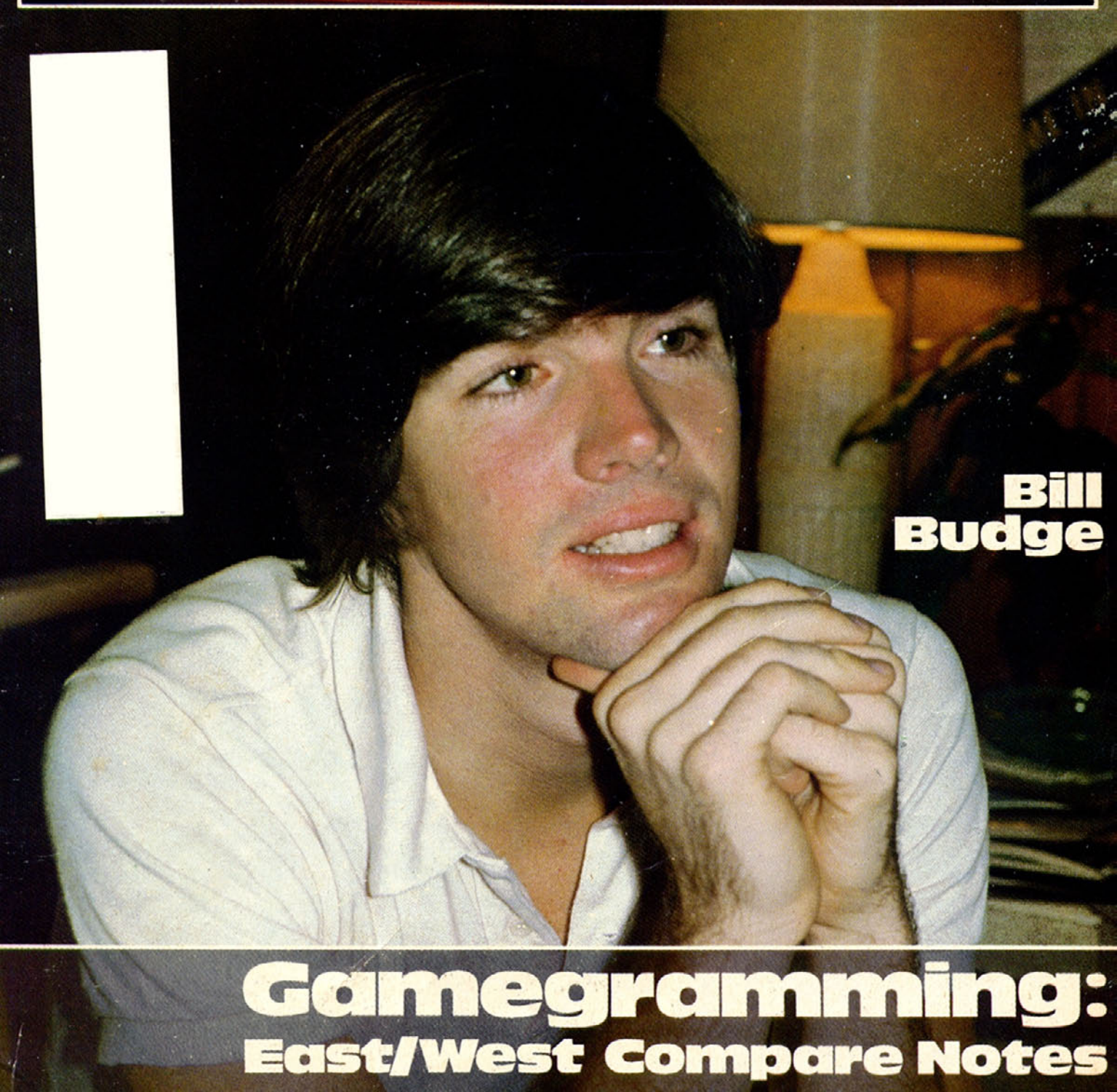
SOFTLINE

VOL. 1 NO. 2 NOVEMBER 1981 $1.50



Bill Budge

Gamegramming: East/West Compare Notes

They have landed and are taking over the city. Steadily they are making their way across the city, destroying everything in their paths. The town has been evacuated and your regiment has retreated leaving you, alone in the city, at the mercy of the aliens.

The aliens have you surrounded, and laser shots fly from all directions. Your movements are confined but you haven't given up. If you're going to live, you'll have to concentrate on where the shots are coming from and where you're going because if you don't, you'll get caught in the **CROSS FIRE**.

CROSS FIRE is a unique new game by JAY SULLIVAN featuring HI-RES graphics and sound, smooth quick animation, and some of the best arcade challenge available anywhere. **CROSS FIRE** runs on any 48K APPLE II/II PLUS DOS 3.2 or 3.3 and is available now for $29.95 on disk from your local computer store or you may order directly from.....

ON-LINE systems

36575 Mudge Ranch Road · Coarsegold, CA 93614 · 209-683-6858

ADD $1.00 FOR DIRECT ORDERS

VISA, MASTER CHARGE, C.O.D. or CHECKS ACCEPTED

SOFTLINE

Editor
Margot Comstock Tommervik

Managing Editor
Jean Varven

Art Director
Kurt A. Wahlner

Softline. Volume 1, Number 2. Copyright © 1981 by Softalk Publishing Inc. All rights reserved. *Softline* is published on the fifteenth day of every other month by Softalk Publishing Inc., 11021 Magnolia Boulevard, North Hollywood, CA 91601 as a continuation of the *On-Line Letter*. Telephone (213) 980-5208. Second class controlled circulation pending at North Hollywood, CA.

Postmaster: Send address changes to Softline, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

Composition by Graphic Typesetting Service, Los Angeles, California. Printing by California Offset Printers, Glendale, California.

Apple is a registered trademark of Apple Computer Inc., Cupertino, California.

Atari is a registered trademark of Atari Inc., Sunnyvale, CA.

Subscriptions: *Softline* subscriptions are free to owners of personal computers. In your subscription request, include your name and address, the brand, model, and serial number of the computer you own, and a brief description of your peripherals. If you do not own a computer, you may subscribe to *Softline* at a rate of $6.00 per year (six issues).

Moving? Send new address and old to Softline Circulation, 11021 Magnolia Boulevard, North Hollywood, CA 91601; telephone, (213) 980-5208.

Softline Staff

Administration

Publisher	Al Tommervik
Associate Publisher	Ken Williams
General Manager	Mary Sue Rennells

Editorial

Editor	Margot Comstock Tommervik
Managing Editor	Jean Varven
Editorial Associate	Craig Stinson
Contributing Editors	Ken Williams Sherwin Steffin Ken Rose
Guest Reviewer	Derek Bang
Assistant Editors	Melissa Milich David Hunter
Copy Editor	Andrew Christie
Special Projects	Greg Voss

Art

Art Director	Kurt Wahlner
Assistant	Donna Siebert

Advertising

Sales and Coordination	David Hunter

Circulation

Associates	Ron Rennells Robert Mann

Advertisers

Directline

A Developing Curiosity

Finally, a computer magazine for the gamesman! My wife and I have just received your first issue and we have devoured it. We both do some programming and use our computer (Apple II) for business use. However, we are both avid adventurers and love our games.

How about a series of articles on Adventure Development Language, mentioned in your preview of *Time Zone*? It should be fascinating to see how my favorite programs are written.

Carl T. Porchey, M.D.
Winston-Salem, NC

Wolfenstein's a Winner

I have a suggestion for your Softline Salutes section. I think Muse Software ought to be saluted for coming out with *Castle Wolfenstein*. This is the most fantastic and interesting game ever published.

Michael J. Pinsel
Northbrook, IL

Patchwork

I have just finished reading my first issue of *Softline*. Congratulations on a job well done.

Perhaps you, On-Line Systems, or one of your readers can advise me on a way to patch *SuperScribe II* to allow loading of "runoff" to the 16K Ramcard.

Most of my text files are letters, so I don't think I need the Ramcard for text file storage.

SuperScribe II is the best word processor I have ever found. However, having to wait for individual programs to load is an irritating drawback to an otherwise powerful piece of software.

David J. Hodgson
Hoffman Estates, IL

Games Are Weighted on Softliners' Scales

Despite "some manufacturers lofty claims," and in spite of some magazine editors' letters, I'd like to present some numbers that contrast to your claim that "the primary use to which personal computers have been put so far is for playing games." The most recent *Byte* readers survey by McGraw-Hill Research shows that 49 percent of those responding say they use their computer for both professional/business and hobby, 12 percent for professional/business, 24 percent for hobby only, and 19 percent for other nonhobby uses.

Personal Computing estimates that the business applications share of market has increased from 30 percent in 1980 to 50 percent in 1982, while the home/entertainment use of personal computers has dropped from 45 percent in 1980 to 25 percent in 1982.

The *InfoWorld* Subscriber Survey found that its readers were overwhelmingly interested in professional applications. Those with both professional and personal interests totalled 77.5 percent, with professional interests alone of 16.3 percent and personal interest of 6.2 percent.

Other examples could be quoted, but I think you get the idea. If you sat down with any of the personal computer manufacturers today, I believe they would tell you that the

vast majority of people buying their computers are buying them primarily for business and professional applications. There is no denying that they also throw in a couple of games for their own enjoyment and that of their family. The manufacturers' input from the marketplace and feedback to you would hardly be considered lofty claims. If there are any lofty claims being made, they are coming from users.

In spite of the fact that the business/professional user dominates the marketplace now and will through 1982, I certainly send best wishes on your new magazine and look forward to reading those reviews that help me buy games for my entertainment and that of my sons!

Bill Langenes, Advertising Manager
Personal Software
Sunnyvale, CA

Business Apple Spends Half Its Time Playing

I both appreciated and enjoyed my first issue of *Softline*. While I use my computer for home finances and in connection with my work as a systems engineering manager, I spend at least half of my Apple time playing games. I was pleased to see that there was to be a publication that concentrated on games.

I would like to see your game reviews include tips on obtaining high scores and, if possible, the scoring algorithm used by the program. I have not been able to figure out the one used by *ABM*.

Michael B. Goldsteen
Brookeville, MD

Kudos, Comments, and Questions on Graphics

I recently received the September 1981 issue of your magazine, and was pleasantly surprised and delighted.

I read your instructive article "Apple II Graphics," by Ken Williams and I applaud your efforts. I want to congratulate Ken Williams on the basic clarity of some of the hardest material for the novice to understand from Apple manuals and other publications.

I bought my Apple II for pleasure and lean toward game software. Most publishers handle game-oriented software as a sideline and therefore do not come close to the depth you give it.

Dean Peters
Lima, OH

Thank you for including me on the mailing list for *Softline*. I enjoyed the entire magazine very much, but I especially devoured the article by Ken Williams on Apple II Graphics. As a newcomer to computers (I have owned my Apple II Plus for seven months), it is this type of article that I usually search for in a computer magazine, and usually don't find. The author has succeeded in clarifying the concepts of addresses and memory locations in a few pages of instruction, where many chapters in many books had failed. I am awaiting with great anticipation the remaining articles in this series and hope it will be a very long series.

I was also very interested in the article on educational software, and if that series is as informative as the initial installment promises, it will also be eagerly awaited and appreciated.

A request: Would it be possible for you to include in a future issue a bibliography of books and magazine articles by these two authors? I am a novice and am not yet familiar with computer literature, but I am sure these are certainly not the first items by these very lucid and readable teachers.

Thomas E. Militello, M.D.
Rancho Palos Verdes, CA

Your magazine came as a pleasant surprise. I am especially looking forward to Ken Williams's subsequent tutorial articles on Apple hi-res graphics, since I have some ideas for games I would like to program and sell.

Some questions I would really like to see answered are:

1. How do you utilize shape tables from assembly language? I can do it in Basic, but assembly proves to be confusing.

2. According to the Apple manuals, a defined shape can be plotted in any of the seven colors (including black and white). The whole shape is plotted in that color. So how do you make multicolored shapes, such as the oranges in Broderbund's *Alien Rain*, the wolves and the ship in *Space Eggs*, and so on? Are they two different shapes plotted next to each other, or one shape with two colors?

3. I have yet to figure out how to make an explosion or collision check. For example, the bullet hits the plane. Both are defined shapes, so how does one tell when the bullet has hit the plane? Also, how do you distinguish whether a bullet hit it, or maybe just falling debris or another paratrooper hit it (in which case it wouldn't explode?) I've tried the PEEK (810) mentioned briefly in the *Programmer's Aid #1* manual (Apple Computer, Inc.), but that doesn't seem to work with Applesoft.

4. I haven't been able to figure out how to, say, make something explode or have a sound effect (music, for instance) happen without stopping all the other action (including your controls) completely.

Mark Seybold
West Covina, CA

It's through free, open exchange of ideas that a craft can become an art: meaningful, significant, and influential. The resources of the computer give gaming this potential—the ability to affect the way people approach life and view the world, if only through increasing their senses of accomplishment and pride in themselves. Hardly an unworthy achievement and definitely the province of art.

Your contributions to the enhancement of computer gaming as an art and simply as a pleasure are very welcome in Directline; your comments and answers to other contributors' letters and questions will be much appreciated.

Through Directline, you'll have the opportunity to communicate with people of like interests directly; people who are pursuing the same goals you are, whether as hobbyist or professional.

Several established gamemakers, such as Doug Carlston and Ken Williams, look forward to tackling some of your questions and problems too. But don't let their names and accomplishments cow you in questioning, commenting on, or criticizing their contributions just as you might those of another reader.

Out of just such open exchange are great ideas formed, progress conceived, and art born.

Send your letters to Directline, Softline, 11021 Magnolia Boulevard, North Hollywood, CA 91601. SL

Preview:
The Arcade Machine

A lot of creative folks who love home-arcade games are feeling stranded. They have ideas for games, inspirations even, but their ideas will never see the light of day. Why? It's a matter of programming.

Many people with ideas for games have no working knowledge of programming. Without such knowledge, they have no hope of implementing their ideas, no way of bringing their own games to life.

Until now. Broderbund's *Arcade Machine*, a program generator for the Apple, can help. It allows gamers with no programming knowledge to create and animate their own fast-action home-arcade games.

According to Doug Carlston, program coauthor, the idea for the program generator came out of a conversation he had one day by chance with Broderbund's newest on-staff programmer, Chris Jochumson. Impressed with Jochumson's skill and creativity at putting together simple programming routines that have a lot of flexibility, Carlston remarked, half-serious, "Wouldn't it be neat if we could come up with a program that could create other programs?" Which is, wishful thinking aside, exactly what they did.

Unwrapping the Package. *The Arcade Machine* package will consist of a write-protected master program disk and separate documentation written by Doug Carlston. Using the master program disk, you'll be able to convert ordinary blank disks into game disks on which to store finished games or games you're in the process of creating; or file disks, on which your own library of shapes and special effects can be stored.

For a demonstration of the kind of game the program generator will enable you to create, default through all the program values (hit return at each point, rather than choosing one of the options). You'll get a sample single-player shoot-'em-up. With the aid of the new program generator, you'll soon be creating your own one or two player home arcade games.

When you're ready to begin in earnest, start again. The screen prompts will make it easy for you to know what will happen if you choose a particular option or hit a given key, and most of the commands you need to know will be displayed on a portion of the screen so you'll be able to use them without having to refer to separate documentation.

A series of paths or courses your game could follow will be presented for you to choose from. Then if you want to make things a little more unpredictable, choose the gravity option for your game paddles. The game paddle controls will now give you a gradual speed-up/slow-down effect, rather than being directly responsive to your input.

The Shape of Things To Come. Next, you'll be asked to decide on the shapes of your players and the shapes of the creatures that inhabit your game. If you don't fancy any of the shapes shown on the program disk, you can elect to create and animate your own. The shapes you draw or choose can be stored on your file disk, enabling you to proceed with the creation of the remainder of your program.

For animation purposes, you'll be asked to draw or position your character shapes seven times from seven different angles. At any point in the drawing process, you'll be able to review your program or selected segments of it to see how things are shaping up so far. Returning to the drawing board, you can correct or modify your drawings as necessary so they'll look just the way you want them to in your finished game. You can refine your program until everything matches your original mental picture.

Your working area is a large square portion of the video screen. To create shapes, you move the cursor around the screen in much the way you do in programs like *EZ Draw*. You can choose the colors you'd like your shapes to be; all six colors on the hi-res screen are available.

You'll even have the opportunity to choose the kinds of missiles and explosion sequences you want in your game. Although you won't be able to create your own sound effects, various suitable sound effects will be provided.

When you complete a game and want to share it with friends, you'll be invited to create a title page for it that lists the name of the game and its author. The game disk will boot automatically when you load it, so this title page will be the first thing your friends see when they sit down to play your game.

In a program such as this one, memory is at a premium. Allowing for the animation possibilities takes up a lot of memory space. This means it's not possible to put your game rules on the disk along with your game, but you can get around this limitation by preparing hard copy of your rules.

You won't be able to use *The Arcade Machine* to create games in which there's a maze that changes or interacts with the rest of your program. A master program that takes all the possible variations into account would be exceedingly complex and would outstrip the memory capability of the Apple. It is possible, though, to make a game that contains a maze, using the two hi-res pages in combination with one another. Through a technique known as *oring*, your maze is placed underneath your game. The maze won't be affected by what's happening on top of it; instead, it will serve as an unchanging backdrop against which your game can be played.

Effects and Expectations. We asked Doug Carlston what effect he thinks *The Arcade Machine* will have on commercial products written by professional programmers and marketed by companies like Broderbund, and on gaming in general. He told us he does not expect games created on *The Arcade Machine* to compete with or affect the market for well-done home-arcade games by professional programmers.

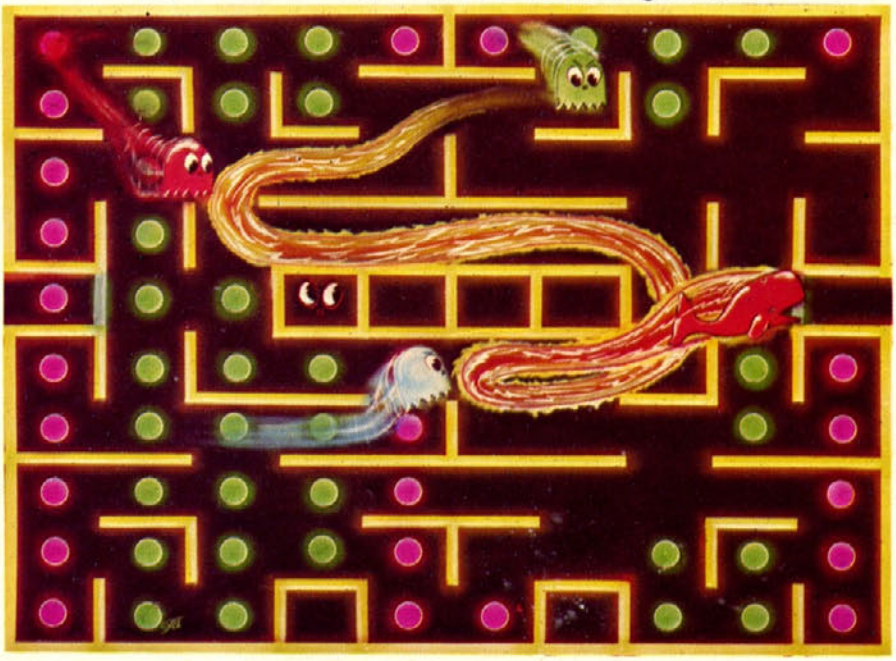
What he does expect is that people who love to play home-arcade games but lack programming knowledge will enjoy making games of their own. Who knows? They may become so curious about the art of gamemaking that they'll decide to learn more about game and graphics programming.

Carlston feels that it's time professional programmers move on from the home-arcade shoot-'em-up and begin to explore new challenges and frontiers. With the introduction of *The Arcade Machine*, Broderbund has offered new vistas to gamers as well. SL

Habit forming

SNACK ATTACK

Betcha can't play just one game!



Try one game and you're caught in its irresistable web of fun. More habit forming than peanuts. More fun than gorging on hot fudge sundaes. More exciting than anything like it.

SNACK ATTACK won't just a-maze you, it'll 3-maze you! Win the first level, and up pops a new, faster version, with a more intricate maze. And for more challenge there are doors you can enter and **they** can't. Doors they can dash thru and **you** can't!

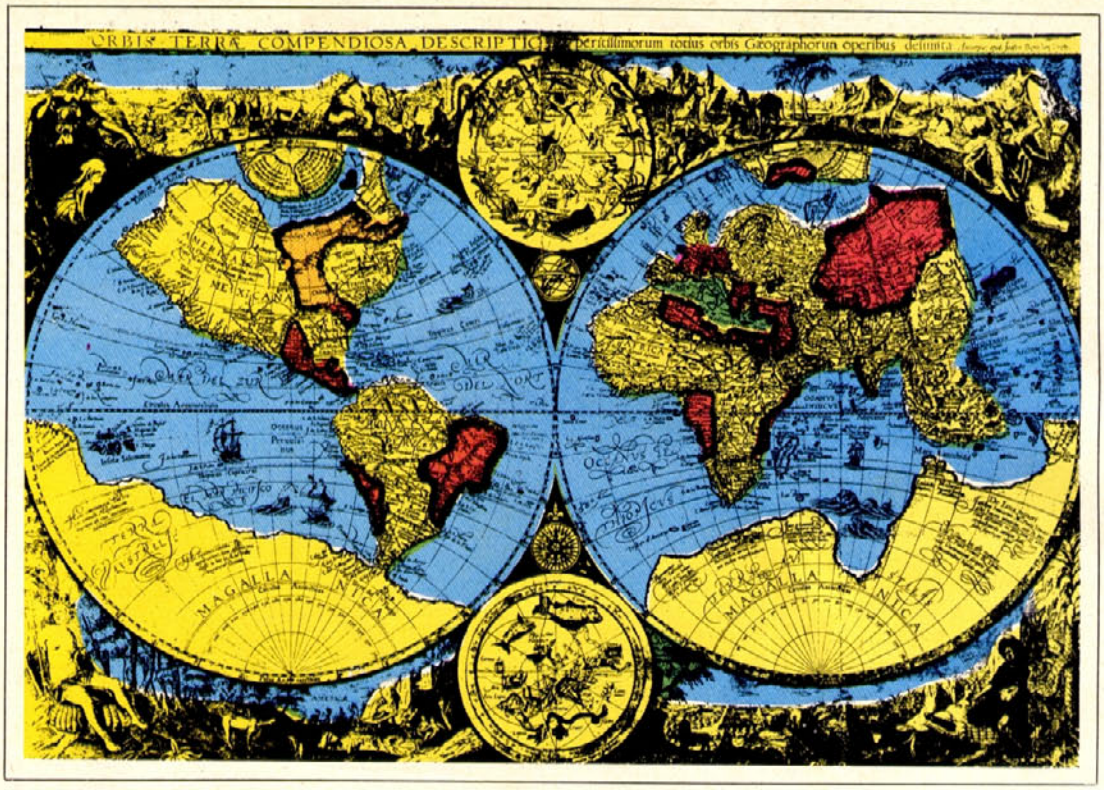
SNACK ATTACK . . . by Dan Illowsky, the game that defies anyone to stop after just one game!

$29.95, for Apple II*
At computer stores, or from:

DATAMOST
19273 Kenya St.
Northridge, CA 91326
(213) 366-7160

VISA/MASTERCHARGE accepted. $1.00 shipping/handling charge. (Calif. residents add 6% tax)

*Apple II is a trademark of Apple Computer, Inc.



Origins of the Adventure

by Ken Rose

I'd finally made it to the interior of the tomb. How fortunate that I'd stumbled across the secret passage. And now, as I stood in front of the low, strangely engraved wooden trestle table, my goal was within reach.

My flickering lamp picked out highlights on the beautifully wrought chalice, worth more than a king's ransom (and more than one adventurer's life). The chalice that could literally work miracles was mine. It was almost too easy.

Wait!

Movement from the corner. Is it a shadow, an illusion? No. Shadows don't move. Shadows don't have coal-red eyes, blazing from beneath a cowl black as the grave. Shadows don't brandish a silver blade and moan softly as they move toward you.

I tore my own blade from its scabbard and parried the apparition's thrust. Its supernatural strength wrenched the sword from my hand, and a mocking laugh reached my ears as the ghastly creature raised its skeletal arm for the final stroke. I had only one hope—one desperate way to save myself. And I took it.

I pressed the reset key.

Adventures. What are they? From whence did they come and where are they headed? What is this affinity between the adventure player and the computer user? Herein we'll explore the world of adventuring and a bit of the world beyond.

An adventure is, after all is said and done, a puzzle or series of puzzles. The successful outcome of an adventure is fixed and does not change from playing to playing. To win in the end, you must have fulfilled the quest, collected the required treasures, or both.

The Square Root of Adventure. The history of puzzle games goes back to the ancient Greeks and Romans. Such games were created and have endured primarily because they appeal to mankind's sense of curiosity and mystery.

Few details of the games survive, and little was known about them in the Dark Ages. A resurgence of interest in these early games took place in the Middle Ages, stimulated in part by the invention of the printing press, which meant that information was more readily available to a greater number of people; in part by the enthusiasm generated by writers of arithmetic texts; and in part by the rivalry and disputations between early algebraists and scholars.

Leonardo of Pisa (Leonardo Fibonacci), born in 1170, was to become the most distinguished mathematician of the middle ages. His bestseller, the *Liber Abaci (Book of the Abacus)*, introduced the use of the Hindu-Arabic numerals during this time. Before the appearance of his book, only a few European intellectuals knew about them.

Fibonacci introduced the numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9, as well as the concept of place value. He is especially remembered for what has come to be known as the Fibonacci sequence:

A pair of rabbits is in a place that's surrounded on all sides by a wall. How many pairs of rabbits can be produced from that pair in a year if it is supposed that every month

each pair begets a new pair which becomes productive from the second month on? (The answers to this and to the other examples appear at the end of this article.)

A Duel without Swords. An example of a rivalry between mathematicians is the one between Niccolo Fontano Tartaglia and Geronimo Cardano. Tartaglia, who had found a way to solve cubic equations, made the mistake of confiding his method to Cardano, who promptly proceeded to publish the discovery as his own. Incensed, Tartaglia challenged Cardano to an algebraic duel, in which each was to propose thirty-one problems to be solved by the other. Cardano accepted the challenge, but sent one of his pupils to solve Tartaglia's problems. The pupil failed. Tartaglia, who solved Cardano's problems himself, succeeded and won.

In medieval times, puzzles were generally of two kinds: those that involved computation, and those that involved the manipulation of objects.

The first type of puzzle resulted in a wide variety of arithmetic/algebraic and geometric/topological puzzles. Some examples of arithmetic/algebraic amusements are number patterns, cryptographic arithmetic, paradoxes, pythagorean triples, and magic squares. Geometric and topological puzzles include optical illusions, tangrams, mazes, and map coloring problems.

The second type of puzzle requires little or no mathematical skill but does require general intelligence and creative thinking. These produced the logical inference puzzle, which calls for deductive inferences based chiefly on logical relationships, and is the primary precursor of today's adventure games.

To Get to the Other Side. The classic logical inference puzzle relies only partially on the various techniques of logic. In the final analysis, success depends in great measure on that elusive quality we call ingenuity.

An example of a logical inference puzzle is the one known as "Difficult Crossing."

In the ninth century, a medieval mathematician posed the following:

A man with a boat must get across a stream with a wolf, a goat, and a bundle of cabbages. His boat is so small that he can only transport one of the three at a time. He cannot leave the wolf alone with the goat, nor the goat alone with the cabbages.

What is the minimum number of crossings he must make to get everything across stream successfully?

The next ingredient that became part of the adventure was the riddle. Part of most folklore from ancient times, riddles often rely upon a deliberately misleading or ambiguous statement, a play on words, or some other device intended to catch the unwary.

Western scholars generally recognize two main kinds of riddles: the descriptive riddle and the shrewd or witty question. Popular in England during the tenth century and during the Renaissance, descriptive riddles deal with appearance, not function. An English favorite was the Greek folktale, in which the Sphinx asks:

"What has one voice and walks on four legs in the morning, two at noon, and three in the evening?"

Shrewd or witty questions are also of ancient origin. An example from classical Greece is:

What is the strongest of all things?

A more recent example is the sort that calls for interpretation:

"ICUR YY 4 me."

From Poe to Parker Brothers. The detective story introduced the public to the step-by-step investigation and solution of a crime. In April of 1841, with the publication of Edgar Allen Poe's short story "The Murders in the Rue Morgue," the modern detective story was born.

Clue, developed by Parker Brothers in the early fifties, brought the first puzzle-solving board game to the public. This outgrowth of the detective story allows interaction among several people in solving a whodunit. Each player takes the part of a character, one of whom may be the murderer. Randomly chosen cards representing the murderer, weapon, and room of the crime are put in a packet on the game board. Through deduction, the players must determine which cards are in the packet. The first correct accusation—including by whom, where, and with what, the murder was committed—wins the game.

Our brief tour through history would not be complete without a mention of Robert E. Howard's *Conan* stories, written in the 1920s. With Conan, Howard conceived one of the greatest larger-than-life characters in American fiction. Conan came from Cimmeria, a prehistoric land of cold and darkness. He set out to make his way and fortune in a world where strength and sorcery rule. Eventually, he rules (and loses) a kingdom.

Conan was not entirely original with Howard. His background lies in the myths of many lands. He also shares ground with Lord Dunsany and J. R. R. Tolkien. But he was the earliest true example of the sword and sorcery genre that's so popular today and so much a part of the adventure game.

Making It All Compute. So what do we have? The puzzle game, the riddle, the whodunit, and swords and sorcery. The only element that's missing is the computer.

In the mid-seventies, Will Crowther and Don Woods shook up these elements and produced a game called *Adventure*. The game, which ran on many mainframes, acquired a cult status. *Adventure* may have made many department managers unhappy, but it delighted programmers and systems designers, who spent surreptitious hours playing the game. As time and computers marched on, the original *Adventure* shrank to a size that could fit into the microcomputer of today.

The original *Adventure* was played in text on the screen. The next development in the adventure genre—many years later—was the addition of graphics. Two early examples of the use of graphics in an adventure game are *Beneath Apple Manor* and *Temple of Apshai*. Produced when Dungeons and Dragons was gaining momentum, these games introduced the concept of conflict into the adventure game on the graphics screen.

Beneath Apple Manor in lo-res graphics and *Temple of Apshai* in hi-res illustrate their characters and show their movement on the screen. Both games involve negotiating

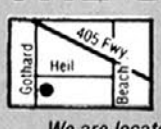
THE SOFTWARE STORE

We have the best prices in the West . . . stop in and see for yourself. We carry a full selection of Business and Utility Programs.

PROGRAM	LIST PRICE	OUR PRICE
A2-FS1 FLIGHT SIMULATOR	35.50	**29.95**
ABM	24.95	**19.95**
ADVENTURE #1 (CLASSIC)	20.95	**16.95**
ADVENTURE #1,2,3	39.95	**33.95**
ADVENTURE #4,5,6	39.95	**33.95**
ADVENTURE #7,8,9	39.95	**33.95**
ADVENTURE (MICROSOFT)	29.95	**23.95**
ADVENTURE 10, 11 & 12	39.95	**33.95**
ADVENTURE IN TIME	29.95	**23.95**
AKALABETH	34.95	**27.95**
ALGEBRA I	39.95	**33.95**
ALIEN RAIN	24.95	**19.95**
ALIEN TYPHOON	24.95	**19.95**
APPLE PANIC	29.95	**23.95**
APPLE-OIDS	29.95	**23.95**
ASTEROID FIELD	24.95	**19.95**
AUTOBAHN	29.95	**23.95**
BATTLE OF SHILOH	39.95	**33.95**
BILL BUDGE'S TRILOGY OF GAMES	29.95	**23.95**
BIRTH OF THE PHOENIX	14.95	**13.95**
CASTLE WOLFENSTEIN	29.95	**23.95**
CAVES OF KARKAN	49.95	**39.95**
COMPUTER AIR COMBAT	59.95	**49.95**
COMPUTER AMBUSH	59.95	**49.95**
COMPUTER BASEBALL	39.95	**33.95**
COMPUTER BISMARCK	59.95	**49.95**
COMPUTER CONFLICT	39.95	**33.95**
COMPUTER NAPOLEONIC	59.95	**49.95**
COMPUTER QUARTERBACK	39.95	**33.95**
COPTS AND ROBBERS	34.95	**27.95**
CRANSTON MANOR	34.95	**27.95**
CREATURE VENTURE	24.95	**19.95**
CYBER STRIKE	39.95	**33.95**
DRAGON FIRE	49.95	**39.95**
EPOCH	34.95	**27.95**
FALCONS	29.95	**23.95**
GAMMA GOBLINS	29.95	**23.95**
GOBBLER (JOYSTICK/KEYBOARD)	24.95	**19.95**
GORGON	39.95	**33.95**
HELLFIRE WARRIER	39.95	**33.95**
HI-RES FOOTBALL	39.95	**33.95**
HI-RES SOCCER	29.95	**23.95**
INTERNATIONAL GRAN PRIX	30.00	**24.00**
KEYS OF ACHERON	19.95	**18.95**
MASTERTYPE (TYPING GAME)	34.95	**29.95**
MISSILE DEFENSE	29.95	**23.95**
MISSION ASTEROID	19.95	**15.95**
MYSTERY HOUSE	24.95	**19.95**
OLYMPIC DECATHLON	29.95	**23.95**
OO-TOPOS	32.95	**27.95**
OPERATION APOCALYPSE	59.95	**49.95**
ORBITRON	29.95	**23.95**
PEGASUS II	29.95	**23.95**
PHANTOMS FIVE	29.95	**23.95**
PLANETOIDS	19.95	**15.95**
POOL 1.5	34.95	**27.95**
PRESIDENT ELECT	39.95	**33.95**
PULSAR II	29.95	**23.95**
RASTER BLASTER	29.95	**23.95**
ROBOTWAR	39.95	**33.95**
SABOTAGE	24.95	**19.95**
SARGON II	34.95	**27.95**
SHUFFLEBOARD	29.95	**23.95**
SNEAKERS	29.95	**23.95**
SNOGGLE (JOYSTICK/KEYBOARD)	32.95	**25.95**
SOFTPORN ADVENTURE	29.95	**23.95**
SPACE EGGS	29.95	**23.95**
SPACE QUARKS	29.95	**23.95**
STAR THIEF	29.95	**23.95**
TAWALA'S LAST REDOUBT	29.95	**23.95**
TEMPLE OF APSHAI	39.95	**33.95**
THE SHATTERED ALLIANCE	59.95	**49.95**
THE VOICE	39.95	**33.95**
THE WARP FACTOR	39.95	**33.95**
THREE MILE ISLAND	39.95	**33.95**
TORPEDO FIRE	59.95	**49.95**
TUES MORNING QUARTERBACK	29.95	**23.95**
TYPING TUTOR II	24.95	**20.95**
ULTIMA	39.95	**33.95**
UPPER REACHES OF APSHAI	19.95	**18.95**
WIZARD & THE PRINCESS	32.95	**25.95**
WIZARDRY	49.95	**39.95**

Mail order requests are welcomed.

THE SOFTWARE STORE



16562 Gothard St.
Huntington Beach, Ca.
(714) 842-0460
VISA – MASTERCARD

HOURS:
SAT.: 10-6 • SUN.: 10-6
MON.-TUES.: CLOSED
WED.: 10-6 • THUR.: 10-6
FRI.: 10-8

We are located approx. 30 miles S. of Los Angeles, just minutes from the BeachBlvd. exit of the 405- Fwy. at Huntington Beach.

one's way through mazes, finding weapons (some of them magic), accumulating treasures, and defeating or being defeated by monsters. Movement and action determined by single keystrokes took the place of verbal interaction with the screen and these games came to be called role-playing fantasies to distinguish them from "riddled" adventures.

The graphicization of the verbally interactive riddle adventures was not long in following. Ken and Roberta Williams broke this ground with *Mystery House*, a true adventure in which individual pictures depict scenes and substitute for word descriptions.

These milestones are only some of the important ones in the development of the unique form known as the adventure game. They should, however, serve as a basis from which to see the elements you'll encounter as you play adventure games.

The End's Only the Beginning. The next development in adventure games will be the multiple endings game. Up until now, games have had only one possible ending. You could accumulate points, but unless you got the maximum number of points in the minimum number of moves, you couldn't truly win. The multiple ending feature will let you complete an adventure, and then, the next time you play, will introduce additional elements that must be dealt with.

This sort of approach has been taken (in a way) in Scott Adams's *Savage Island I* and *Savage Island II*. You can't play the second until you've completed the first. Many of the adventure games being developed now for release in 1982 will have this feature. Some are likely to require several disks to contain their entire scenarios with all their possible outcomes.

The crystal ball for the near future also reveals adventure games combining animated graphics, of the sort rudimentarily available in *Creature Venture*, with full-color pictures like the ones in *Wizard and the Princess*.

Later on, we'll see interactive adventures, in which multiple players will be able to affect each other's success in playing the game. This sort of adventure will differ from war games in that the puzzles and riddles will still have to be solved, but one player will be able to help or hinder the others' quests for the solution. It's possible that this type of game may be played over modems.

From its rudimentary beginnings, the adventure game has acquired a polish and depth that has attracted a wide following. If you haven't yet experienced the joys of adventuring, they're only as far away as your nearest computer store. And, if you're already an ardent adventurer, the best is yet to come. SL

Answers. *Rabbits*: If you really must know. . . 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, in which each number is the sum of the two numbers that precede it. The sequence was the first recursive number sequence (one in which the relation between two or more successive terms can be expressed by a formula) known in Europe. "*Difficult Crossing*": seven. *Riddle of the sphinx*: man, who crawls on all fours in infancy, walks on two feet when grown, and leans on a staff when aged. *Strongest of all things*: Love! Iron is strong, but the blacksmith is stronger, and love can subdue the blacksmith. *Translation*: I see you are too wise for me.

Apple II Graphics: Mapping the Memory Maze

by Ken Williams

In this issue, we'll examine the processes by which the Apple displays information on your video monitor or television screen. We'll also take a brief look at the three possible Apple graphics display modes.

But first, let's talk further about the inner workings of the Apple's memory and about how the Apple produces the text and graphic images that appear on your screen.

Memory Mapped Output. All images the Apple displays on your video monitor screen come about through a process known as *memory mapped output.* A separate piece of hardware inside your Apple is constantly looking at and interpreting the contents of memory in order to determine what should be displayed on the video screen.

When the Apple is displaying a certain kind of information on the video screen, it is said to be in that particular mode. A *mode* can be thought of as a condition or set of conditions under which certain rules apply. The Apple has three methods of interpreting the contents of memory: text mode, lo-res graphics mode, and hi-res graphics mode.

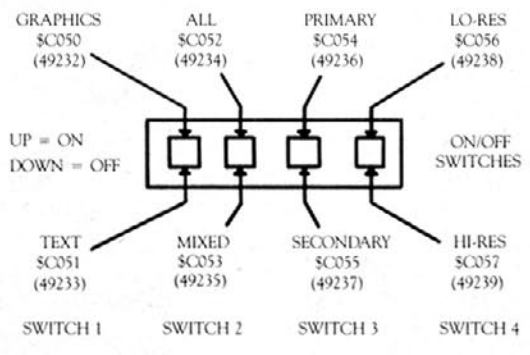
In *text mode,* each byte of memory contains the ASCII code for one character. The monitor screen is mapped into a grid comprising forty horizontal and twenty-four vertical character positions. This translates into twenty-four lines of text, each line containing forty character positons.

In *lo-res graphics mode,* each byte of memory can be used to code for two colored blocks. The monitor screen is mapped into forty-eight horizontal rows and forty vertical columns. The grid that results is made up of colored blocks.

In *hi-res graphics mode,* each byte of memory contains the code for producing seven colored dots. The monitor screen is mapped into a grid made up of two hundred eighty vertical columns and one hundred ninety-two horizontal rows. Each of the first seven bits in every byte of the area of memory that controls the hi-res screen can code for one dot in this 192 by 280 grid.

Knowing Where You're Going. As you've probably surmised, the computer needs some way of knowing which one of these three display modes to go into, as well as what memory locations to draw from. Here's where the so-called soft switches we talked about last time come in.

SOFT SWITCHES FOR THE APPLE



Numbers in parentheses are decimal.

Figure 1

The *soft switches* (four in all) are called soft because they are controlled by the software of the computer. As you'll recall, soft switches can be thought of as on/off switches which activate or deactivate the display modes. Besides the text, lo-res graphics, and hi-res graphics modes, a special mode that mixes graphics with text is provided.

The way to flip a soft switch is to cause the Apple to reference the memory address of its on or off position. This can be done either by entering a memory address from the Monitor or by peeking at the location from Applesoft. For instance, to turn on switch one (select a graphics mode), from Basic you would either enter C050 from the Monitor or PEEK 49232. Remember, the actual value you peek is random gibberish that should be ignored.

Switch one simply serves the function of specifying whether text or a graphics mode is desired. When you first turn on your machine, the Monitor turns off switch one, putting you in text mode. If you flip switch one on, you'll get graphics; which graphics mode is displayed will depend on the settings of the other switches.

If you flip *switch two* to the on position, you're on your way to getting a split screen of graphics and text in which the first three quarters of the screen is in graphics mode, and the bottom quarter is in text mode. But you'll only get this split screen if switch one is on, specifying graphics rather than text mode. This mixed graphics/text mode is very handy and is put to good use in many popular games, including hi-res adventures, in which a player needs to interact with the program in order to play or solve the game.

Switch three permits you to choose between two possible areas of memory through which to display text or graphics.

The System Monitor turns on switch three when you turn on your machine. This switch will become more meaningful to you later on when we explore some animation techniques that use it.

Switch four, which affects the screen display only if switch one is on, specifies whether the computer is to display hi-res or lo-res graphics.

Text mode is the mode in which normal character text is displayed. It's likely that most of the time when you use your Apple, you'll be in text mode, at least at first.

Sometimes, you'll see text faked through the use of the hi-res graphics screen. For instance, some word processing programs rely on this technique in order to be able to represent both the upper and lower-case equivalents of text characters on screen without a lower-case chip.

Translating Text Characters into Numbers. The standard, factory-direct Apple has the capability of displaying only sixty-four different characters. These sixty-four characters are the twenty-six characters of the alphabet (upper case only), twenty-eight special characters (such as parentheses, quotation marks and so on) and ten numerals (zero to nine).

Text coding on the Apple is accomplished by means of the American Standard Code for Information Exchange

!!!! XMAS !!!!
!!!! SUPER PRICES !!!!

TOP
THIRTY
SOFTWARE

APPLE' DISKETTE SOFTWARE

	PROGRAM	RETAIL	TOP 30	✓
1	VISICALC 3.3	$199.95	$147.95	[]
2	GORGON	$ 39.95	$ 32.75	[]
3	CRANSTON MANOR	$ 34.95	$ 27.95	[]
4	POOL 1.5	$ 34.95	$ 27.95	[]
5	RASTER BLASTER	$ 29.95	$ 23.96	[]
6	APPLE PANIC	$ 29.95	$ 23.96	[]
7	DB MASTER 3.3	$229.95	$191.95	[]
8	SNOGGLE	$ 24.95	$ 19.96	[]
9	ROBOT WAR	$ 39.95	$ 31.96	[]
10	SNEAKERS	$ 29.95	$ 23.95	[]
11	EXPEDITER II	$ 99.95	$ 82.95	[]
12	SPACE EGGS	$ 29.95	$ 23.96	[]
13	FLIGHT SIMULATOR	$ 34.95	$ 27.96	[]
14	WIZARD & PRINCESS	$ 32.95	$ 26.36	[]
15	GOBBLER	$ 24.95	$ 19.96	[]
16	ALIEN TYPHOON	$ 24.95	$ 19.95	[]
17	PERSONAL FILING SY	$ 95.00	$ 78.95	[]
18	ORBITRON	$ 29.95	$ 23.96	[]
19	STAR THIEF	$ 29.95	$ 24.95	[]
20	COMPUTER BASEBALL	$ 35.95	$ 28.95	[]
21	SUPERTEXT	$150.00	$120.00	[]
22	VISITREND/VISIPLOT	$259.00	$215.00	[]
23	EPOCH	$ 34.95	$ 27.95	[]
24	PULSAR II	$ 29.95	$ 23.95	[]
25	MISSILE DEFENSE	$ 29.95	$ 23.95	[]
26	WARP FACTOR	$ 39.95	$ 31.95	[]
27	LISA ASSEM LANG	$ 79.95	$ 65.95	[]
28	VISIDEX	$199.95	$166.95	[]
29	TAWALAS LAST REDOU	$ 29.95	$ 23.95	[]
30	COPS & ROBBERS	$ 34.95	$ 27.95	[]

SUB TOTAL : $
SHIPPING & HANDLING : $ 1.95
TOTAL : $

MICROSTAND
P.O. BOX 96
TOLOVANA PARK, OR 97145
(503) 436-1429
VISA MC
* FREE CATALOG *
NOV/DEC PRICES

----- 'TM OF APPLE COMPUTER INC -----

(ASCII). Since the computer can only deal with numbers, it needs some way of translating text characters into numerical form. The ASCII code serves this function.

In ASCII, each possible text character is represented by a number from 0–127. This means that seven bits are sufficient to represent one ASCII value in binary, and that one bit of an eight-bit byte is left over for use as a cue to the computer that a key has been pressed.

Depending on the ASCII code used, characters can be displayed three different ways—inverse, flashing, and normal upper case. Normal text characters are made up of white dots on a black background.

When you're working from the Monitor, ASCII values can be expressed in hexadecimal; when in Applesoft, they can be represented in decimal. A text character, the letter A, for instance, can be produced by poking any one of its three ASCII equivalents, 1, 65, or 193 from Basic, or by entering its hexadecimal equivalents from the Monitor. Each of the three ASCII values will give you A, but each in a slightly different form.

There's a chart on page 15 of the *Apple II Reference Manual* that shows the ASCII values for all the characters you can put on the screen—in normal, flashing, and inverse modes. The chart supplies values in both decimal and hex notation. To get, for example, the ASCII equivalent, in hex, of a flashing G, first locate the character on the chart. Look up to the hex number at the top of the column in which you find the character (in this case $40); now look to the left and find the hex number at the beginning of the row ($7). Add these two numbers and you'll get the appropriate ASCII value ($47).

Text mode uses two different areas of memory. The first, called the *primary screen*, occupies memory locations $400 (1024) to $7FF (2044). The second area, known as the *secondary screen*, occupies $800 (2048) to $BFF (3071). Most of what you do in text mode will rely only on the primary screen area. As mentioned last time, the secondary screen overlaps with the area of memory that's used to store the Apple's Basic programs, so it's hard to use it from Applesoft.

Let's get our feet wet by writing a program that demonstrates how memory is used to represent text. Since we know that screen memory for the primary screen runs from 1024 to 2047, it should be possible to poke things into memory and have them appear on the monitor screen without ever using print statements.

For starters, try typing in and running the following Applesoft program:

```
10  HOME
20  FOR I = 1024 TO 2047
30  POKE I, 193 :REM THE LETTER "A"
40  FOR J = 1 to 30 :REM LET'S SLOW THINGS
     DOWN
50  NEXT J
60  NEXT I
70  CALL 65338 :REM BEEP THE SPEAKER
80  GOTO 80
```

When you run this program, you should see the letter

A filling the screen, working from left to right and top to bottom. The screen will appear to be broken into three pieces that are being filled with As simultaneously. Lines of As begin to form in three different screen areas, and subsequent lines of As begin underneath each of the first three lines until the screen is filled.

Try replacing the number 30 in line 40 of the program with some other number. The higher your replacement number is, the slower the screen will fill. Modifying the program to poke other values can also be fun. For instance, making A equal to 1 in statement 30, rather than to 193, should give you an inverse A.

Now let's examine a few peculiarities about how the screen fills. Why, for instance, does it appear to be broken into three separate pieces?

The best answer we've received is that this effect is tied somehow to the hardware design of the Apple and has to do with the scan rate on standard television sets. Whatever its cause, this oddity has definitely made life more difficult for programmers. Instead of being a simple matter, locating successive lines in memory requires a complex algorithm or a table.

Did you notice that each time a line is drawn on the bottom third of the screen there's a pause before any more As appear? The pause happens because, for no reason we've been able to discover, eight bytes of memory are wasted after each group of three lines has been displayed on the screen.

Making matters even worse, Apple has caught on to this so-called useless memory and has put it to use within DOS. The eight bytes are used by DOS to remember what disk drive was accessed most recently. You'll notice that your next disk access after running the program we just worked with will cause recalibration to occur.

Page 16 of the *Apple II Reference Manual* contains a chart you can use to reference any byte of memory that's part of the text screen. When experimenting, remember that if the screen should scroll after a poke, you'll lose what you just poked into memory. For instance, if you enter "POKE 1024, 193" from the prompt], you might expect to see an A in the upper lefthand corner of the screen. But if pressing return causes the screen to scroll, you'll lose your A.

Let's not delve any further into text mode at this point, since it's used only occasionally in games. We'll return to text mode in later installments of this series when we learn how to implement a scoreboard using mixed-mode graphics.

Lo-Res Graphics. Because of the blocky looking graphics they produce, lo-res graphics are not as popular as they once were. In general, lo-res graphics are put to best use in situations that have special requirements, such as when you want to use the exta colors they afford, or when you need to take advantage of the lower memory requirement and don't mind working with a forty by forty-eight display. For a list of the colors available in lo-res graphics, see Table 8 on page 17 of the *Apple II Reference Manual*.

Lo-res graphics make use of the same memory area ($400 to $7FF) as text graphics do. We can even use the program we wrote earlier as a jumping off point for leaning about lo-res graphics.

Go back to your program and add a Line 5 which says

"5 GR." Now, run the program again. You'll see each character (where the letter A was before) appear as a magenta box overlying a light green box. The screen should fill in exactly the same manner as the text screen did.

If you get a split screen that has text at the bottom, you'll actually see the As appear. To view this example as a full screen of lo-res graphics (no text window), simply turn on Switch two by adding a Line 7 (X = PEEK 49234) to the Applesoft program.

The decimal value 193 we poked into memory is $C1 in hexadecimal. Remember from last time that every location in memory contains one hexadecimal byte, and that every byte can be split into two nibbles. In lo-res graphics, each nibble in a byte corresponds to one colored block on the monitor screen. The nibble on the left corresponds to the bottom block in a character position, the right nibble to the top block.

One of the main drawbacks of working in lo-res is the confusion that can result from trying to deal with two colored blocks at once. Remember, there's no way to poke or peek just a nibble. The smallest amount of memory an Apple can handle at one time is a byte.

Hi-Res Graphics. A great deal of the commercial software written today relies on hi-res graphics. Games, plotting packages, and even some word processors use the hi-res screen for all video output. Virtually everything that follows in this series of articles will deal with different methods of writing to the hi-res screeen.

As do the text and lo-res graphics modes, hi-res graphics uses memory mapped output. Hi-res graphics memory is also divided into primary and secondary screen pages. Considerably more memory (16K bytes in all) is required to support hi-res graphics than is needed for lo-res or text. Each hi-res screen page contains 8,192 memory locations.

In the hi-res graphics mode, the screen has two hundred eighty dots horizontally and one hundred ninety-two dots vertically. One of the advantages of hi-res graphics over lo-res is that you have complete control over any dot. One limitation of hi-res, as we'll see later on, is that although there are six hi-res colors, a given dot can be only one of four of these colors, depending on the dot's location on the screen.

Let's create a simple program to demonstrate how the 8K of memory for the primary hi-res screen, located between $2000 (8192) and $3FFF (16383), is laid out. (In order to understand fully what this program does, you may wish to refer back to September's column, in which the algorithm used for computing the decimal value of the bits in a byte was explained.)

```
10   HGR :REM TURN ON HI-RES
     GRAPHICS
20   X = PEEK (49234): REM SWITCH TWO
     ON
30   FOR I = 8192 TO 16383: REM START AND
     END OF PAGE ONE OF HI-RES
40   FOR J = 1 TO 8: REM BIT # IN BYTE
50   GOSUB 1000:REM CONVERTS BIT # TO
     BINARY VALUE
60   POKE I,X
70   FOR L = 1 to 30:REM SLOW IT DOWN
80   NEXT L
90   NEXT J
100  NEXT I
110  CALL 65338:REM GO BEEP
120  GOTO 120
1000 REM
1010 REM CONVERT BIT # IN J TO VALUE IN
     X
1020 REM
1030 X = 2^(8-J):REM
1040 RETURN
```

The subroutine from lines 1000–1040 will return the decimal value of a byte with only the bit in J turned on. For instance, if you call it with J = 1, it should return a 128.

This program, when run, will poke all the bytes in the 8K that's used to produce hi-res graphics from the primary screen page. It will poke each byte eight times, once for every bit in the byte. The result is that each bit in a given byte will turn on in sucession.

Now run the program. If it seems to behave in an unpredictable—bordering on bizarre—manner, then it probably worked just right.

Watch the One on the Left. You should have seen a dot appear approximately one-half inch into the screen. The dot should then have moved to the left. Just as the dot bumped into the left border of your video screen, a new dot should have appeared, moved to the left a bit and stopped, and then another new dot should have appeared. This process should have repeated until the screen was full of dots.

Your first reaction when all this happened may have been to think that the equation in Line 1030 of our program works backward. It doesn't. The Apple actually uses the bits it finds in a byte in reverse of the way you'd expect (remember our brief discussion last time?).

If you think all this is wild, try slowing down the screen fill process by replacing the 30 in Line 70 with a higher numerical value. If you substitute a high enough value, you should be able to count the dots as they are drawn on the screen. If you count only seven dots per byte, you have verified another of the Apple's idiosyncracies. The leftmost bit in each byte is not drawn to the screen. It has another function, which we'll talk about next time.

You must also have observed that once again the screen appeared to have been cut into thirds. Once again, you're seeing that lines are drawn from various positions on the screen, as well as noticing the pause that results from the fact that eight bytes of memory are "wasted" after every third line is drawn on the hi-res screen.

A chart on page 21 of the *Apple II Reference Manual* shows the memory address for every dot on the hi-res screen. In the next issue, we'll learn more about this area of memory. We'll also discuss hi-res color and outline some techniques that will make dealing with the hi-res screen less confusing. SL

Two New Additions to our Best Selling Adventure Series

Hi-Res Adventure # 3
CRANSTON MANOR
by Ken Williams & Harold DeWitz

ULYSSES and the Golden Fleece

HI-RES ADVENTURE #4
Our First 2 Disk Adventure!

BOB DAVIS, Co-ordinator for TIME ZONE, the "Ultimate" HI-RES Adventure project, has teamed up with KEN WILLIAMS to bring the high adventure of ancient mythology to your APPLE. "CAST-OFF" with ON-LINE SYSTEMS and ULYSSES, and see if you can find the Golden Fleece.

ULYSSES is a HI-RES Adventure epic set in ancient Greece.

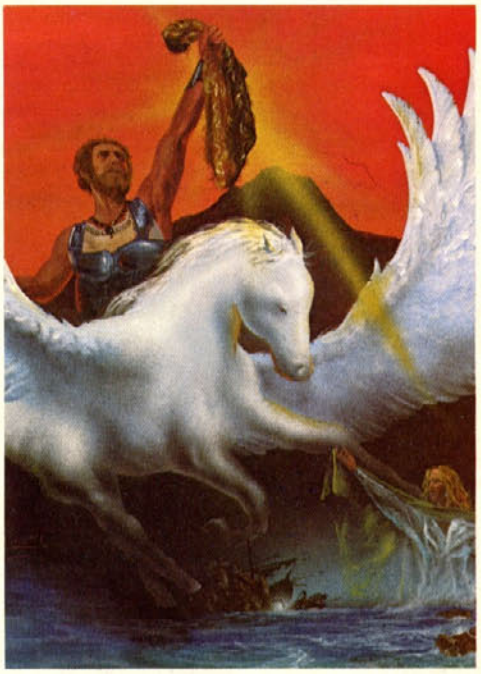
PRICE $34.95

CRANSTON MANOR

HI-RES ADVENTURE #3

In this HI-RES Adventure, you explore the estate of Old man Cranston and find the riches that are hidden throughout the manor and surrounding grounds. Be careful! Rumor has it that the disembodied spirit of the old man roams the estate guarding his fortune.

CRANSTON MANOR is a HI-RES Adventure game set in contemporary times in Coarsegold California.
PRICE $34.95



Hi-Res Adventure #4
ULYSSES
by Bob Davis & Ken Williams

Hi-Res Adventure #0
MISSION ASTEROID
by Ken & Roberta Williams

Written as an introduction to the HI-RES Adventure family of games, MISSION: ASTEROID is designed to acquaint beginning adventure players to the wonderful world of HI-RES Adventures.
PRICE: $19.95

Hi-Res Adventure #1
MYSTERY HOUSE
by Ken & Roberta Williams

When you enter the house, you are pulled into the mystery and intrigue as your companions are murdered one by one. Be careful, you may be next! Can you solve the mystery and leave the house alive? The secret passage way may lead you to the answer.
PRICE: $24.95

Hi-Res Adventure #2
WIZARD AND THE PRINCESS
by Ken & Roberta Williams

After one year on the best sellers list, this adventure is still going strong. To win this game you will have to cross deserts, chasms and oceans in your search for the kings daughter. If you find her, half of the kingdom of Serenia is yours.
PRICE: $32.95

ON-LINE systems

Visa, Master Charge, COD or Checks Accepted
For direct mailing add $1 for shipping.

36575 Mudge Ranch Road • Coarsegold, CA 93614 • 209-683-6858

Gamemakers' Summit

Bill Budge Chats with Star Galaxy

In Japan, nobody makes a living writing game programs. In fact, it would seem that the Japanese consider the title "professional game programmer" a contradiction in terms.

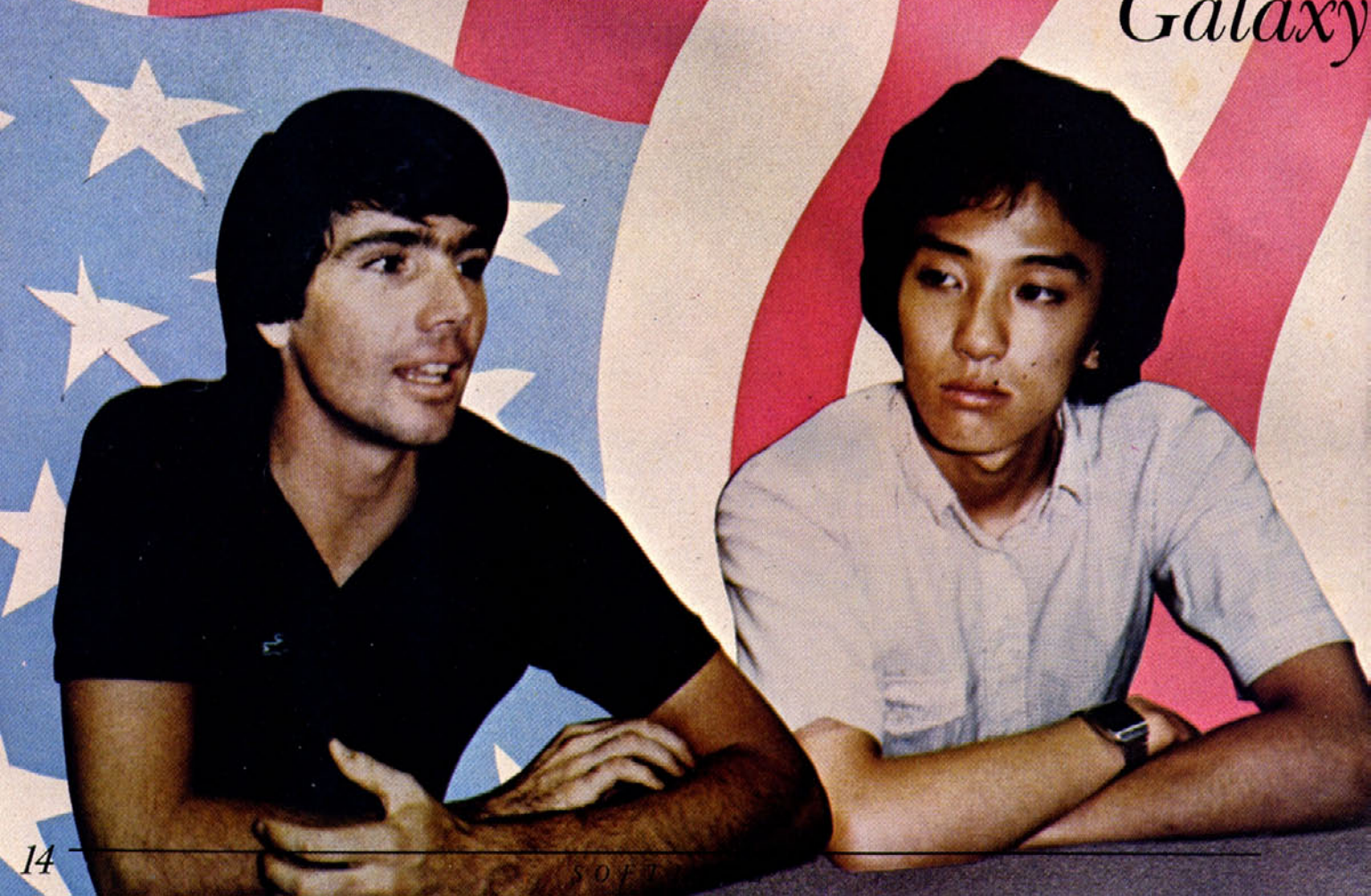
Although writing software for business applications is thought fairly respectable, game programming is held in much lower esteem. It has what the Japanese describe as a "negative aspect" to it. The Japanese view game programming as trivial—kind of like a game itself—not professional, and perhaps not even quite honorable. Certainly, becoming a game programmer is not an appropriate goal for a serious-minded, creative young person in Japan to aspire to.

Another View. Minoru Nakazawa, president of Star Craft, a Japanese software exporting firm, disagrees with this assessment. He wants the situation in his country—both the prevailing attitude toward game programming and game programmers' feelings about themselves—to change. Looking forward to a time when Japanese game programmers work in a positive environment and command the admiration and respect of their countrymen, Nakazawa is doing all he can to put game programming in a new perspective.

A significant step in this direction took place in August. Accompanied by Gary Carlston, of Broderbund Software, Bill Budge, author of *Raster Blaster* and other popular game programs, visited Japan at Nakazawa's invitation. The purpose of Budge's visit was to create an opportunity for dialogue between Japanese game programmers and the American; the setting was an informal gathering to be attended by Nakazawa, Budge, and several Japanese programmers, including Tony Suzuki, author of *Alien Rain*, Jun Wada and Ken Iba, coauthors of *Snoggle*, and Raly Yamaguchi, a talented business programmer. It was Nakazawa's hope that such a meeting would both educate and encourage Japanese game programmers.

With a translator facilitating the discussion, the day's topics ranged from hardware to software, from game programming to business programming, from the conditions of the present to the challenges of the future. And, after the tentative first ten minutes, the language barrier was hardly noticeable, so eager were the participants to ask questions, share experiences, and discuss concerns.

Hardware Breaks the Ice. After welcoming Budge and the other programmers and expressing his hopes for the day, Nakazawa asked Budge to comment on how Japanese hardware is assessed in the United States. Budge assured his



listeners that Japanese technology is well thought of here, so well thought of, in fact, that many people believe the Japanese will eventually take over the American market. As a result, Budge explained, there are almost no hardware companies left. Instead, the response of American companies has been to become software companies. "If a company builds a new computer, ten people work on the hardware and ninety people work on the software."

By contrast, the Japanese market for software is a limited one. Japanese consumers, unlike their counterparts in this country, have not yet begun purchasing microcomputers in great numbers for personal/home use. In addition, so many varieties of microcomputer are produced in Japan that no one, two, or three in particular have emerged as favorites for which the majority of software is designed. Not much game software is made in Japan and most of what is produced there finds its market in the United States; hence, the existence of an export company like Nakazawa's Star Craft.

A major difference between the Japanese and the American game programming climates became apparent when Budge talked about how strange it is to walk into a computer store and see his programs on shelf or to overhear people talking about them. "If any of you go to America, it's really fun to go to a store and see your program and just wait for somebody to come and look at it or buy it." The Japanese programmers could imagine the experience Budge described, but they could not identify with it firsthand. Their own programs are not very likely to be on the shelves of computer stores in Japan.

Instead, both Suzuki and Wada spoke of the lack of confidence they tend to feel in their abilities to make their livings as game programmers. They contrast their perception of the American attitude of confidence with the more tentative attitude they experience in Japan—an unstated but powerful feeling of uncertainty that begins to convince them that they can't do it, that they probably won't succeed, so it's rather pointless to try.

Confidence Takes Time. Budge believes that it is only a matter of time before Japanese consumers catch microcomputer fever, and that once they do, an installed user base will come into being. When this happens, the Japanese demand for software of all sorts will increase dramatically, ensuring programmers of a market for the software they produce. And it's likely that Japanese programmers will be inspired to create more and better software if they feel confident that their efforts will be supported in their own country.

But for right now, Japanese game programmers cannot be sure of such respect and appreciation in Japan. When Budge asked Suzuki and Wada whether they planned to make professional game programming their life's work, the uncertainty about the future prospects for professional game programmers in Japan surfaced again. As much as he enjoys game programming, when it comes to a career, Suzuki thinks about becoming a scientist of some sort, a chemist maybe, instead of a game programmer.

Budge's heartfelt response to Suzuki: "I hope you'll change your mind. You're a very good programmer. In the future game programs will become much harder to write, and I think writing them will be very respectable. That's a challenge to us, because the money that will be there won't be free; you'll have to be able to write the most popular program ten years from now, which none of us may be able to do."

As Budge sees it, the possibilities of game programming and graphics are vast and have just barely been explored. "Very few people," asserts Budge, "are making an effort to extend the idea of a personal home computer game or making the effort to do more sophisticated graphics. The field of computer graphics is very big, and game programmers have used but a small corner of it. . . . I am thinking of personal computer games as being more than what they are now, which mainly involves shooting things that move. Personal computer games would be very different. I think we're just beginning to explore the possibilities."

Games Plus Business Equals Benefits Plus Compromises. According to Nakazawa, Japanese microcomputer manufacturers, like those in the United States, are attempting to include "the best of everything, games and business" in a personal computer. The Japanese realize that such an approach requires design compromises.

Yet the interrelationship developing between business programs and game programs has positive consequences for

Pictured below: Bill Budge, Jun Wada, and Tony Suzuki.

Craft's of Programmers

by Jean Varven

both fields. People's concerns about business applications have resulted in the creation and use of new hardware that cannot help but benefit game programming. And business programs are also changing; graphics are becoming an integral part of business applications programs.

Budge has tried his hand at creating both serious programs and game programs. He enjoys business programming—in fact, he's working right now on developing a new word processor—but says, "I think I find game software more challenging and more fun. It allows you to be more creative and to try more sophisticated techniques; if things don't work, it's not a disaster."

New CPUs Welcome—with Some Reservations. Budge wants to see the development and widespread use of better, more powerful CPUs. Both the graphics involved in game programming and the speed at which game programs operate push hardware to its limits. "A business program—you write it, and it runs as fast as it runs. But a game program, you write it, and, if its not fast enough, you throw it away."

Wada, Suzuki, and Budge all agreed that that its a frustrating experience to come up against the limits of the smaller CPUs. They have the feeling at times that they have simply run out of things to do. This contributes, in Budge's view, to the lack of understanding that some people have about game programming.

Budge sees learning about the capabilities and construction of new microprocessors as one of the essential challenges to game programmers in the future. "The reason I want a faster CPU is because really advanced computer graphics use many data structures. Very complicated techniques like curved surfaces and shading require complicated data structures." A microprocessor like the 6502 that drives the Apple II and the Atari lacks the power that makes games all they can be.

The Japanese programmers expressed some uncertainty at the thought of learning the ins and outs of a whole new microprocessor. "Wouldn't you rather wait until the right computer's on the market so you wouldn't have to keep restudying it?" joked Wada.

While Budge could relate to these feelings, he encouraged a positive attitude. Learning a new microprocessor soon after its introduction gives a programmer certain advantages. He is in on the ground floor of a new market and has plenty of time to develop his expertise and to explore the possibilities.

A Special Blend. Of course, the programmers agreed, there can also be a disadvantage in being too early. When there's not yet a base of computer owners to buy new software programs, some of the people who create programs early on may be passed by. And they also had to agree with Nakazawa's point that Suzuki is a real exception to all the rules; although he started late on the Apple, he surely caught up in a hurry.

And in fact each one of the programmers who met together in Japan this past August is exceptional. As Nakazawa pointed out, "Among the Apple users in Japan, there are quite a few people who are very knowledgeable and who have the ability to write good programs; on the other hand, there are very few among them who, like Tony Suzuki and Jun Wada, complete really good quality, marketable programs."

Budge concurred and in the process of responding, went on to enumerate some of the attributes of a successful game programmer. "I think it's the same in the United States as it is in Japan. Most people don't have the combination of skills—it's kind of specialized. To write a good game, you have to know assembly language, which rules out quite a few people; you have to like to play with graphics, which rules out more people; and in the end, there aren't that many people left who can write games. And the number who can finish a program is another small fraction."

Budge suggests that people are likely to have a deeper understanding of what's involved in graphic game programming if they try some programming themselves. "People send me letters, and basically, the question is: What's the secret of hi-res programming? Not specific questions. They want to know the secret. . . . They see a ball bounce off something and they think, well, that's the way the computer does things; but that's not at all true. It's a great deal of work to make things the way you want them."

Users groups or clubs, common in both America and Japan, can foster a better, more informed understanding of the hard work and creative energy that go into programming. Despite the piracy problems in both countries, Budge and all the others present agreed that clubs are, for the most part, a positive force. They look forward to a time when the users who spend time and effort breaking the codes of existing software programs begin instead to develop their skills and use them to create programs of their own.

Moving into the Future. Game programming continues to evolve. Recently, we've seen software become a consumer product in which program, documentation, packaging, and advertising all work together. As for the future, Budge sees more complex graphics and predicts a shift in the emphasis of personal computer games—away from speedy home-arcade games and toward games with more diverse entertainment value.

Budge also envisions more collaboration in the creation of game programs. He suggests that computer game companies of the future may be modeled after the sort of technical group that Star Wars producer George Lucas has put together to produce computer graphics for films, with computer graphics experts working in teams with systems programmers.

When individuals who want to understand and learn from one another decide to collaborate on a project, some marvelous things can happen. This sharing of knowledge and ideas between people who respect one another's skills may give each individual the opportunity to express himself in a new way.

Finally, deciding whether to make game programming one's profession is not a simple matter. It's an individual decision in which many factors play a part. But whatever career choices they make later on, the programmers who met in Japan this past August know the meaning of professionalism. It has to do with respect, and being able to respect themselves for the integrity of their own programs is what matters most. SL

apple discount software

CALSOFT

Personal—Entertainment—Business SOFTWARE

We have all the latest software—ASK US!

Low Discount Prices / 15 to 25% Discount off List Price
Fast Convenient Service / We ship same or next day
Large Selection of Software / Call or Write for our FREE Catalog!

Call Toll Free: 800-423-5290 In California: 213-991-9641

All products below are on Disk unless otherwise noted

Apple Software

✓ *Also available on cassette*

ADVENTURE INT'L

	Retail Price	Our Price
Adventure -1, 2 & 3	39.95	33.95
Adventure -4, 5 & 6	39.95	33.95
Adventure -7, 8 & 9	39.95	33.95
Adventure - 10, 11 & 12	39.95	33.95
Planetoids - "Asteroids" ✓	19.95	16.95
Adventure 1-12 please specify (cass only)	19.95	16.95

AUTOMATED

Introductory 3-pack (Rescue, Morloc's, and Datestones)	49.95	42.45
Starfleet Orion ✓	24.95	21.20
Invasion Orion ✓	24.95	21.20
Morloc's Tower ✓	19.95	16.95
Datestones of Ryn	19.95	16.95
★ Dragon's Eye	24.95	18.70
Tuesday Morning Quarterback	29.95	25.45
★ Temple of Apshai	39.95	29.95
Hellfire Warrior	39.95	33.95
★ Star Warrior	39.95	29.95
Rescue at Rigel ✓	29.95	25.45
★ Crush, Crumble & Chomp	NEW....29.95	22.45

BRODERBUND

★ Alien Rain	24.95	18.70
Galactic Empire	24.95	21.20
Galactic Trader	24.95	21.20
Galactic Revolution	24.95	21.20
Tawala's Last Redoubt	29.95	25.45
Alien Rain - "Galaxian"	24.95	21.20
Alien Typhoon	24.95	21.20
Payroll	395.00	335.75
General Ledger	495.00	420.75
Space Warrior	24.95	21.20
★ Snoggle	32.95	24.70
★ Apple Panic	NEW....29.95	22.45
Space Quarks	NEW....29.95	25.45
Genetic Drift	NEW....29.95	25.45

BUDGECO

★ Raster Blaster	29.95	22.45

EDU-WARE

Windfall	19.95	16.95
Network	19.95	16.95
Terrorist	29.95	25.45
★ The Prisoner	29.95	22.45

HAYDEN

Sargon II ✓	34.95	29.70
★ Reversal ✓	34.95	26.20
Assembly Lang. Dev. Sys	39.95	33.95
Applesoft Compiler	200.00	170.00

INNOVATIVE DESIGN

Pool 1.5	34.95	29.70
★ Shuffleboard	29.95	22.45

MICRO LAB

Dog Fight	30.00	25.50
★ Data Factory	150.00	112.50
Mini Factory	75.00	63.75

MICROSOFT

	Retail Price	Our Price
Typing Tutor ✓	19.95	16.95
Olympic Decathlon	29.95	25.45
Adventure	29.95	25.45
FORTRAN 80	195.00	165.75
A.L.D.S.	125.00	106.25
BASIC Compiler	395.00	335.75
COBOL-80	750.00	637.50
SoftCard	399.00	339.15
RAMCard	195.00	165.75
TASC Compiler	NEW....175.00	148.75

MUSE

A.B.M.	24.95	21.20
SuperText II	150.00	127.50
Three Mile Island	39.95	33.95
Robot War	NEW....39.95	33.95
★ Castle Wolfenstein	NEW....29.95	22.45

ON-LINE SYSTEMS

HI-RES Adv-0 - Mission Asteroid	19.95	16.95
HI-RES Adv -1 - Mystery House	24.95	21.20
★ HI-RES Adv -2 - Wizard & Princess	32.95	24.70
HI-RES Adv -3 - Cranston Manor	34.95	29.70
HI-RES Adv -4 - Ulysses	NEW....34.95	29.70
HI-RES Adv -5 - Time Zone	NEW....99.95	84.95
★ HI-RES Football	39.95	29.95
HI-RES Soccer	29.95	25.45
HI-RES Cribbage	24.95	21.20
Sabotage	24.95	21.20
★ Cross Fire	NEW....29.95	22.45
★ LISA Educational System	NEW..119.95	89.95
The Dictionary	NEW....99.95	84.95
General Manager	NEW....99.95	84.95
Pegasus II	29.95	25.45
Threshold	39.95	33.95
SuperScribe II Word Processor	129.95	110.45
★ Expediter II Applesoft Compiler	99.95	74.95
LISA Assem. Lang. Dev. System	79.95	67.95

PERSONAL

Monty Plays Monopoly ✓	34.95	29.70
Zork	39.95	33.95
Bridge Partner ✓	24.95	21.20
Desktop Plan II	199.95	169.95
CCA Data Management System	99.95	84.95
Visiplot	179.95	152.95
Visitrend/Visiplot	259.95	220.95
Visidex	199.95	169.95
Visiterm	149.95	127.45
Visicalc 3.3	199.95	169.95
Visifile	NEW....250.00	212.50

STONEWARE

★ D B Master	229.95	172.45

SUB LOGIC

A2 FSI Flight Simulator ✓	33.50	28.45

TG PRODUCTS

Game Paddles (accs)	39.95	33.95
★ Joystick (accs)	59.95	44.95

SIRIUS SOFTWARE

	Retail Price	Our Price
E-Z Draw 3.3	49.95	42.45
Cyber Strike	39.95	33.95
Phantom Five	29.95	25.45
★ Space Eggs	29.95	22.45
Pulsar II	29.95	25.45
Orbitron	29.95	25.45
Autobahn	29.95	25.45
Gamma Goblins	29.95	25.45
Gorgon	39.95	33.95
★ Sneakers	29.95	22.45
★ Epoch	NEW....34.95	26.20
Copts & Robbers	NEW....34.95	29.70
Outpost	NEW....29.95	25.45
Beer Run	NEW....34.95	29.70
Pascal Graphics Editor	NEW....99.95	84.95

STRATEGIC SIMULATIONS

Computer Bismarck	59.95	50.95
Computer Ambush	59.95	50.95
Computer Napoleonic	59.95	50.95
Computer Air Combat	59.95	50.95
Torpedo Fire	59.95	50.95
Operation Apocalypse	59.95	50.95
Computer Conflict	39.95	33.95
★ The Warp Factor	39.95	29.95
The Cartels & Cutthroats	39.95	33.95
Computer Quarterback	39.95	33.95
The Shattered Alliance	59.95	50.95
★ Computer Baseball	39.95	29.95
President Elect	39.95	33.95
Battle of Shiloh	NEW....39.95	33.95
Tigers In The Snow	NEW....39.95	33.95

VERSA COMPUTING

VersaWriter Expansion Pac-I	39.95	33.95
VersaWriter Graphics Tablet (accs)	299.00	254.15

We also carry complete lines from the following companies:

ARTSCI • AVALON HILL
CONTINENTAL
HIGHLANDS • PHOENIX
PICADILLY
CAVALIER • DATASOFT
HOWARD SOFTWARE
SOFTAPE • SENTIENT
SIERRA • SIR-TECH
CONTEXT • L.J.K.
SOFTWARE PUBLISHING CO.
SYNERGYSTICS • IUS
DELTA • DENVER
SENSIBLE • QUALITY
LAZER SYSTEMS • UNITED

If you don't see it, Ask Us!

CALSOFT

6610 Tamarind St.
Agoura, CA 91301

Call Toll Free: 800-423-5290 In California: 213-991-9641

We take Mastercard & Visa. (Please include No. and Expiration Date.) California residents add 6% sales tax. Include $2.00 for shipping. ($3.00 for UPS Blue Label). Checks, COD (+$1.40) or Money Order.

★ Sale prices are through December only! Prices subject to change without notice.

Software To Learn From: How To Choose It

by Sherwin Steffin

When educational software is purchased, the buyer's objective is usually to increase the knowledge or skill of the user. The user (learner) brings a base of prior experience and knowledge into this new situation. After using the software, learners should possess skills or knowledge they didn't have before or should have improved their skill levels.

As more and more software publishers develop and market educational software, both the school and home consumer are faced with some demanding tasks. Selecting among the many available titles and evaluating their usefulness requires time and expertise.

The Program, the Computer, and the User. Before questioning the instructional validity of CAI materials, it's worthwhile to assess whether the software system is reliable—that is, both user-friendly and misuser-resistant.

Many users of CAI are relatively unfamiliar with the computer and like to explore what it will do. As a result, users are likely to try out every keyboard entry imaginable. As the simplest example of this, assume that a program requires the learner to input the answer to a multiplication problem. Instead of typing the numeric answer, the learner may type an alpha key; press the space bar, escape, or return; turn off the computer; or open the disk drive door while disk activity is going on.

The careful programmer anticipates these moves by building in appropriate software devices to circumvent the crashing of programs, either through misuse or by intent. To the degree that this crash-proofing is not present, the usefulness of any educational program is severely impaired.

A second component of the program construction, at a somewhat higher level, is the human engineering provided for the user. An example of this might be a program that directs the user to hit return rather than press the space bar to continue. A look at the keyboard confirms that pressing the space bar is easier than hitting return. Current programming technology also allows single keystroke input on the part of the users in such areas as menu selection.

The Instructional Assignment. Once you've established that a CAI program meets some minimum programming criteria, your next task is to evaluate its potential effectiveness for the learner. If program construction is the foundation of the system, the variables concerned with learning potential represent the floor, the walls, and the framing for the roof.

The first consideration in the construction of an instructional system must be *content accuracy.* A spelling program containing misspelled words does not serve the learner; neither does a geography program that matches state capitals with state names incorrectly. An arithmetic program in which the computations are incorrect is also useless.

Most teachers agree that learners should know what they're expected to learn and that the best measure of gain in learning is to test learners' knowledge both before and after the experience that's designed to teach them.

Learners should be given, either within the program itself or within the accompanying documentation, a precise statement of what they're expected to learn. Whenever possible, programs should have the capability of pretesting learners. This allows the learners to assess their current knowledge and determine what portion(s) of the program will be most useful to them.

In some more elegant instructional systems, pretesting and assignment of components of the program to the student is done by a *learning manager.* The learning manager, set up by the parent or teacher, tailors the needs of the individual learner to the attributes of the instructional system being used. This kind of software management system enhances greatly the flexibility of the entire CAI system; it is a prerequisite for the individualization of instruction to a particular learner.

Tailoring Format to Objectives. Thus far we've been talking about instructional objectives. Before we move on, these objectives deserve further attention, since the way they're developed by the designer of the CAI system has implications for the format in which the system is presented to the learner.

An *instructional objective* can be defined as a precise statement of an intended learning outcome. The achievement of the objective is then evaluated by observing some aspect of the learner's behavior. Several classification systems have been developed that delineate the differences among categories of objectives. The scheme we'll use here is a simplification of one developed by Benjamin Bloom in the late 1960s at the University of Chicago. This scheme has four categories.

1. *Rote memory.* In rote memory learning, the objective is the memorization of simple facts. Learning the multiplication tables by heart, matching the names of the state capitals with the correct state names, and memorizing the spellings of words are all examples of the kind of learning that requires rote memory practice. Drill and practice programs, whether simply or elegantly done, provide the learner with many opportunities for memorization of simple factual material, but do not address the more complex learning needs reflected in the objective categories that follow.

2. *Generalization and abstraction.* In this category, the

learner is asked to rearrange and process information. Verbs and phrases such as *explain, compare and contrast, project a trend,* and so on are all clues as to the behavior expected of the learner in this sort of situation. Achieving this kind of objective requires the learner to do more than merely memorize facts. Therefore, the format of CAI materials intended to foster the abilities of generalization and abstraction necessarily must lead the learner along a sequential path from known territory to new knowledge. Tutorial programs formatted in a way that provides learners with feedback on the appropriateness of their responses are clearly needed here.

3. *Application-level objectives.* This kind of activity involves learning how to solve some specific problem by applying a general set of rules. Almost any algebra problem requires this kind of performance. An example: Given the formula for the area of a right triangle and the lengths of the sides, compute the area of the triangle.

Other instances of such an objective might include formulating the solution of a chemistry equation-balancing problem, finding the subject and verb in a sentence, or correctly dividing one fraction by another. Here again, tutorial problems that present the rule, an example of the use of the rule, and a sample problem offering the opportunity to practice and obtain feedback are appropriate to these kinds of objectives.

4. *Creative problem solving.* Since problem-solving skills require complex behavior on the part of the learner, simple tutorial problems are seldom enough to promote the development of such skills. Simulations that serve as models of real-world interactions have the greatest probability of achieving these ends. It's important to note, however, that human dialogue and interaction between learner and teacher are essential if the potential of simulations for teaching problem-solving skills is to be fully realized.

Some Additional Considerations. So far, we've described some of the components that are essential to an effective CAI structure . Now, let's take a look at some of the trimmings that can make an instructional system highly desirable, interesting, and motivating to the learner.

Legibility and screen cleanliness. A minimum requirement for educational software is that text displays on the screen be available in both upper-case and lower-case formats. This goes far beyond any consideration of esthetics for the learner. Instead, it should be understood that formats using both upper and lower cases are the ones most learners are familiar with and best able to process. Upper-case text alone is distracting to the learner. The initial caps at the beginnings of sentences and on proper nouns provide emphasis; they give learners' eyes a break in routine that helps them process textual information more effectively.

Also of concern is the amount of information on the screen at one time. When large amounts of information are presented all at once, the human processing mechanism may find itself overloaded. The old psychology-based rule of thumb (seven items, plus or minus one) is a useful one to remember when judging how many informational segments ought to be presented to the viewer at one time. If the aim is to maximize the efficiency of learning, presenting a full screen of single-spaced text is inappropriate.

Attention to cognitive style. There are many ways of presenting the same information. Most of us are aware that certain presentation formats are more effective than others in facilitating our learning.

The many methods of processing new information are called *cognitive* (thinking) *styles.* CAI systems geared to serve these differences by providing multiple strategies for learning are far more useful than those that provide only one strategy by which learning is to take place. CAI materials that take cognitive styles into account may put rules, examples, sample problems, definitions, or tests into the control either of the learner or of the teacher who is aware of the most appropriate learning style for the learner.

Summing Up. We've attempted to give you some basic tools by which you may judge the value of instructional software you're thinking of buying. These tools are of essentially two kinds: those that help you assess the program's operating characteristics, and those that help you assess the potential of the instructional system for reaching specified learning outcomes.

Attention should also be paid to the presence of upper-case and lower-case textual materials, the cleanliness of the screen presentation, and the care given by the program designer to the varying cognitive styles of learners.

We have thus established a framework for judging both the essential and the nice-to-have ingredients that make up effective, appropriate CAI materials. In future columns, we'll talk about how to motivate and reinforce learners as they take part in the CAI experience. SL

BEZ
PRESENTS
BEZMAN
© BEZ 1981

Work your way through the Maze, eating dots and scoring points as you go. Certain dots give you invulnerability from the beasties sent to guard the dots. The better you get, the faster the game goes. Have fun but watch out for speedy!

Bezman runs on a 48k Apple] [plus w/disc.
To order, send $22.95 plus $1.50 shipping and handling. CA residents add 6%.

BEZ-4790 Irvine Blvd — Suite 108B
Irvine, California 92714
APPLE II PLUS IS A TRADEMARK OF APPLE COMPUTER, INC.

New Players

PICCADILLY SOFTWARE

Dennis Tolley and Michael Mahoney have worked together for years now, but their new company, Piccadilly Software, came into being only this summer.

The two men met and became friends while both were employed at a bank—Tolley as director of long range planning and Mahoney as head of data processing. Both used big computers like the IBM 370 in the course of their work.

Three key factors—computer expertise, friendship, and the desire to do something more challenging with their lives—motivated the two men to consider giving up their high-power jobs to start their own company. After six months of planning and discussion, they took the plunge or, as Tolley put it, "sold the farm." The result was Stonehenge Computer Corporation, a business software publisher/dealer and computer retailer.

But when the opportunity arose this summer to market a hot new computer game, Tolley and Mahoney elected not to mix business with pleasure. Instead, they formed an entirely new company, Piccadilly Software, for the purpose of creating and marketing entertainment software.

The game they marketed, *Falcons*, is the creation of Eric Varsanyi, a Stonehenge employee, and his friend Thomas Ball. If you've played it, you know it's fast-paced and challenging; if you haven't, you might want to. Piccadilly also has three new games in the works—*Warp Destroyer*, another Varsanyi/Ball collaboration, *Suicide* by Steve Hawley, and *Survival*, an adventure.

Warp Destroyer takes you on an interplanetary journey. The graphics in this home-arcade game have a three-dimensional effect. Objects come into view as they would in real life, seeming to get closer and then to pass you. You'll feel as though you are looking out from your ship into the surrounding space. The illusion of movement is achieved by changes in the sizes of the objects themselves. This gives you a feeling of perspective—and requires more space in memory and complex programming as well.

In *Suicide*, creatures drop from the top of the screen to the bottom and you try to bounce them up again with a bar. This game involves several levels and a variety of creatures. As you become more proficient, the game becomes more difficult. It also has a surprise ending.

Survival involves a plane crash in which you, as the sole survivor, are left in the middle of nowhere. Your task is to decide what direction to go (through the mountains, jungle, or desert), and to survive by using your wits and the few objects you are able to take along with you. Tolley says *Survival* really tests your ability to think creatively. He hints that thinking of ways to put familiar objects to unexpected use may keep you alive.

The *Survival* hint Tolley offers grows out of experience. Creative thinking and the courage to trust their own instincts are what motivated Tolley and Mahoney to set out on the adventure of starting their own business. These essentials should serve them well, ensuring not merely their survival, but their continued success.

BEZ

Sporting a medium build, gray hair, a short beard, wishing he were six feet tall, wearing Ben Franklin octagonal glasses, John F. Besnard is the thirty-four year-old creator of *Bezman*. Married, with two growing boys, Besnard lives in Irvine, California, forty miles south of Los Angeles.

After receiving the bachelor's degree in math from UCLA and the MBA from Cal State Long Beach, Besnard spent the next ten years writing programs for minicomputers, until this year when he left his programming post for a nonprogramming position. "Then I started to suffer from programmer's withdrawal. So I got an Apple early this year.

"The first thing I said was 'what can I do with this?' Balance the checkbook, of course! So I wrote a little program for that." Blessed with programming skills, a fertile imagination, and enough free time, Besnard was ripe for the cosmic interference that would send him on the treacherous path of the software game business.

"One day, Mike, the oldest kid, says, 'You should write a game.' And that's how I came to write *Bezman*." Besnard first tried Applesoft but found the finished game too slow. Experienced with assembly language programming, Besnard tried this mode next and found success.

Once he had the game finished, Besnard realized that it was good enough to sell. "One thing they don't teach you in business school is how to go down to the local software store and peddle your game. It was an eye-opener." Another thing that business school underplayed was the importance of cash flow and accounts receivable.

Although getting started is difficult, Besnard had no trouble selling *Bezman*. *Bezwars* was a different story, with the biggest draw expected to come from kids not yet in high school. "I couldn't go into the store and say 'this is a great game for adults after they've had a few beers.'"

Regardless of how *Bezwars* eventually measures up against *Bezman*, Besnard has several more games in the chute.

The first two to look for are *Bez-Mx* and *Bez-1*, both based on current defense projects that President Reagan has given the go-ahead to in real life. Besnard is excited because he feels they're great strategy and action games. You lay down your strategy at the beginning of the game and then modify that strategy during real-time using game paddles. *Bez-Mx* and *Bez-1* should be available in December.

More games are on the way, one of which is *Bez Wax*. Based on the TV show "Name That Tune," it will be a party game that includes hundreds of songs. *Bez-Off* will be a strict action game with the usual levels of difficulty. The trick is that you can go to any level you want during play.

Optimistic about the future, Besnard loves to dream of the day when there will be a Bezland, USA, and Bez cartoons. But that's still a way off. For the moment, he echoes a philosophy that's surely familiar to the likes of Richard Nixon and Francis Coppola: "There's no such thing as bad publicity." SL

It seemed like a cinch assignment. At least for the galaxy's master thief.

Just sneak inside the Star-Complex Base, dodge a few guardian robots and grab the formula. A piece of cake.

Oh yeah! They didn't say the robots went berserk at the sight of a human. They didn't say I'd have to laser my way through 7 levels. Or, that rampaging robots would keep coming and coming.

And most of all, they didn't bother to say there's some weird blob, an unearthly **something** that tracks you right thru the damn walls.

I made it to the 7th level. I found the formula. But, more robots are massing out there. And I'm wounded. My energy charge is low.

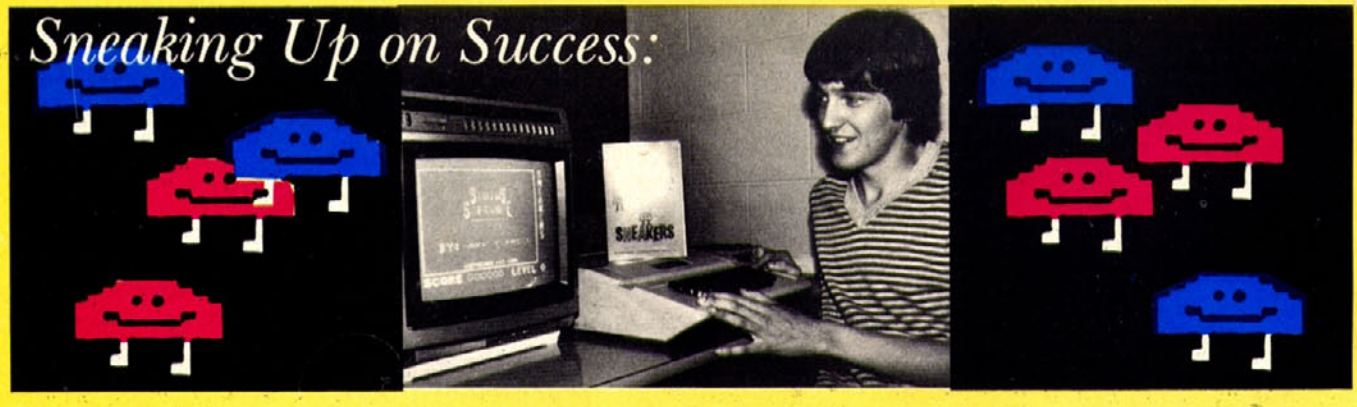
You. Yes, **you**. How good a Thief are you? Prove yourself. Come get me. Come save me.

$29.95 for Apple II*. At computer stores, or from DATAMOST.

VISA/MASTERCHARGE accepted. $1.00 shipping/handling charge. (CA residents add 6% tax)

*Apple II is a trademark of Apple Computer, Inc.

DATAMOST
19273 Kenya St., Northridge, CA 91326, (213) 366-7160



An Interview with Mark Turmell

by Greg Voss

"There are no artesians. But then there are."
Mark Turmell

In the wee, small hours of the morning, long after the doors of the arcade are locked and the last spare change is spent, long after the machines are silent, and the ghosts and phantoms of the game world have begun to slip through the shadows, he works. Huddled in the corner of the basement, his lanky figure looms over the keyboard; his wide eyes reflect the glare of the video tube. A mischievous grin sneaks onto his face.

This man, with the use of his computer, is about to tap into a small fortune.

No, he hasn't figured how to break into the files of the Chase Manhattan Bank—not yet. But if this job goes as well as the last one did, it should be a real steal.

Triumphant laughter rises from the corner. It looks like he's done it again. Mark Turmell has heisted another case of Oly. The artesians are running wild tonight.

Beer Run, Turmell's latest effort, has a hard act to follow in his first game, *Sneakers*.

"That was a fun game to write," Turmell says of *Sneakers*. He had to analyze other games in order to figure out how to program this one. Once he'd worked these things out, it was easy. Coming up with the dome-shaped men who descend upon players like dainty ballerinas, only to stomp them to death with their tennis shoes, was a little more difficult.

In creating *Sneakers*, Turmell was inspired by various arcade games. He gained insight from such favorites as *Astro Blaster* and *Gorf*. He liked the thought of having a meteor shower and having twenty different objects in one game.

"I got the germ of the idea from *Gorf*—which is frog spelled backward. I liked the idea of an object with feet and eyes and a big smile. So I made these round characters with funny smiles, but originally the legs didn't move. It was all right. Then I gave them moving legs and made them come down and stomp—it was great."

Sneaking into the Spotlight. When he'd finished *Sneakers*, Turmell called Sirius Software in Sacramento, California, to see if they were interested in it. After a couple of conversations with Turmell, Sirius flew him to Sacramento.

Released by Sirius last July, *Sneakers* was an instant success. By the end of September, orders had been placed for half of the twenty thousand copies produced. By October, *Sneakers* had gone from nowhere to number nine in the *Softalk* Top Thirty. According to Jim Ackerman of Sirius, sales of *Sneakers* have not yet peaked. "I think *Sneakers* is going to be another *Space Eggs*," says Ackerman.

You can bet the night Turmell completed *Beer Run* wasn't the first time he'd stayed up all night in front of his Apple. He's been burning the candle at both ends ever since he started college—at age fifteen.

College Follows High School—Daily. Turmell has always been ahead of himself. As a sophomore at Western High School in Bay City, Michigan, he began taking computer courses at nearby Delta College because his high school offered no courses in computers. Besides, high school didn't challenge him.

Situated near the edge of Lake Huron's Saginaw Bay in the middle of Michigan's cropland, the two-year college serves the three cities of Saginaw, Midland, and Bay City. Before long, Turmell was spending more time in the Delta College computer room than he was at high school. While in college he taught and worked for the school. You just can't keep a good fanatic down.

Turmell's biggest challenge at Delta was adjusting to college life as a fifteen-year-old. The transition wasn't easy. "I was a little timid at first," Turmell says. "Everybody could tell how young I was." But Turmell took to the computer classes with such vigor that other students soon forgot about his age.

Marjorie Leeson, a data processing professor at the college, became aware almost immediately of Turmell's enthusiasm and talent. "She took me under her wing and helped make things easier," says Turmell. It was recognition of his ability that drew Leeson's attention to the young man. Soon Turmell was writing programs and teaching other students how to work out their computer problems.

During his last three years of high school, Turmell spent the morning in classes at Western, arriving at the Delta campus at two in the afternoon to begin his five classes in data processing. Remembering his habit of working in the computer room of Delta College's H-wing until 10:30 at

night and beyond, Marjorie Leeson comments, "The only problem with having someone like Mark in class is that they never want to go home."

Leeson was the first faculty member to have Turmell in class at Delta. She recalls with a twinge of amusement the difficulty her student had when he tried to check tapes out of the college library. Even though the tapes were required for class assignments, the librarian would not allow him to have them. A driver's license is required for checking out tapes, and Turmell didn't have one. When he told this to the librarian, she insisted he must go out and get one.

"Mark didn't like to tell people he was fifteen," explains Leeson. "I finally had to write him a note so he could check out the tapes."

The peculiar gleam in Turmell's eye was first observed when a friend's father, also a Delta College instructor, brought home a terminal that tied into Delta's central computer. Turmell and a friend logged on and played football, battleship, and other games. Although he took to computer games right away, Turmell recalls that it wasn't much fun playing with a text printout instead of a video display.

How Turmell Broke into His First Job. Turmell purchased his first computer, an Apple, at seventeen. He scraped together some of the money by mowing lawns and borrowed the rest from his parents.

His investment in the Apple and his late nights in the computer room soon began to pay off. He became such a proficient programmer that he landed a job at Delta that would earn him enough money to pay back the loan from his parents. "They had to hire me," Turmell recalls with a chuckle. "I figured out how to break the code to their computer system."

The system contains administrative data, payroll information, and student files. "I just wanted to see if I could do it," Turmell says, innocently.

Delta College figured that, if they couldn't beat him, they'd recruit him. The data processing manager hired Turmell as a system programmer; his job was to write programs, keep the system running, and develop a code that could not be broken by ambitious student programmers.

During the summer, Turmell tried his hand at teaching—at camp. The same year he graduated from high school, he taught computer courses at a computer camp organized by Leeson. Cosponsored by the Saginaw Township Schools and Delta College, the camp focused on the impact computers are having on everyday life.

By the time he took home his high school diploma, Turmell had already completed all the data processing classes Delta College offered.

It was while he was working at Delta that Turmell began writing *Sneakers*; he showed the game to the other programmers for feedback. They gave him suggestions for improvement and even drew sketches for him to try out. They suggested that he add some explosion sequences to the game; Turmell is especially proud of the way the H-wings blow up.

Taking It Easy. At the suggestion of his co-workers, Turmell toned down the game a little. The original version was too hard for the uninitiated to play. "I had more rocks in the field of meteors and I had the H-wings (whose name is taken from the building that houses the Delta College computer room) firing at you.

You can't tell how hard a game is when you're programming it," Turmell explains. "I was playing *Sneakers* every day just to test it out, so I was pretty good. It seemed easy to me."

His fellow programmers at Delta also helped him put the nine screens of *Sneakers* in order from easiest to hardest. "The scrub screen is my favorite," says Turmell. "It was the first screen that seemed to have the feeling of an arcade game."

By the way, Turmell claims top score for *Sneakers* with 93,000, but then he has an advantage. He says he heard of someone in Detroit who scored 65,000, but he can't remember the player's name.

Spicing Up Software. Turmell is becoming known around town for his computer expertise, and local newspapers have picked up on the story. When Spicer Engineering, a local firm assigned to lay out new sections of sewer in Bay City, bought a canned software package, they turned to Turmell for help in debugging it.

"They had to lay out sewers and calculate exactly where to place pumps in the system," Turmell explains. "They were using a map of the city and they needed to calculate flow volumes to figure out where the pumps had to go." Turmell cleaned up the program and, at seventeen, began a new career—consulting.

On his first assignment, Turmell developed a program for inventory for a quick-print shop in the area. Next, he wrote a program for payroll and another to do bidding estimates for a construction company.

His involvement with the computer keeps Turmell busy, but he is a man of many interests. At Ferris College in Big Rapids where he's now a junior, Turmell has a double major in business administration and computer science. Family and friends command his interest as well. Each weekend, he makes the 120-mile drive from Muskegon River to the Bay City to spend time in his home town. "I've always been one to drop anything and go out and do something with my friends," says Turmell.

Basketball's Loss. For fun, Turmell plays racquetball, tennis, and basketball, in addition to playing with his computer. "I live and die sports," he says. In fact, at six feet four inches and still growing, he probably could have made a career out of basketball.

He did give basketball a shot as a sophomore in high school, but the practice requirements—two hours a day—conflicted with the courses he was taking at Delta. It was either basketball or data processing. Although he still enjoys following both pro and college basketball, it didn't take Turmell long to establish his priorities.

Turmell says he didn't study much in high school. It took too much time away from sports and programming.

Instead of cracking the books during the evening hours, Turmell looks for a good poker game. But his real vice overtakes him late at night when he gets that gleam in his eye and feels a compulsive urge to sit in front of a computer terminal.

Lately he's been obsessed with artesians, the bubbly little creatures that allegedly populate his new game, *Beer Run*.

When he started talking about grasping for the rope lowered from a blimp that would whisk him to the top of the Olympia beer building, his parents began to wonder. One night, he was heard mumbling in his sleep about being chased by guzzlers and bouncers.

Although Turmell had been working on the game itself for a while, it was only a couple of months ago that he added the pursuit of beer to the concept. "I decided to call one group of creatures artesians, like the artesian brewing water—I don't know why—it just sounded good."

At this point the game was a climbing game, in the vein of *Apple Panic*. When it was nearly finished, Turmell showed it to the folks at Sirius for suggestions. The artesian creatures inspired Jerry Jewell with the thought of making the game into a beer run. Sirius contacted the Olympia Brewing Company and asked if the Oly logo could be used in the game. Olympia loved the idea, and, by this time, artesians should be bubbling away in some Apples.

Wisdom from an Old Timer. Turmell offers would-be programmers some encouragement. "Don't ever get stopped because of a problem in your program. The main problem is usually with speed. If you keep digging, there's always a way to work the program out."

The poor quality of some of the games on the market disturbs Turmell. "I think quality is really important. A lot of the stuff out there just uses high quality graphics to sell, but the programs aren't very good." More critical game reviews that give readers "fair warning" are a step in the right direction.

One thing Turmell finds especially offensive in game programs is flickering. "You have to do a lot of testing. I have specialized routines for testing. It's important that images on the screen pass through each other well. That worked out very well in *Beer Run*. I would never have believed you could pull off some of the high-speed stuff that's being done today on the Apple.

"Still," says Turmell, "there are too many get-rich-quick programs on the market. Sometimes I wonder why I'm sitting here struggling for months to get depth in a program when I could market it right now and make money. I guess I'm a perfectionist."

Turmell doesn't play a lot of games on the Apple these days. He's just too busy creating games to get serious about playing them. But he sees games as an excellent way to become familiar with the computer.

He does take time out occasionally to visit the arcades. "I like the break-out games and space games." His favorites are *Asteroids*, *Galaxia*, *Scramble*, and *Froggers*.

The Roads Not Taken—Yet. At nineteen, it must be hard for Mark Turmell to decide what to do. With prospects for consulting work, a job offer in California, and a year of college ahead, he has a lot to look forward to. He's already decided not to become a guzzler or a bouncer.

But with the drive and enthusiasm that has characterized his past, he will have no problem breaking into any field he chooses. And if you discover someday that Turmell has been hired as a programmer for the Chase Manhattan Bank, you will have cause to be very, very suspicious. SL

ANIX 1.0

by Randy Hyde

THE MOST POWERFUL SOFTWARE TOOL $49.95 WILL BUY

Time is money. But if you're faced with a deadline, all the money in the world can't buy you a few extra days to complete your project. Therefore, it makes a lot of sense to use your time as wisely as possible. And having the proper tools for the job is the first step towards insuring maximum utilization of your time.

Lazer MicroSystems is proud to announce ANIX 1.0, one of the best software tools money can buy. With many of the features that has made Bell Labs' UNIX operating system so popular, ANIX is a flexible, easy to learn and use operating system that can read and write DOS 3.3 files. Existing files and databases are not obsoleted by ANIX, only complimented.

An operating system without application programs is useless. Supplied on the disk with ANIX are over 30 useful utility programs. The utilities alone are worth far more than the cost of the ANIX package. Some of the utilities provided with the ANIX package include: ADU, DUPDSK, PUTDOS, TYPE, PRINT, STAT, DREN, FREE 2, UNDEL, MAKE, SIZE, WORD COUNT (WC), LINE COUNT (LC), BUILD, ED, INIT, MAKEAUTO, BTOT, and many more!

ANIX 1.0 was written by Randy Hyde, the author of LISA, SPEED/ASM, DISASM/65, TRACE/65, DOSOURCE 3.3, portions of Apple PIE, and several other professional software tools. ANIX is, by far, one of his best efforts to date. For advanced 6502 programmers the source listings of ANIX and all the utilities are also available.

ANIX 1.0 features:

- DOS 3.3 file structure compatable
- UNIX-like console command processor
- I/O can be redirected to a printer, slot, or disk file.
- Over 30 useful utilities and files.
- Smaller than DOS 3.3.

A DOS 3.3-Compatable Operating System with the flavor of UNIX*

Lazer MICRO SYSTEMS INC.

1791-G Capital
Corona, CA 91720
(714)735-1041

Trademarks

ANIX Lazer MicroSystems, Inc.
*UNIX Bell Labs
Apple Apple Computer, Inc.

Lazer is famous for high quality software and hardware

Call or write for our free catalog

NEW FOR THE APPLE II

Gameline Reviews

Bezman

By John Beznard. Bez (Irvine, CA).

Seen any good dots lately? If not, may we suggest a couple rounds of *Bezman*?

Collect the dots in the maze and you score points, but steer clear of the beasties who guard the dots. When you've collected all the dots on the first level, go to the second. More dots, and more beasties who'll pursue you.

Take a big dot—there are three in each maze—and you're empowered, temporarily, to do in the beasties as well. They're delicious, and worth additional points. Be careful, though. The beasties keep coming back and, if your evasive tactics are not successful, they'll get you instead.

A Bezman moves faster than his adversaries, making it possible for you to devise various strategies by which to avoid being killed. And the game itself differs from others you've played. There are three different mazes to maneuver your way through, and as you progress to higher levels, both the speed of the game and the number of beasties you must contend with increases. There are at least twenty-six levels to the game, and after that, who's counting?

As the pace speeds up, the beasties seem to grow more clever, more voracious, and more determined. Odds are, though, if you've made it this far, you're a worthy adversary. Besides, once you start playing this game, you'll probably discover that you have an inexplicable, but persistent craving for dots.

Eat hearty. JEV

Bezman will run on the Apple II, Apple II Plus, and Apple III. Requires 48K and one disk drive. $22.95.

The Game Show

By Geoff Zawolkow, Pete Rowe, and Ted Perry. Computer Advanced Ideas (Berkeley, CA).

Everyone's had the experience of watching a television game show and thinking, "I could do better than that turkey." Well, now, here's your chance. *The Game Show* comes complete with enthusiastic contestants, killer questions, and a diplomatic emcee, and it's fun to play even though there are no glamorous prizes to win.

Some winning graphics distinguish *The Game Show*, including your teammates Joe and May, who move their lips, smile, and throw up their arms in triumph at the proper moments. Your partner gives a clue and it's up to you to get the answer right. The categories available range from computer terms to biology to nursery rhymes to advanced vocabulary. You select a category and the emcee instructs either Joe or May to give the first clue.

With the given categories, *The Game Show* is limited to being a learning tool for children rather than a pastime for adults, although the advanced vocabulary category gets pretty sticky, with words like *pelf*, *moil*, and *yare* (check your dictionary). The biology section is also challenging—unless, of course, you happen to have paid close attention in high school science. If not, *The Game Show* may inspire you to brush up on lipase and systolic pressure.

A feature designed to enable teachers to create puzzles appropriate to their students and subjects also makes it possible for password lovers to build some really good games. It's called "build or change a subject area," and it takes you step-by-step into creating your own subject categories or rewriting existing categories to provide better clues.

Only one thing's missing in *The Game Show*—the applause. This is a serious flaw—after all, how realistic can a game show be without applause? *The Game Show* is for one or two players, and would make a fitting Christmas present for television viewers who long to test their mettle as game show contestants. MM

The Game Show runs on the Apple II, Apple II Plus, and Apple III. Requires 48K and one disk drive. $39.00.

Bug Attack

By Jim Nitchals. Cavalier Computer (Del Mar, CA).

To some of us, the tune is a Civil War song, "When Johnny Comes Marching Home Again," and it has little relevance to a home-arcade game about insects. But to the majority of the Apple community under twenty-five, the words to the tune go, "The ants go marching one-by-one, hurrah, hurrah . . ."—and this has all sorts of relevance to *Bug Attack*.

In fact, the tune plays over and over every time you reach a new level, because your first prey is an army of bright colored, funny, deadly ants waddling through a cactus field. The cacti protect the ants but not you, and each ant you kill turns into another cactus to block your shots from the rest of these pink pests. Once your personal bug has dispensed with the ants, and with the daggers that the ants attempt to drop on you, you are finished with the song for that round.

Now you're treated to a bout with a relentless worm. Each segment of the worm is capable of independence, so you have to take on every segment before the creature winds itself down to you. Worms enjoy dropping daggers even more than ants do, and if they reach the bottom, they'll gobble you up. But the worm has nothing on the medflies.

The medflies enter in formation and maintain an orderly flutter around the screen two or three times; if you haven't significantly depleted their number by then, watch out. Singly or in groups of two or three they break formation, make for a spot nearly above you, and flitter and flutter madly among the cacti, all the while dropping daggers in arhythmic swarms.

When you get hit, you lose a bug and have the pleasure of fighting again from scratch the group that just got you.

Just to keep it all from being a snap, you're dependent on fuel. If your fuel runs out, you lose a ship.

Should you make it through this sequence three times, you earn a reward: a cartoon interval, during which you can sit back and watch without peril. The ant song accompanies the cartoon, but it changes at the climax to the 1812 Overture.

Now do it all over again, except that cacti now are clovers. If your score really soars, you'll see the clovers replaced by flowers.

Still to come in the upper reaches are two more configurations of pest; introduction of new elements even at high

levels adds greatly to the enjoyment of home-arcade games.

Hi-res graphics in *Bug Attack* are richly colorful, with bright pinks and greens, oranges and blues. They have a happy, laughing cartoony quality that will make you smile. Everything is accompanied by well done sound effects. MCT

Bug Attack runs on the Apple II, Apple II Plus, and Apple III. Requires 48K and one disk drive. $29.95.

Crossword Magic

By Larry Sherman. L & S Computerware (Sunnyvale, CA).

Creating a crossword puzzle is a breeze with *Crossword Magic*. Both school and home users are likely to find this program useful, especially if their aim is to make learning more fun.

The program will not compose clues for you, nor will it help you think of words that fit. What it will do—and admirably—is relieve you of the hassle of drawing grids and trying to fit into them the words you specify. This frees you to concentrate on the more creative tasks of thinking of words you want to include in your puzzle and coming up with clever, appropriate clues.

The package consists of a maker disk, on which crossword puzzle grids of from three to twenty squares can be created; a player disk, on which those same puzzles can be played; and documentation. Puzzles can be saved to disk to be played on the computer or printed out for use in the classroom or at home.

If you attempt to solve the puzzle on the computer, the program tracks the errors, but doesn't interrupt you during the solving process. Anything goes, right or not, until you're ready to quit. Then you can choose to have the puzzle with correct answers filled in displayed, along with the number of errors made in solving it. You can also review the clues and match them with the correct answers.

When you propose a word for inclusion in a puzzle, the program puts it in immediately if possible; otherwise, it adds the word to a list from which it will draw later on as opportunities open up.

You can look over the list whenever you like during composition. If you're brainstorming and haven't decided whether you really want to use a word in your puzzle, you may want to keep a pad and pencil handy for listing the possibilities, since once you've suggested a word, it will either go in the puzzle (if it fits) or onto the list. A feature by which you could ask the program to disregard a suggested entry would be a nice enhancement to future versions.

When you're solving a puzzle, you have the option of correcting or changing your answers. If you make a typo or spell a word incorrectly while creating a puzzle, however, the program won't recognize your error. An edit option that would allow you to correct errors you catch yourself making would be another useful addition to an already fine program.

Crossword Magic will print excellent quality printouts on the Silentype and on Epsons with graphics capability. The puzzles you create won't look exactly like the kind found in newspapers and crossword puzzle books—they're likely to have more black squares than you're used to seeing and not all letters will cross. But this is not at all a disadvantage, just a difference. And when you're creating a puzzle containing words that relate to a particular topic or subject area, it's nice not to have to take a lot of time trying to think of unrelated words to fill in the gaps. JEV

Crossword Magic runs on the Apple II, Apple II Plus, and Apple III. Requires 48K and one disk drive. $89.95.

Colorblind

By Bob Johnston and Al Iapicca, Marin Data Systems. Energy Games/Applied Information Dynamics (San Francisco, CA).

Remember—or did you ever hear about—*House of Wax* or *Bwana Devil*? They were two of a small spate of three-dimensional movies in the early fifties. *House of Wax* was the masterpiece: an excellent creepy Vincent Price horror flick with corpses sitting up and cloaked monsters frequenting dark alleys looking for people to feed into cauldrons of boiling wax. It all became many times scarier when the corpse sat up toward you in all three-dimensions. At that moment, the theater was one massive scream.

Then why aren't there plenty of three-dee movies today? It's the glasses. With your ticket to *House of Wax* you were given a pair of cardboard-framed spectacles with one green lens and one red. These quite uncomfortable contraptions had to be endured in order to see the film in three-dee. It wasn't worth it.

It still isn't. Energy Games has a super idea in *Colorblind*, that being to render players blind to the activities of their opponents except when they can legitimately see each other. And Johnston and Iapicca are to be commended for coming up with a way of achieving this effect that works—and it does work.

But the method is to provide two pairs of cardboard framed goggles, one red, one blue, for the respective players. These goggles are significantly more stable and even more uncomfortable than the three-dee glasses of the fifties. But, just as three-dee movies were worth a try, so is *Colorblind*.

The game is no great shakes, but it's only a vehicle for the concept, and the concept is super. It would be great if inventors were not so often commercially ignorant. In this case, a look at the marketplace might have driven home the point that any game really needs at least to approach state-of-the-art in order to sell. MCT

Colorblind runs on the Apple II, Apple II Plus, and Apple III. Requires 48K and one disk drive. $34.95.

SCRAM

Atari (Sunnyvale, CA).

Scram presents a simulation of a nuclear power facility, modeled after the Three Mile Island Nuclear Power Plant Unit 2. The various mechanisms of the plant appear in brilliant blues, reds, greens, and yellows. As you might imagine, the game is considerably more difficult to play in black and white than in color; the valve settings and water circulation levels are harder to see.

After studying the Environmental Impact Report and Final Safety Analysis Report, the Nuclear Regulatory Commission steps in to issue a safety license to the plant. Players with a vehement dislike for government red tape will get a vicarious thrill here; the entire licensing procedure takes only five seconds, hardly leaving protestors time to paint their signs.

The major components of the pressurized water reactor are the reactor core, generator, hyperbolic cooling tower, and three circulating water loops. These are interconnected by a series of pumps and valves. Temperatures at seven major sites are displayed, and the net energy of the system (your score) is indicated at the lower right of the screen.

Left to its own devices, the station, as in reality, will produce megawatt hours of energy at a uniform rate. This would not make for a particularly interesting or eventful game, so Nature steps in to liven things up a bit, providing earthquakes in the form of loud rumbling noises from the audio and tremors in the visual display. Every quake breaks one pump or valve which must be repaired before disaster strikes . . . or rather, melts down. Determining which valve or pump needs repair calls for careful attention to the temperature gauges and testing of all systems to be sure each performs its specified function.

A work force of eighty men stands ready to help. After a break has been located, five men are sent to repair it. Having suffered their maximum allowable exposure to radiation, these men will never return. In locating the trouble spot, you're allowed a total of sixteen guesses, right or wrong; a wrong guess uses men just as irreversibly. After these have been used up, there is no way to repair a break.

The object of *Scram* is to produce as much net energy as possible while repairing all breaks. A successful end to the game is achieved with a cold shutdown, accomplished by dropping the control rods into the reactor core. Be forewarned, however; it takes approximately five minutes to reduce the reactor temperature from 655 degrees to 200, and earthquakes are not obliging enough to allow an uneventful shutdown.

Scram has nine risk/skill levels, with the higher levels experiencing more frequent earthquakes. Producing five hundred megawatt hours of energy at level nine qualifies the player as a senior reactor operator. Most players will make rapid progress through the first several risk levels. A bar that appears above or below each temperature reading indicates rises or drops in temperature and serves as a handy visual aid. When speed is important, the bars appear much faster than degree changes.

There are also two very important pumps which, when inactive, rapidly produce a condition known as steam voiding (very serious), and from there, a quick meltdown. It would obviously be quite helpful to the novice to know the locations of these two pumps ahead of time . . . but why interfere with the thrill of discovery?

The *Scram* instruction manual is needlessly wordy and confusing. The author (whom Atari keeps anonymous) has an annoying habit of using too many abbreviations in too short a time. Only a reader with perfect retention will be able to avoid flipping back and forth, while struggling to recall the meanings of PWR, RCS, HPI, and LOCA. Understanding is desirable, but such a level of comprehension is not essential to a successful run at the game. Since the game itself is a series of cause/effect steps, it's not necessary to remember what a valve is called to know when it needs to be closed. DB

Scram will run on either the Atari 400 or 800 computer. Tape only. $19.95. SL

Still Pending: Micros versus Arcades— Playing for Keeps

Who's ripping off whom? Or is everyone a total innocent? The real questions are, what right does a software author have to his work under law, and what remedies can an author and/or publisher pursue when these rights are abrogated? What about the end users? Are the prices users pay for entertainment software outrageous, or are they truly reflective of real value?

When companies collide over software rights and, in particular, when manufacturers of coin-operated equipment press their claims to software against personal computer software publishers, who's right, who's wrong, and who, besides the public deprived of a desirable product, suffers?

Why does Atari feel it's necessary to spend scarce advertising dollars in personal computer magazines to assert their right to programs they already own? Who's going to court with whom and to what end?

As these questions indicate, all's not exactly well when it comes to the writing and publishing of entertainment software. As the stakes get higher, and good gaming ideas become harder to conceive, companies seem to be turning on each other rather than concentrating on the marketplace.

Some of the very software publishers who scream the loudest about piracy by individual microcomputer owners may have been guilty themselves of borrowing gaming concepts from the manufacturers of coin-operated arcades.

On the other hand, what about people who maintain that it is their right to break protection codes and distribute software? Are they any better than common thieves? Are user groups that trade unlocked software violating federal law by conspiring to defraud the original software publishers?

The legal and ethical constructs that obtain in the world of microcomputer software publishing will be examined in detail in our January issue. The motivations of all members of the microcomputing community will be subjected to scrutiny. Rights, and wrongs, will be discussed at length, in an attempt to put commonplace activities into perspective.

What are the implications for microcomputer owners who prize their possessions for their entertainment value? Will these brouhahas be resolved in a fashion that will lessen the choices in the marketplace, or will the fallout take the shape of new entertainment forms that have less emphasis on arcade-style games?

This will be your chance to find out why some of your favorite games are no longer being sold, why others may never see the light of day, and what impact the current controversy is having on the microcomputer game programming industry. SL

High Scores

Game	Publisher	Player	Score
ABM	Muse	Chuck Jon and Joe Wilson, Sunnyvale, CA	24,500
		Wayland Lim, Cupertino, CA	31,900
Alien Rain	Broderbund Software	Alan Lee, Brighton, MA	17,735,500
Alien Typhoon	Broderbund Software	Robert Young, South Pasadena, CA	67,870
Apple Galaxian	Broderbund Software	Brian Wilson, Sunnyvale, CA	41,360
Apple Panic	Broderbund Software	Richard Smith, San Jose, CA	412,001
Asteroid Field	Cavalier Computer	Gary Aulfinger, Carlisle, PA	174,310
Asteroids in Space	Quality Software	Mark Adams, Salt Lake City, UT	319,315
Autobahn	Sirius Software	Don Smith, Endwell, NY	244 miles
Bezman	Bez	Al Tommervik, North Hollywood, CA	40,171
Bug Attack	Cavalier Computer	Jim Nitchals, program author	60,000
		Mary Taylor Rollo, North Hollywood, CA	29,258
Epoch	Sirius Software	David Boyle, Binghamton, NY	73,660
Falcons	Piccadilly Software	Lisa Poritz, Endwell, NY	25,200
Galaxy Wars	Broderbund Software	Mike McConnell, Endicott, NY	23,250
Gamma Goblins	Sirius Software	Dick Nitto, Binghamton, NY	12,000
Gobbler	On-Line Systems	Dick Nitto, Binghamton, NY	147,710
Gorgon	Sirius Software	Albert Ting, Wilmette, IL	25,000
Head-On	California Pacific	Brian Wilson, Sunnyvale, CA	38,230
Missile Defense	On-Line Systems	Robert Young, South Pasadena, CA	149,080
Mission Escape	CE Software	Donald Brown, Des Moines, IA	2,184
Olympic Decathlon	Microsoft	Eric Casler, Essex Junction, VT	8,905
Orbitron	Sirius Software	Dick Nitto, Binghamton, NY	23,700
Pegasus II	On-Line Systems	Robert Young, South Pasadena, CA	32,020
Pulsar II	Sirius Software	Jon David Nitto, Binghamton, NY	27,540
Raster Blaster	BudgeCo	Chris Reed, St. Louis, MO	2,243,000
Sabotage	On-Line Systems	Steve Wozniak, Scott's Valley, CA	13,678
Sneakers	Sirius Software	Mark Turmell, program author	93,000
		Dick Nitto, Binghamton, NY	42,131
Snoggle	Broderbund Software	Jun Wada, program author	Level 28
		Ron Flickinger, Ft. Wayne, IN	Level 15
Space Eggs	Sirius Software	Mark Adams, Salt Lake City, UT	11,445
Space Quarks	Broderbund Software	Dick Nitto, Binghamton, NY	3,120
Star Thief	Cavalier Computer	Jim Nitchals, program author	13,900
		Dick Nitto, Binghamton, NY	13,200
		Randy Antler, Del Mar, CA and Bill Emerick, La Mesa, CA (partners)	23,480
Stellar Invaders	Apple Computer, Inc.	Kenneth T. Lim, Cupertino, CA	turned over twice
Super Invaders	Creative Software	Matt Fisher, Apalachin, NY	13,100
Threshold	On-Line Systems	Warren Schwader, program author	419,000
		Gary Kevorkian, Inglewood, CA	136,000
Wormwall	Sirius Software	Pam Nitto, Binghamton, NY	34,684

Do you have a high score on your favorite game? If so write to let us know so we can publish your name and score in the next issue of Softline.

ON-LINE SYSTEMS introduces arcade gaming as an art form. THRESHOLD, by WARREN SCHWADER and KEN WILLIAMS, features fast smooth animation, HI-RES graphics, and more challenge than you'll find in any other arcade game on the market.

THRESHOLD is an arcade game with alien attackers galore. In fact, there are more monsters out there than we expect you'll ever see. THRESHOLD was designed to be an arcade game that you won't get bored of, and that means a constantly changing game with a graduated skill level, but more than that, THRESHOLD means constant fun.

THRESHOLD runs on any 48K Apple II or II Plus DOS 3.2 or 3.3. Available now for $39.95 on disk from your local computer store or you may order directly from. . . . .

ON-LINE systems

36575 Mudge Ranch Road - Coarsegold, CA 93614 - 209-683-6858

ADD $1.00 FOR DIRECT ORDERS

Visa, Master Charge, C.O.D. or Checks Accepted

HUNTINGTON COMPUTING

ONE OF THE WORLD'S LARGEST INVENTORIES

Limited Time Only

Visidex List 200.00 **Now 139.00**

(Please mention this ad when ordering) ***Expires Dec. 31, 1981***

Personal Software, Brand New, Shrink Wrapped. Immediate Delivery. $2.00 Shipping (for total Software order)
No Hidden Charges. No Gimmicks.

Also
Atari®
Pet®
TRS-80®

Huge Inventory

The World's Most Remote Computer Store! This is the Huntington Computing gang in the cotton field across the street from our 3300 square foot store in Corcoran, CA. We may be in the country but we believe we have the world's largest selection of microcomputer software. And, we accept School Purchase Orders.

SPECIALS

Visicalc List 200.00 ..... Now **$149.00**
Pegasus II List $29.95 ..... NOW **$22.99**
S-C Assembler II List $55.00 ..... NOW **$46.69**
Microsoft Z-80 Card List $395 ..... NOW **$299.00**
D.C. Hayes Micromodem II List $375 NOW **$299.00**

NEC 12" Green/Black Monitor
List $285, NOW $239.00
Andromeda or Microsoft 16K Expansion Board
List $195, NOW $169.00
All Educational Software
15% off list

Call Toll-Free 800-344-4111 (Outside California)

HUNTINGTON COMPUTING, Dept. MC-1
Post Office Box 787
Corcoran, California 93212

Order by Phone 800-344-4111
In California (209) 992-5411

SUPER DISCOUNTS
apple SOFTWARE

VISA
master charge

We take MasterCard or VISA (Include card # and expiration date). California residents add 6% tax. Include $2.00 for postage. Foreign and hardware extra. Send for free catalog. Prices subject to change.