Vol. 1, Issue 5

Price $2.25

ROM

The Magazine That Brings The ATARI™ Computer to Life!

SPACE STATION Z64

WHY BUY AN ATARI

ARTI HAROUTUNIAN INTERVIEW

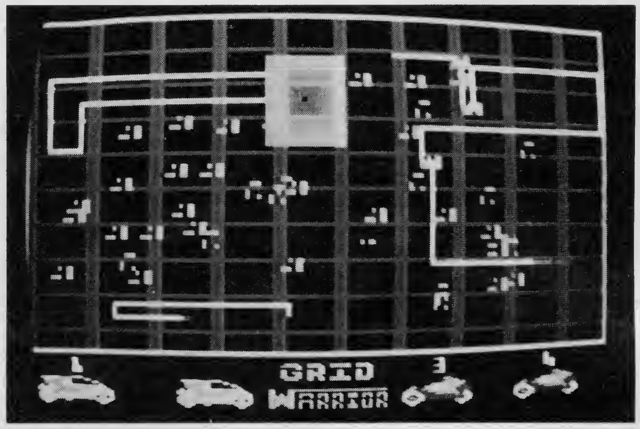
FREE ARCADE GAME INSIDE

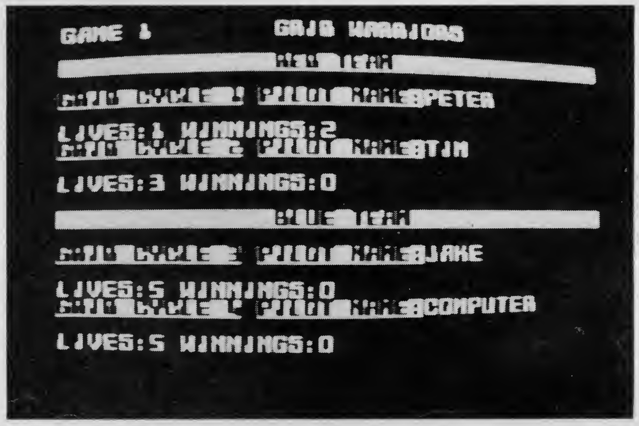
ROM SOFTWARE PROUDLY PRESENTS

GRID WARRIOR

It is the 23rd Century and Maracus Luxor, ruler of Earth, has created the deadliest game known to man. This game called "Grid Warrior" was made in order to eliminate the convicted prisoners and also to entertain the Earth's rulers. Four prisoners are chosen to battle it out on the grid.There are two teams: The red team and the blue team, each having two cycle riders per team, battling it out until death. Shift gears from low to high or vice versa to make turns or trap your enemy. Race your light cycles through a maze of debris or without any obstacles at all. You make the choice, for you're in control.

* 1 to 4 players simultaneously.
* Computer controlled Cycles.
* Twelve different levels of play.
* 100% Machine Language.
* Fast action arcade-like graphics.
* Very Imaginative Sound.



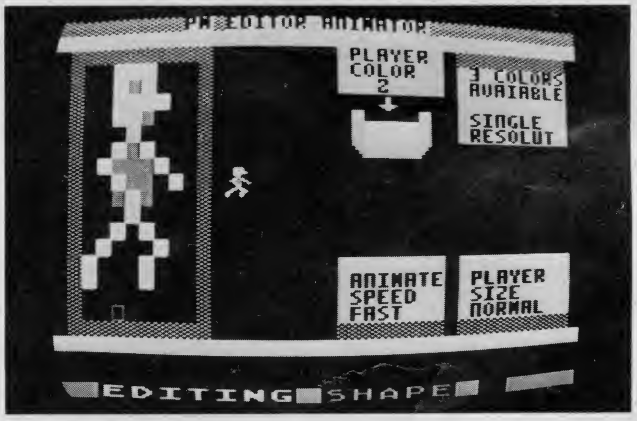


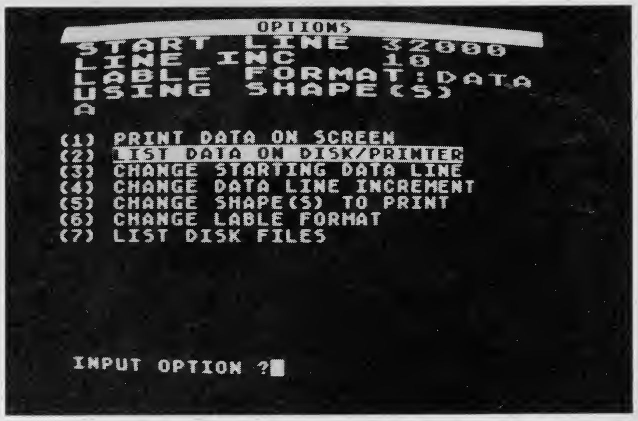
```
GAME 1              GRID WARRIORS
               RED TEAM
GRID CYCLE 1 PILOT NAME:PETER
LIVES:1 WINNINGS:2
GRID CYCLE 2 PILOT NAME:TIM
LIVES:3 WINNINGS:0
               BLUE TEAM
GRID CYCLE 3 PILOT NAME:JAKE
LIVES:5 WINNINGS:0
GRID CYCLE 4 PILOT NAME:COMPUTER
LIVES:5 WINNINGS:0
```

PM/EDITOR/ANIMATOR

The ULTIMATE tool for editing and animating player/missiles on the the ATARI.

* Simple ONE key commands.
* Three color players made easily.
* Animate 2-26 different players.
* Six different animating speeds.
* Print Data to Printer, Disk, or screen.
* Lots of animated player demonstrations.
* For the BASIC and Assembly language programmer.
* A must for any serious game programmer!





```
               OPTIONS
START LINE      32000
LINE INC        10
LABLE FORMAT:DATA
USING SHAPE(S)
A

(1) PRINT DATA ON SCREEN
(2) LIST DATA ON DISK/PRINTER
(3) CHANGE STARTING DATA LINE
(4) CHANGE DATA LINE INCREMENT
(5) CHANGE SHAPE(S) TO PRINT
(6) CHANGE LABLE FORMAT
(7) LIST DISK FILES



INPUT OPTION ?
```

GRID WARRIOR requires 48K and DISK Drive.(Joystick)-$24.95 US
PM/EDITOR/ANIMATOR requires 32K and Disk Drive.(Joystick)-$24.95 US
Both are available from:
ROM SOFTWARE
P.O. BOX 252
MAPLE RIDGE,B.C.
V2X 7G1 CANADA
(604)462-9177 or (604)462-9128
* Dealer Inquiries Invited *

ROM

VOLUME 1,ISSUE 5

FEATURE/ARTICLES

PROGRAMS

COLUMNS

ROM
STAFF

Editor
Peter Ellison

Publishers
Peter Ellison
Bob Cockroft

Art Director
Paul Slevin

Technical Division
Bob Cockroft
Jack Chung

Photographer
Jason Cockroft

Contributors:
Tim Reekie
Geoff Corry
Tom Tran
David Bacon

Domestic Subscriptions:1 yr. 6 issues bimonthly $12; 12 issues $23. 1yr. cassette & magazine $25;1 yr. disk & magazine $40. Send subscription orders to: ROM MAGAZINE, P.O. BOX 252, Maple Ridge,B.C. V2X 7G1 CANADA (604)462-9177 or (604)462-9128

Nothing in this magazine may be reproduced without the written permission of the publisher. Program listings should be in printed form. Articles should be as a type copy in upper and lower case with double spacing. By submitting articles to ROM articles accepted for publication will become the exclusive property of ROM. If article is not accepted, the only way it will be returned is; if a self-addressed envelope is enclosed. PRINTED IN CANADA.

Editorial

Last issue I said that I would make the editorial short, instead I went on for a whole page. This time I won't say a thing and maybe I won't go on for so long.

Piracy, this is one subject that I have been hearing a lot about in the past little while. I don't think piracy is right, but what I think won't changed many peoples' opinions. In my opinion I feel one reason why so much software is being pirated is because the cost of the product. I've heard many arguments by many programmers that say you have to charge forty dollars a copy, or you'll never make any profit. That is very untrue. I think the more that is charged for a program, the more the program will be pirated, instead of being bought. There will probably still be the odd pirate around, wanting a pirated version, but I think all the smaller pirates will probably buy the program, because it is not worth the hassle.(That is if the price is not too high).

One company that has brought their prices down to a reasonable price range is that of GENTRY. They are a division of DATASOFT and have made all of their games priced for under $20. This is quite reasonable when you think the disk, packaging and duplication shouldn't cost more than $2 for each disk. I also hear the argument that, what about all the promotion money that goes into a product. I think if the product is good, hardly any promotion money should have to be put in to it. With so many pirates on the loose, much time has to be spent on making copy protection, instead of spent on making programs. So if you see or hear of anyone pirating please tell them it is themselves that they are ripping off. I feel some programs are worth the $40+ price range but many are not and those are the ones that should think of lowering their prices.

Enough about piracy. This issue of ROM is, I feel, our best yet. We have kept our promise of going full color as you have probably noticed when looking at the cover. Our cover was designed by our photographer Jason Cockroft. He used a piece of carboard to poke holes in the shape of an ATARI symbol. A light was then put behind it to light-up the pin-holes. The symbol was then photographed using a Cokin filter called a "Diffractor Galaxy".

In this issue we have a machine language program called, 'Space Bridge' by Bob Cockroft, that I'm sure you'll all enjoy. Also included in this issue is an interview with 'Arti Haroutunion', author of Kidgrid and Juice. A new column starting up this issue is called 'Making a Game' by Jack Chung, author of Base Hunter from the last issue. We hope this will help both the beginner and advanced become familiar with the programming of a game. The 'War Zone' also has a new name. It will be called, 'Strategy Zone' as it will deal with other games other then war games. We hope that this issue will help you with your programming and will keep you informed on the new things that are coming out. Thanks again for all your letters.

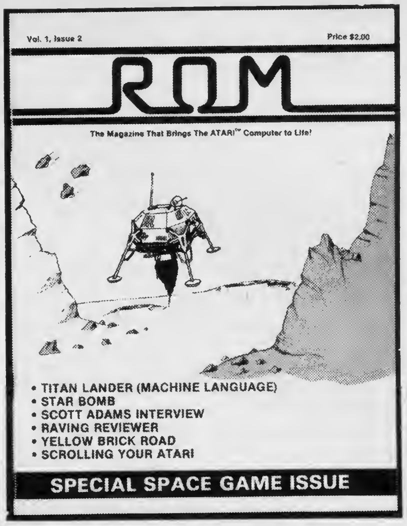
Peter Ellison
Editor

BACK ISSUES NOW AVAILABLE

VOLUME 1,ISSUE 1

* STOCK BROKER
* SCROLLING SHIP
* SATELLITE
* STARTING PAGE
* YELLOW BRICK ROAD

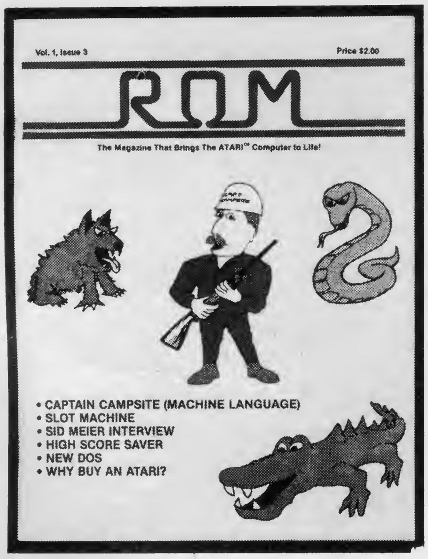
PRICE-$2.50



VOLUME 1,ISSUE 2

* TITAN LANDER(MACHINE LANGUAGE)
* STAR BOMB
* CHART MAKER
* SCOTT ADAMS INTERVIEW
* SCROLLING YOUR ATARI

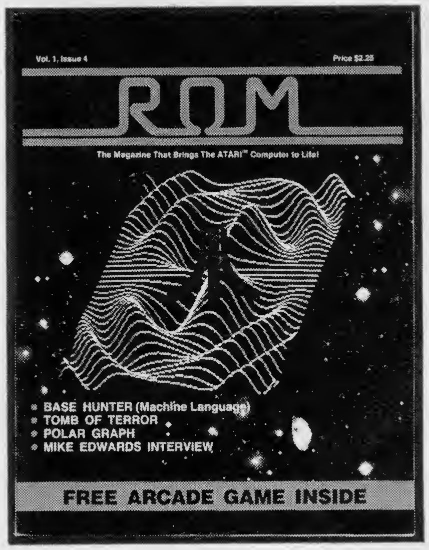
PRICE-$2.50



VOLUME 1,ISSUE 3

* CAPTAIN CAMPSITE(MACHINE LANGUAGE)
* SLOT MACHINE
* HIGH SCORE SAVER
* SID MEIER INTERVIEW
* DISPLAY LIST INTERRUPTS

PRICE-$2.50



VOLUME 1,ISSUE 4

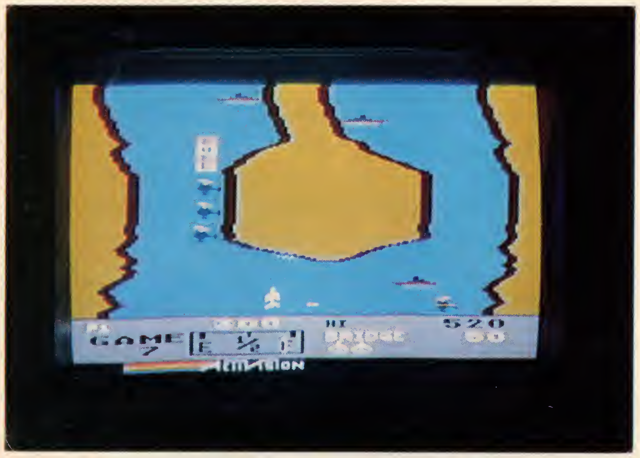
* BASE HUNTER(MACHINE LANGUAGE)
* TOMB OF TERROR
* POLAR GRAPH PLOTTER
* QUICKDOS
* MIKE EDWARDS INTERVIEW

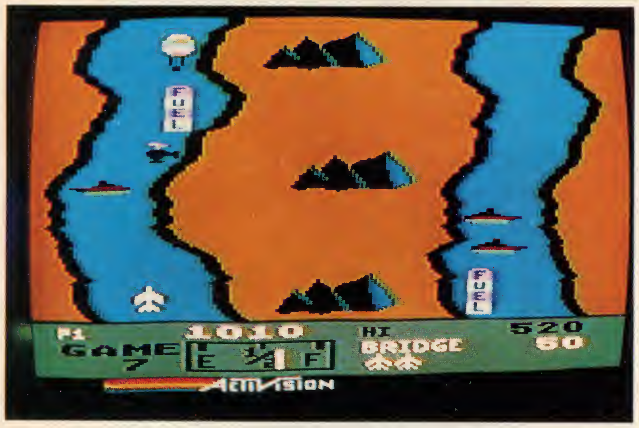
PRICE-$2.50

JAKE THE SOFTWARE DUDE

By JASON COCKROFT

O.K.,so you liked that ad on T.V. about some crazied computer commandos fighting their way up the river. If you like intense, nerve quick, reflex action of combat,(well at least on the computer!), then my friend, you're ready for 'River Raid'. So why don't you sit down with a big bag of pretzels and something cool to drink and I'll tell you all about it.





For the ones that don't know a heck of a lot about River Raid I'll give you a brief rundown. Your goal is to fly up the river canyons and blow the daylights out of everything you see. (It's a real shoot'em up classic.) Your targets may include ships,weather balloons, jets, tanks, bridges, fuel depots and some other goodies I'll just have to keep a secret. But before you get carried away thinking this game was designed for the patience of a two year old, let me tell you about its' challenge and playability.

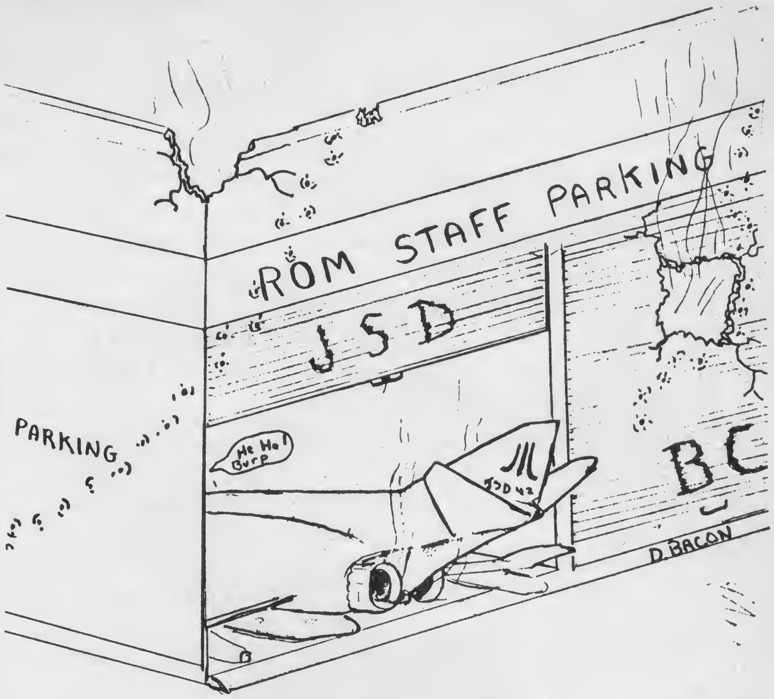
I guess you know by now,this reviewer has always remained a gentle,patient, mature video game sportsman. Some smart guy may comment on the fact that my worst monthly bill is in Joystick Repairs, but since I've bought River Raid my monthly payments have tripled. You see River Raid is a kind of three way struggle.Firstly, as you fight up the river,you are met with increasingly intense opposition in both in quality and quantity. Secondly, as you progress, the river in which you fly in, becomes narrower. Not only that, the channel begins to meander. Margin for error becomes increasingly smaller! Finally, fuel depots, in which you must continually refuel, decrease in their frequency as you fly your gas guzzling jet up river. In my own estimation,I would say these Rivver Raid jets burn as much fuel as my Strato-Chief!This three way battle, as my local Joystick repairman has noticed, combines into an intense, exiting game.

Now that we've all agreed that this is a great game, I suppose ya wanta know a couple of pointers from THE number one master of Video games himself,J.S.D. Well to start with, I guess it would be kinda nice if you had the basic eye/hand coordination down pat before you go on any record breaking performance. Well my advice to you is to start at bridge five and practice shooting the jets that fly across the screen. Secondly, practice refueling,then immediately destroying the given fuel depots. If your a real videosport,you'll appreciate the Extra points award this technique will bring you over the long haul. Finally, as horrid as this may sound, don't be greedy. Even when you start calling yourself a River Raid Pro, your greed to get the maximum kills will al-

J.S.D. cont'd

ways bring you to your end. The secret to this game is Endurance. If you stick to the center and just blow yourself a path through then you'll end up beating your opponent everytime.

To put a capper on this little review,(I'm sorry but I've run out of pretzels), I'd like to say that the boys,(and gals),down at Activision, sure know their stuff. I'd recommend this game to any ex-commando or any VIDIOT!



RIVER RAID

Playability:9
Challenge:9
Graphics:9
Sound:8
Documentation:8
Overall:8.7

P.S. Anyways my top score is 85,410. If you can beat that, I'd like to hear from you. What you can do is drop me a line in care of this magazine, leaving your address or phone number, and I'll get back to you.

Letters

Dear ROM:

I would like to know if it were necessary in programs like "Captain Campsite" (issue 3) or others where you show an assembler listing to type in both? Or is just one sufficient? If not, is the assembler cartridge or macro assembler required? This is the 1st issue of yours that I've gotten, and was very impressed with it.

Richard Rezza
Picataway,NJ

Dear Richard:

This is a question which has been asked by many who have purchased our magazine. All that is needed to be typed in, is the BASIC listing, for this has the machine code inside the program in the form of DATA statements. It is then called in BASIC by the USR function. You could also just ype in the assembler listing, but you may not get all of the sound and scoring, because a lot of this is set-up in BASIC. The assembler listing is mostly given so that you can understand assembly language a little better and incorporate these subroutines into your own programs. The assembler which we use is the SynAssembler from Synapse because we feel it is the fastest and easiest to use for its price. It is available from Synapse Software for $49.95 on disk or a special ROM version for $89.95.

In this issue we have a program that will convert your assembler language listing into DATA statements so that you can put these subroutines into your own BASIC programs. This is the program in which we use for all of our assembler listing programs. Thankyou for your compliment and we hope we can improve with every issue.

Dear ROM:

I enjoyed your magazine and disk No. 4, but I am unable to load BASE.ASM. Is this because I have an ATARI assembler editor and not SynAssembler? The BASE.HUN program called "Base Hunter" is easily beaten if you ram the force shield and go through it. You have a good magazine, but at the beginning of each program you should indicate how much memory is needed, and whether or not a disk drive is required. It would also be nice to have articles or programs for people with small 16K memory.

J.A.Williams
Regina,Sask.

Dear Mr. Williams,

Thankyou for your compliment on issue 4. The answer to your first question is "Yes". You do require the SynAssembler as I said in regards to the first letter. You also said Base Hunter could be easily beaten by passing through the shield. After play testing it, we also have come to the same conclusion. In the next issue there will be an update making your ship explode if it comes in contact with the shield. Your last suggestion is a good From now on we're going to say at the beginning the program how many Bytes(K) is required. Also in this issue we have some games that all will run on 16K computers.

Dear ROM:

I would first like to say how happy I am to see an ATARI supporting magazine published right here in Canada!

The magazine is very well written and very understandable. It is written as if you would be talking right to me. I like that! I never did understand re-defined character sets, until I picked up my copy of Vol. 1, Issue 4 and read it in there.

Can you come up with a type checker similar to other maga-

Continued on next page

zines of your type. I go crazy debugging your programs, especially DATA statements.

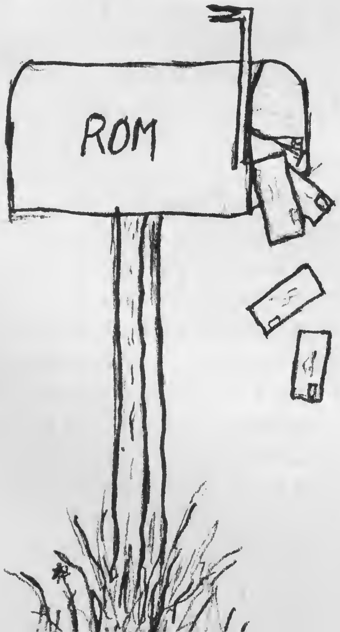
Roger Wein
Hamilton,Ont.

Dear Roger:
I am glad that we could help you with the learning of character graphics. Also we thank you for your compliment about our writing. We try to make our articles as understandable to the reader as possible. In the next issue of ROM we will be printing a Type checker, so it will make typing in our programs a lot easier.

Dear ROM:
Thanks for publishing a GREAT magazine. I picked up issues 3 and 4 one day not knowing what to expect. I was pleasantly surprised at the quality of your magazine and was anxious to get my hands on issues 1 and 2.

Darren Layne
Seweren,NJ

Dear Darren:
I'm glad that you like our publication.We're trying our best to make ROM, the number one ATARI magazine. By the time this issue is on the newstands, issue one reprints will be for sale.



INTERVIEW: ARTI HAROUTUNIAN

Interviewed By PETER ELLISON

Arti Haroutunian works for Tronix,Inc. and has to his credit two excellent original games for the Atari computer. The are 'Kid Grid' and his latest 'Juice'. He has also translated both of them to run on the Commodore 64. We at ROM were able to chat with him while he was at his office.

Q. Arti, when did you first become interested in computer programming?

A. I became interested back in 1978 when I bought a TRS-80 Model I.

Q. Did you go to school for programming or did you learn your programming skills on your own?

A. I have a Masters in Computer Engineering, however most of the microcomputer skills I have learned on my own.

Q. What was the first program that you sold commercially?

A. My first program was a Text Adventure called 'Micro-World' for the TRS-80. It was published in 1980.

Q. When was Tronix started and by whom?

A. Tronix was started in September of 1982 by a gentleman by the name of John Reece.

Q. How long did it take you to program 'Kid Grid'?

A. It took me two months.

Q. And what assembler did you use?

A. The Assembler/Editor by ATARI.

Q. How long did it take you to program 'Juice'?

A. Juice took me four and a half months long, because Juice originally started out as a totally different game. I don't really like to sit down and define a game right down to the last detail and then find out the game, once coded, is boring. So I come up with a basic idea and start coding it, adding features as I code. With Juice I hit lots of dead ends.I would come up with a program like Basketball, but when I see a program isn't going anywhere I have to drop the past two weeks of work and retrace my steps, going back to the base I had which was the playfield of Juice. Then take it again from there. Designing new characters, new action and all those things.

Q. I noticed that Juice had more complicated graphics then Kid Grid. Was this because you spent more time in the designing of it?

A. Well,commercial programming for me was a whole new field. I started out with a text adventure, then I got my ATARI and Kid Grid was my first major graphic work. Being my first venture into graphics in general and after looking at Kid Grid I wanted to do something better then it, more graphically demanding. The next original piece of work I do on th ATARI or any machine would be even more graphically better then Juice.

Q. What one feature do like on the ATARI over all the other personal computers on the market?

A. The feature that I like best about the ATARI is that it is the best documented machine in its' class. Basically my attitude is that you can do wonders with every machine if you can program it, however, ATARI makes it much easier for you, because they document everything that the machine does, thus opening the field to you, not leaving anything up to guessing or trial and error.

Q. What do you like to do when your not writing programs?

A. I like listen to music, play the piano, and read. Anything I can get my hands on I read.

Q. What do you do to get all your ideas on writing a game?

A. First of all I look at what is out on the market. I look at was has been done on other computers or in the arcades, and also try to look at what traditional games, board games, kept people amused before there were computers. I often go into an arcade and play the games and then think to myself,"What did I like about it, or what I didn't like about a certain game." I then try to avoid those pitfalls when I'm writing a game because one of the main problems that as a programmer, usually the programmer has the attitude that whatever he's doing at the moment is the most perfect and best thing there is. He gets locked into thinking, "I have the perfect game." When it is really a piece of garbage. A programmer has to be very objective about his own work and it is difficult, especially if you've put in a couple of months of coding, to sit down and tell yourself,"This is not a good piece of work." You have to go back and retrace your steps and change things. Also what I try to do is get a lot of suggestions once I have started the game. These suggestions are good from people who know nothing about computers. They'll sit in front of a half finished game and say, "I don't like this color and this sound is too loud, etc." Most of the time what they say is not usable, but they steer me in the right direction.

Q. What direction do you see computer games going?

A. The days of where a single programmer can write a game in a couple of months and make a big hit out of it, those days are numbered. The reason for this is because most machines now come with 64K and soon to be 128K therefore, lots of people are going to start wanting games of this size. We're going to get to the point where 64K machine language games are standard and any one person in any appreciable amount of time cannot write a good 64K program. This means programs are going to be collaberated efforts between two, three, four,or a whole group of programmers. So I think the days of a single office are going to come to an end and producing software will become like producing any other product, one person cannot do it anymore. More people will have to pool their time and their talents to put applications and games together.

Q. What role do you see ATARI playing in the microcomputer world?

A. I'm looking forward to the ATARI comeback soon because I have looked at their new machines and their fantastic.I really hope that they get back to the point to where they were before.

Q. What advice do you give to someone thinking about writing a program to sell?

A. Most color computer type machines can be programmed similar to produce very good games. It is a lot like playing a piano. A composer might be a real good composer, but if his talents are limited on the piano, he is just going to write a work that he can perform himself. If he is not a very good performer then his compositions aren't going to be very good, and it is the same about writing an original computer program. You may have very grandiose ideas, but you may not be able to implement them and therefore you will cut them out of your program. I think a problem that most people are into this 'get rich quick' theme in which they write something in two or three months and think it's the ultimate, it is not. I think before anyone should sit down and write a program they should first have a very clear understanding of the language they are working in. It may be Assembly, Pascal, Basic, or any language, and then have a very clear understanding of the machine's capabilities. Then they should plunge into writing a game.

MAGIC MAIL

Reviewed By Peter Ellison

This is the mailing list that we have all been waiting for. This program called "Magic Mail" from ABBS is so simple to use even a small child could use it. When I first received it I was surprised to find out how simple it was to operate. I barely looked at the 54 page instruction booklet to get a pretty good understanding of the product.

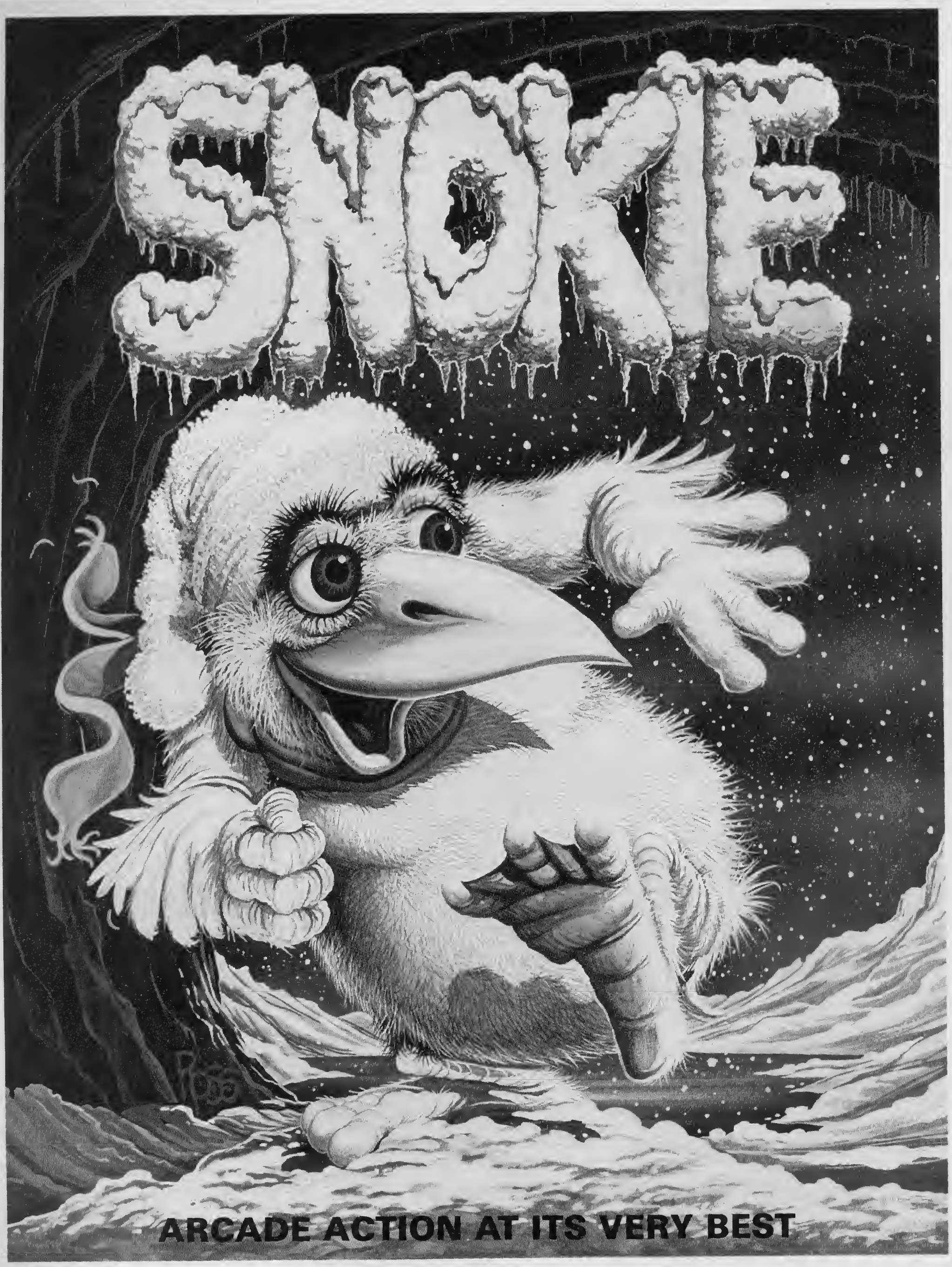
This program will make life a lot easier for either the person in the home or business, with its' great speed (100% machine language) to store or retrieve names and addresses. There are eight different fields that information can be entered in. They are Name, Street Address, City/Town, State, Zipcode, Phone number(including area code), Birthday(Day/date/year), and an Entry status.There are lines after each field showing how much space is available for each field. When you get near the end of the current Data field, Magic Mail even lets out a quiet beep to let you know.

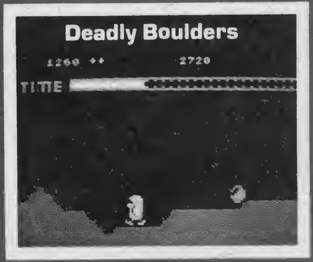
You can store over a 1000 entries on each single-sided Density Disk and can also have an unlimited number of DATA Disks in your system. There are so many features in this program that when you are using it, Magic Mail makes you feel like you're using a highly sophisticated business computer. One of these excellent features is the FIND command.This command allows the user to find any entry in a matter of a second. I have used many mailing lists and have waited many hours to find one of my addresses. One thing that makes the FIND command so fast is, it can search right through the current Data File using multiple Data Fields as the search criteria.The 'OR' or 'AND' is user specified.

Another feature that I have found useful, not usually found in most Databases, is the ability to edit by using the cursor control keys to go back up a field that needs correction or was left out. Also when booting up the disk, it didn't take more than 15 seconds, when many of the others took nearly a minute. Once the disk is booted, a Built-In help menu appears, making it easy to get started quickly. This menu can be accessed at any time by just pressing "<" while in the data entry mode. By using this menu, one can sort or reorganize mailing list I have used, which took nearly a minute. Once the disk is booted, a Built-In help help menu appears, making it easy to get started quickly. This menu can be accessed at any time by just pressing a "<" while in the data entry mode. By using this menu, one can Sort or reorganize the entries in any Database by any field.The limit on the number of entries that you may sort is determined by the amount of memory on your computer. The average sort time for a very large file is only 20-30 seconds.

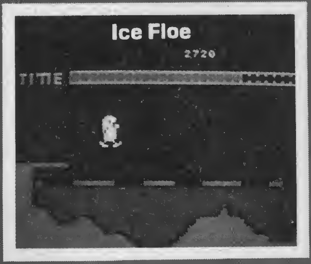
This product has so many different features I could write a ten page review of it and still have more to say. For the moment, however, I will just tell about one last feature. This is one that allows the computer to do all of your phone dialing. It will automatically dial any number in your database by simply placing the touch tone phone over or near your Television monitor speaker and press Control 'D' while the desired entry is displayed on your screen. Just think of all the time that can be saved never having to dial another number. In closing, all I can say is that 'Magic Mail' is a very excellent Mailing List program and is the last one you'll ever need. This program is available from:

ABBS
P.O. BOX 2452
SYRACUSE,NEW YORK
13220


SNOKIE
Ross
ARCADE ACTION AT ITS VERY BEST



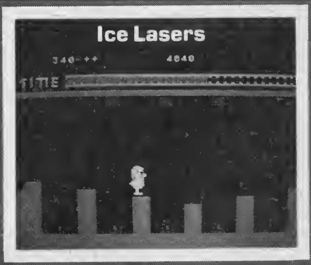
1260 ++
2720
TIME



2720
TIME

4840
TIME

200 +
4840
TIME



340 ++
4840
TIME

MAKING A GAME

By JACK CHUNG

Welcome to the world of programming. This series of articles will teach a BASIC programmer the basics of making a game. I have been programming for 4 years now on the ATARI and through trial and error, I have found a lot of special techniques. I will try to pass some of these techniques through the article also.

One of the phrases from a BASIC programmer I hear from alot of people is:

"I just learned BASIC but I don't know how to make objects move around the screen with a joystick."

One of the simplest techniques to move objects around the screen with BASIC is with the POSITION command. So to start out the program,we should first clear the screen.

```
     10 GRAPHICS 0
```

When you clear the screen, the cursor will still be there. It doesn't look that good to have a cursor flying around the screen so we'll shut it off by:

```
     20 POKE 752,1
```

The screen doesn't look that good in blue so the program should turn the color to black to look like space(hint!)

```
     30 SETCOLOR 2,0,0
```

If your confused about some of the basic graphic commands this article is using, refer to the "BASIC REFERENCE MANUAL".

Now we start to get to the main part of the program:

```
     40 S=STICK(0)
     45 IF S=15 THEN 40
     50 IF S=14 THEN Y=Y-1:REM MO
VE OBJECT UP
     60 IF S=7 THEN X=X+1:REM MO
VE OBJECT RIGHT
     70 IF S=13 THEN Y=Y+1:REM M
OVE OBJECT DOWN
     80 IF S=11 THEN X=X-1:REM M
OVE OBJECT LEFT
```

After we put in the joystick checks, the program will print out the image by:

```
     100 POSITION  X,Y:?  "*":REM
BETWEEN THE  QUOTES,YOU  CAN  PUT
ANYTHING YOUR HEART DESIRES.

     200 GOTO 40
```

Now type that program in and run it on the computer.

"But I don't want to have that funny trail everytime I move the joystick."

Ok,to make the object clear the trail, you add these lines:

```
     41 DX=X:DY=Y:REM DUMMY LOCAT
IONS
     90 POSITION DX,DY:? "  ":REM
THE " " CLEARS THE TRAIL
```

Now run the program. If the program still leaves the trail, you probably typed something in the program wrong.

If the object runs around the screen too fast for you, slow it down by adding this line:

```
     120 FOR T=1 TO 50:NEXT T:REM
FOR-NEXT LOOP
```

"That's great! But in my game, how would the program know if my object hit anything?"

Good question. To check for collisions,type in the following lines:

```
     85 LOCATE X,Y,Z
     86 IF Z<>32 THEN PRINT "BOOM
":END:REM GET THE IDEA!
```

To see what Z is, check the ASCII code. Line 86 sees if the object hits anything. The 32 means blank

Continued on page 68

FANCY WRITER

A Data Based Font Writer for the Atari®

Fancy Writer© is a unique program which permits printing of five different fonts (supplied with the program) on your Prowriter®, NEC®, C.Itoh®, Epson®, Gemini®, AT/GX100® and Gorilla Banana® dot printer. Fancy Writer© will print five sizes on the Prowriter®, NEC®, and C. Itoh®; two sizes on the Epson®, Gemini®; one size on the AT/GX100® and Gorilla®. Fonts available are Old English, Cursive writing, Countdown, Futura and Babyteeth.

This is Old English

AND HERE IS BABYTEETH

This is Cursive Writing

THIS IS COUNTDOWN

HERE IS FUTURA STYLE

- ★ Easy to use . . . Menu driven.
- ★ Not a dump or downloader.
- ★ Fonts were custom made by a professional artist especially for computer printing.
- ★ Print from disk file.
- ★ File text on disk.
- ★ Direct keyboard entry.
- ★ Great for newsletters, correspondence, letterheads.

$34.95

SEE REVIEW OF OUR PRODUCTS IN JAN. '84 ISSUE OF ANTIC

HUMPTY DUMP

Put Your Art in Print

Humpty Dump© is a RAM resident screen dump for Graphics modes 0 (yes! zero!), 7, 7+, 8, 9, 10 and 11. Once booted into your Atari 400/800® it is always ready to dump your computer art or mode 0 listings to a Prowriter®, NEC®, C.Itoh®, Epson®, or Gemini® graphics printer. It will even dump a screen while a BASIC program is running!!! Humpty Dump© works with either the BASIC or ASSEMBLER/EDITOR cartridges. Now you can show Aunt Sara your Art without hauling your equipment over to her house!!

$34.95

Humpty Dump® not available for AT/GX100® or Gorilla.®

AT LAST !

BEARTRAP

PRINT YOUR KOALA PICTURES

Beartrap© is a new translator utility from Kidstuff Software which converts Koala® picture files to raw screen files for loading and dumping with programs on Kidstuff's famous HUMPTY DUMP©!! Koala® files are left unharmed because BEARTRAP© places the converted files on a separate disk under the filename of your choice. Just think! You can now combine the best of two worlds, Koala Pad®, the most exciting peripheral on the market today, and HUMPTY DUMP©, the famous RAM resident screen dumper from Kidstuff Software.

$14.95

**Koala Pad is a trademark of Koala Technologies Corp.

Kidstuff Software
3736 Ferndale Drive
Ft. Wayne, Indiana 46815
U.S.A.
(219) 485-2923

KIDSTUFF SOFTWARE

Dear Kidstuff,

Please send me the following:

_____ Fancy Writer ($34.95 ea.)

_____ Humpty Dump ($34.95 ea.)

_____ Beartrap ($14.95 ea.)

I am enclosing check or money order in the amount of $__________. Please Rush !! my order to . . .

Name ____________________________

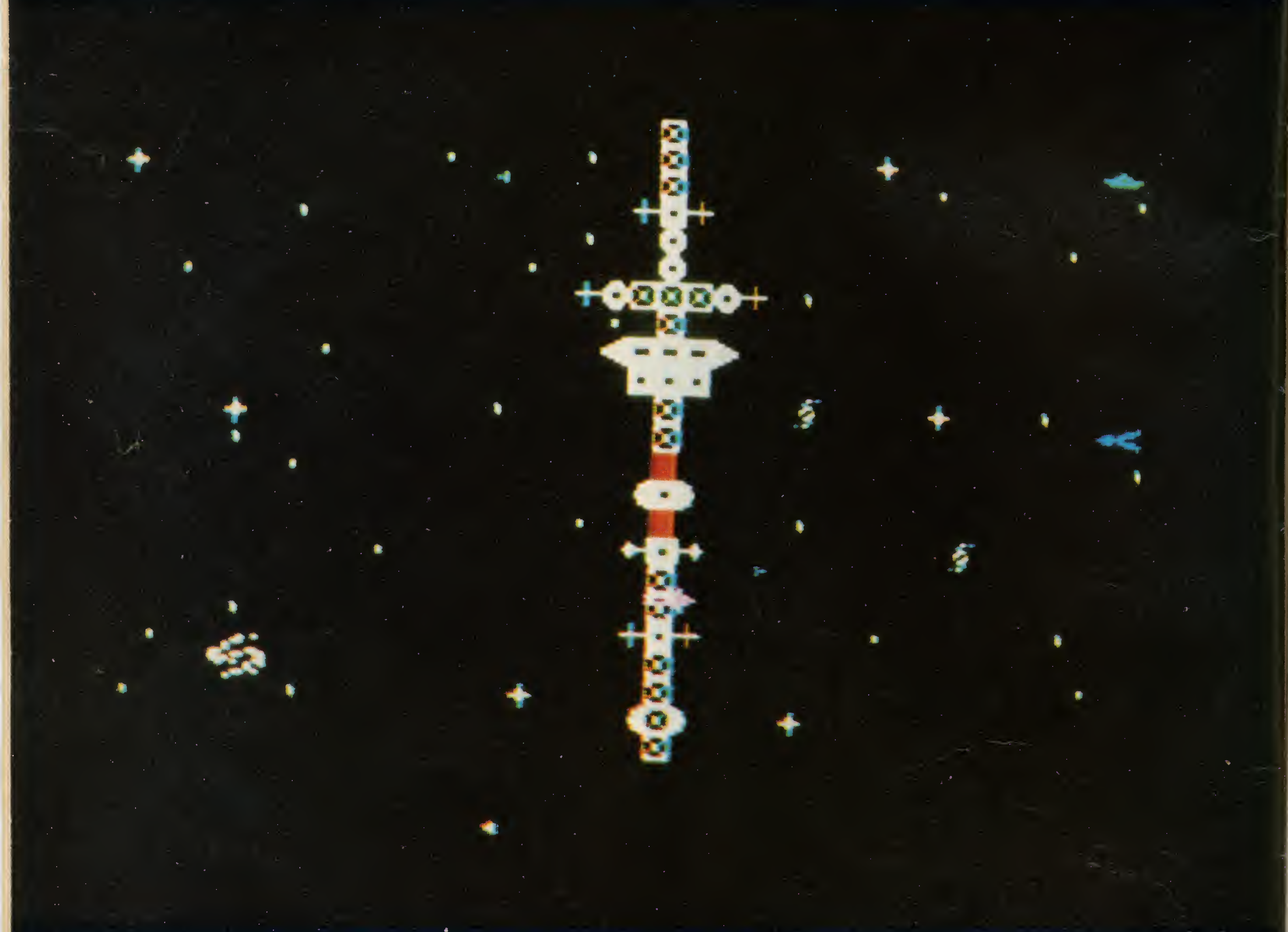
Address ____________________________

City________________ St_____ Zip_______

Country ____________________________

SPACE STATION Z64

By Bob Cockroft



Space Station Z64 cont'd

The anxious faces of the six member crew of space station Z64 stared silently into the circular radar scope as the once distant blips approached their location with high velocity. In the emptiness of space, the possibility of an attack from the Atitrons became an unspoken concern. Gazing into the pulsing monitor, they began to think that the attempts of the Earth Federation to avoid a war must have failed. Yet the fact that no message had been received from headquarters, notifying of any danger, puzzled them.

Suddenly C.A.T., the central computer system, sounded an alert. The once quiet bridge now buzzed with the noise of people rushing to their assigned positions. C.A.T. automatically displayed normally hidden weapon panels and illuminated the interior of the station with pulsing yellow lights. Consumed with nervous energy, the crew frantically began to prepare their defense systems. Because they were a minor installation, no aid from Earth Federation was expected. Being isolated and located near Atitron territory, they all knew that they would become easy prey for the enemy forces, should war break-out. The blips were drawing closer.

The only significant defense would be their mobile missile launcher. Mounted on guide rails that extend the length of the station, this weapon would be able to fire horizontally upon oncoming spacecraft. Although missiles could be fired with deadly accuracy, the rate of launching is relatively slow, partically if the target is a distance away.

C.A.T. was now automatically transmitting messages to recall all the one-man research pods. These pods were used by scientists to study the surrounding space. Unfortunely, because no immediate attack was previously expected, they had been sent out to do research. As a result of the pods distance from the Space Station, they would not be able to return until the Atitrons had arrived. They will need to be recovered by the mobile missile launcher, the pods will be recovered, and the keyboard speaker will "click" and your score will be increased.One does not need to worry about destroying the pods before they reach the station. The pods are equiped with special defence shields that neutralize the effects of friendly missiles.

Taking controls of the missile launcher, C.A.T. briefs you on typical Atitron strategies. The best way to destroy the Station is to severe the central power cable that runs vertically along the middle of the structure. This cable supplies all the weapon systems with power. Without it the Station would have no means of defending itself and as a result become completely neutralized. Knowing this, the Atitrons will attempt to have their ships crash into the cable. The early waves of an Atitron attack are usually ineffective. Using slow and obsolete spacecraft , these are usually handled even by the most inexperienced player. The attacking spacecraft are little more than easy points and a means to practice skills. However, each wave becomes increasingly dangerous. With growing speed and intensity , Atitrons will become more than a match for anyone. The only thing a player can do is delay the destruction of the Space Station, thus maximizing his or her score.

The blips were nearly upon them. All was quiet and ready. You sit silently in the missile laucher gazing into the black emptiness of space. Suddenly, you get a visual on the first wave of the attack. Thumb hovering over the fire button, you take aim. GOOD LUCK!

Space Station Z64 cont'd

```
1 REM * REQUIRES 48K AND ONE JOYS
TICK*
2 GRAPHICS 0:SETCOLOR 2,16,1:POKE
752,1:POSITION 11,8:? "Loading Ma
chine Data":DIM RAN1$(20),YN$(4)
3 POSITION 11,10:? "Please Wait
1:45 min"
4 FOR X=1 TO 3355
5 READ D:IF X>180 THEN POKE 3400
0-181+X,D
6 NEXT X
7 C=20:PLAY=1:LIVES=1:LVL1=0:LVL2
=0:LVL3=0:LVL4=0:LEVEL=1:LVEL1=1:
LVEL2=1:LVEL3=1:LVEL4=1:POKE 752,
0:RESTORE
8 NON1=5:NON2=5:NON3=5:NON4=5:ESP
1=27:ESP2=27:ESP3=27:ESP4=27:CH1=
0:CH2=0:CH3=0:CH4=0:SC1=0:SC2=0:S
C3=0:SC4=0
9 POKE 88,64:POKE 89,156:POKE 106
,160
10 GRAPHICS 1:SETCOLOR 2,16,1
11 ? "Use OPTION & SELECT buttons
or START to continue"
12 POSITION 1,4:? #6;"How many
players?"
13 POSITION 18,4:? #6;PLAY
14 IF PEEK(53279)=3 AND PLAY<4
THEN PLAY=PLAY+1
15 IF PEEK(53279)=5 AND PLAY>1
THEN PLAY=PLAY-1
17 FOR X=1 TO 100:NEXT X
18 IF PEEK(53279)<>6 THEN 13
20 POSITION 1,6:? #6;"How many
lives?"
22 POSITION 18,6:? #6;LIVES
24 IF PEEK(53279)=3 AND LIVES<5
THEN LIVES=LIVES+1
26 IF PEEK(53279)=5 AND LIVES>1
THEN LIVES=LIVES-1
28 FOR X=1 TO 100:NEXT X
30 IF PEEK(53279)<>6 THEN 22
40 LV1=LIVES:LV2=LIVES:LV3=LIVES:
LV4=LIVES
99 REM *CHARACTER BASE FOR THE RO
M SET
100 ROMSET=57344
105 REM*SET BASE VALUE OF THE NEW
SET
110 RAMT=156
120 NSET=RAMT*256
125 REM * RESERVE MEMORY SPACE *
130 POKE 106,PEEK(106)-5
140 GRAPHICS 0
141 POKE 752,1:SETCOLOR 2,16,1
142 POSITION 15,4:? "Please WAIT"
```

```
144 POSITION 8,6:? "(Loading Char
acter Data)"
145 REM * MOVE ROM SET TO NESET
LOCATION *
150 FOR L=1 TO 1024
155 POKE NSET+L-1,PEEK(ROMSET+L-1
)
160 NEXT L
165 REM * SET CHARACTER BASE
REGISTER TO NEW SET LOCATION *
166 POKE 752,0
170 POKE 756,NSET/256
175 REM * POKE IN VALUES FOR
MODIFIED CHARACTERS *
180 FOR L1=1 TO 20
190 READ LOC
200 SET=NSET+LOC*8
210 FOR L2=0 TO 7:READ D
220 POKE SET+L2,D
230 NEXT L2
240 NEXT L1
500 GOTO 2000
1000 POKE 88,64:POKE 89,156
1001 GRAPHICS 0:SETCOLOR 2,16,1
1002 POKE 752,1
1005 POKE 756,NSET/256
1010 POSITION C,1:? "K":POSITION
C,2:? "K":POSITION C,3:? "J":POSI
TION C-1,3:? "N":POSITION C+1,3:?
"M"
1015 POSITION C,0:? "K"
1020 POSITION C,4:? "L":POSITION
C,5:? "L":POSITION C-1,6:? "K":PO
SITION C,6:? "K":POSITION C+1,6:?
"K"
1030 POSITION C-2,6:? "L":POSITIO
N C+2,6:? "L":POSITION C-3,6:? "N
":POSITION C+3,6:? "M"
1040 POSITION C,7:? "K":POSITION
C-2,8:? "C":POSITION C-1,8:? "I":
POSITION C,8:? "I":POSITION C+1,8
:? "I"
1050 POSITION C+2,8:? "D"
1060 POSITION C-1,9:? "J":POSITIO
N C,9:? "J":POSITION C+1,9:? "J"
1070 POSITION C,10:? "K"
1080 POSITION C,11:? "K"
1090 POSITION C,12:? "Q"
1100 POSITION C-1,13:? "E":POSITI
ON C,13:? "J":POSITION C+1,13:? "
F"
1110 POSITION C,14:? "Q"
1120 POSITION C,15:? "K"
1130 POSITION C-1,15:? "H":POSITI
ON C,15:? "J":POSITION C+1,15:? "
G"
```

Basic Listing cont'd

```
1140 POSITION C,16:? "K"
1150 POSITION C,17:? "K"
1160 POSITION C-1,18:? "N":POSITI
ON C,18:? "J":POSITION C+1,18:? "
M"
1170 POSITION C,19:? "K"
1180 POSITION C,20:? "K"
1190 POSITION C-1,21:? "E":POSITI
ON C,21:? "K":POSITION C+1,21:? "
F"
1200 POSITION C,22:? "K"
1299 REM * DRAW BACKGROUND*
1300 POSITION 10,15:? "S":POSITIO
N 15,5:? "S":POSITION 7,12:? "S":
POSITION 8,8:? "S"
1305 POSITION 25,10:? "S":POSITIO
N 28,18:? "S":POSITION 35,4:? "S"
:POSITION 36,20:? "S":POSITION 24
,8
1310 POSITION 18,7:? "S":POSITION
17,14:? "S"
1315 POSITION 5,11:? "T":POSITION
25,6:? "T":POSITION 25,10:? "T":P
OSITION 30,10:? "O":POSITION 34,1
0:? "T"
1325 POSITION 17,4:? "T":POSITION
12,1:? "S":POSITION 17,1:? "T"
1330 POSITION 1,20:? "S":POSITION
2,18:? "S":POSITION 5,17:? "T":PO
SITION 7,20:? "T"
1335 POSITION 1,1:? "O":POSITION
5,10:? "O":POSITION 15,20:? "O":P
OSITION 25,21:? "O":POSITION 28,1
:? "O"
1340 POSITION 25,10:? "R":POSITIO
N 35,18:? "T":POSITION 30,2:? "S"
:POSITION 38,2:? "S":POSITION 38,
12:? "T"
1345 POSITION 25,14:? "T":POSITIO
N 3,5:? "S":POSITION 31,15:?"R":P
OSITION 7,3:? "T":POSITION 14,10:
? "T"
1490 POKE 77,0
1500 X=USR(33999)
1501 POKE 752,0
1505 SCR1=PEEK(1607)*2*LEVEL5+(PE
EK(1545)*LEVEL5+INT(RND(0)*5+1))
1520 IF P=1 THEN SC1=SC1+SCR1
1525 IF P=2 THEN SC2=SC2+SCR1
1530 IF P=3 THEN SC3=SC3+SCR1
1535 IF P=4 THEN SC4=SC4+SCR1
1599 GOTO 4040
1999 REM * STATUS SECTION *
2000 FOR P=1 TO PLAY
2002 IF P=1 AND LV1<1 THEN 4120
2003 IF P=2 AND LV2<1 THEN 4120
2004 IF P=3 AND LV3<1 THEN 4120
2005 IF P=4 AND LV4<1 THEN 4120
2007 SOUND 0,0,0,0:SOUND 1,0,0,0:
SOUND 2,0,0,0:SOUND 3,0,0,0
2008 POKE 53248,0:POKE 53249,0:PO
KE 53250,0:POKE 53251,0:POKE 5325
2,0:POKE 53253,0:POKE 53254,0:POK
E 53255,0
2009 GRAPHICS 1:SETCOLOR 2,16,1
2010 POSITION 7,7:? #6;"PLAYER";:
? #6;P
2012 IF P=1 THEN L5=LV1
2013 IF P=2 THEN L5=LV2
2014 IF P=3 THEN L5=LV3
2015 IF P=4 THEN L5=LV4
2020 POSITION 7,9:? #6;"LIVES ";:
? #6;L5
2022 IF P=1 THEN LEVEL5=LVEL1
2024 IF P=2 THEN LEVEL5=LVEL2
2026 IF P=3 THEN LEVEL5=LVEL3
2028 IF P=4 THEN LEVEL5=LVEL4
2030 POSITION 7,11:? #6;"LEVEL";:
? #6;LEVEL5
2040 FOR X=1 TO 200:NEXT X
2100 IF  CH1=1  AND  ESP1>2  THEN
NON1=NON1+2:ESP1=ESP1-2:CH1=0:POK
E 1608,NON1:POKE 1609,ESP1
2110 IF  CH2=1  AND  ESP2>2  THEN
NON2=NON2+2:ESP2=ESP2-2:CH2=0:POK
E 1608,NON2:POKE 1609,ESP2
2120 IF  CH3=1  AND  ESP3>2  THEN
NON3=NON3+2:ESP3=ESP3-2:CH3=0:POK
E 1608,NON3:POKE 1609,ESP3
2130 IF  CH4=1  AND  ESP4>2  THEN
NON4=NON4+2:ESP4=ESP4-2:CH4=0:POK
E 1608,NON4:POKE 1609,ESP4
2180 IF P=1 THEN POKE 1608,NON1:P
OKE 1609,ESP1
2182 IF P=2 THEN POKE 1608,NON2:P
OKE 1609,ESP2
2184 IF P=3 THEN POKE 1608,NON3:P
OKE 1609,ESP3
2186 IF P=4 THEN POKE 1608,NON4:P
OKE 1609,ESP4
4020 POKE 33999,104
4030 GOTO 1000
4040 IF PEEK(1621)<>1 THEN 4100
4050 IF P=1 AND LV1>0 THEN LV1=LV
1-1
4060 IF P=2 AND LV2>0 THEN LV2=LV
2-1
4070 IF P=3 AND LV3>0 THEN LV3=LV
3-1
4080 IF P=4 AND LV4>0 THEN LV4=LV
4-1
4090 GOTO 4120
```

Space Station Z64 cont'd

Basic Listing cont'd

```
4100 IF P=1 THEN LVEL1=LVEL1+1:CH
1=1
4104 IF P=2 THEN LVEL2=LVEL2+1:CH
2=1
4106 IF P=3 THEN LVEL3=LVEL3+1:CH
3=1
4108 IF P=4 THEN LVEL4=LVEL4+1:CH
4=1
4120 NEXT P
4122 IF PLAY=1 AND LV1<1 THEN 700
0
4124 IF PLAY=2 AND LV1<1 AND LV2<
1 THEN 7000
4126 IF PLAY=3 AND LV1<1 AND LV2<
1 AND LV3<1 THEN 7000
4128 IF PLAY=4 AND LV1<1 AND LV2<
1 AND LV3<1 AND LV4<1 THEN 7000
4130 P=1:GOTO 2000
7000 POKE 53248,0:POKE 53249,0:PO
KE 53250,0:POKE 53251,0:POKE 5325
2,0 :POKE 53253,0:POKE 53254,0:PO
KE 53255,0
7002 SOUND 0,0,0,0:SOUND 1,0,0,0:
SOUND 2,0,0,0:SOUND 3,0,0,0
7005 GRAPHICS 0
7010 SETCOLOR 2,16,1
7020 POSITION 15,2:? "Score Resul
ts"
7030 POSITION 3,4:? "Player":POSI
TION 13,4:? "Score":POSITION 25,4
:? "Rank"
7045 Y=8
7050 FOR X=1 TO PLAY
7060 IF X=1 THEN SE=SC1
7070 IF X=2 THEN SE=SC2
7080 IF X=3 THEN SE=SC3
7090 IF X=4 THEN SE=SC4
7100 IF SE<750 THEN RAN1$="Laser
Waster"
7110 IF SE>749 THEN RAN1$="Statio
n Steward"
7120 IF SE>1500 THEN RAN1$="Rooki
e"
7130 IF SE>2000 THEN RAN1$="Crew
Member"
7140 IF SE>2500 THEN RAN1$="Stati
on Technician "
7150 IF SE>3000 THEN RAN1$="Star
Centurion"
7160 IF SE>4000 THEN RAN1$="STATI
ON STAR MASTER"
7170 POSITION 2,Y:? "PLAYER";:? X
7180 POSITION 14,Y:? SE
7190 POSITION 21,Y:? RAN1$
7200 Y=Y+2
7210 NEXT X
```

```
8000 ? :? "Do you want to play
again?"
8010 INPUT YN$
8020 IF YN$="Y" THEN 7
10250 DATA 33,96,48,56,255,56,48,
96,0
10260 DATA 34,6,12,28,255,28,12,6
,0
10270 DATA 35,3,15,63,255,255,63,
15,3
10280 DATA 36,192,240,252,255,255
,252,240,192
10290 DATA 37,3,15,15,31,31,15,15
,3
10300 DATA 38,192,240,240,248,248
,240,240,192
10310 DATA 39,128,132,142,255,142
,132,128,0
10320 DATA 40,1,33,113,255,113,33
,1,0
10330 DATA 41,255,255,255,195,195
,255,255,255
10340 DATA 42,255,255,255,231,231
,255,255,255
10350 DATA 43,255,195,165,153,153
,165,195,255
10360 DATA 44,60,126,255,231,231,
255,126,60
10370 DATA 45,8,8,8,255,8,8,8,0
10380 DATA 46,16,16,16,255,16,16,
16,0
10390 DATA 47,0,16,16,56,254,56,1
6,16
10400 DATA 48,0,0,24,126,255,255,
126,24
10410 DATA 49,170,170,170,170,170
,170,170,170
10420 DATA 50,13,16,28,50,38,28,4
,88
10430 DATA 51,0,0,0,24,24,0,0,0
10432 DATA 52,0,0,16,24,24,8,0,0
20000 DATA 169,62,141,47,2,169,88
,141,192,2,169,60,141,193,2,169,1
12,141, 194,2,169,150,141,195
20005 DATA 2,169,64,141,7,212,169
,3,141,29,208,169,127,141,1,6,169
,125,141,0,6,169,1,141
20010 DATA 30,208,169,0,160,0,153
,58,6,153,62,6,153,67,6,200,192,4
,208,242,141,57,6,141
20015 DATA 9,6,141,7,6,141,2,6,14
1,11,6,141,9,6,141,71,6,173,1,6,1
41,0,208,162
20020 DATA 0,169,0,141,55,6,141,5
6,6,157,28,6,157,45,6,232,224,4,2
08,245,165,106,160,0
```

Space Station Z64 cont'd

Basic Listing cont'd

```
20025 DATA 169,0,141,85,6,153,0,6
7,153,0,68,153,0,69,153,0,70,153,
0,71,200,192,255,208
20030 DATA 236,169,0,141,8,210,16
9,3,141,15,210,169,100,141,79,6,1
69,80,141,78,6,169,60,141
20035 DATA 77,6,169,40,141,76,6,1
69,20,141,75,6,169,10,141,74,6,16
9,4,141,80,6,76,32
20040 DATA 135,173,71,6,205,72,6,
208,1,96,169,0,141,13,6,24,105,1,
162,0,232,224,255,208
20045 DATA 251,205,80,6,208,241,1
73,120,2,141,10,6,201,6,208,3,76,
227,133,201,7,208,5,208
20050 DATA 222,76,227,133,201,5,2
08,3,76,227,133,201,10,208,3,76,2
35,133,201,11,208,3,76,235
20055 DATA 133,201,9,208,3,76,235
,133,76,243,133,169,0,141,2,6,76,
243,133,169,1,141,2,6
20060 DATA 76,243,133,173,10,6,20
1,10,208,3,76,35,134,201,14,208,3
,76,35,134,201,6,208,3
20065 DATA 76,35,134,201,9,208,3,
76,48,134,201,13,208,3,76,48,134,
201,5,208,3,76,48,134
20070 DATA 76,61,134,172,0,6,192,
24,240,3,206,0,6,76,61,134,172,0,
6,192,215,240,3,238
20075 DATA 0,6,76,61,134,76,139,1
38,173,11,6,201,0,240,6,173,8,6,7
6,130,134,173,132,2
20080 DATA 201,0,240,11,169,0,141
,0,210,141,1,210,76,142,133,173,1
20,2,201,13,240,246,201,14
20085 DATA 240,242,201,15,240,238
,141,8,6,174,0,6,142,4,6,174,1,6,
142,5,6,162,1,142
20090 DATA 11,6,169,90,141,0,210,
169,166,141,1,210,174,57,6,224,0,
208,10,238,7,6,172,7
20095 DATA 6,192,75,208,25,160,0,
140,5,6,174,4,6,140,4,6,140,11,6,
140,7,6,169,0
20100 DATA 141,57,6,157,4,67,173,
8,6,201,10,208,3,76,230,134,201,1
1,208,3,76,230,134,201
20105 DATA 9,208,3,76,230,134,201
,6,208,3,76,242,134,201,7,208,3,7
6,242,134,201,5,208,3
20110 DATA 76,242,134,76,142,133,
206,5,6,173,5,6,141,4,208,76,254,
134,238,5,6,173,5,6
20115 DATA 141,4,208,76,254,134,1
73,13,6,24,105,3,141,13,6,172,4,6
,169,0,153,2,67,153
20120 DATA 3,67,153,5,67,153,6,67
,169,3,153,4,67,76,142,133,238,55
,6,162,0,232,224,4
20125 DATA 208,16,173,55,6,205,73
,6,208,5,169,0,141,55,6,76,241,13
8,189,58,6,201,0,240
20130 DATA 3,76,206,140,173,55,6,
205,73,6,240,3,76,37,135,189,28,6
,201,0,208,3,76,99
20135 DATA 135,160,0,200,192,200,
208,251,76,179,135,238,71,6,169,1
,157,28,6,173,10,210,157,19
20140 DATA 6,160,0,152,221,19,6,2
40,21,200,192,29,208,245,160,255,
152,221,19,6,240,16,136,192
20145 DATA 215,208,245,76,158,135
,169,29,157,19,6,76,158,135,169,2
15,157,19,6,76,158,135,173,10
20150 DATA 210,16,8,169,30,157,14
,6,76,179,135,169,230,157,14,6,76
,179,135,189,14,6,201,127
20155 DATA 208,3,76,216,135,48,3,
76,205,135,254,14,6,169,1,157,32,
6,76,216,135,222,14,6
20160 DATA 169,2,157,32,6,76,216,
135,189,14,6,201,127,208,6,169,1,
141,85,6,96,189,14,6
20165 DATA 157,0,208,224,1,208,3,
76,252,135,224,2,208,3,76,53,137,
76,99,138,169,60,141,193
20170 DATA 2,189,23,6,24,105,80,1
57,23,6,189,23,6,74,74,74,74,74,7
4,201,0,208,3,76
20175 DATA 43,136,201,2,208,3,76,
252,136,201,3,208,3,76,177,136,76
,106,136,189,19,6,168,169
20180 DATA 0,153,255,68,153,10,69
,169,125,153,0,69,169,232,153,1,6
9,169,128,153,2,69,153,3
20185 DATA 69,169,236,153,4,69,16
9,126,153,5,69,169,50,153,6,69,16
9,2,153,7,69,169,22,153
20190 DATA 8,69,169,188,153,9,69,
76,37,135,189,19,6,168,169,0,153,
254,68,153,11,69,169,10
20195 DATA 153,255,68,169,24,153,
0,69,169,32,153,1,69,153,2,69,153
,10,69,169,198,153,3,69
20200 DATA 169,155,153,4,69,169,2
25,153,5,69,169,83,153,6,69,169,5
,153,7,69,169,4,153,8
20205 DATA 69,169,26,153,9,69,76,
37,135,189,19,6,168,169,0,153,254
,68,153,11,69,169,8,153
20210 DATA 255,68,169,116,153,0,6
```

```
9,169,38,153,1,69,169,98,153,2,69
,169,49,153,3,69,169,24
20215 DATA 153,4,69,169,142,153,5
,69,169,4,153,6,69,169,166,153,7,
69,169,104,153,8,69,169
20220 DATA 52,153,9,69,169,96,153
,10,69,76,37,135,189,19,6,168,169
,0,153,255,68,153,10,69
20225 DATA 153,9,69,153,0,69,153,
1,69,169,134,153,2,69,169,11,153,
3,69,169,201,153,4,69
20230 DATA 153,6,69,169,147,153,5
,69,169,104,153,7,69,169,49,153,8
,69,76,37,135,169,117,141
20235 DATA 194,2,189,23,6,24,105,
35,157,23,6,189,32,6,201,1,208,3,
76,216,137,189,23,6
20240 DATA 74,74,74,74,74,74,201,
0,208,3,76,110,137,201,2,208,3,76
,154,137,201,3,208,3
20245 DATA 76,182,137,76,182,137,
189,19,6,168,169,0,153,255,69,153
,7,70,169,1,153,0,70,153
20250 DATA 6,70,169,7,153,1,70,15
3,5,70,169,110,153,2,70,153,4,70,
169,248,153,3,70,76
20255 DATA 37,135,189,19,6,168,16
9,0,153,1,70,153,5,70,169,96,153,
2,70,153,4,70,169,255
20260 DATA 153,3,70,76,37,135,189
,19,6,168,169,0,153,0,70,153,1,70
,153,5,70,153,6,70
20265 DATA 169,111,153,2,70,153,4
,70,169,248,153,3,70,76,37,135,18
9,23,6,74,74,74,74,74
20270 DATA 74,201,0,208,3,76,249,
137,201,2,208,3,76,71,138,201,3,2
08,3,76,37,138,76,37
20275 DATA 138,189,19,6,168,169,0
,153,255,69,153,7,70,169,128,153,
0,70,153,6,70,169,224,153
20280 DATA 1,70,153,5,70,169,118,
153,2,70,153,4,70,169,31,153,3,70
,76,37,135,189,19,6
20285 DATA 168,169,0,153,1,70,153
,5,70,153,0,70,153,6,70,169,246,1
53,2,70,153,4,70,169
20290 DATA 31,153,3,70,76,37,135,
189,19,6,168,169,0,153,1,70,153,5
,70,169,6,153,2,70
20295 DATA 153,4,70,169,255,153,3
,70,76,37,135,169,150,141,195,2,1
89,19,6,168,169,0,153,255
20300 DATA 70,153,4,71,169,8,153,
0,71,169,28,153,1,71,169,119,153,
2,71,169,62,153,3,71
20305 DATA 76,37,135,173,2,6,201,
0,208,3,76,192,138,172,0,6,169,0,
153,0,68,153,8,68
20310 DATA 169,4,153,1,68,153,7,6
8,169,12,153,2,68,153,6,68,169,25
4,153,3,68,153,5,68
20315 DATA 169,31,153,4,68,76,64,
134,172,0,6,169,0,153,0,68,153,8,
68,153,255,67,153,9
20320 DATA 68,169,32,153,1,68,153
,7,68,169,48,153,2,68,153,6,68,16
9,127,153,3,68,153,5
20325 DATA 68,169,248,153,4,68,76
,64,134,238,56,6,173,56,6,201,8,2
40,3,76,40,140,169,0
20330 DATA 141,56,6,162,0,232,224
,4,208,3,76,40,140,189,45,6,201,0
,208,3,76,26,139,76
20335 DATA 124,139,169,1,157,45,6
,173,10,210,157,37,6,74,74,74,74,
201,0,208,8,169,29,157
20340 DATA 37,6,76,62,139,201,5,2
08,5,169,210,157,37,6,173,10,210,
16,15,173,10,210,16,5
20345 DATA 169,26,76,94,139,169,1
9,76,94,139,173,10,210,16,5,169,2
28,76,94,139,169,235,157,41
20350 DATA 6,76,100,139,189,41,6,
48,11,76,108,139,169,2,157,50,6,7
6,124,139,169,1,157,50
20355 DATA 6,76,124,139,189,41,6,
201,10,240,7,201,245,240,3,76,161
,139,169,0,157,45,6,189
20360 DATA 37,6,168,169,0,153,0,6
7,153,1,67,153,2,67,76,5,139,189,
50,6,201,2,208,6
20365 DATA 254,41,6,76,177,139,22
2,41,6,189,41,6,157,4,208,224,1,2
08,3,76,200,139,224,2
20370 DATA 208,3,76,232,139,76,8,
140,189,37,6,168,189,50,6,201,2,2
08,5,169,8,76,218,139
20375 DATA 169,4,153,0,67,153,2,6
7,169,12,153,1,67,76,5,139,189,37
,6,168,189,50,6,201
20380 DATA 2,208,5,169,32,76,250,
139,169,16,153,0,67,153,2,67,169,
48,153,1,67,76,5,139
20385 DATA 189,37,6,168,189,50,6,
201,2,208,5,169,128,76,26,140,169
,64,153,0,67,153,2,67
20390 DATA 169,192,153,1,67,76,5,
139,173,8,208,74,74,74,201,0,240,
28,162,3,169,0,157,28
20395 DATA 6,142,57,6,189,19,6,16
8,169,1,157,58,6,173,8,208,56,233
```

Space Station Z64 cont'd

Basic Listing cont'd

```
,8,76,81,140,173,8
20400 DATA 208,74,201,2,240,11,20
1,1,240,31,201,0,208,3,76,139,140
,162,2,168,169,1,157,58
20405 DATA 6,169,0,157,28,6,142,5
7,6,152,201,3,240,3,76,139,140,16
2,1,169,1,157,58,6
20410 DATA 169,0,157,28,6,142,57,
6,76 ,139,140,173,9,208,201,1,208
,5,162,1,76,172,140,173
20415 DATA 10,208,201,1,208,5,162
,2,76,172,140,173,11,208,201,1,20
8,28,162,3,238,9,6,169
20420 DATA 0,157,45,6,141,31,208,
189,37,6,168,169,0,153,0,67,153,1
,67,153,2,67,169,1
20425 DATA 141,30,208,76,145,133,
189,62,6,205,79,6,208,14,169,0,15
7,58,6,157,62,6,157,67
20430 DATA 6,76,52,145,224,1,208,
3,76,245,140,224,2,208,3,76,112,1
42,76,206,143,254,62,6
20435 DATA 189,62,6,205,78,6,208,
8,1 69,5,157,67,6,76,77,142,205,7
7,6,208,8,169,4,157
20440 DATA 67,6,76,22,142,205,76,
6,208,8,169,3,157,67,6,76,220,141
,205,75,6,208,8,169
20445 DATA 2,157,67,6,76,160,141,
205,74,6,208,8,169,1,157,67,6,76,
94,141,189,67,6,201
20450 DATA 5,208,3,76,77,142,201,
4,208,3,76,22,142,201,3,208,3,76,
220,141,201,2,208,3
20455 DATA 76,160,141,76,94,141,1
69,62,141,193,2,189,19,6,168,169,
42,153,2,69,153,6,69,169
20460 DATA 28,153,3,69,153,5,69,1
69,126,153,4,69,169,0,153,7,69,15
3,8,69,153,9,69,153
20465 DATA 10,69,153,0,69,153,1,6
9,153,255,68,169,24,141,2,210,169
,14,141,3,210,76,52,145
20470 DATA 169,60,141,193,2,189,1
9,6,168,169,42,153,1,69,169,73,15
3,2,69,169,28,153,3,69
20475 DATA 153,5,69,169,247,153,4
,69,169,8,153,6,69,169,65,153,7,6
9,169,34,153,8,69,169
20480 DATA 32,141,2,210,169,14,14
1,3,210,76,52,145,169,58,141,193,
2,189,19,6,168,169,24,153
20485 DATA 1,69,153,6,69,169,36,1
53,2,69,153,5,69,169,67,153,3,69,
169,194,153,4,69,169
20490 DATA 8,153,7,69,169,0,153,8
,69,169,100,141,2,210,169,5,141,3
,210,76,52,145,169,55
20495 DATA 141,193,2,189,19,6,168
,169,66,153,0,69,153,7,69,169,129
,153,1,69,153,6,69,169
20500 DATA 0,153,2,69,153,3,69,15
3,4,69,153,5,69,153,8,69,169,60,1
41,2,210,169,10,141
20505 DATA 3,210,76,52,145,189,19
,6,168,169,52,141,193,2,169,0,153
,255,68,153,0,69,153,1
20510 DATA 69,153,6,69,153,7,69,1
41,2,210,141,3,210,76,52,145,254,
62,6,189,62,6,205,78
20515 DATA 6,208,8,169,5,157,67,6
,76,174,143,205,77,6,208,8,169,4,
157,67,6,76,122,143
20520 DATA 205,76,6,208,8,169,3,1
57,67,6,76,64,143,205,75,6,208,8,
169,2,157,67,6,76
20525 DATA 4,143,205,74,6,208,8,1
69,1,157,67,6,76,217,142,189,67,6
,201,5,208,3,76,174
20530 DATA 143,201,4,208,3,76,122
,143,201,3,208,3,76,64,143,201,2,
208,3,76,4,143,76,217
20535 DATA 142,169,62,141,194,2,1
89,19,6,168,169,42,153,2,70,153,6
,70,169,28,153,3,70,153
20540 DATA 5,70,169,126,153,4,70,
169,24,141,4,210,169,14,141,5,210
,76,52,145,169,60,141,194
20545 DATA 2,189,19,6,168,169,42,
153,1,70,169,73,153,2,70,169,28,1
53,3,70,153,5,70,169
20550 DATA 247,153,4,70,169,8,153
,6,70,169,65,153,7,70,169,34,153,
8,70 ,169,32,141,4,210
20555 DATA 169,14,141,5,210,76,52
,145,169,58,141,194,2,189,19,6,16
8,169,24,153,1,70,153,6
20560 DATA 70,169,36,153,2,70,153
,5,70,169,67,153,3,70,169,194,153
,4,70,169,8,153,7,70
20565 DATA 169,0,153,8,70,169,100
,141,4,210,169,5,141,5,210,76,52,
145,169,55,141,194,2,189
20570 DATA 19,6,168,169,66,153,0,
70,153,7,70,169,129,153,1,70,153,
6,70,169,0,153,2,70
20575 DATA 153,3,70,153,4,70,153,
5,70,169,60,141,4,210,169,10,141,
5,210,76,52,145,189,19
20580 DATA 6,168,169,52,141,194,2
,169,0,153,0,70,153,1,70,153,6,70
,153,7,70,141,4,210
```

Space Station Z64 cont'd

Basic Listing cont'd

```
20585 DATA 141,5,210,76,52,145,25
4,62,6,189,62,6,205,78,6,208,8,16
9,5,157,67,6,76,20
20590 DATA 145,205,77,6,208,8,169
,4,157,67,6,76,224,144,205,76,6,2
08,8,169,3,157,67,6
20595 DATA 76,166,144,205,75,6,20
8,8,169,2,157,67,6,76,106,144,205
,74,6,208,8,169,1,157
20600 DATA 67,6,76,55,144,189,67,
6,201,5,208,3,76,20,145,201,4,208
,3,76,224,144,201,3
20605 DATA 208,3,76,166,144,201,2
,208,3,76,106,144,76,55,144,169,6
2,141,195,2,189,19,6,168
20610 DATA 169,42,153,2,71,153,6,
71,169,28,153,3,71,153,5,71,169,1
26,153,4,71,169,0,153
20615 DATA 1,71,153,0,71,169,24,1
41,6,210,169,14,141,7,210,76,52,1
45,169,60,141,195,2,189
20620 DATA 19,6,168,169,42,153,1,
71,169,73,153,2,71,169,28,153,3,7
1,153,5,71,169,247,153
20625 DATA 4,71,169,8,153,6,71,16
9,65,153,7,71,169,34,153,8,71,169
,32,141,6,210,169,14
20630 DATA 141,7,210,76,52,145,16
9,58,141,195,2,189,19,6,168,169,2
4,153,1,71,153,6,71,169
20635 DATA 36,153,2,71,153,5,71,1
69,67,153,3,71,169,194,153,4,71,1
69,8,153,7,71,169,0
20640 DATA 153,8,71,169,100,141,6
,210,1 69,5,141,7,210,76,52,145,1
69,55,141,195,2,189,19,6
20645 DATA 168,169,66,153,0,71,15
3,7,71,169,129,153,1,71,153,6,71,
169,0,153,2,71,153,3
20650 DATA 71,153,4,71,153,5,71,1
69,60,141,6,210,169,10,141,7,210,
76,52,145,189,19,6,168
20655 DATA 169,52,141,195,2,169,0
,153 ,0,71,153,1,71,153,6,71,153,
7,71 ,141,6,210,141,7
20660 DATA 210,76,52,145,76,37,13
5
```

If this program seems too long to type in then send us a blank disk or cassette and $3.00 and we'll send you a copy.Send it to:ROM Magazine
P.O. BOX 252
MAPLE RIDGE,B.C.
V2X 7G1 CANADA

Assembler Listing

```
00010  .LI OFF
00011  .OR $84D0
00012  .TA $2600
00014  .TF "D:SPST.OBJ"
00020 VERT .EQ $600  ;VERT POS MML
00021 HORZ .EQ $601  ;HORZ POS MML
00022 HDIR .EQ $602  ;HORZ DIR PLO
00023 STICK .EQ $603
00025 PLAYERO .EQ $4000  ;LOC PLAYERO
00026 PLAYER1 .EQ $4100  ;LOC PLAYER1
00027 PLAYER2 .EQ $4200  ;LOC PLAYER2
00028 PLAYER3 .EQ $4300  ;LOC PLAYER3
00029 MVERT .EQ $604  ;VERT POS F MISSILE
00030 MHORZ .EQ $605  ;HORZ F MISSILE
00031 MCOUNT .EQ $607  ;MISSILE DIST COUNTER
00032 MSTICK .EQ $608  ;MISSILE DIRECTION
00033 SCORE .EQ $609
00034 PSTICK .EQ $60A  ;PLAYER DIRECTION
00035 MIP .EQ $60B  ;MISSILE IN PROGRESS FLAG
00037 MADD .EQ $60D
00038 STRIG .EQ $284  ;BUTTON 0
00040 HPLAY .EQ $60E   ;HORZ POS PLAYER
00042 VPLAY .EQ $613  ;VERT POS PLAYER
00044 CFRA .EQ $617  ;FRAME COUNTER
00046 LIV .EQ $61C  ;LIFE FLAG
00050 RND .EQ $D20A  ;RANDOMIZER
00052 WAY .EQ $620    ;E SHIP DIR
00054 MLI .EQ $62D  ;MISSILE LIFE FLAG
00055 MVPLA .EQ $625  ;POD VERT POS
00056 MHPLA .EQ $629  ;POD HORZ POS
00057 MWAY .EQ $632   ;POD DIR
00058 ECON .EQ $637   ;E SHIP COUNTER
00059 EMMC .EQ $638    ;POD SPEED
00060 MCOL .EQ $D008  ;MISSILE COL
00061 DMS .EQ $639  ;DESTROYED SHIP FLAG
00062 EPX .EQ $63A  ;EXP COUNTER
00063 EXC .EQ $63E   ;EXPLOSION COUNTER
00064 EXT .EQ $643  ;EXPLOSION TYPE
00065 SCORE1 .EQ $647
00066 NOM .EQ $648   ;NUMBER OF E SHIPS
00067 ECS .EQ $649  ;E SHIP COUNTER
00068 EXSP1 .EQ $64A  ;EXP VAR1
00069 EXSP2 .EQ $64B   ;EXP VAR2
00070 EXSP3 .EQ $64C   ;EXP VAR3
00071 EXSP4 .EQ $64D    ;EXP VAR4
00072 EXSP5 .EQ $64E   ;EXP VAR5
00073 EXSP6 .EQ $64F  ;EXP VAR6
00074 GAMS .EQ $650  ;GAME SPEED
00076 DBUS .EQ $655
00500 BEGIN LDA #62      ;SET UP PM GRAPHICS
00505  STA $22F
00510  LDA #88
00520  STA $2C0     ;COLOR PLO
00522  LDA #60
00525  STA $2C1     ;COLOR PL1
00526  LDA #112
```

Space Station Z64 cont'd

Assembler Listing cont'd

```
00530  STA $2C2     ;COLOR PL2
00532  LDA #150
00535  STA $2C3     ;COLOR PL3
00545  LDA /PLAYER0
00550  STA $D407
00555  LDA #3
00560  STA $D01D
00565  LDA #127
00570  STA HORZ      ;HORZ POS MML
00575  LDA #125
00580  STA VERT     ;VERT POS MML
00581  LDA #1
00582  STA $D01E   ;CLEAR COLLISIONS
00583  LDA #0
00584  LDY #0
00585 DD52 STA EPX,Y   ;EXP COUNTER
00586  STA EXC,Y  ;EXPLOSION COUNTER
00587  STA EXT,Y   ;EXPLOSION TYPE
00590  INY
00591  CPY #4
00592  BNE DD52
00596  STA DMS ;DESTROYED SHIP FLAG
00600  STA SCORE
00601  STA MCOUNT  ;MISSILE DIST COUNTER
00602  STA HDIR    ;HORZ DIR PLO
00604  STA MIP  ;MISSILE IN PROGRESS FLAG
00605  STA SCORE
00606  STA SCORE1
00607  LDA HORZ      ;HORZ POS MML
00610  STA $D000
00620  LDX #0
00625  LDA #0
00626  STA ECON     ;E SHIP COUNTER
00627  STA EMMC
00630 LL40 STA LIV,X  ;LIFE FLAG
00631  STA MLI,X    ;MISSILE LIFE FLAG
00632  INX
00635  CPX #4
00640  BNE LL40
00767  LDA $6A
00800  LDY #0
00805  LDA #0
00806  STA DBUS
00810 L60 STA PLAYER0+$300,Y ;CLEAR PLAYERS
00815  STA PLAYER0+$400,Y
00820  STA PLAYER1+$400,Y
00830  STA PLAYER2+$400,Y
00835  STA PLAYER3+$400,Y
00840  INY
00845  CPY #255
00850  BNE L60
00910  LDA #0
00915  STA $D208
00920  LDA #3
00925  STA $D20F
00936  LDA #100
```

```
00938  STA EXSP6       ;EXP VAR6
00940  LDA #80
00942  STA EXSP5      ;EXP VAR5
00944  LDA #60
00946  STA EXSP4          ;EXP VAR4
00948  LDA #40
00950  STA EXSP3       ;EXP VAR3
00952  LDA #20
00954  STA EXSP2      ;EXP VAR2
00956  LDA #10
00958  STA EXSP1      ;EXP VAR1
00960  LDA #4
00962  STA GAMS
00999 *
01000 START JMP ENC     ;CONTROL SECTION
01001 START1 LDA SCORE1
01004  CMP NOM    ;NUMBER OF E SHIPS
01005  BNE DD85
01007  RTS
01019 DD85 LDA #0
01020  STA MADD
01040 DD7 CLC
01042  ADC #1
01050  LDX #0
01060 DD8 INX
01070  CPX #255
01080  BNE DD8
01090  CMP GAMS
01100  BNE DD7
01110  LDA $278         ;CHECK STICK
01120  STA PSTICK  ;PLAYER DIRECTION
01130  CMP #6
01140  BNE L1
01150  JMP L2
01160 L1 CMP #7
01170  BNE L3
01180  BNE DD7
01190  JMP L2
01200 L3 CMP #5
01210  BNE L4
01220  JMP L2
01230 L4 CMP #10
01240  BNE L5
01250  JMP L6
01260 L5 CMP #11
01270  BNE L7
01280  JMP L6
01290 L7 CMP #9
01300  BNE L8
01310  JMP L6
01320 L8 JMP L9
01330 *
01340 L2 LDA #0
01350  STA HDIR      ;HORZ DIR PLO
01360  JMP L9
01370 *
```

```
01380 L6 LDA #1
01390  STA HDIR    ;HORZ DIR PLO
01400  JMP L9
01410 *
01420 L9 LDA PSTICK  ;PLAYER DIRECTION
01430  CMP #10
01440  BNE L10
01450  JMP UPM
01460 L10 CMP #14
01470  BNE L11
01480  JMP UPM
01490 L11 CMP #6
01500  BNE L12
01510  JMP UPM
01520 L12 CMP #9
01530  BNE L13
01540  JMP DOWNM
01550 L13 CMP #13
01560  BNE L14
01570  JMP DOWNM
01580 L14 CMP #5
01590  BNE L15
01600  JMP DOWNM
01610 L15 JMP CONT
01620 *
01630 UPM LDY VERT     ;VERT POS MML
01640  CPY #24
01650  BEQ L17
01660  DEC VERT
01670 L17 JMP CONT
01680 *
01690 DOWNM LDY VERT  ;VERT POS MML
01700  CPY #215
01710  BEQ L18
01720  INC VERT
01730 L18 JMP CONT
01740 *
01750 CONT JMP DRAWP
01760 *
01770 MISS LDA MIP     ;MISSILE SECTION
01780  CMP #0
01790  BEQ L29
01800  LDA MSTICK  ;MISSILE DIRECTION
01810  JMP L28
01820 L29 LDA $284
01830  CMP #0
01840  BEQ L27
01842  LDA #0
01844  STA $D200
01846  STA $D201
01850 DD80 JMP START
01860 L27 LDA $278
01861  CMP #13
01862  BEQ DD80
01863  CMP #14
01864  BEQ DD80
```

Space Station Z64 cont'd

Assembler Listing cont'd

```
01865  CMP #15
01866  BEQ DD80
01870  STA MSTICK  ;MISSILE DIRECTION
01880  LDX VERT   ;VERT POS MML
01890  STX MVERT  ;VERT POS F MISSILE
01900  LDX HORZ     ;HORZ POS MML
01901  STX MHORZ      ;HORZ F MISSILE
01902  LDX #1
01903  STX MIP  ;MISSILE IN PROGRESS FLAG
01905 L28 LDA #90
01925  STA $D200
01926  LDA #166
01927  STA $D201
01932  LDX DMS ;DESTROYED SHIP FLAG
01934  CPX #0
01936  BNE DD20
01940  INC MCOUNT  ;MISSILE DIST COUNTER
01950  LDY MCOUNT
01960  CPY #50
01970  BNE L30
01990 DD20 LDY #0
02000  STY MHORZ
02005  LDX MVERT
02010  STY MVERT  ;VERT POS F MISSILE
02020  STY MIP
02030  STY MCOUNT  ;MISSILE DIST COUNTER
02040  LDA #0
02042  STA DMS ;DESTROYED SHIP FLAG
02050  STA PLAYER0+$304,X
02060 L30 LDA MSTICK  ;MISSILE DIRECTION
02070  CMP #10
02080  BNE L31
02090  JMP MLEFT
02100 L31 CMP #11
02110  BNE L32
02120  JMP MLEFT
02130 L32 CMP #9
02140  BNE L33
02150  JMP MLEFT
02160 L33 CMP #6
02170  BNE L34
02180  JMP MRIGHT
02190 L34 CMP #7
02200  BNE L35
02210  JMP MRIGHT
02220 L35 CMP #5
02230  BNE L50
02240  JMP MRIGHT
02250 L50 JMP START
02260 *
02270 MLEFT DEC MHORZ   ;HORZ F MISSILE
02280  LDA MHORZ
02290  STA $D004
02300  JMP MDRAW
02310 *
02320 MRIGHT INC MHORZ  ;HORZ F MISSILE
```

```
02820 LL3 TYA
02830  CMP VPLAY,X
02840  BEQ LL2
02850  INY
02860  CPY #29
02870  BNE LL3
02880  LDY #255
02890 LL5 TYA
02900  CMP VPLAY,X     ;VERT POS PLAYER
02910  BEQ LL4
02920  DEY
02930  CPY #215
02940  BNE LL5
02950  JMP GEN1
02960 *
02970 LL2 LDA #20
02980  STA VPLAY,X     ;VERT POS PLAYER
02990  JMP GEN1
03000 *
03010 LL4 LDA #215
03020  STA VPLAY,X     ;VERT POS PLAYER
03030  JMP GEN1
03040 *
03050 GEN1 LDA RND
03060  BPL LL7
03070  LDA #30
03080  STA HPLAY,X     ;HORZ POS PLAYER
03090  JMP MOVPLAY
03100 LL7 LDA #230
03110  STA HPLAY,X     ;HORZ POS PLAYER
03120  JMP MOVPLAY
03130 *
03140 MOVPLAY LDA HPLAY,X  ;HORZ POS PLAYER
03150  CMP #127
03160  BNE LL9
03170  JMP CHECK
03180 LL9 BMI MRIGHTP
03190  JMP MLEFTP
03200 *
03210 *
03220 MRIGHTP INC HPLAY,X  ;HORZ POS PLAYER
03230  LDA #1
03240  STA WAY,X   ;E SHIP DIR
03250  JMP CHECK
03260 *
03270 MLEFTP DEC HPLAY,X   ;HORZ POS PLAYER
03280  LDA #2
03290  STA WAY,X   ;E SHIP DIR
03300  JMP CHECK
03310 *
03320 CHECK LDA HPLAY,X    ;HORZ POS PLAYER
03330  CMP #127
03340  BNE DRPLAY
03342  LDA #1
03344  STA DBUS
03350  RTS
```

```
03920  STA PLAYER1+$407,Y
03930  LDA #22
03940  STA PLAYER1+$408,Y
03950  LDA #188
03960  STA PLAYER1+$409,Y
03970  JMP ENSECP1
03980 *
03990 EPL1F2 LDA VPLAY,X   ;DRAW PLAYER1 F2
04000  TAY
04010  LDA #0
04020  STA PLAYER1+$3FE,Y
04030  STA PLAYER1+$40B,Y
04040  LDA #10
04050  STA PLAYER1+$3FF,Y
04060  LDA #24
04070  STA PLAYER1+$400,Y
04080  LDA #32
04090  STA PLAYER1+$401,Y
04100  STA PLAYER1+$402,Y
04110  STA PLAYER1+$40A,Y
04120  LDA #198
04130  STA PLAYER1+$403,Y
04140  LDA #155
04150  STA PLAYER1+$404,Y
04160  LDA #225
04170  STA PLAYER1+$405,Y
04180  LDA #83
04190  STA PLAYER1+$406,Y
04200  LDA #5
04210  STA PLAYER1+$407,Y
04220  LDA #4
04230  STA PLAYER1+$408,Y
04240  LDA #26
04250  STA PLAYER1+$409,Y
04260  JMP ENSECP1
04270 *
04280 EPL1F4 LDA VPLAY,X   ;DRAW PLAYER1 F4
04290  TAY
04300  LDA #0
04310  STA PLAYER1+$3FE,Y
04320  STA PLAYER1+$40B,Y
04330  LDA #8
04340  STA PLAYER1+$3FF,Y
04350  LDA #116
04360  STA PLAYER1+$400,Y
04370  LDA #38
04380  STA PLAYER1+$401,Y
04390  LDA #98
04400  STA PLAYER1+$402,Y
04410  LDA #49
04420  STA PLAYER1+$403,Y
04430  LDA #24
04440  STA PLAYER1+$404,Y
04450  LDA #142
04460  STA PLAYER1+$405,Y
04470  LDA #4
```

Space Station Z64 cont'd
Assembler Listing cont'd

```
02330  LDA MHORZ
02340  STA $D004
02350  JMP MDRAW
02360 *
02370 MDRAW LDA MADD
02380  CLC
02390  ADC #3
02400  STA MADD
02410  LDY MVERT  ;VERT POS F MISSILE
02420  LDA #0
02430  STA PLAYER0+$302,Y
02440  STA PLAYER0+$303,Y
02450  STA PLAYER0+$305,Y
02460  STA PLAYER0+$306,Y
02470  LDA #3
02480  STA PLAYER0+$304,Y
02490  JMP START
02500 *
02510 ENC INC ECON   ;E SHIP COUNTER
02580 *
02590 ENSECP LDX #0
02600 ENSECP1 INX
02610  CPX #4
02620  BNE LPLAYER
02622  LDA ECON    ;E SHIP COUNTER
02623  CMP ECS   ;E SHIP COUNTER
02624  BNE DD88
02625  LDA #0
02626  STA ECON
02630 DD88 JMP EMM
02640 *
02650 *
02660 LPLAYER LDA EPX,X
02662  CMP #0
02664  BEQ DD50
02668  JMP EXPL
02670 DD50 LDA ECON  ;E SHIP COUNTER
02671  CMP ECS
02672  BEQ DD5
02673  JMP ENSECP1
02674 DD5 LDA LIV,X   ;LIFE FLAG
02680  CMP #0
02690  BNE LL1
02700  JMP GEN
02710 LL1 LDY #0
02720 LLL1 INY
02730  CPY #200
02740  BNE LLL1
02750  JMP MOVPLAY
02760 *
02770 GEN INC SCORE1
02775  LDA #1
02780  STA LIV,X    ;LIFE FLAG
02790  LDA RND     ;RANDOMIZER
02800  STA VPLAY,X    ;VERT POS PLAYER
02810  LDY #0
```

```
03360 *
03370 DRPLAY LDA HPLAY,X   ;HORZ POS PLAYER
03380  STA $D000,X
03390  CPX #1
03400  BNE LL21
03410  JMP EPL1
03420 LL21 CPX #2
03430  BNE LL22
03440  JMP EPL2
03450 LL22 JMP EPL3
03460 *
03470 EPL1 LDA #60
03472  STA $2C1
03480  LDA CFRA,X  ;FRAME COUNTER
03482  CLC
03490  ADC #80
03500  STA CFRA,X  ;FRAME COUNTER
03530  LDA CFRA,X  ;FRAME COUNTER
03540  LSR
03550  LSR
03560  LSR
03570  LSR
03580  LSR
03590  LSR
03600  CMP #0
03610  BNE LL25
03620  JMP EPL1F1
03630 LL25 CMP #2
03640  BNE LL26
03650  JMP EPL1F3
03660 LL26 CMP #3
03670  BNE LL27
03680  JMP EPL1F4
03690 LL27 JMP EPL1F2
03700 *
03710 *
03720 *
03730 EPL1F1 LDA VPLAY,X   ;DRAW PLAYER1 F1
03740  TAY
03750  LDA #0
03760  STA PLAYER1+$3FF,Y
03770  STA PLAYER1+$40A,Y
03780  LDA #125
03790  STA PLAYER1+$400,Y
03800  LDA #232
03810  STA PLAYER1+$401,Y
03820  LDA #128
03830  STA PLAYER1+$402,Y
03840  STA PLAYER1+$403,Y
03850  LDA #236
03860  STA PLAYER1+$404,Y
03870  LDA #126
03880  STA PLAYER1+$405,Y
03890  LDA #50
03900  STA PLAYER1+$406,Y
03910  LDA #2
```

```
04480  STA PLAYER1+$406,Y
04490  LDA #166
04500  STA PLAYER1+$407,Y
04510  LDA #104
04520  STA PLAYER1+$408,Y
04530  LDA #52
04540  STA PLAYER1+$409,Y
04550  LDA #96
04560  STA PLAYER1+$40A,Y
04570  JMP ENSECP1
04580 *
04590 EPL1F3 LDA VPLAY,X   ;DRAW PLAYER1 F3
04600  TAY
04610  LDA #0
04620  STA PLAYER1+$3FF,Y
04630  STA PLAYER1+$40A,Y
04640  STA PLAYER1+$409,Y
04650  STA PLAYER1+$400,Y
04660  STA PLAYER1+$401,Y
04670  LDA #134
04680  STA PLAYER1+$402,Y
04690  LDA #11
04700  STA PLAYER1+$403,Y
04710  LDA #201
04720  STA PLAYER1+$404,Y
04730  STA PLAYER1+$406,Y
04740  LDA #147
04750  STA PLAYER1+$405,Y
04760  LDA #104
04770  STA PLAYER1+$407,Y
04780  LDA #49
04790  STA PLAYER1+$408,Y
04800  JMP ENSECP1
04810 *
04820 EPL2 LDA #117
04821  STA $2C2
04822  LDA CFRA,X  ;FRAME COUNTER
04823  CLC
04824  ADC #35
04825  STA CFRA,X
04830  LDA WAY,X   ;E SHIP DIR
04840  CMP #1
04850  BNE LL60
04860  JMP WAY2
04870 LL60 LDA CFRA,X  ;FRAME COUNTER
04880  LSR
04890  LSR
04900  LSR
04910  LSR
04920  LSR
04930  LSR
04940  CMP #0
04950  BNE LL32
04960  JMP EPL2F1
04970 LL32 CMP #2
04980  BNE LL33
```

Assembler Listing cont'd

```
04990  JMP EPL2F3
05000 LL33 CMP #3
05010  BNE LL34
05020  JMP EPL2F2
05030 LL34 JMP EPL2F2
05040 *
05050 EPL2F1 LDA VPLAY,X    ;DRAW PLAYER2 F1
05060  TAY
05070  LDA #0
05080  STA PLAYER2+$3FF,Y
05090  STA PLAYER2+$407,Y
05100  LDA #1
05110  STA PLAYER2+$400,Y
05120  STA PLAYER2+$406,Y
05130  LDA #7
05140  STA PLAYER2+$401,Y
05150  STA PLAYER2+$405,Y
05160  LDA #110
05170  STA PLAYER2+$402,Y
05180  STA PLAYER2+$404,Y
05190  LDA #248
05200  STA PLAYER2+$403,Y
05210  JMP ENSECP1
05220 EPL2F3 LDA VPLAY,X    ;DRAW PLAYER2 F3
05230  TAY
05240  LDA #0
05250  STA PLAYER2+$401,Y
05260  STA PLAYER2+$405,Y
05270  LDA #96
05280  STA PLAYER2+$402,Y
05290  STA PLAYER2+$404,Y
05300  LDA #255
05310  STA PLAYER2+$403,Y
05320  JMP ENSECP1
05330 EPL2F2 LDA VPLAY,X    ;DRAW PLAYER2 F2
05340  TAY
05350  LDA #0
05360  STA PLAYER2+$400,Y
05370  STA PLAYER2+$401,Y
05380  STA PLAYER2+$405,Y
05390  STA PLAYER2+$406,Y
05400  LDA #111
05410  STA PLAYER2+$402,Y
05420  STA PLAYER2+$404,Y
05430  LDA #248
05440  STA PLAYER2+$403,Y
05450  JMP ENSECP1
05460 *
05470 *
05480 WAY2 LDA CFRA,X  ;FRAME COUNTER
05490  LSR
05500  LSR
05510  LSR
05520  LSR
05530  LSR
05540  LSR
```

```
06102  STA $2C3
06108  LDA VPLAY,X  ;VERT POS PLAYER
06110  TAY
06120  LDA #0
06130  STA PLAYER3+$3FF,Y
06140  STA PLAYER3+$404,Y
06150  LDA #8
06160  STA PLAYER3+$400,Y
06170  LDA #28
06180  STA PLAYER3+$401,Y
06190  LDA #119
06200  STA PLAYER3+$402,Y
06210  LDA #62
06220  STA PLAYER3+$403,Y
06230  JMP ENSECP1
06240 *
06250 DRAWP LDA HDIR  ;HORZ DIR PLO
06260  CMP #0
06270  BNE L20
06280  JMP DRAWPF2
06290 L20 LDY VERT   ;VERT POS MML
06300  LDA #0
06310  STA PLAYER0+$400,Y
06320  STA PLAYER0+$408,Y
06330  LDA #4
06340  STA PLAYER0+$401,Y
06350  STA PLAYER0+$407,Y
06360  LDA #12
06370  STA PLAYER0+$402,Y
06380  STA PLAYER0+$406,Y
06390  LDA #254
06400  STA PLAYER0+$403,Y
06410  STA PLAYER0+$405,Y
06420  LDA #31
06430  STA PLAYER0+$404,Y
06440  JMP MISS
06450 DRAWPF2 LDY VERT  ;VERT POS MML
06460  LDA #0
06470  STA PLAYER0+$400,Y
06480  STA PLAYER0+$408,Y
06490  STA PLAYER0+$3FF,Y
06500  STA PLAYER0+$409,Y
06510  LDA #32
06520  STA PLAYER0+$401,Y
06530  STA PLAYER0+$407,Y
06540  LDA #48
06550  STA PLAYER0+$402,Y
06560  STA PLAYER0+$406,Y
06570  LDA #127
06580  STA PLAYER0+$403,Y
06590  STA PLAYER0+$405,Y
06600  LDA #248
06610  STA PLAYER0+$404,Y
06620  JMP MISS
06630 *
06640 EMM INC EMMC
```

```
07210 *
07220 MMR LDA #2
07230  STA MWAY,X  ;POD DIR
07240  JMP MCH
07250 *
07260 MML LDA #1
07270  STA MWAY,X
07280  JMP MCH
07290 *
07300 MCH LDA MHPLA,X   ;POD HORZ POS
07310  CMP #10
07320  BEQ DD
07330  CMP #245
07340  BEQ DD
07350  JMP MDRA
07360 DD LDA #0
07370  STA MLI,X   ;MISSILE LIFE FLAG
07380  LDA MVPLA,X  ;POD VERT POS
07390  TAY
07400  LDA #0
07410  STA PLAYER0+$300,Y
07420  STA PLAYER0+$301,Y
07430  STA PLAYER0+$302,Y
07440  JMP MISEC1
07450 *
07460 MDRA LDA MWAY,X
07470  CMP #2
07480  BNE DD1
07490  INC MHPLA,X   ;POD HORZ POS
07500  JMP MDRA1
07510 DD1 DEC MHPLA,X   ;POD HORZ POS

07520 MDRA1 LDA MHPLA,X
07530  STA $D004,X
07540  CPX #1
07550  BNE M8
07560  JMP MSL1
07570 M8 CPX #2
07580  BNE M9
07590  JMP MSL2
07600 M9 JMP MSL3
07610 *
07620 MSL1 LDA MVPLA,X
07630  TAY
07640  LDA MWAY,X  ;POD DIR
07650  CMP #2
07660  BNE M10
07670  LDA #8
07680  JMP M11
07690 M10 LDA #4
07700 M11 STA PLAYER0+$300,Y  ;DRAW MISSILE
07710  STA PLAYER0+$302,Y
07720  LDA #12
07730  STA PLAYER0+$301,Y
07740  JMP MISEC1
07750 *
07760 MSL2 LDA MVPLA,X
```

Space Station Z64 cont'd

Assembler Listing cont'd

```
05550  CMP #0
05560  BNE LL50
05570  JMP EPL2W1
05580 LL50 CMP #2
05590  BNE LL51
05600  JMP EPL2W3
05610 LL51 CMP #3
05620  BNE LL52
05630  JMP EPL2W2
05640 LL52 JMP EPL2W2
05650 *
05660 EPL2W1 LDA VPLAY,X   ;DRAW PLAYER2 F1W
05670  TAY
05680  LDA #0
05690  STA PLAYER2+$3FF,Y
05700  STA PLAYER2+$407,Y
05710  LDA #128
05720  STA PLAYER2+$400,Y
05730  STA PLAYER2+$406,Y
05740  LDA #224
05750  STA PLAYER2+$401,Y
05760  STA PLAYER2+$405,Y
05770  LDA #118
05780  STA PLAYER2+$402,Y
05790  STA PLAYER2+$404,Y
05800  LDA #31
05810  STA PLAYER2+$403,Y
05820  JMP ENSECP1
05830 EPL2W2 LDA VPLAY,X   ;DRAW PLAYER2 F2W
05840  TAY
05850  LDA #0
05860  STA PLAYER2+$401,Y
05870  STA PLAYER2+$405,Y
05880  STA PLAYER2+$400,Y
05890  STA PLAYER2+$406,Y
05900  LDA #246
05910  STA PLAYER2+$402,Y
05920  STA PLAYER2+$404,Y
05930  LDA #31
05940  STA PLAYER2+$403,Y
05950  JMP ENSECP1
05960 EPL2W3 LDA VPLAY,X   ;DRAW PLAYER2 F2W
05970  TAY
05980  LDA #0
05990  STA PLAYER2+$401,Y
06000  STA PLAYER2+$405,Y
06010  LDA #6
06020  STA PLAYER2+$402,Y
06030  STA PLAYER2+$404,Y
06040  LDA #255
06050  STA PLAYER2+$403,Y
06060  JMP ENSECP1
06070 *
06080 *
06090 *
06100 EPL3 LDA #150
```

```
06650  LDA EMMC
06660  CMP #8
06670  BEQ DD6
06680  JMP COL
06690 DD6 LDA #0
06700  STA EMMC
06710 *
06720 MISEC LDX #0    ;MISSILE SECTION
06730 MISEC1 INX
06740  CPX #4
06750  BNE MSLI
06760  JMP COL
06770 *
06780 MSLI LDA MLI,X  ;MISSILE LIFE FLAG
06790  CMP #0
06800  BNE M1
06810  JMP MGEN
06820 M1 JMP MCH
06830 *
06840 MGEN LDA #1
06850  STA MLI,X   ;MISSILE LIFE FLAG
06860  LDA RND    ;RANDOMIZER
06870  STA MVPLA,X  ;POD VERT POS
06880  LSR
06890  LSR
06900  LSR
06910  LSR
06920  CMP #0
06930  BNE M2
06940  LDA #29
06950  STA MVPLA,X  ;POD VERT POS
06960  JMP MGEN1
06970 M2 CMP #5
06980  BNE MGEN1
06990  LDA #210
07000  STA MVPLA,X  ;POD VERT POS
07010 *
07020 MGEN1 LDA RND
07030  BPL M6
07040  LDA RND    ;RANDOMIZER
07050  BPL DD2
07060  LDA #26
07070  JMP DD3
07080 DD2 LDA #19
07090  JMP DD3
07100 M6 LDA RND  ;RANDOMIZER
07110  BPL DD4
07120  LDA #228
07130  JMP DD3
07140 DD4 LDA #235
07150 DD3 STA MHPLA,X  ;POD HORZ POS
07160  JMP MMOV
07170 *
07180 MMOV LDA MHPLA,X  ;POD HORZ POS
07190  BMI MML
07200  JMP MMR
```

```
07770  TAY
07780  LDA MWAY,X
07790  CMP #2
07800  BNE M12
07810  LDA #32
07820  JMP M14
07830 M12 LDA #16
07840 M14 STA PLAYER0+$300,Y
07850  STA PLAYER0+$302,Y
07860  LDA #48
07870  STA PLAYER0+$301,Y
07880  JMP MISEC1
07890 *
07900 MSL3 LDA MVPLA,X
07910  TAY
07920  LDA MWAY,X  ;POD DIR
07930  CMP #2
07940  BNE M15
07950  LDA #128
07960  JMP M16
07970 M15 LDA #64
07980 M16 STA PLAYER0+$300,Y
07990  STA PLAYER0+$302,Y
08000  LDA #192
08010  STA PLAYER0+$301,Y
08020  JMP MISEC1
08030 *
08040 *
08050 COL LDA MCOL   ;COLL
08060  LSR
08070  LSR
08080  LSR
08090  CMP #0
08100  BEQ W1
08110  LDX #3
08120  LDA #0
08121  STA LIV,X   ;LIFE FLAG
08122  STX DMS ;DESTROYED SHIP FLAG
08123  LDA VPLAY,X  ;VERT POS PLAYER
08124  TAY
08132  LDA #1
08133  STA EPX,X  ;EXP COUNTER
08140  LDA MCOL    ;MISSILE COL
08150  SEC
08160  SBC #8
08170  JMP W2
08180 W1 LDA MCOL  ;MISSILE COL
08190 W2 LSR
08200  CMP #2
08210  BEQ W3
08220  CMP #1
08230  BEQ W4
08240  CMP #0
08250  BNE W3
08260  JMP POD
08270 W3 LDX #2
```

```
08280  TAY
08282  LDA #1
08283  STA EPX,X  ;EXP COUNTER
08290  LDA #0
08291  STA LIV,X   ;LIFE FLAG
08292  STX DMS ;DESTROYED SHIP FLAG
08310  TYA
08320  CMP #3
08330  BEQ W4
08340  JMP POD
08350 *
08360 W4 LDX #1
08362  LDA #1
08363  STA EPX,X  ;EXP COUNTER
08370  LDA #0
08380  STA LIV,X   ;LIFE FLAG
08385  STX DMS ;DESTROYED SHIP FLAG
08399  JMP POD
08400 *
08410 POD LDA $D009
08420  CMP #1
08430  BNE W5
08440  LDX #1
08450  JMP W6
08460 W5 LDA $D00A
08470  CMP #1
08480  BNE W7
08490  LDX #2
08500  JMP W6
08510 W7 LDA $D00B
08520  CMP #1
08530  BNE W8
08540  LDX #3
08550 *
08560 W6 INC SCORE
08570  LDA #0
08580  STA MLI,X   ;MISSILE LIFE FLAG
08585  STA $D01F
08590  LDA MVPLA,X  ;POD VERT POS
08600  TAY
08610  LDA #0
08620  STA PLAYER0+$300,Y
08630  STA PLAYER0+$301,Y
08640  STA PLAYER0+$302,Y
08650 W8 LDA #1
08660  STA $D01E
08670  JMP START1
20340 EXPL LDA EXC,X    ;EXPLOSION SECTION
20341  CMP EXSP6      ;EXP VAR6
20342  BNE DD51
20343  LDA #0
20344  STA EPX,X  ;EXP COUNTER
20345  STA EXC,X  ;EXPLOSION COUNTER
20346  STA EXT,X   ;EXPLOSION TYPE
20347  JMP WNG
20350 DD51 CPX #1
```

```
20940  LDA #28
20950  STA PLAYER1+$403,Y
20960  STA PLAYER1+$405,Y
20970  LDA #126
20980  STA PLAYER1+$404,Y
20981  LDA #0
20982  STA PLAYER1+$407,Y
20983  STA PLAYER1+$408,Y
20984  STA PLAYER1+$409,Y
20985  STA PLAYER1+$40A,Y
20986  STA PLAYER1+$400,Y
20987  STA PLAYER1+$401,Y
20988  STA PLAYER1+$3FF,Y
20990  LDA #24
21000  STA $D202
21010  LDA #14
21020  STA $D203
21030  JMP WNG
21040 *
21050 EX1F2 LDA #60  ;EXPLOSION FRAME2
21060  STA $2C1
21070  LDA VPLAY,X  ;VERT POS PLAYER
21080  TAY
21090  LDA #42
21100  STA PLAYER1+$401,Y
21110  LDA #73
21120  STA PLAYER1+$402,Y
21130  LDA #28
21140  STA PLAYER1+$403,Y
21150  STA PLAYER1+$405,Y
21160  LDA #247
21170  STA PLAYER1+$404,Y
21180  LDA #8
21190  STA PLAYER1+$406,Y
21200  LDA #65
21210  STA PLAYER1+$407,Y
21212  LDA #34
21213  STA PLAYER1+$408,Y
21220  LDA #32
21230  STA $D202
21240  LDA #14
21250  STA $D203
21260  JMP WNG
21270 *
21280 EX1F3 LDA #58  ;EXPLOSION FRAME3
21290  STA $2C1
21300  LDA VPLAY,X  ;VERT POS PLAYER
21310  TAY
21320  LDA #24
21330  STA PLAYER1+$401,Y
21340  STA PLAYER1+$406,Y
21350  LDA #36
21360  STA PLAYER1+$402,Y
21370  STA PLAYER1+$405,Y
21380  LDA #67
21390  STA PLAYER1+$403,Y
```

```
21940 A10 CMP EXSP4    ;EXP VAR4
21950  BNE A11
21960  LDA #4
21970  STA EXT,X
21980  JMP EX2F4
21990 A11 CMP EXSP3   ;EXP VAR3
22000  BNE A12
22010  LDA #3
22020  STA EXT,X   ;EXPLOSION TYPE
22030  JMP EX2F3
22040 A12 CMP EXSP2  ;EXP VAR2
22050  BNE A13
22060  LDA #2
22070  STA EXT,X
22080  JMP EX2F2
22090 A13 CMP EXSP1  ;EXP VAR1
22100  BNE A14
22110  LDA #1
22120  STA EXT,X   ;EXPLOSION TYPE
22130  JMP EX2F1
22140 A14 LDA EXT,X
22150  CMP #5
22160  BNE A50
22170  JMP EX2F5
22180 A50 CMP #4
22190  BNE A51
22200  JMP EX2F4
22210 A51 CMP #3
22220  BNE A52
22230  JMP EX2F3
22240 A52 CMP #2
22250  BNE A53
22260  JMP EX2F2
22270 A53 JMP EX2F1
22280 *
22290 EX2F1 LDA #62   ;EXPLOSION FRAME1
22300  STA $2C2
22310  LDA VPLAY,X  ;VERT POS PLAYER
22320  TAY
22330  LDA #42
22340  STA PLAYER2+$402,Y
22350  STA PLAYER2+$406,Y
22360  LDA #28
22370  STA PLAYER2+$403,Y
22380  STA PLAYER2+$405,Y
22390  LDA #126
22400  STA PLAYER2+$404,Y
22410  LDA #24
22420  STA $D204
22430  LDA #14
22440  STA $D205
22450  JMP WNG
22460 *
22470 EX2F2 LDA #60   ;EXPLOSION FRAME2
22480  STA $2C2
22490  LDA VPLAY,X  ;VERT POS PLAYER
```

Assembler Listing cont'd

```
20360  BNE A1
20370  JMP B1
20380 A1 CPX #2
20390  BNE A2
20400  JMP B2
20410 A2 JMP B3
20420 *
20430 B1 INC EXC,X  ;EXPLOSION COUNTER
20440  LDA EXC,X
20470  CMP EXSP5     ;EXP VAR5
20480  BNE A4
20490  LDA #5
20500  STA EXT,X   ;EXPLOSION TYPE
20510  JMP EX1F5
20520 A4 CMP EXSP4     ;EXP VAR4
20530  BNE A5
20540  LDA #4
20550  STA EXT,X
20560  JMP EX1F4
20570 A5 CMP EXSP3     ;EXP VAR3
20580  BNE A6
20590  LDA #3
20600  STA EXT,X   ;EXPLOSION TYPE
20610  JMP EX1F3
20620 A6 CMP EXSP2    ;EXP VAR2
20630  BNE A7
20640  LDA #2
20650  STA EXT,X   ;EXPLOSION TYPE
20660  JMP EX1F2
20670 A7 CMP EXSP1    ;EXP VAR1
20680  BNE A8
20690  LDA #1
20700  STA EXT,X   ;EXPLOSION TYPE
20710  JMP EX1F1
20720 A8 LDA EXT,X
20730  CMP #5
20740  BNE A60
20750  JMP EX1F5
20760 A60 CMP #4
20770  BNE A61
20780  JMP EX1F4
20790 A61 CMP #3
20800  BNE A62
20810  JMP EX1F3
20820 A62 CMP #2
20830  BNE A63
20840  JMP EX1F2
20850 A63 JMP EX1F1
20860 *
20870 EX1F1 LDA #62  ;EXPLOSION FRAME1
20880  STA $2C1
20890  LDA VPLAY,X  ;VERT POS PLAYER
20900  TAY
20910  LDA #42
20920  STA PLAYER1+$402,Y
20930  STA PLAYER1+$406,Y
```

```
21400  LDA #194
21410  STA PLAYER1+$404,Y
21420  LDA #8
21430  STA PLAYER1+$407,Y
21432  LDA #0
21433  STA PLAYER1+$408,Y
21440  LDA #100
21450  STA $D202
21460  LDA #5
21470  STA $D203
21480  JMP WNG
21490 *
21500 EX1F4 LDA #55  ;EXPLOSION FRAME4
21510  STA $2C1
21520  LDA VPLAY,X  ;VERT POS PLAYER
21530  TAY
21540  LDA #66
21550  STA PLAYER1+$400,Y
21560  STA PLAYER1+$407,Y
21570  LDA #129
21580  STA PLAYER1+$401,Y
21590  STA PLAYER1+$406,Y
21600  LDA #0
21610  STA PLAYER1+$402,Y
21620  STA PLAYER1+$403,Y
21630  STA PLAYER1+$404,Y
21640  STA PLAYER1+$405,Y
21642  STA PLAYER1+$408,Y
21650  LDA #60
21660  STA $D202
21670  LDA #10
21680  STA $D203
21690  JMP WNG
21700 *
21710 EX1F5 LDA VPLAY,X  ;EXPLOSION FRAME5
21720  TAY
21730  LDA #52
21740  STA $2C1
21750  LDA #0
21752  STA PLAYER1+$3FF,Y
21760  STA PLAYER1+$400,Y
21770  STA PLAYER1+$401,Y
21780  STA PLAYER1+$406,Y
21790  STA PLAYER1+$407,Y
21800  STA $D202
21810  STA $D203
21820  JMP WNG
21830 *
21840 *
21850 B2 INC EXC,X  ;EXPLOSION COUNTER
21860  LDA EXC,X
21890  CMP EXSP5     ;EXP VAR5
21900  BNE A10
21910  LDA #5
21920  STA EXT,X    ;EXPLOSION TYPE
21930  JMP EX2F5
```

```
22500  TAY
22510  LDA #42
22520  STA PLAYER2+$401,Y
22530  LDA #73
22540  STA PLAYER2+$402,Y
22550  LDA #28
22560  STA PLAYER2+$403,Y
22570  STA PLAYER2+$405,Y
22580  LDA #247
22590  STA PLAYER2+$404,Y
22600  LDA #8
22610  STA PLAYER2+$406,Y
22620  LDA #65
22630  STA PLAYER2+$407,Y
22632  LDA #34
22634  STA PLAYER2+$408,Y
22640  LDA #32
22650  STA $D204
22660  LDA #14
22670  STA $D205
22680  JMP WNG
22690 *
22700 EX2F3 LDA #58   ;EXPLOSION FRAME3
22710  STA $2C2
22720  LDA VPLAY,X  ;VERT POS PLAYER
22730  TAY
22740  LDA #24
22750  STA PLAYER2+$401,Y
22760  STA PLAYER2+$406,Y
22770  LDA #36
22780  STA PLAYER2+$402,Y
22790  STA PLAYER2+$405,Y
22800  LDA #67
22810  STA PLAYER2+$403,Y
22820  LDA #194
22830  STA PLAYER2+$404,Y
22840  LDA #8
22850  STA PLAYER2+$407,Y
22855  LDA #0
22856  STA PLAYER2+$408,Y
22860  LDA #100
22870  STA $D204
22880  LDA #5
22890  STA $D205
22900  JMP WNG
22910 *
22920 EX2F4 LDA #55   ;EXPLOSION FRAME4
22930  STA $2C2
22940  LDA VPLAY,X  ;VERT POS PLAYER
22950  TAY
22960  LDA #66
22970  STA PLAYER2+$400,Y
22980  STA PLAYER2+$407,Y
22990  LDA #129
23000  STA PLAYER2+$401,Y
23010  STA PLAYER2+$406,Y
```

Space Station Z64 cont'd

Assembler Listing cont'd

```
23020  LDA #0
23030  STA PLAYER2+$402,Y
23040  STA PLAYER2+$403,Y
23050  STA PLAYER2+$404,Y
23060  STA PLAYER2+$405,Y
23070  LDA #60
23080  STA $D204
23090  LDA #10
23100  STA $D205
23110  JMP WNG
23120 *
23130 EX2F5 LDA VPLAY,X
23140  TAY
23150  LDA #52
23160  STA $2C2
23170  LDA #0
23180  STA PLAYER2+$400,Y
23190  STA PLAYER2+$401,Y
23200  STA PLAYER2+$406,Y
23210  STA PLAYER2+$407,Y
23220  STA $D204
23230  STA $D205
23240  JMP WNG
23250 *
23260 *
23270 B3 INC EXC,X  ;EXPLOSION COUNTER
23280  LDA EXC,X
23290  CMP EXSP5     ;EXP VAR5
23300  BNE A114
23310  LDA #5
23320  STA EXT,X   ;EXPLOSION TYPE
23330  JMP EX3F5
23340 A114 CMP EXSP4   ;EXP VAR4
23350  BNE A15
23360  LDA #4
23370  STA EXT,X
23380  JMP EX3F4
23390 A15 CMP EXSP3   ;EXP VAR3
23400  BNE A16
23410  LDA #3
23420  STA EXT,X   ;EXPLOSION TYPE
23430  JMP EX3F3
23440 A16 CMP EXSP2  ;EXP VAR2
23450  BNE A17
23460  LDA #2
23470  STA EXT,X
23480  JMP EX3F2
23490 A17 CMP EXSP1  ;EXP VAR1
23500  BNE A18
23510  LDA #1
23520  STA EXT,X   ;EXPLOSION TYPE
23530  JMP EX3F1
23540 A18 LDA EXT,X
23550  CMP #5
23560  BNE A70
23570  JMP EX3F5
```

```
23580 A70 CMP #4
23590  BNE A71
23600  JMP EX3F4
23610 A71 CMP #3
23620  BNE A72
23630  JMP EX3F3
23640 A72 CMP #2
23650  BNE A73
23660  JMP EX3F2
23670 A73 JMP EX3F1
23680 *
23690 EX3F1 LDA #62    ;EXPLOSION FRAME1
23700  STA $2C3
23710  LDA VPLAY,X  ;VERT POS PLAYER
23720  TAY
23730  LDA #42
23740  STA PLAYER3+$402,Y
23750  STA PLAYER3+$406,Y
23760  LDA #28
23770  STA PLAYER3+$403,Y
23780  STA PLAYER3+$405,Y
23790  LDA #126
23800  STA PLAYER3+$404,Y
23802  LDA #0
23803  STA PLAYER3+$401,Y
23804  STA PLAYER3+$400,Y
23810  LDA #24
23820  STA $D206
23830  LDA #14
23840  STA $D207
23850  JMP WNG
23860 *
23870 EX3F2 LDA #60    ;EXPLOSION FRAME2
23880  STA $2C3
23890  LDA VPLAY,X  ;VERT POS PLAYER
23900  TAY
23910  LDA #42
23920  STA PLAYER3+$401,Y
23930  LDA #73
23940  STA PLAYER3+$402,Y
23950  LDA #28
23960  STA PLAYER3+$403,Y
23970  STA PLAYER3+$405,Y
23980  LDA #247
23990  STA PLAYER3+$404,Y
24000  LDA #8
24010  STA PLAYER3+$406,Y
24020  LDA #65
24030  STA PLAYER3+$407,Y
24032  LDA #34
24034  STA PLAYER3+$408,Y
24040  LDA #32
24050  STA $D206
24060  LDA #14
24070  STA $D207
24080  JMP WNG
```

```
24090 *
24100 EX3F3 LDA #58     ;EXPLOSION FRAME3
24110  STA $2C3
24120  LDA VPLAY,X  ;VERT POS PLAYER
24130  TAY
24140  LDA #24
24150  STA PLAYER3+$401,Y
24160  STA PLAYER3+$406,Y
24170  LDA #36
24180  STA PLAYER3+$402,Y
24190  STA PLAYER3+$405,Y
24200  LDA #67
24210  STA PLAYER3+$403,Y
24220  LDA #194
24230  STA PLAYER3+$404,Y
24240  LDA #8
24250  STA PLAYER3+$407,Y
24252  LDA #0
24254  STA PLAYER3+$408,Y
24260  LDA #100
24270  STA $D206
24280  LDA #5
24290  STA $D207
24300  JMP WNG
24310 *
24320 EX3F4 LDA #55     ;EXPLOSION FRAME4
24330  STA $2C3
24340  LDA VPLAY,X  ;VERT POS PLAYER
24350  TAY
24360  LDA #66
24370  STA PLAYER3+$400,Y
24380  STA PLAYER3+$407,Y
24390  LDA #129
24400  STA PLAYER3+$401,Y
24410  STA PLAYER3+$406,Y
24420  LDA #0
24430  STA PLAYER3+$402,Y
24440  STA PLAYER3+$403,Y
24450  STA PLAYER3+$404,Y
24460  STA PLAYER3+$405,Y
24470  LDA #60
24480  STA $D206
24490  LDA #10
24500  STA $D207
24510  JMP WNG
24520 *
24530 EX3F5 LDA VPLAY,X   ;EXPLOSION FRAME5
24540  TAY
24550  LDA #52
24560  STA $2C3
24570  LDA #0
24580  STA PLAYER3+$400,Y
24590  STA PLAYER3+$401,Y
24600  STA PLAYER3+$406,Y
24603  STA PLAYER3+$407,Y
24610  STA $D206
24620  STA $D207
24630  JMP WNG
24640 *
24650 WNG JMP ENSECP1
```

YELLOW BRICK ROAD

By PETER ELLISON

The Cassette, like the disk is a good and inexpensive way to store computer programs, and when used in the right way, can even become an asset to those already owning a disk drive.

The Cassette Handler

The Cassette Handler is similar to the disk manager in that files can be written to or read from the cassette. Programs can be loaded by using BASIC or using an autoload command and the buffer can be opened for get and put commands. One thing you must remember, there are still a few big differences between a disk drive and a cassette. One is:the price. The second is: when a program is saved to cassette it is saved in one continuous stream. You cannot specify tracks as you can point to sectors and bytes on a disk. The cassette can only read what is passing under the cassette head, meaning that you cannot point to only one section of tape as you can read in one sector from a disk.

Write a short program (it doesn't matter how long) and then save it to your cassette recorder. After you have the program saved, rewind the tape and enter these commands in the direct mode (meaning you don't need a line number):

```
OPEN #7,4,0,"C:":FOR A=1 TO 128
:GET #7,B:? B;"   ";:NEXT A:CLOSE
#7
```

The cassette is now open for us to read from it. Now press the play button on the recorder and return on the keyboard. You will then hear one tone. Press the return key again. After you hear a record of data being read in, you will see numbers being printed onto the screen. These numbers are the ones for which the program was saved to the cassette. The computer is printing it from the cassette buffer in the same way it was printed from the disk buffer.

Now enter the line below without clearing the screen:

```
FOR A=0 TO 130:? PEEK(1021+A);"
";:NEXT A
```

You will have noticed that the first three bytes printed on the screen are different from the first three bytes that were printed from the buffer. Then, if you will look at the fourth byte, you'll notice both numbers are the same. Memory location 1021 is where the cassette buffer begins. The buffer is 131 bytes long. The first two bytes(85) are fixed and are used by the computer to measure the speed of the cassette. The third byte is the control byte,meaning if this byte is 252, the record is full-128 bytes. A 250 in this position indicates a partially full record and a 254 is an end of file record.

The first bytes after the speed and control bytes for the first record consist of the table entries. This includes the variable table, the value table,etc. The values are adjusted by the amount of memory in your computer. This is why a program that requires more storage memory than your machine has, will produce an error message immediately, when you try loading it, rather than half way through the load.

The last byte that the computer reads from the cassette is a checksum. When the program was saved to a cassette, every byte that was sent to it in the 128 byte record was added to the previous bytes. The two markers are also included in the addition. After the two markers, the control byte, and the record were sent to the cassette, the sum of the bytes were also sent. When the records are read in,the bytes are added together again. If the sum of the bytes match the checksum byte that is read in, then the record is assumed to be cor-

Yellow Brick Road cont'd

rect and the computer will continue with the loading process. If the two byes do not match, the load will stop and an error message will appear on the screen.

Each record of 128 bytes is considered to be frame. The speed at which the data is sent or received is called the baud rate. The ATARI uses a baud rate of 600 (600 bits per second(each byte is 8 bits)). The two marker bytes are read in by the computer. It then determines how long it took to read them in and calculates the correct baud rate for the tape. It does this with every frame of data that is read in. The input baud rates can be adjusted to read faster or slower. This adjustment process allows for variations in motor speeds, stretched tape, etc. It does not, however, allow for alignment problems between the recorder that the program was originally saved on and the one that it is being read on.

You probably noticed that when you CSAVE a program to cassette, the records seem to be sent faster than when you used the list command. There are two different modes that the computer can use when sending records to the cassette. One is called the normal IRG(Inter-Record Gap), and the other is the short IRG. The computer uses the short IRG for CSAVE and CLOAD commands. The computer reads in the record, checks the checksum, places it in RAM, and goes back for another record. The recorder is running the entire time. The computer must do its work so that it won't miss the next record.

When the computer uses the normal IRG for saving records to the tape, the recorder stops after every record is read into the computer. The information can be processed; then the next record can be read in. The baud rate for saving and reading the data is the same for both modes. It is the time between records that varies. In the normal mode there there is about 3 seconds of tone between records. In the short mode there is about 1/4 of a second of tone time between records. The IRG is set up by the computer when the save or load command is entered. If the wrong command is entered for the tape, the program will not load and an error message will be displayed on the screen.

That ends the first part of a two part tutorial on the cassette recorder. In the next issue I will talk about making an autoboot on your cassette and making your recorder sing.

DATUM
COMPUTER CENTRE
A DIVISION OF DATUM ELECTRONICS INC.

We've Brought the Computer Age Home...™ ATARI® 600XL™

ATARI HOME COMPUTER PROGRAMS LET YOU PLAY AS HARD AS YOU WORK.

Thanks to the ever-growing selection of ATARI software programs, owners of ATARI Home Computers can make sure they play as hard as they work.

Choose from nearly 40 different inexpensive, yet invaluable, ATARI programs including sophisticated business-related programs, languages, and the very latest in entertaining and challenging computer games.

Plus, there's no limit to what the future holds. Because the best minds at ATARI are working night and day on new software programs, all of which can be utilized with your **ATARI 600XL™** Home Computer.

So add to your ATARI program library today. Because no home computer should be all work and no play.

©1982 ATARI, INC. All rights reserved
Stock analysis available only for the ATARI 800™ Home Computer

ATARI®
A Warner Communications Company

Available now at:

DATUM
COMPUTER CENTRE
A DIVISION OF DATUM ELECTRONICS INC.
PERSONAL AND BUSINESS COMPUTERS

Westwood Mall
No. 11-3000 Lougheed Hwy.
Port Coquitlam, B.C.
Canada V3B 1C5
(604) 464-4424

Authorized
ATARI Service Center

WATER FROM HEAVEN

By PETER ELLISON

Deep inside darkest Africa there lives a tribe called the Moegabians and for centuries have kept an ancient custom. The cuscalled "Moneg-ehds-sayalk" or in English "Water From Heaven." Every year, just before harvest, each tribe member would wait outside his or her mud hut with a strainer waiting for the rains to come. When the rains begin, the entire tribe would huddle together around the farmland, strainers in hand, and begin moving back and forth to catch the raindrops. The reason for this was, if the rain touched the ground without first going through the strainer, the piece of ground would become unholy, and the ground was their livelihood. For this reason something had to be sacrificed inorder to appease the gods. But if this rain called "moguna" or "Heaven Rain" was strained and landed on the ground, the ground would be blessed from heaven,giving a fruitful harvest.This meant that the tribe had to move quickly back and forth as one unit, so that no rain would fall without first being strained.

The story that you just read in the beginning paragraph is a short outline of this game. This game, called "Water From Heaven", is one in which you move your strainer back and forth, inorder to catch the raindrops that fall from the top of the screen. If you catch the drop and it passes through your strainer and hits the ground, you then receive some points. If you don't strain the rain and it hits the ground, then you lose one life. Sometimes you will have to hold the strainer in one position to get the drop or you'll lose a life.

This game is written in BASIC with a short little assembly language suroutine to move the raindrops down the screen quickly and smoothly. This game demonstrates the potential your ATARI computer has in BASIC to produce arcade-type action. The game was written in graphics mode three because this was the fastest graphics mode for the Player/missile graphics. I used Player(0) for the strainer and Players(1-3) for the raindrops. All of the raindrop's shapes were setup in the assembly language subroutine, so none of them had to be declared in BASIC. To begin with, the prowill ask you how many Lives(Up to nine) and Level(1-7). The difference in the levels, is the speed in which the raindrops fall. The machine-code takes a couple of seconds to load in at the beginning of the game.

All that is needed to play the game is that you type in the BASIC listing. The Assembly language listing is just so those people who want to see how the players(raindrops) were moved down the screen. There are really two short Assembly language subroutines for this program, but I have only listed one. The reason for this is the only difference between the two is what I made VPOS equal to. In the one not listed, VPOS is equal to $652 instead of $601. This subroutine had to be different for a 16K machine and a 48K computer. This is because PMBASE is different with the amount of memory your machine has. This is the reason for the two different DATA listings. If you have 48K then type in the DATA statements that are at the end of the 16K listing.

I hope you enjoy this game and if you have any modifications or suggestions, feel free to send them in. Have fun, and don't forget to strain the water.

```
10 REM WATER FROM HEAVEN
20 REM ROM MAGAZINE 1984
30 REM REQUIRES 16K AND ONE JOYST
ICK
40 GOTO 120
50 REM READ DATA FOR MUSIC
60 READ A
70 IF A=31 THEN GOTO 180
```

```
,B,C,D:SC=SC+PO
835 REM DETERMINE LEVEL
840 IF PEEK(53260)>1 THEN NUM=NUM
+1
850 IF NUM=10 THEN LEV=2:IF NUM=2
0 THEN LEV=3:IF NUM=30 THEN LEV=4
:IF NUM=40 THEN LEV=5:IF NUM=50 T
HEN LEV=6
860 IF NUM=60 THEN LEV=7
870 ? ↑↑':? "SCORE:";SC
880 POKE 85,1
890 IF ST=11 OR ST=10 OR ST=9 THE
N M=M-8:POKE 53248,M
900 IF ST=7 OR ST=6 OR ST=5 THEN
M=M+8:POKE 53248,M
910 IF M<=85 THEN M=190
920 IF M>=191 THEN M=86
930 X=USR(1538)
940 X=USR(1638)
950 POKE 53278,0
960 SOUND 1,0,0,0
970 SOUND 0,0,0,0
980 GOTO 720
990 FOR R=1 TO 80:READ S:POKE 153
8+R-1,S:NEXT R
1000 DATA 104,173,82,6,24,105,1,1
41,82,6,172,82,6,169,0,153,120,58
,153,121,58,153,122,58
1010 DATA 153,123,58,153,124,58,1
53,125,58,153,126,58,153,127,58,1
69,4,153,128,58,169,4,153,129
1020 DATA 58,169,4,153,130,58,169
,4,153,131,58,169,14,153,132,58,1
69,31,153,133,58,169,14,153
1030 DATA 134,58,169,4,153,135,58
,96
1040 FOR H=1 TO 80:READ K:POKE 16
38+H-1,K:NEXT H
1050 DATA 104,173,1,6,24,105,1,14
1,1,6,172,1,6,169,0,153,248,58,15
3,249,58,153,250,58
1060 DATA 153,251,58,153,252,58,1
53,253,58,153,254,58,153,255,58,1
69,4,153,0,59,169,4,153,1
1070 DATA 59,169,4,153,2,59,169,4
,153,3,59,169,14,153,4,59,169,31,
153,5,59,169,14,153
1080 DATA 6,59,169,4,153,7,59,96
1090 GOTO 720
1100 REM COLLISION SUBROUTINE
1110 POKE 712,PEEK(53770)
1120 Q=Q+1
1130 FOR X=10 TO 20:SOUND 0,X,6,1
0:SOUND 1,X,8,8:SOUND 2,X,10,12:N
EXT X
1140 SOUND 1,0,0,0:SOUND 2,0,0,0
```

```
1150 IF Q>3 THEN GOTO 1170
1160 GOTO 1110
1170 Q=0
1180 X=USR(1538):X=USR(1638):POKE
712,40
1190 K=INT(RND(0)*100)+50:IF K<90
THEN K=90
1200 L=INT(RND(0)*140)+50:IF L<12
0 THEN L=120
1210 O=INT(RND(0)*140)+50:IF O<12
0 THEN O=120
1220 RETURN
1230 REM GAME OVER
1240 REM ENDING SONG
1250 GOSUB 1410
1260 READ A
1270 IF A=31 THEN END
1280 SOUND 0,A,10,10:SOUND 1,243,
10,6:FOR W=1 TO 400 STEP 20:NEXT
W
1290 GOTO 1490
1300 END
1310 DATA 121,121,121,102,108,108
,121,121,128,121,162
1320 DATA 60,60,60,50,53,53,60,60
,64,81
1330 DATA 121,121,121,102,108,108
,121,121,128,121,162
1340 DATA 60,60,60,50,53,53,60,60
,64,81
1350 DATA 121,121,121,102,108,108
,121,121,128,121,162
1360 DATA 60,60,60,50,53,53,60,60
,64,81
1370 DATA 121,121,121,102,108,108
,121,121,128,121,162
1380 DATA 60,60,60,50,53,53,60,60
,64,81
1390 DATA 121,121,121,102,108,108
,121,121,128,121,162
1400 DATA 60,60,60,50,53,53,60,60
,64,81,29,29,29,31
1410 PLOT 12,5:DRAWTO 9,5:DRAWTO
9,10:DRAWTO 12,10:DRAWTO 12,8:DRA
WTO 11,8
1420 PLOT 17,10:DRAWTO 17,5:DRAWT
O 14,5:DRAWTO 14,10:PLOT 16,7:DRA
WTO 15,7
1430 PLOT 23,10:DRAWTO 23,5:DRAWT
O 21,7:DRAWTO 19,5:DRAWTO 19,10
1440 PLOT 28,10:DRAWTO 25,10:DRAW
TO 25,5:DRAWTO 28,5:PLOT 26,7:DRA
WTO 28,7
1450 PLOT 9,13:DRAWTO 9,18:DRAWTO
12,18:DRAWTO 12,13:DRAWTO 9,13
1460 PLOT 14,13:DRAWTO 16,18:DRAW
```

Water From Heaven cont'd

```
80 SOUND 0,A,10,10:SOUND 1,243,10
,6:SOUND 2,204,10,6:SOUND 3,173,1
0,6
90 FOR Z=1 TO 90:NEXT Z
100 GOTO 60
110 DATA 96,81,60,81,96,81,121,31
120 GRAPHICS 2+16
130 POSITION 2,2
140 ? #6;"WATER from "
150 POSITION 2,4:? #6;"BY Peter E
llison"
160  POSITION  0,6:?   #6;"(c)ROM
Magazine 1984"
170 GOSUB 60
180 C=PEEK(53279)
190 SOUND 0,0,0,0:SOUND 1,0,0,0:S
OUND 2,0,0,0:SOUND 3,0,0,0
200 IF C=6 THEN GOTO 240
210 LIV=1:SCORE=0:LEV=1:SC=0
220 POSITION 0,8:?   #6;"PRESS
START TO BEGIN"
230 GOTO 180
240 ? #6;"↰"
250 P=PEEK(53279)
260 IF P=5 THEN LIV=LIV+1:IF LIV>
=10 THEN LIV=1
270 IF P=3 THEN LEV=LEV+1:IF LEV>
7 THEN LEV=1
274 IF LEV=2 THEN NUM=10:IF LEV=3
THEN NUM=20:IF LEV=4 THEN NUM=30:
IF LEV=5 THEN NUM=40:IF LEV=6 THE
N NUM=50
276 IF LEV=7 THEN NUM=60
280 IF STRIG(0)=0 THEN GOTO 350
290 POSITION 2,4:? #6;"lives:";LI
V
300 POSITION 2,6:? #6;"";LEV
310 POSITION 0,8:? #6;"press sele
ct for lives"
320 POSITION 0,9:? #6;"PRESS OPTI
ON FOR LEVEL"
330 POSITION 0,10:? #6;"TO BEGIN
PRESS BUTTON"
340 GOTO 250
350 POKE 752,1
360 SCORE=0
370 A=1:B=5:C=12:D=11
380 REM START GAME
390 GRAPHICS 3
400 COLOR 2
410 SETCOLOR 2,16,1
420 ? "**************************
***********"
430 ? "
           "

450 POKE 712,40:POKE 708,35
```

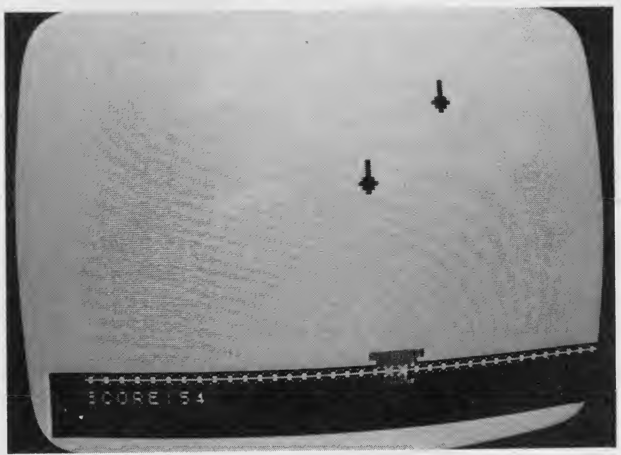
```
460 REM SET UP PLAYER/MISSILE ARE
A
470 POKE 53256,1
480 POKE 623,1
490 I=PEEK(106)-8
500 POKE 53277,3:POKE 54279,I
510 PMBAS=I*256
520 POKE 704,72:POKE 705,150:POKE
706,100:POKE 707,22
530 POKE 559,46
540 REM RAIN DROPS
550 X1=160:Y1=20
560 POKE 53249,X1
570 X2=100:Y2=20
580 POKE 53250,X2
590 X3=145:Y3=150
600 POKE 53251,X3
610 M=100:Y=92
620 REM DROP STRAINER
630 POKE 53248,M
640 FOR L=PMBAS+512 TO PMBAS+640:
POKE L+X,0:NEXT L
650 FOR L=1 TO 8
660 READ G
670 POKE PMBAS+512+Y+L,G
680 NEXT L
690 DATA 255,255,126,126,60,60,60
,126
700 GOTO 990
710 REM CLEAR COLLISIONS
720 POKE 53278,0
730 REM MAIN GAME LOOP
740 ST=STICK(0)
744 REM POINTS PER LEVEL
745 IF LEV=1 THEN PO=1
746 IF LEV=2 THEN PO=2
747 IF LEV=3 THEN PO=3
748 IF LEV=4 THEN PO=4
749 IF LEV=5 THEN PO=5
750 IF LEV=6 THEN PO=6
751 IF LEV=7 THEN PO=7
755 X=USR(1538):X=USR(1638)
760 REM COLLISIONS:POKE 53278,0
770 IF PEEK(53260)<1 AND PEEK(532
53)>1 THEN GOSUB 1110:POKE 53249,
K:X=USR(1538):LIV=LIV-1
780 IF PEEK(53260)<1 AND PEEK(532
54)>1 THEN GOSUB 1110:POKE 53250,
L:X=USR(1538):LIV=LIV-1
790 IF PEEK(53260)<1 AND PEEK(532
55)>1 THEN GOSUB 1110:POKE 53251,
O:X=USR(1638):LIV=LIV-1
800 IF LIV=0 THEN GOTO 1230
810 POKE 752,1
820 POKE 1544,LEV+1:POKE 1644,LEV
830 IF PEEK(53260)>1 THEN SOUND A
```

```
TO 18,13
1470 PLOT 23,18:DRAWTO 20,18:DRAW
TO 20,13:DRAWTO 23,13:PLOT 21,15:
DRAWTO 23,15
1480 PLOT 26,15:DRAWTO 28,15:DRAW
TO 28,13:DRAWTO 25,13:DRAWTO 25,1
8:PLOT 29,18:DRAWTO 26,16
1490 POKE 712,PEEK(53770):GOSUB 1
260
```

These lines below are to be type in instead, if your machine has 48K:

```
990 FOR R=1 TO 80:READ S:POKE 153
8+R-1,S:NEXT R
1000 DATA 104,173,1,6,24,105,0,14
1,1,6,172,1,6,169,0,153,120,155,1
53,121,155,153,122,155
1010 DATA 153,123,155,153,124,155
,153,125,155,153,126,155,153,127,
155,169,4,153,128,155,169,4,153,1
29
1020 DATA 155,169,4,153,130,155,1
69,4,153,131,155,169,14,153,132,1
55,169,31,153,133,155,169,14,153
1030 DATA 134,155,169,4,153,135,1
55,96
1040 FOR H=1 TO 80:READ K:POKE 16
38+H-1,K:NEXT H
1050 DATA 104,173,82,6,24,105,0,1
41,82,6,172,82,6,169,0,153,120,15
4,153,121,154,153,122,154
1060 DATA 153,123,154,153,124,154
,153,125,154,153,126,154,153,127,
154,169,4,153,128,154,169,4,153,1
29
1070 DATA 154,169,4,153,130,154,1
69,4,153,131,154,169,14,153,132,1
54,169,31,153,133,154,169,14 ,153
1080 DATA 134,154,169,4,153,135,1
54,96
```

```
00005  *ASSEMBLER SUBROUTINE FOR*
00006  *MOVING RAINDROPS DOWN THE*
00007  *SCREEN*
00010  .LI OFF
00020  .OR $84D0
00030  .TA $2600
00035  *TARGET ADDRESS *
00040  .TF "D:DROP.PL1"
00045  *VPOS IS $601 OR $652*
00046  *DEPENDING ON WHICH PLAYER*
00050 VPOS      .EQ $601
00055  *PLAYER MISSILE BASE ADDRESS
00060 PMBASE    .EQ $3800
00070           PLA
00080           LDA VPOS
00090           CLC
00100           ADC #1
00110           STA VPOS
00120           LDY VPOS
00130           LDA #0
00150           STA PMBASE+$278,Y
00160           STA PMBASE+$279,Y
00170           STA PMBASE+$27A,Y
00180           STA PMBASE+$27B,Y
00190           STA PMBASE+$27C,Y
00200           STA PMBASE+$27D,Y
00210           STA PMBASE+$27E,Y
00220           STA PMBASE+$27F,Y
00225  *LOAD IN PLAYER DATA*
00230           LDA #4
00240           STA PMBASE+$280,Y
00270           LDA #4
00280           STA PMBASE+$281,Y
00290           LDA #4
00300           STA PMBASE+$282,Y
00310           LDA #4
00320           STA PMBASE+$283,Y
00330           LDA #14
00340           STA PMBASE+$284,Y
00350           LDA #31
00360           STA PMBASE+$285,Y
00370           LDA #14
00380           STA PMBASE+$286,Y
00390           LDA #4
00400           STA PMBASE+$287,Y
00410           RTS
```



BEGINNER'S LINE
CHARACTER GRAPHICS – PART IV
By GEOFF CORRY

Well here we are again, in a new year and for some of you, a new computer. This is the last in the series on Character Graphics. This series started in Issue 2 of ROM, and I have noticed some back issues in some stores around the lower mainland; try your local drug store. If you can't find some back issues then we have some available at $2.50 per copy, which includes mailing. Leave a voice message at (604) 462 9177, or write to us, enclosing your payment, to ROM BOX 252, Maple Ridge B.C. V2X 7G1, and we will mail them out to you.

This issue we will start a project that will take us on to character animation, screen flipping, scrolling and other programming techniques that the Atari carries out so well.

Two small but important BOO BOO's crept into the short program on page 7 of the last issue. Lines 510 and 530 should be as follows:-

```
510 DATA 0,224,252,252,255,254,248,240

530 DATA 240,248,252,254,255,252,224,0
```

This should make 'an old friend' more recognizable.

The name of the publication was omitted from one of the Font programs listed near the top of Page 8. The series on COMPUTER ANIMATION is in The Atari Connection, starting in the Spring '83 edition, and continuing. Some computer stores have a counter copy of this magazine for you to look at, or you may want to subscribe at $10/year for this useful publication, aimed at the new Atari owner. Another address for you;- Subscriptions, c/o ATARI CONNECTION, P.O. Box 50047, San Jose, California, U.S.A. 95150.

Now that's over, lets get on with character graphics. I am going to describe how to layout shapes using the modified character set to form larger picture elements on the screen. Then we will introduce the project that will include character graphics, as well as other dynamic programming features for your Atari.

In the Christmas greeting program in Issue 4, the trees that were 'planted' to form 'Merry Christmas', were each created by altering the lower case letter 'o'. It looked a bit lopsided, so I added one more pixel on the right, using a modified 'p' character.

We used another trick to get more color on the GR. 0 screen by a process called artifacting. The trick uses odd or even columns to locate the pixels.

The green color comes from using only columns 1,3,5,& 7. The holly berries used only columns 2,4,6,& 8 to give the red color. This happens because this fine vertical line only illuminates the green or red phosphors on your color T.V. By shifting the vertical pattern one pixel width either right or left gives you a green or red overall color. Here is a 16 by 8 grid to show you how the trees are formed:-

```
          ___________GREEN_____________
          v   v   v   v   v   v   v   v
            ____________RED______________
            v   v   v   v   v   v   v   v
    /  --------------------------------    \
    ¦  8¦ ¦ ¦ ¦ ¦X¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦  0 ¦
    ¦   --------------------------------    ¦
    ¦  8¦ ¦ ¦ ¦ ¦X¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦  0 ¦
    ¦   --------------------------------    ¦
 SEE¦  8¦ ¦ ¦ ¦ ¦X¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦  0 ¦ SEE
    ¦   --------------------------------    ¦
LINE¦ 42¦ ¦ ¦X¦ ¦X¦ ¦X¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦  0 ¦LINE
    ¦   --------------------------------    ¦
 640¦ 42¦ ¦ ¦X¦ ¦X¦ ¦X¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦  0 ¦ 650
    ¦   --------------------------------    ¦
    ¦170¦X¦ ¦X¦ ¦X¦ ¦X¦ ¦X¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ 128¦
    ¦   --------------------------------    ¦
    ¦ 12¦ ¦ ¦ ¦ ¦X¦X¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦  0 ¦
    ¦   --------------------------------    ¦
    ¦ 12¦ ¦ ¦ ¦ ¦X¦X¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦  0 ¦
     \  --------------------------------    /
        1 6 3 1 8 4 2 1 1 6 3 1 8 4 2 1
        2 4 2 6         2 4 2 6
        8               8
        \__modified___/\__modified___/
          letter 'o'     letter 'p'
```

The short trunk in the bottom two rows use two adjacent pixels (code 12 in line 640) which take on the brown color against the white background. The background is controlled by Color Register 2 in Gr.0 Mode (SETCOLOR 2,0,10 in line 100). The holly leaves and berries took the most time to set up and they used up 14 characters, 'a' to 'n'. Take my word for it, spikey holly leaves are tough.

-- GOING TO TOWN PROJECT --

Now for the new project. Just to set the stage, lets imagine you live on a farm, some distance from the town where you, and your family, do your shopping. You have decided to pick up some things in town, and your appointment with the dentist has just come up. First you get in your car. All you can see is the back inside wall of the garage (character graphics), you look behind you, out the rear window (page flipping). Back out slowly, you don't want to flatten any chickens that are wandering about. You turn (horizontal scrolling). Oh Boy! You you turned in the wrong direction, look forward (page flip) and turn back into the garage. Careful don't hit the door posts (P/M graphics, collision registers). O.K. Back out and turn the other way. What's that squark ? (sound effects). Looks like you will have to buy another chick to replace the one you squished. Now drive to the gate and stop.

Time to check the list. Hit 'L' (keyboard handler), you see the shopping list displayed. There are groceries, feed for the hens, parts for the tractor that broke down today, and one chick. Also there are books to return to the library and a dentist appointment at 2:P.M. Lets see how much money we have. Hit 'W' for wallet. The amount of cash you are carrying is displayed as well as your bank password number and checking account balance. Looks like we don't have enough, so better go to the bank first. (While we are checking the list, the next part of the program is loading.)

This section will look down on a view of the roads around your farm and will scroll when you drive along the road (character graphics, scrolling, and player missile graphics). Finally when you see the freeway, you drive to the intersection before the freeway access. The scene changes, you are now back in the car, looking ahead. There is a red traffic signal (we will give a short demo of this) which remains red until the freeway portion of the program loads.

The light turns green and we enter the freeway access. You hear a beep, somebody wants by. You steer into the right lane and let him pass. Oh,oh, there's a accident up ahead, better slow down. Finally you can see the sign showing the turnoff to the town. Drive off the freeway exit and wait at the traffic signal. While we are waiting, the town portion of the program loads. The map of the town is displayed, again use the joystick to control the two way scrolling. We can now check all the locations that we have to go to. First, drive to a car park that's convenient.

Now we are walking to the bank. They have one of those automatic tellers. Follow the directions as they come up. We enter our password-- remember, the one we saw when we checked how much money we had. We key in the amount we want to withdraw. Now we can check our wallet again for the amount we now have and also our new bank balance. Hit 'T' for time, it's almost two o'clock. better find the dentist's office. (Some nice dentist noises would come about now. Brrrr.) After that's over, we can continue with the shopping. Don't forget the chick. Then back to the car park. Pay for the parking, and then drive back to the farm.

Now this kind of game would be 'booooring' if there wasn't some variety in what goes on. We will access a data file from time to time, and pick at random some of the suitable things to do or see. Maybe the tractor is in front of the garage when we want to drive out. The list of things to get and appointments to keep would change each time the game is played. We may have to detour along another road because a bridge is washed out. It's up to our imagination, system capabilities, and time factors, how this works out. I welcome your suggestions, and as we get along with this ,feel free to try out some of your ideas. Call us at the number at the beginning of this article.

Charles Brannon, Program Editor for COMPUTE! magazine (Feb. '84) gives the following requirements for a good game:-

Selling Power (hope it keeps your interest up,
and make you a regular reader of ROM).
Style - things should happen in a reasonable way
and also be creative.
Telling Tales - this game will tell a different
story,each time it is run, in the context
of 'a trip to town'.
More Depth - innovative ways to operate the
various episodes that happen in the game.
Using Capabilities - The Atari has great graphic
abilities that can be used in so many ways.
Finding A Center - Using graphics and sound to
guide the way that the game develops
during play.
Playability - Each episode operates in a different
way, and must be challanging but not
confusing.
Sound Effects - The car motor, the chick squark,
sirens on the freeway, music from the record
store, etc. to add realism to the game.

We will keep these points in mind as we develop the various modules in this tutorial/game. An added interest will be the different techniques that we will need to find out and use as the situations come up. At this time (Feb. 1984) this game does not exist, but we will build it through this series of articles. You will get a chance to add or modify some parts to form your own version.

Here is a small segment. The traffic signals before the freeway access and at the entrance into town will use ideas that we have already met in in the series on character graphics. Highway traffic signals have large borders around them to make them more visable. Our signal in this demonstration will be 4 characters wide and 10 characters long. The supporting pole will extend down the side of the screen. Here is the short program to try out. First we will use the familiar character move routine to get our characters down into an area of memory where we can change them:-

```
10 RAMTOP=106:CHBAS=756:CHORG=57344
20 GRAPHICS 0:POKE 752,1:SE. 1,0,0:
SE. 2,0,10:REM See next paragraph
30 RAMNEW=PEEK(RAMTOP)-8
40 START=RAMNEW*256
45 IF PEEK(START+1022)=16 THEN 80:REM
Check to see if we have already moved
it down. Why wait each time we run it
50 FOR CH=0 TO 1023
60 POKE START+CH,PEEK(CHORG+CH)
70 NEXT CH
80 POKE CHBAS,RAMNEW
```

Here's a trick to save you time. These lines are mainly the same as the first few lines of last issue's 'Greeting' program, but we don't need the rest of the listing. If you 'saved' this program, 'load' it again and write down the counter number on your recorder. Now type in the following:-

```
   LIST 20
```

Now add the setcolor statements in line 20 as shown above. We will then save lines 10 to 80 as follows:-

```
   LIST"C:",10,80
```

Press RECORD and PLAY on your recorder, and hit RETURN twice. For you disk drivers, type in:-

```
   LIST"D:SIGNAL.ROM",10,80
```

and hit RETURN.

This will copy up to line 80 only (less the REM statements) onto your tape or disk. Now type 'NEW' and hit RETURN to clear the memory in your computer. Now rewind your tape and type:-

```
   ENTER"C:".
```

Press PLAY and hit RETURN twice. Diskers, type:-

```
   ENTER"D:SIGNAL.ROM"
```

and hit RETURN.

This is a good procedure to use, when you want to select any group of lines from an existing BASIC program to use in a new program.

To save memory space, something that will be very important as we get on with this project, we are going to show you two ways to put the shape information into a program. The first way is the procedure we used in the 'greeting' program, with POSITION statements followed by the actual converted characters in their relative positions. Please type this in:-

```
100 FOR I=0 TO 135:READ A
110 POKE(START+(97*8)+I),A
120 NEXT I
130 POS. 19,04:? "aaaabbbbcd"
140 POS. 19,05:? "afga     e"
150 POS. 19,06:? "ahia     e"
160 POS. 19,07:? "aaaa     e"
170 POS. 19,08:? "aaaa     e"
180 POS. 19,09:? "aaaa     e"
190 POS. 19,10:? "aaaa     e"
200 POS. 19,11:? "aaaa     e"
210 POS. 19,12:? "         e"
220 POS. 19,13:? "         e"
230 POS. 19,14:? "         e"
240 POS. 19,15:? "         a"
250 POS. 19,16:? "         a"
260 POS. 19,17:? "         a"
270 POS. 19,18:? "         a"
280 POS. 19,19:? "         a"
290 POS. 19,20:? "         a"
```

To make things clearer, the 'a's are converted to a black square, the 'b's form a horizontal black line, which is the support arm. The 'c's and 'd's form the bend in the pole, which is made up by the 'e's. We make the pole thicker by substituting 'a's in lines 240 to 290. Letters 'f,g,h, and i' make up the red light. (Artifacting -- remember?) Lets go on:-

```
300 IF PEEK(764)=61 THEN GOSUB 440:?
FRE(0)
310 IF PEEK(764)=62 THEN GOSUB 330:GO
SUB 390:GOSUB 330:GOSUB 340
320 POKE 764,255:GOTO 300
330 FOR W=1 to 1000:NEXT W:RETURN
340 POS. 19,7:? "aaaa"
350 POS. 19,8:? "aaaa"
360 POS. 19,9:? "afga"
370 POS. 19,10:? "ahia"
380 POKE 764,255:RETURN
390 POS. 19,9:? "aaaa"
400 POS. 19,10:? "aaaa"
410 POS. 19,7:? "anoa"
420 POS. 19,8:? "apqa"
430 POKE 764,255:RETURN
440 POS. 19,5:? "aaaa"
450 POS. 19,6:? "aaaa"
460 POS. 19,9:? "ajka"
470 POS. 19,10:? "alma"
480 POKE 764,255:RETURN
```

Beginner's Line cont'd

These lines operate the lights. Lines 300 and 310 check the keyboard, waiting for you to either press a 'G' for GO, or an 'S' for STOP. Here's something to add a bit of confusion. Remember we talked about ASCII coding for the characters, and ATASCII code, which is Atari's version to include the graphic shapes. Then we went on to talk about internal coding, which is ATASCII coding rearranged in blocks of 32 characters, so that the alphanumeric characters are all together for the larger print styles (Graphics 1 and 2).

Now when you look at the keyboard, you can see that the letters are not arranged in alphabetic order. When you press a key, you are making contact at one of the crosspoints of the four rows and 14 columns. Memory location 764 holds the code for the last key pressed, and this code directly relates to the way that those contacts are wired out on your keyboard. The January '84 issue of A.N.A.L.O.G (page 103) has a complete table of the ATASCII, internal, and keycode numbers for all the 128 characters and special keys on your computer. We also listed out most of these keycodes on page 16 of Issue 1, if you have it.

Now the code for 'g' is 61 and the code for 's' is 62. Due to the literal way this code is read, it is not necessary to go to the lower case (CAPS LOWR) to get this to work in our program. Here is a four liner you can use to see the code for any key pressed:-

```
1 OPEN #1,4,0,"K:"
2 CODE=PEEK(764):IF CODE=255 THEN 2
3 GET #1,R:FOR W=1 TO 250:NEXT W:? "
THE CODE FOR ";CHR$(R);" IS ";CODE
4 GOTO 2
```

Type this in later (don't mix it with our program) and press CTRL CLEAR and then type RUN.

I seem to get sidetracked, so lets go back to the main program. All those GOSUB's in lines 300 and 310 go to the subroutines that print over lights to switch them off, and then print another light either above or below.

The one line subroutine, line 330, provides a 4 second delay when the light changes from green, through amber, to red. The poke 764,255 resets the keyboard handler, when the keys are pressed.

Now lets end this program with the data statements that modify the characters:-

```
500 DATA 255,255,255,255,255,255,255,
255:REM 'a' is changed to a black square for the light
border and pole base.(Don't type in these remark
statements).
510 DATA 255,255,0,0,0,0,0,0:REM 'b' is changed to a
thick horizontal line to make up the signal arm.
520 DATA 252,255,3,0,0,0,0,0
530 DATA 0,0,192,224,112,56,24,24:REM 'c' and 'd' form
the curve in the arm to the pole.
540 DATA 24,24,24,24,24,24,24,24:REM 'e' is the thick
vertical line which makes up the pole.
550 DATA 253,245,245,213,213,85,85,85
560 DATA 95,95,87,87,85,85,85,85
570 DATA 85,85,85,213,213,245,245,253
580 DATA 85,85,85,85,87,87,95,95
```

These 4 lines make up the red light, top left, top right, bottom left, & bottom right respectively. These are one pixel vertical lines with one pixel separation in the round area of the red light.

```
590 DATA 255,250,234,234,170,170,170
,170
600 DATA 255,191,175,175,171,171,171
,171
610 DATA 170,170,170,170,234,234,250
,255
620 DATA 171,171,171,171,175,175,191
,255
```

These four lines make up the green light in the same way that the red light is formed, except the vertical lines are moved over one pixel width.

```
630 DATA 255,249,249,249,153,153,153
,153
640 DATA 255,159,159,159,153,153,153
,153
650 DATA 153,153,153,153,249,249,249
,255
660 DATA 153,153,153,153,159,159,159
,255
```

These four lines form the amber light. The artifacting does not work too well for this small area. The yellow effect is obtained by having vertical bars, two pixels wide, with one pixel separation.(Move the pattern over one pixel width and the color would be light blue).

Now you have typed in this program, rewind your recorder back to the counter number you wrote down earlier, and CSAVE it (SAVE"D:SIGNAL.ROM for you diskers), and then RUN it. If everything is O.K., the screen should turn white, and after a few moments, the traffic signal will appear with the red light on. To change the light, press the 'G' key for a count of three. Pressing the 'S' key for a count of three will cause the light to change back, through amber, to red.

When you push the 'G' button, you will notice some numbers come up on the left. This is provided courtesy of the 'FRE(0)' statement in line 300, and shows how much free memory is left in your system. When have jotted down this number, type 'NEW' and RETURN to clear this program. Type:- ? FRE(0) and RETURN and you will see the total memory available. Subtract the number that came up in the program to see how much memory your program took. This version took approximately 2340 bytes.

LOAD the program back in. Type in the following lines as an alternate way to program this traffic light demo, using strings. According to all accounts, the use of strings to store information saves memory space. This is one of the criteria for the development of the 'going shopping' project. Here are some lines that will replace existing lines:-

```
130 F=4:DIM LIGHT$(30),ARM$(10),
BORDER$(F),POLE$(1),BASE$(1)
140 RED1$(F),RED2$(F),AMBER1$(F),
AMBER2$(F),GREEN1$(F),GREEN2$(F)
```

The length of all strings must be dimensioned with ATARI basic. A little trick to save some memory was used here. All the colors and the border used 4 characters (see lines 340-370 in the original program). Using the variable 'F' seven times, instead of repeating the '4', saves 6 bytes in the DIM statements.

```
150 LIGHT$="aaaabbbbcdeafgahiajkalmano
apqa"
```

You can see all the sets of characters used in the last program embedded in this 30 character string

```
160 ARM$=LIGHT$(1,10):BORDER$=LIGHT$
(1,4):POLE$=LIGHT$(11,11):BASE$=LIGHT
$(1,1):RED1$=LIGHT$(12,15)
```

'ARM$' is the same character format in the first program (line 130), as well as being the first 10 characters in 'LIGHT$'.

```
170 RED2$=LIGHT$(15,18):AMBER1$=LIGHT
$(24,27):AMBER2$=LIGHT$(27,30):GREEN1
$=LIGHT$(18,21):GREEN2$=LIGHT$(21,24)
```

Here are all the characters in lines 360,370,410,420,460,and 470 in the original program. Now to place them on the screen:-

```
180 POS. 19,4:? ARM$:POS. 19,5:? RED1
$:POS. 19,6:? RED2$
190 FOR X=7 TO 11:POS. 19,X:? BORDER$
:NEXT X:REM Light assembly and arm.
200 FOR X=5 TO 14:POS. 28,X:? POLE$:
NEXT X:REM Build the pole
210 FOR X=15 TO 20:POS. 28,X:? BASE$
:NEXT X:REM Assemble the base
```

We don't need lines 220 to 290 in the original program but we will deal with that later. Lines 300 to 330,380,430, and 480 to 660 are the same in both programs, so why retype them.

```
340 POS. 19,7:? BORDER$
350 POS. 19,8:? BORDER$
360 POS. 19,5:? RED1$
370 POS. 19,6:? RED2$
390 POS. 19,9:? BORDER$
400 POS. 19,10:? BORDER$
410 POS. 19,7:? AMBER1$
420 POS. 19,8:? AMBER2$

440 POS. 19,5:? BORDER$
450 POS. 19,6:? BORDER$
460 POS. 19,9:? GREEN1$
470 POS. 19,10:? GREEN2$
```

Whew!! That's it.

Now, two more things before we run it. There are some lines we don't need in this new version. Type 220 to 290 with a RETURN after each number. 'LIST 200,300' to make sure lines 220 to 290 have gone. Lets ' CSAVE ' or ' SAVE"D:SIGNAL.STR" ' before there is a power failure, or the dog chews up your T.V. cord. (Result--'Digital Dog'). Hey, good name for a program!

O.K. 'RUN' it. Hope it is the same as last time. It should be, except the number will be different. Strange, this number is larger than the last one by about 80 bytes. So much for using strings to save memory. Hold on, before you write me a nasty letter for putting you to so much work. We did learn a new method, and if our screen was loaded with character graphics instead of just one little old traffic signal, we would have saved memory as well as writing a more compact program.

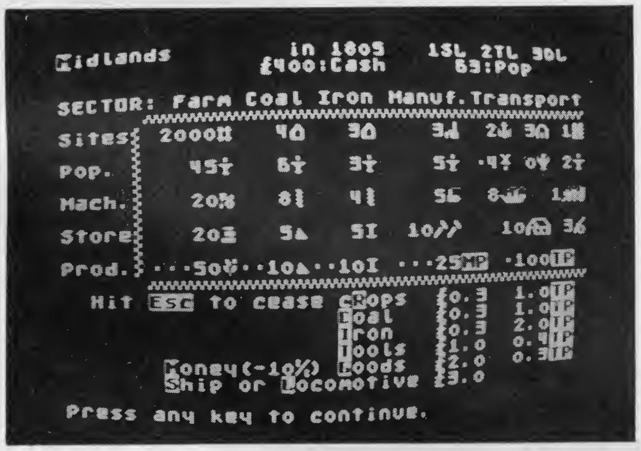
That's it for now. Next time we will delve into display lists and horizontal scrolling; after all, we have to get that car out of the garage without running over any chicks.

STRATEGY ZONE

By BOB COCKROFT

Chancellor of the Exchequer
MACH-INA,INC.
161 ARAGON AVE.
CORAL GABLES,FL 33134

Since the time I got my Atari computer back in 1981,I have noticed a large number of economic strategy games on the market. The "run country" scenario has always fascinated me. So in great anticipation,I bought or typed in these games, only to be disapointed by their simplicity. However, until recently this has all changed.The reason is 'Chancellor of the Exchequer by Mach-ina.This game is terrific, being infinitely better than other early attempts at an economic strategy game that I have seen.



```
Midlands          in 1805     1SL 2TL 3DL
                £400:Cash         63:Pop

SECTOR: Farm Coal Iron Manuf.Transport
Sites   2000   4    3      3    2  3  1
Pop.      45   6    3      5   .4  0  2
Mach.    20%   8    4     56    8     1
Store    20    5    5I  10     10    36
Prod.  ...50 ..10 ..10I   ...25MP  .100IP
   Hit ESC to cease Crops    {0.3    1.0IP
                    Coal     {0.3    1.0IP
                    Iron     {0.3    2.0IP
                    Tools    {1.0    0.4IP
        Money(-10%) Goods    {2.0    0.3IP
        Ship or Locomotive   {3.0
  Press any key to continue.
```

Chancellor of the Exchequer places you in control of Britain at the beginning of the Industrial Revolution.(1805) At this time your country has for the most part an agrarian economy, lacking many production facilities. It will be your goal to make Britain an industrial power. Success is determined by your rating for Great Britain on the Analysis Program Graph. High ratings are achieved by merging all eight regions with Great Britain, increasing popoulation,maximize machine utilization and have at least 55,000 pounds sterling in the Exchequer by the game's end.

At the start of the game Britain is divided into eight regions which have their own strengths and weaknesses. Each region has its own display screen with all vital statistics. As Chancellor it is your job to distribute surplus resources so that all regions operate efficiently. To assist the player in using resources, an Analysis program is provided. This program will indicate how well you are employing your people and how industrialized a region has become. When a region becomes industrialized enough it will automatically merge with Great Britain. This will help to unite the economy and increase your rating.

Chancellor of the Exchequer's greatest strength is its' ability to create a situation where one strategy is not the only realistic option. Different economic problems that the game produces can be corrected using many different methods. There is no built-in pattern that can be followed to insure success. Each game can be a different experience.

The game is structured so that it forces on experienced players to make decisions on a level higher than the "this or that" basis. The decision to produce either product 1 or product 2 does not end with production, it could have a rippling effect on the entire country's economy. For example, a player could be forced to make a decision to either produce goods or coal goods that could be sold abroad, the amount of money in the treasury will increase.There will be money for the many expenses. However,by selling consumer goods instead of making prducer goods one may be sacrificing growth for current prosperity. This is only a very simple illustration of what decisions must be made. Because it causes the player to be concerned with all the repercussions of his decisions, the "rippling effect" is one of the more interesting aspects of the game. Because this game is designed to promote many alternative strategies one may find that it is a little complex at first. During my first game I quickly became lost in the many symbols that are used. In addition, the instruction pamphlet, although useful as an outline, not clearly explain the mechanics of the game. Fortunately, Chancellor of Exchequer comes with an additional program called "Help". This assistance program does a fair job of instructing the player how to play the game. The various screens and symbols are explained here.

Being an economic game, not 'shoot'em up' arcade game, there are no dramatic graphic displays. However, the region statistic displays are made with some degree of creativity. Using modified character sets, the symbols graphically depict the information that is being presented. The overall display is fairly understandable, certainly for one who has gone through the Help Program. Statistics from any or all regions can be printed out on Epson printers. The print-out is a summary, including the name of the region, date, economic statistics and cost of various manufacturable items. It would be best to have the graphic chip for your printer, because without it some of the artistry cannot be printed. However, it is not essential, all important data can printed from any Epson printer. Because Chancellor of the Exchequer is a well designed strategic simulation of the industrial revolution, it is a game that I would highly recommend.

Strategy Tips

Legionnare

In Legionnare, the game is ancient warfare between Rome and the barbarian tribes of success. While fighting difficult tribes, the battle must unfold exactly how you want it to. Although easier levels can be played without much thought, the Romans must use their environment and legionnares to their maximum effiecency in order to survive. The best method of being successfull in this game can be outlined in three words: preparation, deployment and execution.

Preparation consists of deciding where you want the battle to occur and moving the legions to this location without fighting the enemy. One cavalry legion should move to the far side of the barbarian army in order to draw it away from the rest of the troops. While this is happening the main body of legionnares should move to a predetermined defensive position. The defensive position should be one in which the legionnares would be fighting downhill.The effect of this would be two-fold. One, while fighting downhill the legionnares' fighting ability will be enhanced.Two, the enemy will tire as it climbs the hill to get to your location. The battlefield should be one on which the barbarians attack from only one direction.By doing this, the Romans will create a vacant area which enemy units could not penetrate. Being an essential part of any strategy this area could be used as a movement corridor for the reshuffling of units during the heat of battle,an area where depleted legions may recuperate and a safe place to retreat in case all else fails.

There are a few places on the playfield that meet my criteria. The first, is the hill located on the bottom left corner of the playfield. Making a horizontal line with the legions, the player should be able to draw the barbarians from the north. The second, is a steep hill that is located half way down the left side of the board. To maximize the effect of the hill, draw the opposition towards this location from the east.

The legions must be deployed so that the defensive effect can be maximized. Legions fight best when attacked from only one side. As a result,a tight box-like formation is always most effective when fighting the more difficult tribes.The barbarian cavalry will attempt to pry between the legions, thus destroying the box. Once this happens it usually means the Romans' inevitable defeat. By grouping the legions ina rectangular box,there is a twofold effect. One, the legions,except for the corners, can be attacked from only one side. Two, the barbarians, would not be able to pry between the center legions because they are grouped tightly together.

The only legions that could be moved are the ones on the end of the line. Because of this the end units need to be the strongest. Casear Men or Cicero legions would work well in this location. The end unit's flank should be protected by one of the two Roman cavalry legions. Acting like a fire brigade, the cavalry would push back any opposition unit that moves to the flank of the legions . After performing its task, the cavalry should retreat one square to avoid being attacked. When a barbarian moves back to the flank of an end legion, this process should be repeated. By doing this, all or at least most legions should be at-from only one side. Although the legions may be deployed effectively, a battle strategy must be planned and executed. Because the legions fight better while on defence, in the early stages of the battle all Roman infantry should only defend. The Roman commander should allow the aggressive barbarians to tire while he is preparing for a counter-attack. Because of fatigue, the number of barbarians fighting in relation to the total number of barbarians in the field is quite small.What the Romans need to do is defeat the barbarians who are fighting, rendering the others who are resting helpless. When the barbarians become weakened as a result of fatigue,the Romans should counter-attack. The entire line of legions should advance, overwhelming the weakened opposition. If timed correctly, this strategy should enable the Romans to eliminate several barbarian formations in one strike.

WHY BUY AN ATARI?

By PETER ELLISON

In the past few issues I have written mostly about the technical use of the ATARI computer. in this issue I'm going to talk about what type of support it has apart from just the computer itself.This support is like having a toll free number in CANADA (112-800-268-1784) and the United States (1-800-268-1404) to talk direct with ATARI to find out information on user groups, new products, or anything to do with the ATARI. The people are friendly to talk to and it doesn't cost a penny to speak with them. These phone numbers are just the tip of the iceberg when it comes to the ATARI computer support.

Another type of support that is springing up all over the world is a thing called an "ATARI Computer Users' Group". An ATARI Computer Users' Group is an organization of people who own ATARI computers. Their main purpose is to share information about ATARI computer related programs and to help each other expand the usefulness and enjoyment of their computing experience.

There are many reasons why you would like to become involved in a Users' Group. Would you like to make your free public domain software grow or find a commercial software just right for your application? Or would you like to know what peripherals will work best with your ATARI computer, or do you need help in hooking up the new hardware you've just purchased? A User group should be able to answer all of these questions and more. Even if there isn't a User Group near you it is possible of starting your own. All you need is five or more people who want to share information about their ATARI computers and two or more people who are will-to get things started.

In the planning stages of your User Group it is a good idea to write the people at the ATARI Users' Group Support Program or call 800-672-1404 and ask for their "Users' Group Starter Package".

ATARI,Inc.
Computer Users' Support Program
60 East Plumeria
San Jose,Ca 94134

Earl Rice, of ATARI Users' Group Support said there are currently over 25,000 active User Group members worldwide-and more joining everyday. For a very excellent book with much of the Free Software available for your ATARI get the book called, "Free Software For Your ATARI" from Enrich.This book also gives a listing of over 250 user groups worldwide.

Besides getting free public domain software in your Users' Group,it is possible to get software over the phone by the use of a modem. A "MODEM" is the abbreviated combination of the two words: MOdulator and DEModulator. (Modulator=Transmitter and Demodulator=Receiver).A MODEM is used to connect your computer to a telephone line. It enables your computer to communicate with other computers. A diverse network of services and recreations is as close to you and your ATARI computer as your telephone. Hundreds of information services and electronic bulletin boards (BBS) are standing by at this moment waiting for you. For a very large list of different bulletin boards on which you can use your MODEM, get the book which I mentioned in the last paragraph, "Free Software For Your ATARI".

Another very useful way of getting information about your computer is through the use of magazines.These programs are free for the typing in. There are three publications that I know of. They are:

Continued on page 68

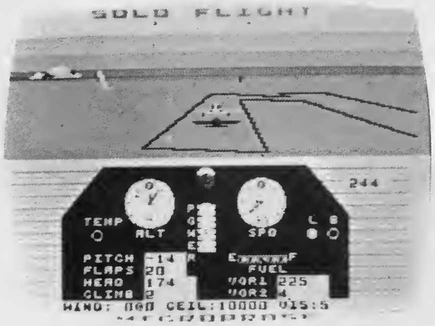
AT LAST!!! A REAL FLIGHT SIMULATOR!!!

— From The Creators of HELLCAT ACE!

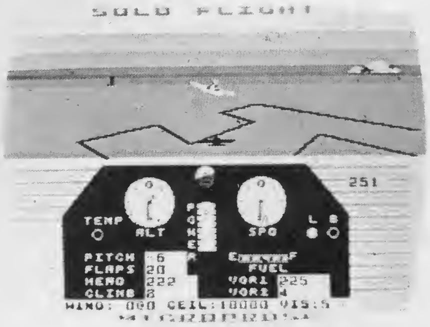
A real flight simulator allows you the pilot to accomplish Takeoffs, Landings, Cross-Country Navigation, and Emergency Procedure practice under Clear (VFR), Cloudy (IFR), and Cross-Wind conditions. SOLO FLIGHT is such an advanced simulation that uses realistic three dimensional terrain graphics, actual configuration instrument panel, multiple air navigation maps, and accurate aircraft performance characteristics to provide the challenge, thrill, excitement, and joy of the flight experience!!!

- **Extensive flight manual including Basic Flying, Instrument Flying, Maps, and Charts.**
- **Three geographical air navigation areas each with multiple airports and multiple runways.**
- **Dual Radio Navigation (VOR) and Instrument Landing System (ILS).**
- **On-screen flight path review between flights.**
- **Multiple Landing Approach Charts and Design-your-own approach feature.**

- **Takeoffs and Landings under all weather conditions.**
- **Cross-Country Navigation via Landmarks, VOR Stations and Dead Reckoning.**
- **Emergency Procedure Practice.**
- **Air Mail delivery game to test your flying skills.**
- **Landing ratings for smoothness and speed.**
- **For all Atari computers, Commodore 64, Disk and Cassette . . . $34.95.**



```
                                 244
PITCH -14
FLAPS 20
HEAD  174      VOR1 225
CLIMB 2        VOR2 4
WIND: 000 CEIL:10000 VIS:5
```



```
                                 251
PITCH 6
FLAPS 20
HEAD  222      VOR1 225
CLIMB 8        VOR2 4
WIND: 000 CEIL:10000 VIS:5
```

Look for MicroProse's two new exciting real time combat simulations — MIG ALLEY ACE & NATO COMMANDER. Write or call for our Free Catalog.

If you cannot find our games at your local store, you can order by MasterCard or VISA, Money Order, COD or Check. Add $2.50 for Postage and Handling. Maryland Residents add 5% Sales Tax.

MicroProse Software

DEALER INQUIRIES INVITED.

10616 Beaver Dam Road, Hunt Valley, MD 21030
(301) 667-1151

ATARI and Commodore 64 are the registered trademarks of ATARI, Inc. and Commodore Business Machines

BOOKS ON THE SHELF

By PETER ELLISON

In this issue I'm going to to review three different books. They are "Free Software For Your ATARI"(See Why Buy An AtARI?), "6502 Assembly Language subroutines", and "Beyond Games:Systems Software for Your 6502 Personal Computer".

FREE SOFTWARE FOR YOUR ATARI

The first book, "Free Software for your Atari is a book every Atari user should own. This book deals with many things. From Free Public Domain Software to a list of user groups.

If you have just bought a MODEM or are planning on buying one, this book gives an overview on the different ones on the market. It also shows you how a person should go about setting one up. It then shows the differences between an Acoustic MODEM and a Direct - Connect MODEM. It then gives you some buying tips for when you're ready to buy one.After giving a list of twenty different MODEMS on the market it then shows you how to operate it. It gives you a short program called "MINIATRM", which when run can be used to communicate with ATARI BBS's (Bulletin Board Services).

The next chapter then deals with communicating with a BBS once their on the line. It gives you a list of typical BBS Instructions. Example:H-Help.Prints this list or Q-Quick scan. Scans message titles only.Once you have all this down pat the next little section gives you a directory of Atari Bulletin Boards. It then gives a list of 14 pages of Public Access Message Systems that give different types of information.

Besides just helping you get started with a MODEM this book has a list of over a hundred User Groups around the world. It also shows you how simple it is for you to become a User Group. This book, I feel,is one the most valuable books that I have picked up for my ATARI. This book retails for $8.95 and can be bought direct from:

ENRICH/OHAUS
2325 Paragon Drive
San Jose,CA 95131

6502 ASSEMBLY LANGUAGE SUBROUTINES

The more familiar that you become with the BASIC language, the more you realize how slow the speed is. So to speed things up you'll need some machine language subroutines. This is the book for you. This book identifies the strengths and weaknesses of the 6502 instruction set, and allows you to make instant use of 6502 assembly language.

Although this book is not written specifically for the ATARI, all that is said in this book can be used since the ATARI has a 6502 microprocessor. If you want to use a specific assembly language routine, learn assembly language quickly, or improve your programming skills, 6502 Assembly Language Subroutines is for you. It provides code for more than 40 common 6502 subroutines, including code conversion, array manipulation, arithmetic, bit manipulation, string processing, input/output, and interrupts. It describes general 6502 programming methods(including a quick summary for experience programmers), and tells,how to add instructions and addressing modes. It even discusses common 6502 assembly language programming errors.

With over five hundred pages this book can help the beginner to the advanced improve his or her programming skills in assembly language. This book is available from:

OSBORNE/McGraw-Hill
630 Bancroft Way
Berkeley,California 94710

BEYOND GAMES:SYSTEMS SOFTWARE

FOR YOUR 6502 PERSONAL COMPUTER

This book is also directed to the learning of assembly language, but does have some referto the ATARI computer. This book deals with the Atari, Apple, Pet, and OSI because they all use the 6502 microprocessor.

Beyond Games moves through a fast but surprisingly complete course in assembly language programming. Having mastered these these fundamentals, the reader is introduced to many useful subroutines and programming tools, such as screen utilities, print utilities, a machine language monitor, a hexadecimal dump tool, a disassembler, and a simple, screen-based text editor.This book written by Ken Skier, a systems analyst for Wang Laboratories, can also be used as a reference quide for the other 6502 computers. Since it gives examples for the Apple, Pet, and OSI it makes it possible to understand how the other computers function.

I recommend both these books on assembly language,because they they are both clear in the writing,making assembly not that dificult to understand.I'm not saying it is really easy to understand assembly language, but with the help of these two books it will give you somewhere to start. Beyond games is available from:

BYTE/McGraw-Hill
Book Division
70 Main St.
Peterborough,NH 03458

MISSILE MATH

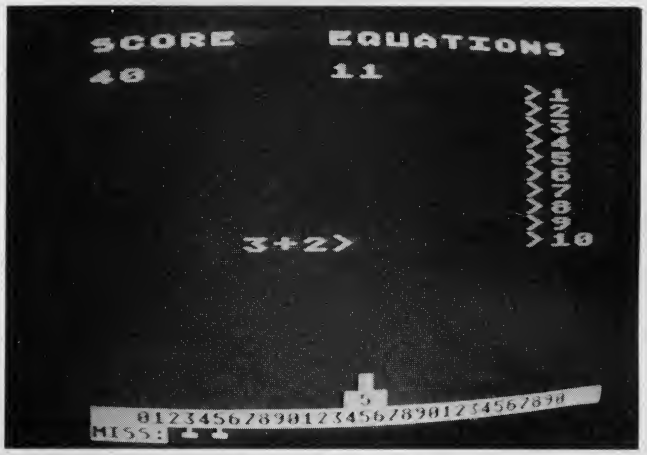
By TUONG (Tom) TRAN

Many times it is very hard to get a youngster to do any type of school work. The program below is a combination educational-arcade type game. The game requires the child to use both his brains and arcade skill to get a good high score. The game needs 16K and a joystick. In the game different math equations move across the top of your screen, and with your joystick you position your number shooter over the answer at the bottom of the screen and pull back. This will then pick up the number and display it on the bottom of your Number Shooter.

Once you have the correct answer displayed on your shooter, position the answer (with joystick), directly under the equation. Then push the red button and joystick forward, and the number will be fired at the equation. If the answer is correct you'll hear a big explosion and ten points will be added to your score. If you are incorrect then part of the equation will break off and fall on your Number Shooter and destroy it. This will subtract one life and ten points from your score.

The game begins by giving you a few options. They are EQUAT (Equations, as many as you wish), LEVEL(Up to 15), LIVES(Up to 15), START(Begin game). Each option is selected or changed by the movement of the joystick. Once the flashing arrow is beside the option you want, press the red button. This will then allow you to change the option by pushing the joystick up(for higher) or down (for lower). The higher the level that you choose, the faster the equations cross the screen. This makes it very difficult, because the child has to first think of the correct answer and then pick the number up with the joystick.

This game was written entirely in BASIC without any assembler subroutines whatsoever. This program shows how easy it is to make a good educational tool without having to learn assembly language. If you have any changes that might make this program more enjoyable,I'd be happy to hear of them. Have fun and remember this game is for Kids!(Right?)



```
10 REM ---BY--TOM--TUONG--TRAN---
15 REM ---FOR ROM ISSUE 5--------
20 DIM O$(1),B(3),L(14):B(1)=20:B
(2)=1:B(3)=1:F=128:FOR Q=0 TO 14:
L(Q)=0:NEXT Q:GOTO 810
30 C=0:GRAPHICS 0:X=20:SCREEN=PEE
K(88)+256*PEEK(89):P=SCREEN+X+40*
21
40 GRAPHICS 1:POKE 710,96:SETCOLO
R 2,0,0:POKE 752,1
50 ? "[illegible]" :POKE 85,1:? "▓▓01234567890123
45678901234567890▓▓":GOSUB 560
60 ? #6;"score    equations":? #6:
? #6:LEVEL=1:FOR W=1 TO 10:? #6;"
          >";W:NEXT W
70 GOSUB 80:GOTO 200
80 FOR Q=1 TO 50:NEXT Q:B(1)=B(1)
-1:POSITION 9,2:? #6;"      ":POS
ITION 9,2:? #6;B(1):IF B(1)<=0 TH
EN 810
90 I=1:A=INT(RND(0)*6):B=INT(RND(
0)*5):RN=INT(RND(0)*2):IF RN=0 TH
EN D=A+B:O$="+":RETURN
100 IF B>A THEN TE=A:A=B:B=A
110 D=A-B:O$="-":RETURN
120 FOR PO=SD TO SD+27 STEP 5:VOL
=14-PO/20:SOUND 0,PO,0,VOL:SOUND
1,PO,8,VOL:SOUND 2,PO+15,2,VOL:NE
XT PO
130 SD=SD+20:RETURN
150 POSITION 1,13-LEVEL:SOUND 1,7
0-I-B(2)*2,8,B(2)/15*2+0.5
```

```
160 POKE 85,I:? #6;"     ";:I=I+1
170 POKE 85,I:? #6;A;O$;B;">";
180 IF I>=13 THEN POKE 85,I-1:? #
6;"     ";:SCORE=SCORE-(11-LEVEL):
SOUND 1,0,0,0:GOSUB 80:GOTO 470
190 RETURN
200 GOSUB 150
210 FOR Q=1 TO 15-B(2):GOSUB 310:
NEXT Q
220 GOTO 200
230 IF T=11 THEN X=X-1:GOTO 260
240 IF T=7 THEN X=X+1:GOTO 260
250 IF T=13 THEN F=PEEK(P+40)
260 IF X<=1 THEN X=X+1
270 IF X>=38 THEN X=X-1:RETURN
280 POKE P+1,0:POKE P-1,0:POKE P,
0:POKE P-40,0:P=SCREEN+X+40*21
290 POKE P-1,128:POKE P,F:POKE P+
1,128:POKE P-40,128
300 RETURN
310 IF STRIG(0)=0 AND C=0 THEN H=
P-60+(20-INT(X/2+0.5)):GOSUB 150:
GOSUB 340
320 T=STICK(0):IF T<>15 THEN GOSU
B 230:RETURN
330 FOR Q=1 TO 250-B(2)*12:NEXT Q
:RETURN
340 SD=20:SOUND 1,0,0,0
350 C=C+1:H=H-20:SOUND 0,SD,4,5:S
D=SD+5
360 IF PEEK(H)<>0 THEN SD=5:GOTO
400
370 IF C>20 THEN GOSUB 480:C=0:SO
UND 0,0,0,0:RETURN
380 POKE H,F-128
390 GOTO 340
400 REM
410 SOUND 0,0,0,0:GOSUB 480:IF D=
F-128-16 THEN 430
420 GOSUB 490:GOTO 200
430 GOSUB 570
440 POKE 85,I-1:? #6;"     ";:GOS
UB 80
450 SCORE=SCORE+10*LEVEL:IF SCORE
>100*LEVEL AND LEVEL<10 THEN LEVE
L=LEVEL+1
460 IF B(2)<15 AND SCORE>1000*B(2
) AND LEVEL>=10 THEN B(2)=B(2)+1:
LEVEL=1
470 C=0:POSITION 1,2:? #6;"      "
:POSITION 1,2:? #6;SCORE:RETURN
480 FOR W=C TO 1 STEP -1:POKE H+W
*20,0:NEXT W:RETURN
490 A1=PEEK(H):FOR W=1 TO 200:NEX
T W
500 FOR W=0 TO C-1:POKE H+20*W,A1
```

```
510 FOR Q=X1 TO X1+10:X1=X1+10:SO
UND 0,Q,10,8:NEXT Q:X1=0
520 POKE H+20*W,0:NEXT W:GOSUB 80
0
530 POKE P,0:POKE P-40,0:POKE 85,
I-1:? #6;"     ":C=0
540 FOR W=1 TO 10:POKE P+W,128:PO
KE P-W,128:GOSUB 120:POKE P+W,0:P
OKE P-W,0:NEXT W:GOSUB 600
550 ? "‡":? "MISS!!";:FOR Q=1 TO B(3)
:? "  ";:NEXT Q:? "‡":B(3)=B(3)-1
:IF B(3)<=0 THEN 810
560 ? "MISS!!"    ;:FOR Q=1 TO  B(3) :
? "⊥";:NEXT Q:POKE P,F:POKE P-40,
128:POKE P+1,128:POKE P-1,128:I
=1:RETURN
570 FOR W=0 TO 5
580 GOSUB 800
590 NEXT W
600 FOR Q=0 TO 3:SOUND Q,0,0,0:NE
XT Q:SD=0:RETURN
610 GRAPHICS 2+16:S=PEEK(88)+PEEK
(89)*256+20*5+1:? #6;" **********
******";
620 ? #6;"** ":? #6;"    missile-
math":? #6:? #6;" BY: TOM TUONG T
RAN ":? #6:? #6; "EQUATN:";B(1)
630 POKE S+20,30:? #6;"LEVEL: ";B(
2):? #6;"START ":";B(3):? #6;"LIVES:
":? #6:? #6;" *****************"
:C=1
640 IF STRIG(0)=0 THEN 690
650 T=STICK(0):IF T=15 THEN 640
660 POKE S+C*20,0:IF T=14 AND C>1
THEN C=C-1:GOTO 680
670 IF T=13 AND C<4 THEN C=C+1
680 POKE S+C*20,30:FOR Q=1 TO 50:
NEXT Q:GOTO 640
690 IF C=4 THEN V1=B(1):V2=B(2):V
3=B(3):GOTO 30
700 FOR Q=S+C*20+1 TO S+C*20+7:PO
KE Q,PEEK(Q)-128:NEXT Q:FOR Q=1 T
O 100:NEXT Q:S2=0
710 IF STRIG(0)=0 THEN 790
720 T=STICK(0):IF T=15 THEN 710
730 S1=0:IF T=14 THEN S1=1
740 IF T=13 AND B(C)>0 THEN S1=-1
750 IF C=1 AND (B(1)>=16 OR T=14)
THEN 780
760 IF C>1 AND (B(C)<=14 OR S1=-1
) THEN 780
770 GOTO 710
780 B(C)=B(C)+S1:POSITION 9,5+C:?
#6;"     ":POSITION 9,5+C:? #6;B
(C):FOR Q=1 TO 20:NEXT Q:GOTO 720
790 FOR Q=S+C*20+1 TO S+C*20+7:PO
```

Continued on page 59

MACTODAT

By Bob Cockroft

Have you ever wanted to make a machine language subroutine for a Basic program using an assembler, but did not want to waste time rewriting the Operand (Op) codes in data statements? Your problems are over! This is a program which automaticlly rewrites the machine code you have created with an assembler into data statements. By doing this the code can be loaded with the BASIC,and integrated into your program. Before any data lines are created, the program will ask a few simple questions about the machine code and how you want it presented. You will need to know where your machine code starts and ends,at what line number your data statements are to start, and how much you wish the lines incremented by. By using untokenized files(LIST,ENTER) it is possible to add data statements to programs. First, save the MACTODAT program listed below, using the LIST statement, thus creating a untokenized file. Then do the same to the program to which you wish to add data statements.Then, clear the memory of any BASIC program with the NEW statement. LOAD in the machine code you want converted. Now reLOAD in the MACTODAT program with the ENTER command(tokenized files) and RUN it. Be careful not to have your machine code in memory locations used by the BASIC program. The computer will now ask you if you wish to erase the MACTODAT program so that it does not interfere with the program you wish to add the data statements to. After the MACTODAT program has finished, ENTER the program to which you want to add the DATA statements. If you have done all these steps correctly, you will have successfully merged your program with the newly created DATA statements.

The MACTODAT program creates the DATA statements by PRINTing them on the screen. What makes this program interesting is its use of the memory location 842. While the DATA statements are being drawn, memory location 842 contains a 12, the value of "write to the screen" mode. However, after a screen of DATA statements has been created, the cursor is placed at the top of the screen, and a value of 13 is POKEd into the memory location in order to put the computer into the "return key mode". What you will see is the cursor at the top left corner of the screen, then moving down as if the RETURN key were repetitively pressed. Theoretically this is what is happening. By poking 13 into location 842 Dec., one tricks the computer into thinking the RETURN key is pressed. As the cursor passes each line of data, the line is added to your program as if you had manually typed a line in and RETURNed the cursor yourself.

It is important to note that at the bottom of a screen of data is a line that reads:

```
POKE 842,12:GOTO
```

This line tells the computer to return to "write mode". In other words the cursor is prevented from moving any farther down the screen and allows the computer to function in its normal way. The second part of this line tells the computer where to go to next.

Although this program did not take me very long to make, I have found it helpful in making machine code programs for ROM,and hopefully, it will be equally helpful to you.

```
1 GRAPHICS 0:? :? " RETURNS DATA
LINES FROM MACHINE CODE":?
2 ? "Print START and END address
of machine code":INPUT A1,A2:GRAP
HICS 0:POKE 842,12:POSITION 2,2
842,12:POSITION 2,2
3 DIM AN$(10)
4 ? "print beginning DATA line nu
```

```
mber(must be greater than 180)":I
NPUT LOW
5 ? "line numbers incremented by
what?":INPUT INCR
6 ? "do you want to ERASE the
orginal PROGRAM?":INPUT AN$
7 GRAPHICS 0
25 FOR LINE=LOW TO 32000 STEP INC
R
30 POSITION PEEK(82),3*COUNT+3:I=
0:COUNT=COUNT+1:PRINT LINE;"DATA"
;
82 I=I+1
88 IF I=24 THEN ? PEEK(A1);:A1=A1
+1:GOTO 120
90 PRINT PEEK(A1);",";:A1=A1+1
92 IF A1=A2 THEN ? PEEK(A1);:? :?
:?"POKE 842,12:GOTO 160":GOTO 155
95 GOTO 82
120 ? :IF  COUNT=6 THEN ?:? "POKE
842,12:GOTO 160":GOTO 155
150 NEXT LINE
155 POSITION PEEK(82),0
159 GOTO 180
160 IF A1>=A2 THEN 162
161 COUNT=0:I=0:GRAPHICS 0:GOTO 1
50
162 IF AN$="N" THEN END
163 GRAPHICS 0:POSITION PEEK(82),
8
164 ? "2":? "25":? "30":? "82":?"
88":? "90":? "159":? "92":? "95":
? "120":? "150":? "155"
165 ? "160":? "162":? "163":? "16
4":? "165":? "180":? "161":? "POK
E 84 2,12:GOTO 175"
170 GOTO 176
175 GRAPHICS 0:POSITION PEEK(82),
8:? "1":? "3":? "4":? "5":? "6":?
"7":? "176":? "175":? "170":? "PO
KE 842,12:LIST"
176 POSITION PEEK(82),0:POKE 842,
13
180 POKE 842,13
```

THE RAVING REVIEWER

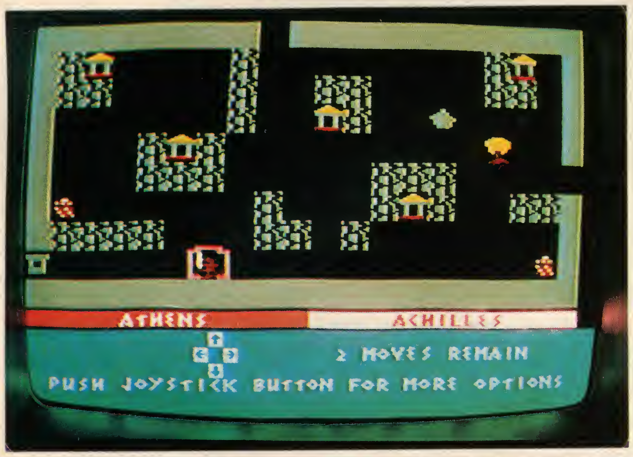
By TIM REEKIE

The Return Of Heracles
Quality Software
6660 Reseda Blvd.
Reseda,CA 91335

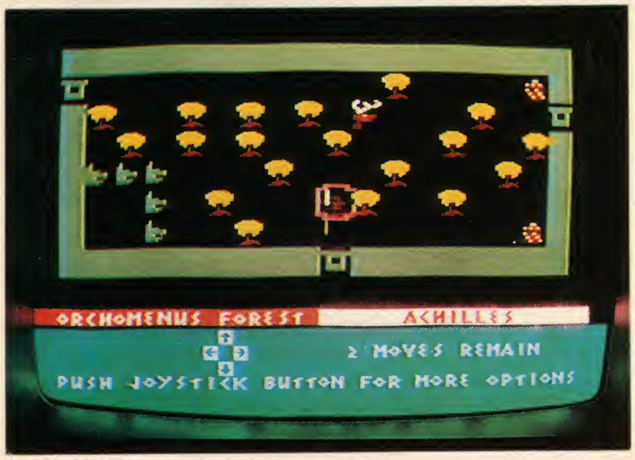
This is not so much the "Return of Heracles" as it is the return of Stuart Smith to the software scene. After the successful release of "Ali Baba" in 1981, Stuart went into seclusion with what must have been a library full of Greek Mythology and history. The product of his toils is well worth the wait.

The Game

The Return of Heracles(Heracles, for short) is set in ancient Greece. Under the tutelage of Zeus, each player may guide as many heroes as he or she wishes to follow in the footsteps of Heracles(a.k.a. Hercules).



ATHENS ACHILLES
2 MOVES REMAIN
PUSH JOYSTICK BUTTON FOR MORE OPTIONS



ORCHOMENUS FOREST ACHILLES
2 MOVES REMAIN
PUSH JOYSTICK BUTTON FOR MORE OPTIONS

Before you lie the twelve tasks of Heracles from the Oracle of Zeus, your heroes receive instructions per your task. If you wish, you may travel to the Oracle of Delphi, deposit a contribution and receive hints and clues regarding your task. The more money donated, the better the clues will be (You pays your moneys, you takes your chances). When I say "heroes", I am not implying gender, nor even species, but merely describing a heroic specimen. For example:Pegasus,the winged horse, is, for our intents and purposes, a hero.

Now that we have our definitions straight, which heroes are the best choices? From our experience(The Adventure Duo et. al.), if there are less than five, players, multiple characters is a good idea. Throughout the game, unless played exceedingly well, some player is going to snuff it (that's part of the game folks) and the readying of new heroes is time consuming. What with the gathering of gold to buy a new sword and dagger (the originals are invariably cheap) among other things, the acquisition of new players takes up time. Therefore, the back up hero is a good idea, unless there are quite a few people playing.

That still doesn't clarify who the best choices are. I may hear arguments from the wings on these choices, but here they are anyway:

(1)Achilles:Achilles is the power-house among the heroes; he is not only fast, but also strong, healthy, and dextrous. He already has excellent armor (the River Styx etc.) and armed with a poisoned sword and dagger, he is almost invincible(more about this later). To further the argument, along with Achilles, comes his inseperable companion and cousin, Patroclus, who is also an excellent character.

(2)Thesus:Having inherited a fine suit of armor and sword he almost rivals Achilles. The rest of the

Continued on next page

heroes fall into the also-ran category with the Great Ajax and Perseus topping the list.

Furthermore, what the Adven-Duo has found to be good strategy, is to acquire two additional draft choices:Asclepius(the healer) and Pegasus. These two,heroes are basically wimps but their forte is not only the power to heal himself one point per round, but if he stops on top of a hero, or vice-versa, that hero is completely healed;a handy asset in a battle. Pegasus is a mule. Actually he is a winged horse, but he is not only able to carry great amounts of gold, but can carry alot of gold at high speeds.This, in effect, makes Pegasus a great pack-horse. A warning here, however: the more gold piled on, the slower any given hero will be able to move.

Even with heroes like Achilles, death at any time is possible. Besides the usual deaths at the hands of monsters (or anti-heroes), there are needles in chests and the like. Fortunately, (or unfortunately) there can be no invincible characters for, along the way, there are "Insta-Deaths". This is my name for the unavoidable slayings that include fates such as quicksand and intervention from the gods. We once had an adventure when Achilles (Played by J.S.D.) found a forgetfulness potion and inconviently forgot who he was, thereby wandering aimlessly for the balance of the game.The trouble was, because he was being controlled by J.S.D., no one noticed that Achilles was a mindless droid until we found J.S.D. listening to Guy Lombardo on his stereo and singing,"should old acquaintance be forgot...". We then began to suspect!

Unlike Ali-Baba,once a character dies in Heracles he/she/it can not be reincarnated. This keeps the game on a more realistic level, or, at least, as realistic as possible, because it encourages players to nurture their heroes along and protect them more.(unless,of course,your player is the worst and you want to kill it off as soon as possible!)

The game itself is played along the lines of Ali Baba. You move your joystick or keyboard and may attack enemies by wrestling(on same square) or swordplay (between neighbouring spots). The documentation includes the strike probability table used in Heracles,for those interested in such things. However, in Heracles, it is impossible(much to J.S.D. chagrin) to strike another player character.Hapless characters need no longer fear for their lives when the J.S.D's of this world are on the prowl. If it is at all possible to avoid encounters, I suggest doing so,as killing creatures brings no rewards except when 1)it happens to be a task or 2)you are a sadist and it makes you feel good inside. There are, however, a few beasts that block your path and must then "pay the debt which cancels all others".

There is, in Ali Baba,a feature called the Random Monster Generator, which enabled players to select the chance of a monster appearing in the room.It was generally agreed that this was an excellent idea as it made for better playability, but curiously was not incorporated into Heracles. Fortunately there are more than enough enemies to carve your way through, without having wandering monsters to contend with. What features does this game have then?, you ask. Someone must have sat Stuart down and said:

"Stu baby,Ali Baba is an excellent game but hey, between you and me, it can be deadly slow at times. In a room full of beasts, each one must move and/or fight. I mean, who cares what happens in a fight between a bear and a rat if we're ready to leave jump out the door,anyway?And the messages:

Raving Reviewer cont'd

When one comes up there is a barrage of button pushing to get back to the game!"

The result of this wee wordie is that in Heracles, there are two additional commands to control monster and message speed. These two features really speed up the game.

Besides the primary objective of completing the twelve different tasks of Zeus,there is also a secondary objective of getting the most points while doing these tasks. A score can approach but never reach 10,000 points. While we have achieved a score of just over 8600, Stuart Smith says he has gotten just over 9500.

Heracles has a different atmosphere than does Ali Baba. The music (I will not call it sound) reflects the Greek background as the music in Ali Baba reflected the Arabian background. Of course the graphics are basically similar, although Heracles does have animation of a sort.

If anything, Heracles is the further adventures of Mr. Stuart Smith into the world of myth and adventure.

Welcome back to the software scene, Stuart. I hope it isn't another couple of years before your next game goes on sale.

Ratings
Return of Heracles

Playability......	9.4
Challenge........	8.9
Graphics.........	9.7
Music(Sound).....	9.8
Documentation....	9.0
Overall Rating...	9.4

SPEEDING UP YOUR JOYSTICK

By BOB COCKROFT

Because the Atari Basic Language is so slow, any method to make programs operate faster can be quite useful. Joystick commands are an area which technique can minimize some of the languages weaknesses.

The first step is to set a variable equal to the stick position.

```
ST=STICK(0)
```

Next determine whether the stick is pressed in the direction of the vertical(y) or horizontal (x) axis.

```
SY=(ST=13)-(ST=14)
SX=(ST=7)-(ST=11)
```

For example, if the stick is pushed in the direction which causes it to register a '7', SX will equal 1. Conversely if the stick registers an '11' SX will equal -1. In this manner, variables SY and SX will have provided a program with a more efficient means of registering joystick's position.

The following is a program that applies the principles mentioned to create a simple screen painter. The only significant difference from the example above is that this program provides for diagonal movement.

```
10 REM * JOYSTICK SUBROUTINE *
12 X=50:Y=50
20 GRAPHICS 8:COLOR 1:SETCOLOR 2,
16,1:POKE 752,1
25 ? "        Press button to era
se"
30 ST=STICK(0)
40 SY=(ST=13 OR ST=9 OR ST=5)-(ST
=14 OR ST=10 OR ST=6)
50 SX=(ST=7 OR ST=6 OR ST=5)-(ST=
11 OR ST=10 OR ST=9)
60 Y=Y+SY:X=X+SX
70 PLOT X,Y
75 IF STRIG(0)=0 THEN COLOR 0:GOT
O 30
77 COLOR 1
80 GOTO 30
```

DISK SPEED CHECKER

By BOB COCKROFT

If your disks get many format errors, or your programs cannot be read by someone elses drive, your drive could be rotating at the wrong speed. Some disk drives, particularly ones manufactured before 1982, have been known to get out of speed adjustment. The drive was intended to spin disks at 288 revolutions per minute(RPM). However, because of a poor data separator, the rotation speed may vary from its correct setting. Although not perfect, the disk should still be able to function correctly when the speed ranges anywhere from 285-290 RPM.

When a drive rotates too slowly, data is stored so close together that it becomes difficult to read. The drive that wrote the data, because it is already running at a slow speed,may be able to read it, but a faster drive would not. Because the inner tracks on a disk have a smaller circumference,the rotation is slower, and therefore the data here is stored closer together. These inner tracks are represented as the higher sector. As a result, higher sectors are more susceptible to error on slow disk drives than lower sectors.

When a drive rotates too quickly, it may not write to all 18 sectors on each rotation. The drive could skip a sector,resulting in destruction of the item being saved.

The following program is a utility that will determine whether or not your disk drive is operating at the correct speed. The BASIC section of the program only sets up the environment for the machine code that does all the work. To determine how the machine code operated refer to the assembler listing and read on.

The first part of the program is self explanatory; lines 15-24 defines the variable used in the assembler program and line 10 clears the stack.

DUNIT(hex 301) (dec 769) is the location that indicates the drive that is to be used. The user must POKE the value (1-4) representing the wanted disk drive.

DBUFLO/HI (hex 304,305) (dec 772,773) is the data buffer address of the destination of the data to be transfered.

DCOMND(hex 302)(dec 770) is the location that indicates the disk operation to be performed. The options are:

Command	hex	dec
Read	52	82
Write	57	87
Status	83	83
Put	50	80
Format	21	33
Download	22	32
Read address	54	84
Read spin	81	81
Motor on	55	85
Verify sector	56	86

DAUX 1/2 (hex 30A ,30B) (dec 778,779). This location indicates the disk sector number which will be read or written to. This program sets this value to one.

Basic Listing

```
5 DIM YN$(3)
10 GRAPHICS 0:SETCOLOR 2,16,1
20 POSITION 10,8:? "Loading Machi
ne Code"
30 FOR X=0 TO 62:READ D:POKE 1536
+X,D:NEXT X
40 GRAPHICS 0:SETCOLOR 2,16,1
50 POSITION 12,5:? "DRIVE SPEED T
EST"
60 ? :? :? "Test Which Drive?"
70 INPUT N
80 POKE 1616,1
90 ? :? "The test will take 18 se
c"
100 X=USR(1536)
110 MIN=(PEEK(1618)+256*PEEK(1619
))/3600
120 RPM=INT(85/MIN+0.5)
130 ? :? ,"     ";:? RPM;:? "  RP
M":?
140 IF RPM=288 THEN ? "Drive spee
```

Disk Speed Checker cont'd

```
d is perfect":GOTO 200
150 IF RPM<280 THEN ? "Drive spee
d is too slow":GOTO 200
160 IF RPM>290 THEN ? "Drive spee
d is too fast":GOTO 200
180 ? "Drive speed is o.k."
200 ? :? "Would you like a retest
?"
210 INPUT YN$
240 IF YN$="Y" THEN 40
500 DATA 104,173,80,6,141,1,3,169
,5,141,5,3,169,82,141,2,3,169,1,1
41,10,3,160,0
510 DATA 140,11,3,140,4,3,169,85,
141,81,6,32,83,228,160,0,132,19,1
32,20,32,83,228,206
520 DATA 81,6,208,248,166,20,165,
19,141,83,6,142,82,6,96
```

Assembler Listing

```
00010  .LI OFF
00011  .OR $600
00012  .TA $4000
00014  .TF "D:RPM.OBJ"
00015 DAUX1 .EQ $30A     ;DISK SEC
TOR NO. (LO)
00016 DAUX2 .EQ $30B     ;(HI)
00017 DBUFLO .EQ $304    ;DATA BUF
FER ADDRESS (LO)
00018 DBUFHI .EQ $305    ;(HI)
00019 DUNIT .EQ $301    ;DISK UNIT
NUMBER
00020 DCOMND .EQ $302    ;DISK COM
MAND
00022 RTCLOK1 .EQ $13    ;CLOCK
00024 RTCLOK2 .EQ $14    ;CLOCK
00100 BEGIN PLA       ;CLEAR STACK
00110  LDA $650
00120  STA DUNIT      ;DRIVE NUMBER
00130  LDA #5
00140  STA DBUFHI     ;RESERVE BUFF
ER AREA
00150  LDA #82        ;SET DISK OPE
RATION
00160  STA DCOMND
00170  LDA #1         ;SECTOR NO. 1
00180  STA DAUX1
00190  LDY #0
00200  STY DAUX2
00210  STY DBUFLO
00220  LDA #85        ;NO. OF DISK
SPINS
00230  STA $651
00240  JSR $E453      ;GET DISK UP
TO SPEED
00250  LDY #0        ;START CLOCK
00260  STY RTCLOK1
00270  STY RTCLOK2
00280 L2 JSR $E453   ;DISK HANDLE
R ENTRY
00290  DEC $651
00300  BNE L2
00310  LDX RTCLOK2   ;RECORD TIME
00320  LDA RTCLOK1
00330  STA $653      ;STORE TIME
00340  STX $652     ;BASIC PROGRAM
00350  RTS
```

Missile Math cont'd

```
KE Q,PEEK(Q)+128:NEXT Q:FOR Q=1 T
O 100:NEXT Q:GOTO 640
800 SETCOLOR 2,3,3:SETCOLOR 4,3,3
:POKE 755,4:GOSUB 120:SETCOLOR 2,
0,0:SETCOLOR 4,0,0:POKE 755,2:GOS
UB 120:RETURN
810 GRAPHICS 1+16:IF B(1)<15 THEN
B(1)=16
820 ? #6;" ▒▒▒▒▒▒▒▒▒▒ ":? #6;"
missile-math":? #6:? #6:?  #6;"
2+5-4+3-2+1-8+ ":? #6:? #6
830 ? #6;"▒▒▒▒▒▒▒▒▒▒": ? #6:? #
6;" BY:TOM TUONG TRAN":? #6
840 IF SCORE>HI THEN HI=SCORE:H1=
HI:C1=1:GOSUB 920

850 ? #6:? #6;"      HI-SCORE":?
#6: POKE 85,(20-C1)/2:? #6;HI:? #
6:? #6;"  SCORE:";SCORE:? #6;" EQ
UATIONS:";B(1)
860 ? #6;"  LEVEL:";B(2):? #6;" L
IVES:";B(3):? #6:? #6;" *********
*********"
870 IF STRIG(0)=0 THEN 610
880 GOSUB 940:POSITION 2,5:? #6;L
(1);"+";L(2);"-";L(3);"+";L(4);"-
";L(5);"+";L(6);"-";L(7);"+";L(8)
890 FOR Q=1 TO 50:NEXT Q
900 POSITION 2,5:? #6;"-";L(1);"+
";L(2);"-";L(3);"+";L(4);"-";L(5)
;"+";L(6);"-";L(7);"+":FOR Q=1 TO
50:NEXT Q
910 GOTO 870
920 H1=H1/10:IF H1<1 THEN RETURN
930 C1=C1+1:GOTO 920
940 FOR Q=8 TO 1 STEP -1:L(Q)=L(Q
-1):NEXT Q:L(1)=INT(RND(0)*10):RE
TURN
```

NEW PRODUCTS

GRAPHICS HARDCOPY

The Graphics Hardcopy program is an all new method of printing high resolution graphics on Epson, Okidata, Gemini, NEC, Centronics, Trendcom or IDS Paper Tiger printers.The Graphics Hardcopy program will let the user get a hardcopy of just about anything that appears on the display screen of an Atari computer.

This program will print any graphic or text mode, including mixed mode with custom display lists. The horizontal and vertical size of the printout can be increased or decreased. The chart or picture can be composed on the screen then printed in several sizes,from quite small to several feet wide. When the width exceeds the width of the paper the picture is automatically printed in strips.There is an automatic gray scale simulation which the user may select to represent either grey scale simulation which the user may select to represent either color or luminence levels.

This program requires 32K and can be bought for $34.95 from:

Macrotronics,Inc.
1126 N. Golden State Blvd.
Turlock,CA 95380

BEAT THE BEATLES

This new text game from Interactive Software is a challenging one for all ages. It comes with a 30 day warranty and a 24 hour User assistance hotline. A Free subscription to the Beat the Beatles Network is also included with the game. This game although only text should keep any person or persons occupied for hours.

This program requires 48K and can be bought for $24.95 from:

Interactive Software
P.O. Box 991
BALA-CYNWYD,PA 19044

NEW KID ON THE BLOCK

A new company called "Starfire Games" has just released two new games for the ATARI.They are Time Machine 1 and Global Thermonuclear War. The first of the two, is a game where you're the pilot of a USAF/NASA experimental vehicle in search of the first three pilots,who have disappeared into a different time zone. The second game involves you and your ATARI into World III hoping you'll be able make the right decisions.

TWO NEW PROGRAMS FROM ROM

The two programs are Grid Warrior and PM EDITOR/ANIMATOR. The first program is a challenging four player arcade game. It features high resolution graphics with twelve different levels of play. Each player steers his Light Cycle to avoid hitting the tail of another ones bike. Play against the computer or against three other human opponents. This game should keep any arcade goer amused for many hours to come.

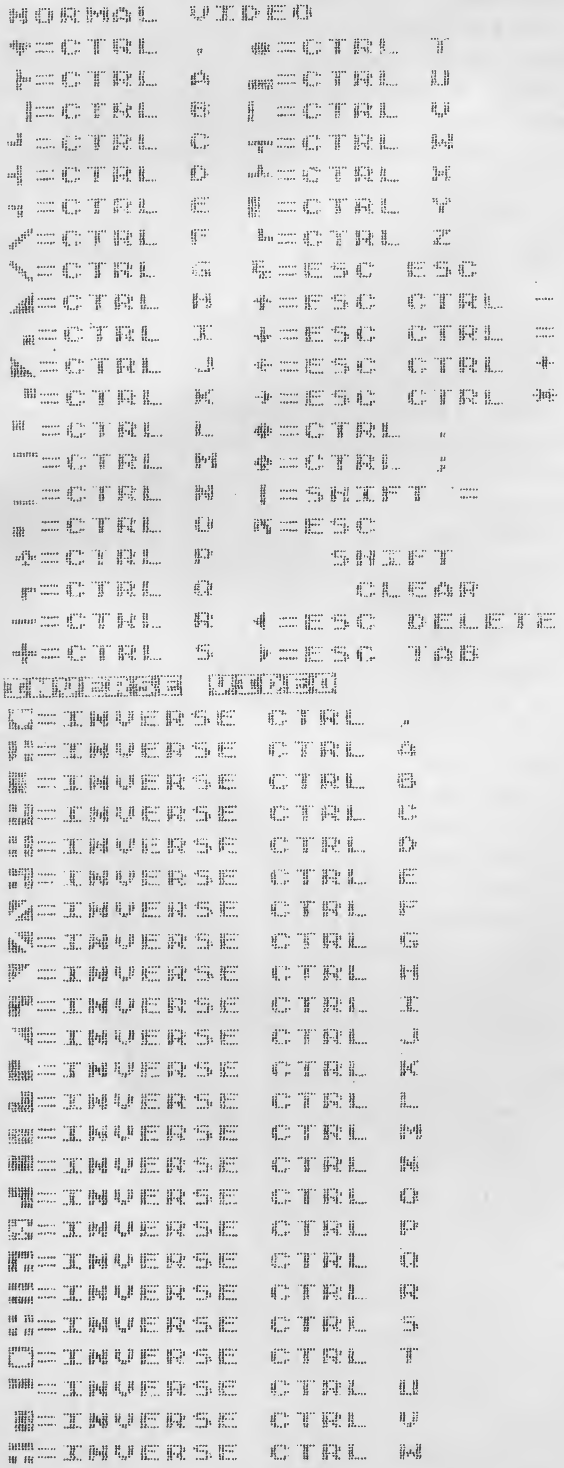
The second program of the two is a Player/Missile editor with many features not found on most editors. One of these includes some excellent demos that show the user how he or she can use the players created by this program in their games. Also a lot of the commands are joystick controlled making it very easy to become familiar with the commands. This Utility is one that no programmer should go without.

Grid Warrior requires 48K and PM/EDITOR/ANIMATOR 32K and they both sell for $24.94 from:

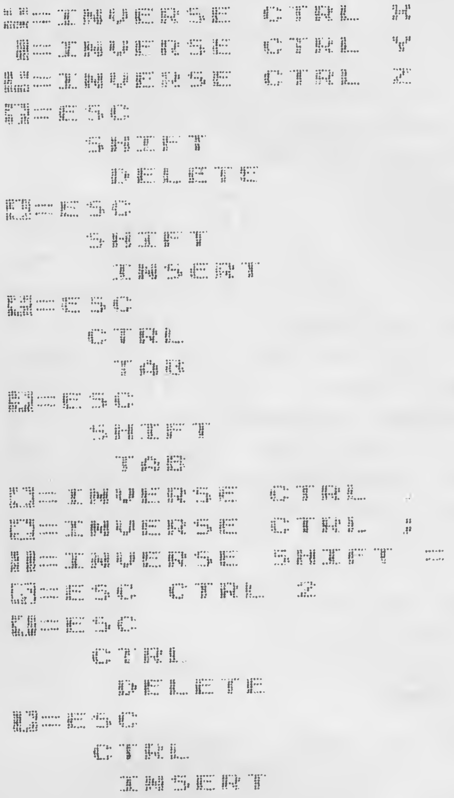
ROM SOFTWARE
P.O. BOX 252
MAPLE RIDGE,B.C.
V2X 7G1 CANADA

Listing Conventions

This page is in every magazine in order to show the different CONTROL(CTRL) characters that are listed in some of our programs. It first shows how the character will look on the screen and then following the equal sign an explanation of how to type it in. For example; to type in the clear key you would first push ESC(top left hand corner of keyboard), then hold the shift key and push clear.



```
NORMAL VIDEO
♥=CTRL ,    ●=CTRL T
├=CTRL A    ▄=CTRL U
▕=CTRL B    ▎=CTRL V
┘=CTRL C    ┬=CTRL W
┤=CTRL D    ┴=CTRL X
┐=CTRL E    ▌=CTRL Y
╱=CTRL F    └=CTRL Z
╲=CTRL G    ␛=ESC ESC
◢=CTRL H    ↑=ESC CTRL -
▗=CTRL I    ↓=ESC CTRL =
◣=CTRL J    ←=ESC CTRL +
▝=CTRL K    →=ESC CTRL *
▘=CTRL L    ◆=CTRL ,
▔=CTRL M    ♠=CTRL ;
▁=CTRL N    |=SHIFT =
▖=CTRL O    ↰=ESC
♣=CTRL P          SHIFT
┌=CTRL Q           CLEAR
─=CTRL R    ◀=ESC DELETE
┼=CTRL S    ▶=ESC TAB
INVERSE VIDEO
♥=INVERSE CTRL ,
├=INVERSE CTRL A
▕=INVERSE CTRL B
┘=INVERSE CTRL C
┤=INVERSE CTRL D
┐=INVERSE CTRL E
╱=INVERSE CTRL F
╲=INVERSE CTRL G
◢=INVERSE CTRL H
▗=INVERSE CTRL I
◣=INVERSE CTRL J
▝=INVERSE CTRL K
▘=INVERSE CTRL L
▔=INVERSE CTRL M
▁=INVERSE CTRL N
▖=INVERSE CTRL O
♣=INVERSE CTRL P
┌=INVERSE CTRL Q
─=INVERSE CTRL R
┼=INVERSE CTRL S
●=INVERSE CTRL T
▄=INVERSE CTRL U
▎=INVERSE CTRL V
┬=INVERSE CTRL W
```



```
┴=INVERSE CTRL X
▌=INVERSE CTRL Y
└=INVERSE CTRL Z
▣=ESC
   SHIFT
    DELETE
▣=ESC
   SHIFT
    INSERT
▣=ESC
   CTRL
    TAB
▣=ESC
   SHIFT
    TAB
◆=INVERSE CTRL .
♠=INVERSE CTRL ;
|=INVERSE SHIFT =
▣=ESC CTRL 2
▣=ESC
   CTRL
    DELETE
▣=ESC
   CTRL
    INSERT
```

New Products cont'd

YBM

YBM stands for "Your BASIC Machine". This program enables you to implement, modify or interface with your existing Atari computer-programs:swiftly, easily and inexpensively.

The YBM package writes Atari programs that create and abstract information from data bases you set up. Since all YBM programs are written in bug-free, hassle-free BASIC, you can quickly interface them into existing Atari sorts and statistical programs. All file organization, disk I/O and file structure are handled on disk smoothly and efficiently.

This package comes with YBM source code, complete operating instructions,and a complete flow chart of a generated program. This program comes on disk from:

Somers Corporation
P.O. BOX 295
St. Mary's,OH 45885

SPRUCING UP YOUR DISPLAY

By TOM TRAN

I have written a short little assembly program below that when put in a BASIC program makes a very colorful display. The assembly program(in the form of data statements) interrupts the computer giving it 128 colors at the same time. This program can be used to spruce up any program that may have a rather dull beginning. Put this little program at the start of any program and impress your friends. By using the FILL command (on page 54 of your BASIC reference manual) I am able to get the colors within the ROM logo.

```
10 REM ROM SYMBOL BY TOM TRAN
20 REM VOLUME 1,ISSUE 5
30 POKE 559,0:FOR I=53248 TO 5325
1:POKE I,0:NEXT I
40 DIM UP$(32),DOWN$(32)
50 T=1.75
60 GOSUB 350
70 GOSUB 210
80 FOR I=1 TO 50:NEXT I
90 SETCOLOR 2,0,0:SETCOLOR 4,0,0
100 SETCOLOR 0,8,2
110 UP$(15,15)=CHR$(22)
120 X=USR(ADR(UP$),T)
130 UP$(15,15)=CHR$(26)
140 X=USR(ADR(UP$),T)
150 SETCOLOR 0,3,4
160 DOWN$(15,15)=CHR$(22)
170 DOWN$(15,15)=CHR$(26)
180 X=USR(ADR(DOWN$),T)
190 GOTO 600
200 REM DRAW ROM LOGO
210 GRAPHICS 7+16
220 SETCOLOR 0,0,0:SETCOLOR 1,0,1
4:SETCOLOR 2,0,0:SETCOLOR 4,0,0
230 COLOR 2
240 PLOT 25,25:DRAWTO 25,55:DRAWT
O 30,55:DRAWTO 30,40:DRAWTO 50,55
:DRAWTO 55,55:DRAWTO 35,40:DRAWTO
55,40
250 DRAWTO 55,25:DRAWTO 25,25
260 POSITION 25,55:POKE 765,1:XIO
18,#6,0,0,"S:"
270 PLOT 35,40:POSITION 30,40:POK
E 765,1:XIO 18,#6,0,0,"S:"
280 PLOT 65,25:DRAWTO 65,55:DRAWT
O 95,55:DRAWTO 95,25:DRAWTO 65,25
290 PLOT 65,55:POSITION 65,25:POK
E 765,1:XIO 18,#6,0,0,"S:"
300 PLOT 105,25:DRAWTO 105,55:DRA
WTO 110,55:DRAWTO 110,30:DRAWTO 1
20,40:DRAWTO 130,30
310 DRAWTO 130,55:DRAWTO 135,55:D
RAWTO 135,25:DRAWTO 130,25:DRAWTO
120,35:DRAWTO 110,25:DRAWTO 105,2
5
320 POSITION 105,55:POKE 765,1:XI
O 18,#6,0,0,"S:"
330 PLOT 135,55:DRAWTO 135,25:DRA
WTO 130,25:POSITION 130,55:POKE 7
65,1:XIO 18,#6,0,0,"S:"
340 REM READ IN MACHINE DATA
350 RESTORE 410
360 FOR I=1 TO 32
370 READ C:UP$(I)=CHR$(C)
380 NEXT I
390 DOWN$=UP$:DOWN$(12,12)=CHR$(2
29)
400 RETURN
405 REM PULL THREE BYTE OFF THE S
TACK
410 DATA 104
420 DATA 104
430 DATA 104
440 DATA 72
450 DATA 162,57
460 DATA 160,0
470 DATA 173,0,210
480 DATA 101,20
490 DATA 141,22,208
500 DATA 141,10,212
510 DATA 136
520 DATA 208,242
530 DATA 202
540 DATA 208,237
550 DATA 104
560 DATA 56
570 DATA 233,1
580 DATA 208,228
590 DATA 96
600 GRAPHICS 18:SETCOLOR 4,8,0
610 POKE 764,255:FOR I=1 TO 30:IF
PEEK(764)=255 THEN NEXT I
620 GRAPHICS 18:SETCOLOR 4,8,0:PO
SITION 3,4
630 POKE 764,255:FOR I=1 TO 30:IF
PEEK(764)=255 THEN NEXT I
640 POKE 752,1
```

ADVENTURE GAMES — PART IV

By PETER ELLISON

In the last issue of this article (Issue 3), I made a machine language subroutine in order to move Lancelot (Player0) around the screen. In this issue I have redefined the character set in ANTIC MODE 4 to produce high resolution graphics. I used this ANTIC mode because of the little memory it takes up. Although ANTIC MODE 4 is recognized as a text mode, it does support multicolor characters. Each character in this mode is eight pixels wide, but the pixels are turned on or off in pairs. In effect, it actually gives us a character four bits wide. The color for each character is determined by the bit combination of every pair of bits in the byte. By turning them on or off in a certain way, this will produce different colors. See diagram below. The first bit pair is 11. The pixels which are turned on for this part of the character would produce the color equivalent to color register 3 or (Peek 711). For the second pair of bits, 01, this part of the character would equal color register 1 or (Peek 709). The third combination, 10, would produce the color of color register 2 or (Peek 710). And for the last bit combination, 00, it would produce the color of the background, color register 0 or (Peek 708).

```
--------------------------------
[ 1 [ 1 [ 0 [ 1 [ 1 [ 0 [ 0 [ 0 ]
--------------------------------
```

By designing the characters in this mode, each character is able to be one color or many. ANTIC mode 4 gives the appearence of being in graphic mode 7.

Lines 5000 to 7900 are the same as in the third issue of ROM, as not to leave anyone out. Those lines just move Lancelot around the screen with a joystick and player/missile graphics with a short machine language subroutine.

In the beginning of this program a short machine language subroutine was used to move the character set from ROM to RAM. Once this is accomplished the character set is redefined by the use of data statements. By defining different characters, I was able to produce different pictures on the screen. In this program I have drawn a boarder all the way around the playfield. These bricks were made by using the asterisk. The asterisk didn't even need to be redefined to make such a nice brick. The next thing that I drew with characters, was a Snake. To the left of it I wrote the word "snake",if you can read it. Next I drew a centipede and then last some trees. These are just some of the characters that I will be using in the adventure game to come. In the next issue I will have the characters moving and chasing poor Lancelot. Until next issue, Happy Adventuring!

```
10 REM DIMENSION ALL CHARACTERS
15 DIM    P1$(20),CEN$(20),SNA$(2
0),TRE$(20)
20 REM INITIALIZE CHARACTER BASE
FOR NEW CHARACTER SET
25 A=PEEK(106)-8:CB=A*256
30 REM STORE THE NEW ROM AND CHAR
ACTERSET ADDRESS
35 POKE 204,A:POKE 206,224
40 REM LOAD MACHINE CODE TO CHANG
E CHARACTER SET
45 FOR M=1 TO 20:READ B:P1$(M,M)=
CHR$(B):NEXT M
50 DATA 104,162,4,160,0,177,205,1
45,203,200,208,249,230,206,230,20
4,202,208,242,96
55 U=USR(ADR(P1$))
60 FOR X=24 TO 79:READ D:POKE CB+
X,D:NEXT X
65 REM MODIFY THE CHARACTER #
70 DATA 0,28,8,119,119,8,28,0
75 REM MODIFY THE CHARACTER $
80 DATA 0,120,143,191,191,143,120
,0
85 REM MODIFY THE CHARACTER %
90 DATA 0,192,240,252,255,255,255
```

```
,204
95 REM MODIFY THE CHARACTER &
100 DATA 3,7,14,28,56,112,224,192
105 REM MODIFY THE CHARACTER '
110 DATA 192,224,112,56,28,14,7,3
115 REM MODIFY THE CHARACTER (
120 DATA 255,255,0,0,0,0,0,0
125 REM MODIFY THE CHARACTER )
130 DATA 16,56,124,254,16,0,0,0
135 GRAPHICS 16
140 REM CHANGE FIRST LINE TO ANTI
C MODE 4
145 DLIST=PEEK(560)+PEEK(561)*256
150 POKE DLIST+3,68
155 REM CHANGE THE REST OF THE LI
NES TO ANTIC MODE 4
160 FOR X=DLIST+6 TO DLIST+28
165 POKE X,4
170 NEXT X
175 POKE 756,A
180 REM TREES
185 TRE$="))))))))))))"
190 REM CENTIPEDE
195 CEN$="$######"
200 REM SNAKE
205 SNA$="('&'&"
210 POSITION 0,1
215 ? "**************************
**************"
220 FOR WW=1 TO 21:POSITION 0,WW:
? "*":POSITION 39,WW:? "*":NEXT W
W:IF WW>=21 THEN GOTO 230
225 GOTO 220
230 POSITION 0,21:? "************
****************************"
235 POSITION 10,15:? "TREES=":POS
ITION 17,15:? TRE$
240 POSITION 11,14:? "CENTIPEDE="
:POSITION 22,14:? CEN$
245 POSITION 15,13:? "SNAKE=":POS
ITION 22,13:? SNA$
4950 REM PLAYER POSITION
5000 X=100:Y=50
5050 REM SET UP PLAYER/MISSILE AD
DRESS
5100 I=PEEK(106)-8:POKE 54279,I:P
MBASE=I*256
5200 FOR Z=PMBASE+512 TO PMBASE+6
40
5300 POKE Z,0
5400 NEXT Z
5500 POKE 559,46:POKE 53277,3
5600 POKE 53248,X
5700 POKE 704,135
5800 FOR Z=0 TO 8:READ W:POKE PMB
ASE+512+Y+Z,W
5900 NEXT Z
5950 REM CHARACTER DATA
6000 DATA 49,50,36,56,32,32,56,23
2,140
6100 GOTO 6800
6200 A=STICK(0)
6250 REM JOYSTICK FOR PLAYER MOVE
MENT
6300 IF A=11 THEN X=X-9:POKE 5327
9,0:POKE 53248,X:A=15:FOR P=1 TO
100:NEXT P:IF A=15 THEN MOVE=MOVE
-1:GOTO 6300
6400 IF A=7 THEN X=X+9:POKE 53279
,0:POKE 53248,X:A=15:FOR P=1 TO
100:NEXT P:IF A=15 THEN MOVE=MOVE
-1:GOTO 6400
6500 IF A=13 THEN R=USR(1700):Y=Y
+9:POKE 53279,0:A=15:FOR D=1 TO 1
00:NEXT D:IF A=15 THEN MOVE=MOVE-
1:GOTO 6500
6600 IF A=14 THEN Q=USR(1600):Y=Y
-9:POKE 53279,0:A=15:FOR E=1 TO 1
00:NEXT E:IF A=15 THEN MOVE=MOVE-
1:GOTO 6600
6650 IF X>=55 AND Y<15 THEN X=55
AND Y=100:POKE 53248,X
6700 GOTO 6200
6800 FOR Q=1 TO 88:READ W:POKE 16
00+Q-1,W:NEXT Q
6850 REM UPCODE
6900 POKE 1537,Y
7000 DATA 104,173,1,6,56,233,9,14
1,1,6,172,1,6,169,0,153,9,154,153
,10,154,153,11,154,153,12,154,153
,13,154
7100 DATA 153,14,154,153,15,154,1
53,16,154,153,17,154,169,49,153,0
,154,169,50,153,1,154,169,36,153,
2,154,169,56
7200 DATA 153,3,154,169,32,153,4,
154,169,32,153,5,154
7300 DATA 169,56,153,6,154,169,23
2,153,7,154,169,140,153,8,154,96
7400 FOR R=1 TO 88:READ S:POKE 17
00+R-1,S:NEXT R
7450 REM DOWNCODE
7500 DATA 104,173,1,6,24,105,9,14
1,1,6,172,1,6,169,0,153,247,153,1
53,248,153,153,249,153,153,250,15
3,153,251,153
7600 DATA 153,252,153,153,253,153
,153,254,153,153,255,153,169,49,1
53,0,154,169,50,153,1,154,169,36,
153,2,154,169,56
7700 DATA 153,3,154,169,32,153,4,
154,169,32,153,5,154
```

Continued on page 67

STOCK ANALYZER AND FILER

By BOB COCKROFT

This program is a Stock Analyzer and Filer, that I made up for myself, because I was always having trouble keeping track of each one of my stocks. This program, written in BASIC, has saved me many hours of looking through pages of stock records trying to find certain stocks.

This program is similar to the MACTODAT program in this issue, in that it saves the stock values as DATA. By positioning the cursor at the top of the screen and have it make returns, it then saves your stocks on file. When you run the program it takes a few seconds to initialize, but once its up and running, it's quite fast. The program begins with a menu of six different options. They are: E-Erase Data, D-Directory, I-Input Data, O-Output Data,S-Save Data, and L-Load Data.

The first option:E-Erase, is one used to erase certain stocks that are no longer of interest to yourself. It then asks you the name of the stock.(each stock can only be three letters long) Once you have inputed it, searching will be printed to the screen and if it finds it, it will print, "I found it! It has been erased."The next line will then say, "What Next?" If nothing is typed in here and a RETURN is pressed,then the program will end. If you type in one of the menu commands then the program will go to the command that you typed in.

The second option: I-Input Data, is the heart of the program. This is where all of the information on the stock is entered. If this data is entered incorrectly, then all of the data on this stock will be wrong. The first question that the program asks you is the name of the stock. This name can only be three letters long, as I said in the last paragraph. Once the stock name is entered in,the program then asks you for the market price. This is the price that the stock sells for on the market.The next question is, "What is the book price?" The book value is equal to the net assets per share of the stock. The next question then asks "Monthly Volume?" This is the amount of shares traded in a month. Then the last question is, "The amount of outstanding shares?" This means the amount of shares that are held by the public.

The third option: O-Output Data, is where the analysis program comes into play. The program first asks you if you would like to see any stock in particular.If you say no, it will list out the first stock it has on file.If you push the space bar you can look at the rest of the stocks on file. It will first give you the name of the stock, then the market value, book value, monthly volume, outstanding shares, public interest rating, and analysis rating. The last two names you didn't type in because they were generated through your input. The first one,Public Interest Rating, is calculated by putting the monthly volume(amount of people buying the stock) over the total shares outstanding. Public Interest, in other words, is the demand for the stock. The last output-Analysis Rating,is the monthly volume over the share outstanding, multiplied by the book rate, over the market value.

The fourth option S-Save, is where your stocks that have been entered in are saved in data statements. These Data statements are inputted into the end of the program and can be loaded into the program through the L-Load command. Also, after a person has finished entering stocks, he or she can save the data statements on disk by typing in:LIST "D:DATA",11000,B. Where B is equal to the last data line with information on it. Remember when saving something you must press break when you see the program writing out just a lot of zeros. I wish you well on the stock market, and let me know how you like the program.

```
9 REM * SETING UP *
10 DIM NAME(20,50),ST$(20),MAR(50
),BOOK(50),VOL(50),OUT(50),TOT(50
),SR$(20),SR1$(20),NAME1(20,1),NA
ME2(20)
13 GRAPHICS 1:COLOR 1:SETCOLOR 2,
16,1 :POSITION 2,6:? #6;" STOCK
ANALYZER"
14 POSITION 4,8:? #6;" AND FILER"
15 FOR Y=1 TO 50:MAR(Y)=0:BOOK(Y)
=0:VOL(Y)=0:OUT(Y)=0:NEXT Y
16 FOR B=1 TO 50:FOR B1=1 TO 20:N
AME(B1,B)=0:NEXT B1:NEXT B:GOTO 2
000
20 GRAPHICS 1:COLOR 1:SETCOLOR 2,
16,1:POSITION 2,6:? #6;" STOCK A-
NALYZER"
22 POSITION 4,8:? #6;" AND FILER"
24 POSITION 6,14:? #6;"loading":P
OSITION 5,16:? #6;"data lines"
50 FOR Y=1 TO 50:READ D:MAR(Y)=D:
NEXT Y
52 FOR Y=1 TO 50:READ D:BOOK(Y)=D
:NEXT Y
54 FOR Y=1 TO 50:READ D:VOL(Y)=D:
NEXT Y
56 FOR Y=1 TO 50:READ D:OUT(Y)=D:
NEXT Y
58 FOR Y=1 TO 50:FOR X=1 TO 20:RE
AD D:NAME(X,Y)=D:NEXT X:NEXT Y
75 GOTO 2000
98 REM * INPUT DATA SECTION *
99 GOTO 2000
100 GRAPHICS 0:FOR Y=1 TO 50:IF N
AME(1,Y)=0 THEN 110
105 NEXT Y
110 ? "NAME OF THE STOCK?"
120 INPUT ST$
125 IF ST$="P" THEN 1000
126 IF ST$="D" THEN 2000
130 IF ST$="E" THEN 800
135 TRAP 120
140 L=LEN(ST$)
150 FOR LL=1 TO L:NAME(LL,Y)=ASC(
ST$(LL)):NEXT LL
160 ? :? "The market price?"
170 INPUT A
175 MAR(Y)=A
176 IF A=0 THEN 170
180 ? :? "The book price?"
182 INPUT A1
184 BOOK(Y)=A1
190 ? :? "The monthly volume?":IN
PUT A2:VOL(Y)=A2
195 ? :? "The amount of outstandi
ng shares?":INPUT A3:OUT(Y)=A3
500 GOTO 100
610 POSITION 2,P:? "POKE 842,12:G
OTO 10"
799 REM * ERASE SECTION *
800 GRAPHICS 0:? :? "Which stock
do you want to erase?"
805 INPUT SR$
806 TRAP 805
807 ? :? "            Searching":
?
810 L1=LEN(SR$):FOR L2=1 TO L1:NA
ME1(L2,1)=ASC(SR$(L2)):NEXT L2
830 FOR Y=1 TO 50
840 X=1
850 IF NAME1(X,1)<>NAME(X,Y) THEN
860
855 X=X+1
856 IF X<4 THEN 850
858 ? "I found it!":FOR X=1 TO 20
:NAME(X,Y)=0:NEXT X:? :? "It has
been erased":GOTO 880
860 NEXT Y
870 ? "SORRY! file not found"
880 ? :? "What next?"
885 INPUT SR$:IF SR$="E" THEN 800
890 IF SR$="D" THEN 2000
899 END
999 REM * OUTPUT SECTION *
1000 GRAPHICS 0
1001 ? :? "Would you like to see
any particular  stock?":INPUT SR$
:IF SR$="N" THEN 1022
1002 TRAP 1001
1003 ? :? "What stock?":INPUT SR$
:? ,"Searching":L10=LEN(SR$):FOR
LL=1 TO L10:NAME2(LL)=ASC(SR$(LL)
):NEXT L
1004 FOR Y=1 TO 50:X=1
1005 IF NAME2(X)<>NAME(X,Y) THEN
1009
1006 X=X+1
1007 IF X<4 THEN 1005
1008 ? "I found it!":GOTO 1022
1009 NEXT Y
1010 ? "stock not on file"
1022 Y=0
1023 Y=Y+1:FOR X=1 TO 20
1024 IF NAME(X,Y)=0 THEN 1060
1025 ? CHR$(NAME(X,Y));
1060 NEXT X
```

Stock Analyzer cont'd

```
1065 ? :? "   MARKET
 ";MAR(Y)
1070 ? :? "   BOOK VALUE
 ";BOOK(Y)
1075 ? :? "   MONTHLY VOLUME
 ";VOL(Y)
1080 ? :? "   OUTSTANDING SHARES
 ";OUT(Y)
1085 ? :? "   PUBLIC INTEREST RA
TING ";(INT((VOL(Y)/(OUT(Y)+1.0E-
03))*100000))/1000;
1095 ? :? :? "   ANALYSIS RATING
  ";(INT((VOL(Y)/(OUT(Y)+1.0E-03
)*BOOK(Y)/(MAR(Y)+1.0E-03))*10000
0))/1000
1200 ? :? "----------------------
----------"
1205 IF PEEK(764)=58 THEN 2000
1210 IF PEEK(764)<>33 THEN 1210
1220 IF Y<50 THEN 1023
1999 REM * DIRECTORY SECTION *
2000 GRAPHICS 0:POSITION 13,1:? "
STOCK ANALYZER":POSITION 15,2
2010 ? "DIRECTORY"
2015 ? :? ,,"   PRESS"
2020 ? :? "     ERASE:
E"
2030 ? "       DIRECTORY:
D"
2040 ? "       SAVE:
S"
2050 ? "       INPUT DATA:
I"
2055 ? "       OUTPUT DATA:
O"
2060 ? "       LOAD DATA LINES:
L"
2100 INPUT SR$
2110 IF SR$="E" THEN 800
2120 IF SR$="S" THEN 6000
2130 IF SR$="I" THEN 100
2140 IF SR$="O" THEN 1000
2160 IF SR$="L" THEN 20
2200 GOTO 2100
2999 END
5999 REM * SAVING DATA LINES SECT
ION *
6000 GRAPHICS 0:POSITION 2,2:LINE
=11000:FOR L=1 TO 50:TOT(L)=MAR(L
):NEXT L:COUNT=1
6010 P=2
6020 X=0
6030 FOR Y=1 TO 50
6040 IF X=0 THEN POSITION 2,P:? L
INE;" DATA ";
6045 IF Y=50 THEN ? TOT(Y):GOTO 6
090
6050 X=X+1
6070 IF X>20 THEN ? TOT(Y):X=0:P=
P+3:LINE=LINE+5:GOTO 6080
6075 ? TOT(Y);",";
6080 IF P>15 THEN X=0:GRAPHICS 0:
GOTO 6100
6090 NEXT Y
6100 POSITION 2,P+3:? "POKE 842,1
2:GOTO 6150":GOTO 30000
6150 IF Y<50 THEN 6090
6160 COUNT=COUNT+1:IF COUNT=2 THE
N 6200
6165 IF COUNT=3 THEN 6210
6170 IF COUNT=4 THEN 6220
6180 IF COUNT=5 THEN 7000
6200 GRAPHICS 0:POSITION 2,2:LINE
=12000:FOR L=1 TO 50:TOT(L)=BOOK(
L):NEXT L:GOTO 6010
6210 GRAPHICS 0:POSITION 2,2:LINE
=13000:FOR L=1 TO 50:TOT(L)=VOL(L
):NEXT L:GOTO 6010
6220 GRAPHICS 0:POSITION 2,2:LINE
=14000:FOR L=1 TO 50:TOT(L)=OUT(L
):NEXT L:GOTO 6010
7000 GRAPHICS 0:POSITION 2,2
7001 P=2
7005 LINE=15000
7010 ? LINE;" DATA ";
7020 FOR Y=1 TO 50
7024 FOR X=1 TO 20
7025 IF P>15 THEN P=2:END :GRAPHI
CS 0:GOTO 5030
7026 IF X=20 THEN ? NAME(X,Y):GOT
O 7029
7027 ? NAME(X,Y);:? ",";
7029 NEXT X
7030 P=P+3
7031 LINE=LINE+5:IF P>15 THEN POS
ITION 2,P:? "POKE 842,12:GOTO 704
0":GOTO 30000
7032 POSITION 2,P:? LINE;" DATA "
;
7035 NEXT Y
7036 GOTO 8000
7040 GRAPHICS 0:P=2:POSITION 2,P:
? LINE;" DATA ";:GOTO 7035
8000 END
30000 POSITION 2,0:POKE 842,13
```

Adventure Games cont'd

```
7800 DATA 169,56,153,6,154,169,23
2,153,7,154,169,140,153,8,154,96
7900 GOTO 6200
```

Why Buy An ATARI cont'd

ROM
P.O. BOX 252
Maple Ridge,B.C.
V2X 7G1

ANTIC
297 Missouri Street
San Francisco,CA 94107

A.N.A.L.O.G. Computing
565 Main Street
Cherry Valley,MA 01611

These magazines are useful for anyone, from the beginner, on up to the advanced system programmer. They are also very useful in that they help keep the programmer updated with what is available on the market. We hope that our magazine can help even the BASIC programmer get a grasp in the programming of Assembly language.

Advertiser List

COMPUTER PALACE	IBC
COMPUTER SOFTWARE SERVICES	54
DATUM	53
DRAPER SOFTWARE	49
DYNASTY ENGINEERING	45
FRONTRUNNER	32
FUNSOFT,INC	11
GARDNER COMPUTING	5,53
KIDSTUFF SOFTWARE	13
MICROPROSE	47
O.S.S. PRECISION SOFTWARE	OBC
RADICAL SYSTEMS	7
ROM SOFTWARE	IFC

Subscribe To ROM and SAVE!!

SEND TO:
ROM MAGAZINE
P.O. BOX 252
Maple Ridge,B.C.
V2X 7G1 CANADA

___1 year magazine(6 issues) Only $12

___2 year Magazine(12 issues)Only $23

___1 year Magazine+Cassette Only $25

___1 year Magazine+Disk Only $40

Name______________________________

Address___________________________

City/Prov.________________________

Pos/Zip CODE______________________

__Check enclosed __Money Order

Making a Game cont'd

space. So if Z<>32 means the object hit something other than blank space!

That'S all to make a object move around the screen. Isn't that easy. If you have any questions about anything in this article, write to me at ROM. Next issue, this article will explore further capabilities of the ATARI to make a game more exciting.

WE SPECIALIZE IN ATARI COMPUTERS SO WE KNOW WHAT WORKS BEST!

COMPUTER PALACE

CALL US AT (503) 683-5361 FOR INFORMATION ON THE LATEST AND BEST FOR ATARI

EXODUS
ULTIMA III

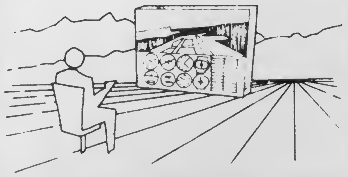
- 2-disk set!
- A lot of game for your money!

$52.50

New from Origin Systems.

Here is the latest in the famous series of Fantasy/Role-Playing games. This one is the best yet!

Flight Simulator II



So what if you haven't gotten your pilot's license yet . . . You can take off with this super-scrolling flight simulator and fly to your heart's content! The disk even contains a WW I flying ace dog fight game so that you can brush up on your aerial combat!

48K Disk From Sublogic **$44.90**

DIMENSION-X

What has unlimited power . . . Scrolling perspective . . . and super excitement? THIS DOES!!!

32K Disk/Tape from Synapse Only **$31.50**

FREE Catalog

With any order, or send $1 (refundable with purchase).

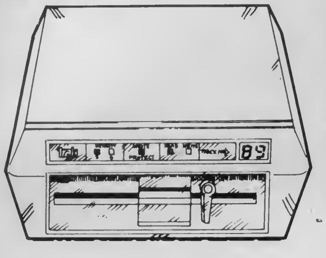
Send us your name & address for FREE FLYERS with Special Offers!

Use Your Credit Cards!

We honor VISA, American Express, & MasterCard.

"IF YOU DON'T HAVE OUR CATALOG... YOU'RE MISSING OUT!"

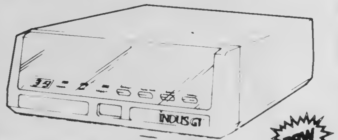
TRAK DISK DRIVE



- DOUBLE DENSITY
- BUILT-IN PARALLEL PRINTER INTERFACE & BUFFER

$398.00 Plus Frt.

INDUS-GT DISK DRIVE



new

New Low Price **$398.00** + Frt.

- Quiet & Reliable
- Faster I/O than Atari 1050 Drive
- True Double Density Operation
- Includes Free Software: DOS—Word Processing—Data-Base—Spread Sheet
- Includes Carrying Case

ONE YEAR WARRANTY!!!

Now You Can Run CP/M On Your Atari!

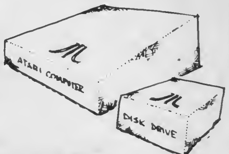
Priced from **$299.00** Plus Freight (16K model, not CP/M compatible)

ATR-8000 Expander

- 64K, Z80 computer
- Runs any combination of 5¼" and 8" drives in single or double density
- Built-in serial and parallel ports
- Built-in printer buffer

64K CP/M model	**$449.00**
2-DDDS 5¼" add-on drives	**595.00**
1-DDDS 5¼" add-on drive	**379.00**

Protect Your Investment with these durable and attractive



DUST COVERS

Custom Fitted For:

- **ATARI 400,800,600XL,800XL,1200XL, 410,810, 1010,1050,1025 and 1027**
- **EPSON MX-80 and FX-80**
- **GEMINI 10**
- **PROWRITER 8510**
- **PERCOM**
- **RANA**

Please specify **Only $8.95 Each**

SPECIAL! ***2 for $16.95***

BOOKS & MAGAZINES

COMPUTE PUBLICATIONS:	
1st Book of Atari	12.95
2nd Book of Atari	12.95
1st Book of Atari Graphics	12.95
1st Book of Robots & Computers	12.95
1st Book of Atari Games	12.95
Atari Basic Source Book	12.95
Inside Atari DOS	19.95
Mapping the Atari	14.95
Machine Language for Beginners	14.95
Machine Language Programming:	
6502 Assembly Language Prgm.	18.95
6502 Assembly Language Subroutines	17.95
The Atari Assembler	14.95
De Re Atari (All About Atari)	17.90
Machine Language for Beginners	14.95
Basic Programming Applications:	
Games Atari's Play	14.95
Kids and the Atari	19.95
Your Atari Computer	17.95
Magazines:	
ANTIC-vol. 2, no. 11	2.50
Back issues-vol2, no. 2, 3, 4, 9, 10 ea.	2.50
ANALOG-no.16	3.00
Back issues-no. 13, 14, 15 ea.	3.00
HiRes-no.3	2.95
Back issues-no. 1, 2 ea.	2.95

SUPER SPECIALS!

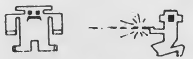
LIMITED QUANTITIES—WHILE SUPPLIES LAST!

(PLEASE LIST YOUR SECOND-CHOICE WHEN ORDERING)

MOUNTAIN KING

16K cartridge
From CBS
Reg. $39.95
Only **$17.95**

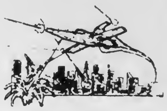
K-Razy Shoot Out



Reg. $39.95 **$17.95**

cartridge from CBS

METEOR STORM



Reg. $29.95 **$9.95**

16K Disk from Royal Software

Boulders & Bombs

From CBS

Reg. $39.95

$17.95 16K CARTRIDGE

	Regular	SPECIAL
Chicken (D, T)	35.00	**$19.95**
Guardian of Gorn (D)	35.00	**9.95**
Genetic Drift (D)	30.00	**14.95**
Claim Jumper (D, T)	35.00	**19.95**
Protector II (D, T)	35.00	**$19.95**
David's Midnite Magic (D)	35.00	**17.95**
Race In Space (T)	30.00	**19.95**
Megalegs (T)	35.00	**9.95**

SOFTWARE

Joust (C)	44.90
Robotron (C)	43.50
Blue Max (D, T)	31.50
Odesta Chess (D)	59.50
Pole Position (C)	44.90
Donkey Kong (C)	44.90
Monkey Wrench II (C)	54.00
Ultima II (D)	52.50
Dig Dug (C)	44.90
B/Graph (D)	89.90
The Tax Advantage (D)	62.50
Letter Perfect (C)	89.90
Spell Perfect (D)	69.50
Home Accountant (D)	67.50
Q-Bert (C)	44.90
Valforth (D)	54.50

HARDWARE (plus freight)

Astra 1620 Dual Drive	529.00
Rana 1000 Drive	349.00
Super-5 Printer	329.00
Atari 1027 Printer	319.00
Gemini & Epson Printers	CALL
Atari Graphics Tablet	79.10
Koala Pad Tablet	87.90
Power Pad Tablet	87.90
Axiom Printer Interface	99.00

COMPUTER PALACE

OPEN Monday - Saturday, 9 to 6
2160 W. 11th Avenue Eugene, Oregon 97402

USE YOUR CREDIT CARD & CALL
Toll Free 1-800-452-8013
(Orders Only, Please)
For Information Call (503) 683-5361

Shipping Note: UPS & Parcel Post-$2.90. UPS Air-$4.75
Shipping on hardware & paper supplies-extra. Call.
24 & 48 hour shipping available—Call (503) 683-5361

We specialize in *ATARI Compatible Products*
So we know what works best!!!
Call for the latest info. . . (503)683-5361

Craftsmen Need Precision Tools . . . Programmers! Demand Precision Software!

BASIC XL has twice the speed and twice the power of Atari® BASIC. And yet, as befits a fine craftsman's tool, **BASIC XL** is even easier to use and more dependable, while including such outstanding major additions as structured programming, string arrays, programming aids, enhanced graphics, and business capabilities.

Atari BASIC is a good starting point. We should know. We wrote it in 1978. Buy **BASIC XL.** Take advantage of five more years of experience!

So, prepare yourself for some exploration into imaginative programming with **BASIC XL!** Cartridge, excellent tutorial, reference manual . . . $99.

OSS™
Precision Software Tools
1173D S. Saratoga/Sunnyvale Road
San Jose, CA 95129 • (408) 446-3099

Functional, honest, and beautiful describe the simple lines of a craftsman's tools. For the jeweler these tools are an extension of the human hand to better execute complex designs. For you, the programmer, Precision Software Tools keep complications out of your programming while allowing you to produce intricate programs.

See the complete collection of OSS Precision Software Tools!™

MAC/65:	The fastest 6502 macro assembler/editor package on cartridge . . . $99.
BUG/65:	A powerful debugger. On disk, with OS/A+ . . . $35.
C/65:	The first native mode "small c" compiler for Atari and Apple® computers. On disk . . . $80.
ACTION!:	The fastest, small computer language ever. A feature-packed cartridge at only . . . $99.

All products on disk include OS/A+ and also require 48 K.

SEE YOUR LOCAL DEALER!

Call or write for informative brochures.

ATARI and APPLE II, are trademarks of Atari, Inc. and Apple Computer, Inc., respectively. MAC/65, BUG/65, C/65, BASIC XL, OS/A+, and OSS PRECISION SOFTWARE TOOLS™ are trademarks of O.S.S. INC. ACTION! is a trademark of Action Computer Products.